

ALIBABA CLOUD

# 阿里云

## 容器服务Kubernetes版 快速入门

文档版本：20220507

 阿里云

## 法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.入门概述	05
2.通过控制台快速使用ACK	07
3.通过kubectl快速使用ACK	13
4.更多入门教程	18
4.1. 快速创建Kubernetes托管版集群	18
4.2. 使用镜像快速创建无状态Deployment应用	23
4.3. 创建有状态工作负载StatefulSet	34
4.4. 通过Helm CLI一键部署WordPress	46
5.使用指引	50

# 1.入门概述

阿里云容器服务Kubernetes版（简称容器服务ACK）提供高性能的容器化应用管理服务，让您轻松高效地在云端运行Kubernetes容器化应用。本文介绍容器服务ACK的快速使用流程、快速使用方式、文档使用指引以及相关的常见问题，帮助您快速上手容器服务ACK。

## 快速使用流程

ACK的快速使用流程如下图所示。



## 使用方式

ACK的完整使用流程支持以下两种方式：

- [通过控制台快速使用ACK](#)
- [通过kubectl快速使用ACK](#)

此外，您还可以通过API、SDK、CLI、Terraform方式使用ACK，请参见[使用Kubernetes API](#)、[SDK说明](#)、[新版阿里云CLI工具ARC使用说明](#)和[Terraform概述](#)。

## 体验实验室

### 通过控制台快速部署容器应用

通过本实验您可以体验到：

- 容器服务Kubernetes版真实场景。
- 如何快速部署并监控容器应用。
- 如何通过容器服务控制台监控应用。

[立即体验](#)

## 入门必读

- [我想使用ACK运行我的一个应用，但是不知道如何把它做成Docker镜像，是否有相应指导？](#)

ACR为您提供便捷的容器镜像构建功能。关于制作应用Docker镜像的详细步骤，请参见[在Dockerfile中使用多阶段构建打包Java应用](#)。同时您也可以使用开源工具Derrick降低您应用容器化的复杂度。

- 我不会制作镜像，是否可以使用ACK?

除了支持部署您在容器镜像服务中托管的镜像，ACK也支持选择官方镜像，及您在容器镜像服务上收藏或者搜索到的公开镜像。具体操作，请参见[使用镜像快速创建无状态Deployment应用](#)。

- 创建集群前如何规划网络?

创建ACK集群时，VPC CIDR、Service CIDR和Pod CIDR不允许重叠。通常情况下，您可以选择自动创建的专有网络，使用默认的网络地址即可。但在某些复杂的场景下，需要您自主规划ECS地址、Kubernetes Pod地址和Service地址。具体操作，请参见[Kubernetes集群网络规划](#)。

- 创建集群时如何选择Terway和Flannel网络插件?

Flannel使用的是简单稳定的社区CNV插件，但功能偏简单，不支持基于Kubernetes标准的Network Policy。Terway是阿里云容器服务ACK自研的网络插件，支持Network Policy和容器的带宽限流，网络性能相对Flannel也更高。具体操作，请参见[使用Terway网络插件](#)。

- 集群创建失败，该如何处理?

您可以查看集群日志，然后根据相应的解决方法解决问题。具体操作，请参见[Kubernetes集群创建失败的解决方法](#)。

- 如何通过公网访问工作负载?

ACK支持五种公网访问方式：

- 节点访问 (NodePort)
- [负载均衡 \(LoadBalancer\)](#)
- [Nginx Ingress](#)
- [域名 \(External DNS\)](#)
- DNAT网关 (DNAT)

- 同个集群中的多个工作负载，如何进行互相访问?

集群内多个工作负载间如需相互访问，可以通过集群内部域名或者ClusterIP的方式。

例如：同个集群中的A工作负载需要访问B工作负载，您需要为B工作负载添加ClusterIP（集群内访问）型的服务。具体操作，请参见[管理服务](#)。A工作负载可以通过以下两种方式访问B工作负载：

- `<自定义的服务名称>.<工作负载所在命名空间>.svc.cluster.local:<端口号>`
- `ClusterIP:<端口号>`

- 使用负载均衡暴露服务时需要注意些什么?

当Service的类型设置为Type=LoadBalancer时，ACK的CCM（Cloud Controller Manager）组件会为该Service自动创建或配置阿里云负载均衡SLB（Server Load Balancer）。如果手动在SLB控制台配置SLB可能会导致Service不可访问。更多信息，请参见[Service的负载均衡配置注意事项](#)。

- 如何拉取私有镜像?

推荐使用aliyun-acr-credential-helper组件。该组件会默认安装在所有ACK集群中，支持在ACK集群中免密拉取容器镜像服务ACR中的私有镜像。具体操作，请参见[使用免密组件拉取容器镜像](#)。

## 2.通过控制台快速使用ACK

阿里云容器服务Kubernetes版（简称容器服务ACK）提供高性能的容器化应用管理服务，让您轻松高效地在云端运行Kubernetes容器化应用。本文将指导您如何通过控制台在ACK集群中快速部署并公开一个容器化Demo应用，并监控应用的运行情况。

### 背景信息

- 本教程中所使用的Demo应用ACK-Cube为一个线上魔方游戏，该游戏将通过容器镜像部署到ACK Pro版集群中。完成本教程后，您将得到一个ACK Pro版集群和魔方游戏应用。



- Demo应用的容器镜像基于[开源项目](#)构建，镜像地址为 `registry.cn-hangzhou.aliyuncs.com/acr-toolkit/ack-cube:1.0`。
- ACK标准版和Pro版都是托管集群。相比于标准版，本教程所创建的Pro版集群具有更高的可靠性和安全性，并提供可赔付的SLA。关于ACK集群及其所使用云资源的收费情况，请参见[产品计费](#)。

### 前提条件

- 您已注册阿里云账号并完成实名认证，请参见[注册阿里云账号](#)和[个人实名认证](#)。
- 您已了解Kubernetes的相关基本概念，请参见[与原生Kubernetes名词对照](#)。你还可以通过[CNCF x Alibaba云原生技术公开课](#)，深入学习Kubernetes。

### 操作流程



### 操作视频

#### 步骤一：开通并授权容器服务ACK

首次使用时，您需要开通容器服务ACK，并为其授权相应云资源的访问权限。

1. 登录[容器服务ACK开通页面](#)。

2. 阅读并选中容器服务ACK服务协议。
3. 单击立即开通。
4. 登录[容器服务管理控制台](#)。
5. 在容器服务需要创建默认角色页面，单击前往RAM进行授权进入[云资源访问授权](#)页面，然后单击同意授权。

完成以上授权后，刷新控制台即可使用容器服务ACK。

## 步骤二：创建ACK Pro版集群

本步骤指导您如何快速创建一个ACK Pro版集群，因此参数配置大多以默认为主。关于创建集群的详细参数描述，请参见[创建ACK Pro版集群](#)。

1. 登录[容器服务管理控制台](#)。
2. 在控制台左侧导航栏中，单击集群。
3. 在集群列表页面中，单击页面右上角的创建集群。
4. 在ACK托管版页签下，配置以下相关的集群参数，未说明参数保留默认设置即可。

参数	说明
集群名称	填写集群的名称。示例：ACK-Demo。
集群规格	选择集群规格。示例：Pro版。关于ACK Pro版集群的详细信息，请参见 <a href="#">ACK Pro版集群概述</a> 。
地域	选择集群所在的地域。示例：华北2（北京）。
专有网络	Kubernetes集群仅支持运行于专有网络，因此您需要为集群指定专有网络VPC，且该VPC必须与集群处于同一地域。 示例：在华北2（北京）地域下创建名为vpc-ack-demo的VPC。通过单击创建专有网络进行创建，详情请参见 <a href="#">创建和管理专有网络</a> 。
虚拟交换机	选择用于集群节点间通信的交换机。示例：在vpc-ack-demo的VPC下创建一个名为vswitch-ack-demo的虚拟交换机，并选择使用该交换机。 通过单击创建虚拟交换机进行创建，详情请参见 <a href="#">使用交换机</a> 。
API Server访问	设置集群API Server是否可在公网访问，当您需要从公网远程管理集群时，需要配置弹性公网IP（EIP）。 示例：选中使用EIP暴露API Server。

5. 单击下一步：节点池配置，配置以下相关参数，未说明参数保留默认设置即可。

参数	说明
实例规格	为集群选配所使用的节点。为了保证集群的稳定性，建议的实例规格为：vCPU ≥ 4核，内存 ≥ 8 GiB。关于如何选型以及规格介绍，请参见 <a href="#">ECS选型和实例规格族</a> 。 您可通过设置vCPU和内存的大小，或者直接搜索实例规格，选用该规格的节点。
数量	根据需要设置集群的节点数量。示例：2个，防止单点故障。

参数	说明
系统盘	选择节点所使用的系统盘。示例：ESSD云盘，40 GiB（最小规格）。
登录方式	选择登录节点的方式，并输入密码。示例：设置密码。

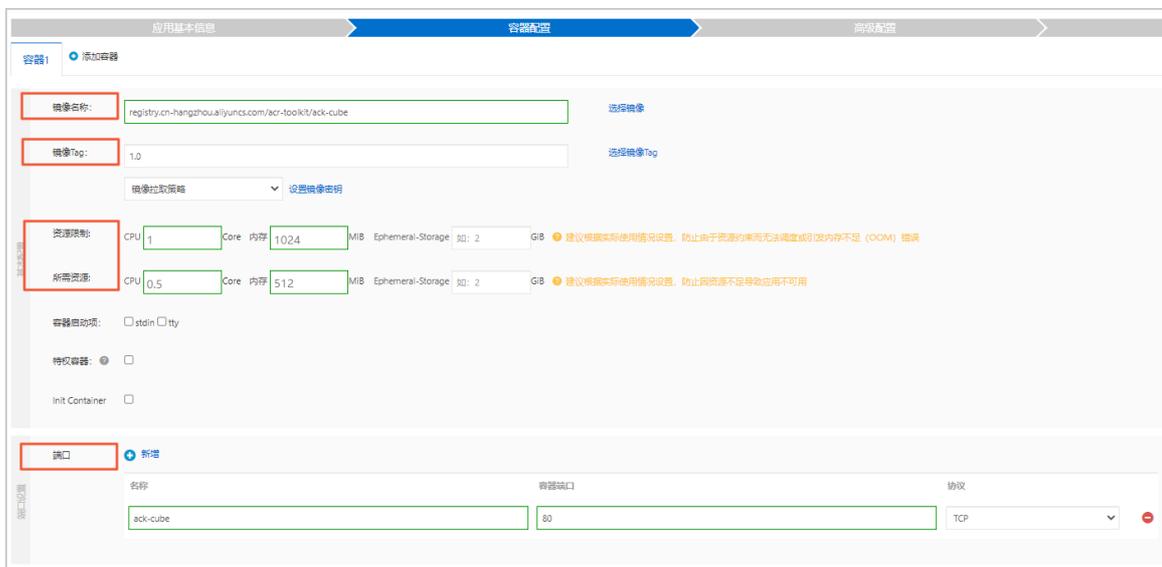
- 单击下一步：组件配置，所有组件使用默认配置。
- 单击下一步：确认配置，然后选中并阅读服务协议，单击创建集群。

**说明** 集群的创建时间一般约为10分钟。创建完成后，在集群列表页面，可以看到新创建的集群。

### 步骤三：部署并公开应用

本步骤指导您如何在新创建的ACK集群中快速部署一个无状态应用（Deployment），即魔方游戏，并将该应用向公网公开。关于创建Deployment的详细参数描述，请参见[创建无状态工作负载Deployment](#)。

- 在集群列表页面中，单击目标集群名称（即ACK-Demo）。
- 在集群管理页左侧导航栏中，选择工作负载 > 无状态。
- 在无状态页面中，单击使用镜像创建。
- 在应用基本信息页签，设置应用名称为ack-cube。
- 单击下一步，在容器配置页签，配置容器的相关参数。



参数	说明
镜像名称	直接输入不包含镜像Tag的镜像地址或通过单击选择镜像来选择所需的镜像。 示例：输入 <code>registry.cn-hangzhou.aliyuncs.com/acr-toolkit/ack-cube</code>
镜像Tag	单击选择 <b>镜像Tag</b> 选择镜像的版本。若不指定，默认为最新版。示例：1.0。

参数	说明
资源限制	根据需为应用指定所能使用的资源上限，防止占用过多资源。 示例：1 Core，内存 1024 MiB，Ephemeral Storage为空。
所需资源	根据需为应用指定预留的资源额度，防止因资源不足而导致应用不可用。 示例：0.5 Core，内存 512 MiB，Ephemeral Storage为空。
端口	设置容器的端口。示例：ack-cube，80，TCP。

- 单击下一步，在高级配置页签，单击服务（Service）右侧的创建。
- 在创建服务对话框中，设置服务的相关参数，单击创建，以通过该服务公开ack-cube应用。

创建服务
✕

名称:

命名空间: default

类型:
负载均衡
公网访问

新建SLB
简约型 (slb.s1.small) [修改](#)

⚠ 请根据自己业务选择SLB规格，SLB计费详情请参考[产品定价](#)；自动新建的SLB在Service删除时会被删除。

外部流量策略: Local

端口映射:
+ 添加

名称	服务端口	容器端口	协议	
<input style="width: 80%;" type="text"/>	80	80	TCP	-

注解: + 添加

标签: + 添加

创建
取消

参数	说明
名称	输入服务的名称。示例：ack-cube-svc
类型	选择服务类型，即服务访问的方式。依次选择负载均衡 -> 公网访问 -> 新建SLB -> 单击修改选择所需的SLB规格。 示例：使用默认规格，简约型I (slb.s1.small)。

参数	说明
端口	设置服务端口和容器端口。容器端口需要与后端的Pod中暴露的容器端口一致。 示例：皆为80。

8. 在高级配置页签，单击页面右下角的**创建**。

创建成功后，默认进入创建完成页面，会列出应用包含的对象，您可以单击查看应用详情进行查看。



### 步骤四：测试应用

本步骤指导您如何通过服务（Service）来访问新部署的容器化应用。

1. 在集群列表页面中，单击目标集群名称（即ACK-Demo）。
2. 在集群管理页左侧导航栏中，选择**网络 > 服务**。
3. 在服务列表页面，找到新创建的服务（即ack-cube-svc），单击**外部端点**列的IP地址，即可访问魔方游戏。

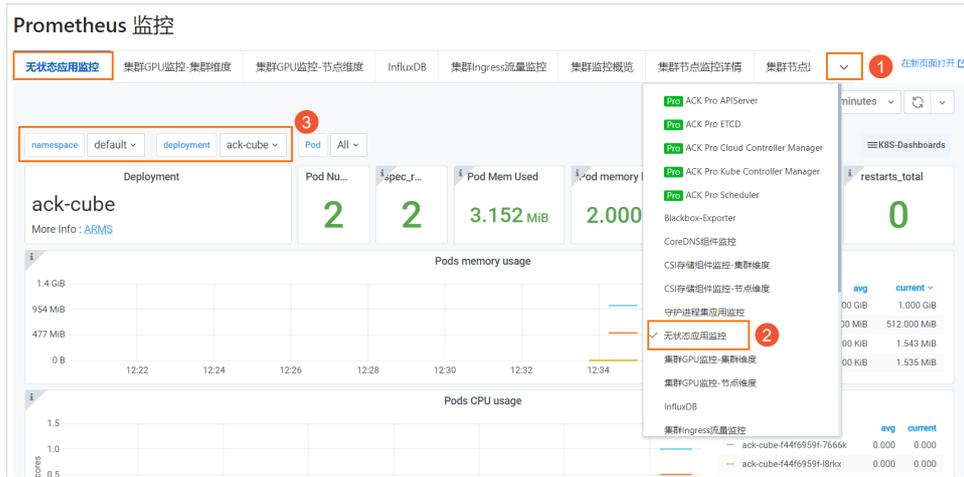


### 步骤五：监控应用

本步骤指导您如何监控应用的运行状况，如CPU使用率、内存利用率、网络I/O压力等指标。

1. 在集群列表页面中，单击目标集群名称（即ACK-Demo）。
2. 在集群管理页左侧导航栏中，选择**运维管理 > Prometheus监控**。
3. 在Prometheus监控页面，单击**无状态应用监控**页签，选择namespace为default，选择deployment为ack-cube。

您可以查看应用的资源使用情况，包括创建应用时所设置的资源阈值，即所需资源（对应图中request）和资源限制（对应图中limit）。



4. 在Prometheus监控页面，单击集群Pod监控页签，选择namespace为default，选择Pod为待监控Pod。

通过集群Pod监控，您可以查看单个Pod的资源使用情况。

### 相关文档

- 为了保证应用能够动态调整所需容器资源，您可以配置容器水平伸缩（HPA）、定时容器水平伸缩（CronHPA）、容器垂直伸缩（VPA）等，详情请参见[弹性伸缩概述](#)。
- 除了通过服务（Service）公开应用，您还可以通过路由（Ingress）实现对应用的七层网络路由控制，详情请参见[创建Ingress路由](#)。
- 除了观测容器性能，您还可以观测集群基础设施、应用性能和用户业务，详情请参见[可观测性体系概述](#)。
- 为了避免产生不必要的费用，请及时清理不需要的集群，详情请参见[删除集群](#)。

## 3.通过kubect l快速使用ACK

阿里云容器服务Kubernetes版（简称容器服务ACK）提供高性能的容器化应用管理服务，让您轻松高效地在云端运行Kubernetes容器化应用。本文将指导您如何通过kubect l在ACK集群中快速部署并公开一个容器化Demo应用，并监控应用的运行情况。

### 背景信息

- 本教程中所使用的Demo应用ACK-Cube为一个线上魔方游戏，该游戏将通过容器镜像部署到ACK Pro版集群中。完成本教程后，您将得到一个ACK Pro版集群和魔方游戏应用。



- Demo应用的容器镜像基于[开源项目](#)构建，镜像地址为 `registry.cn-hangzhou.aliyuncs.com/acr-toolkit/ack-cube:1.0`。
- Kubect l是标准的Kubernetes命令行管理工具，通过kubect l您可以连接和管理ACK集群。关于kubect l命令行的更多信息，请参见[kubect l](#)。
- CloudShell是阿里云推出的云命令行工具，您可以在容器服务ACK控制台上打开CloudShell，无需任何安装或配置操作，可以直接在CloudShell中通过kubect l管理集群。

### 前提条件

- 您已注册阿里云账号并完成实名认证，请参见[注册阿里云账号](#)和[个人实名认证](#)。
- 您已了解Kubernetes的相关基本概念，请参见[与原生Kubernetes名词对照](#)。你还可以通过[CNC F × Alibaba云原生技术公开课](#)，深入学习Kubernetes。

### 操作流程



### 操作视频

#### 步骤一：开通并授权容器服务ACK

首次使用时，您需要开通容器服务ACK，并为其授权相应云资源的访问权限。

1. 登录[容器服务ACK开通页面](#)。
2. 阅读并选中容器服务ACK服务协议。
3. 单击**立即开通**。
4. 登录[容器服务管理控制台](#)。
5. 在容器服务需要创建默认角色页面，单击前往RAM进行授权进入[云资源访问授权](#)页面，然后单击**同意授权**。

完成以上授权后，刷新控制台即可使用容器服务ACK。

## 步骤二：创建ACK Pro版集群

本步骤指导您如何快速创建一个ACK Pro版集群，因此参数配置大多以默认为主。关于创建集群的详细参数描述，请参见[创建ACK Pro版集群](#)。

1. 登录[容器服务管理控制台](#)。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在集群列表页面中，单击页面右上角的**创建集群**。
4. 在ACK托管版页签下，配置以下相关的集群参数，未说明参数保留默认设置即可。

参数	说明
集群名称	填写集群的名称。示例：ACK-Demo。
集群规格	选择集群规格。示例：Pro版。关于ACK Pro版集群的详细信息，请参见 <a href="#">ACK Pro版集群概述</a> 。
地域	选择集群所在的地域。示例：华北2（北京）。
专有网络	Kubernetes集群仅支持运行于专有网络，因此您需要为集群指定专有网络VPC，且该VPC必须与集群处于同一地域。 示例：在华北2（北京）地域下创建名为vpc-ack-demo的VPC。通过单击 <b>创建专有网络</b> 进行创建，详情请参见 <a href="#">创建和管理专有网络</a> 。
虚拟交换机	选择用于集群节点间通信的交换机。示例：在vpc-ack-demo的VPC下创建一个名为vswitch-ack-demo的虚拟交换机，并选择使用该交换机。 通过单击 <b>创建虚拟交换机</b> 进行创建，详情请参见 <a href="#">使用交换机</a> 。
API Server访问	设置集群API Server是否可在公网访问，当您需要从公网远程管理集群时，需要配置弹性公网IP（EIP）。 示例：选中使用EIP暴露API Server。

5. 单击**下一步：节点池配置**，配置以下相关参数，未说明参数保留默认设置即可。

参数	说明
实例规格	为集群选配所使用的节点。为了保证集群的稳定性，建议的实例规格为：vCPU ≥ 4核，内存 ≥ 8 GiB。关于如何选型以及规格介绍，请参见 <a href="#">ECS选型</a> 和 <a href="#">实例规格族</a> 。 您可通过设置vCPU和内存的大小，或者直接搜索实例规格，选用该规格的节点。

参数	说明
数量	根据需要设置集群的节点数量。示例：2个，防止单点故障。
系统盘	选择节点所使用的系统盘。示例：ESSD云盘，40 GiB（最小规格）。
登录方式	选择登录节点的方式，并输入密码。示例：设置密码。

- 单击下一步：**组件配置**，所有组件使用默认配置。
- 单击下一步：**确认配置**，然后选中并阅读服务协议，单击**创建集群**。

 **说明** 集群的创建时间一般约为10分钟。创建完成后，在集群列表页面，可以看到新创建的集群。

### 步骤三：连接集群

本步骤指导您如何通过kubect l客户端或CloudShell快速连接到ACK集群。更多信息，请参见[通过kubect l工具连接集群](#)和[在CloudShell上通过kubect l管理Kubernetes集群](#)。

#### 方式一：通过kubect l客户端连接集群

- 登录[容器服务管理控制台](#)。
- 在控制台左侧导航栏中，单击**集群**。
- 在集群列表页面中，单击目标集群名称（即ACK-Demo）。
- 在**集群信息**页面，单击**连接信息**页签，复制**公网访问**页签下的内容（即集群的公网访问凭证）。
- 将复制的集群凭证内容粘贴至 $\$HOME/.kube$ 目录下的`config`文件中，保存并退出。

 **说明** 如果 $\$HOME/$ 目录下没有`.kube`目录和`config`文件，请自行创建。

- 执行kubect l命令验证集群的连通性。  
以查询命名空间为例，执行以下命令。

```
kubect l get namespace
```

预期输出：

NAME	STATUS	AGE
arms-prom	Active	4h39m
default	Active	4h39m
kube-node-lease	Active	4h39m
kube-public	Active	4h39m
kube-system	Active	4h39m

#### 方式二：通过CloudShell连接集群

- 登录[容器服务管理控制台](#)。
- 在控制台左侧导航栏中，单击**集群**。
- 在集群列表页面中，选择目标集群右侧操作列下的**更多 > 通过CloudShell管理集群**。  
等待数秒启动成功之后，您就可以在CloudShell界面通过kubect l命令来管理集群和应用。

## 步骤四：部署并公开应用

本步骤指导您如何通过kubectI在新创建的ACK集群中快速部署一个无状态应用（Deployment），并将其通过LoadBalancer类型服务（Service）公开。关于服务公开的详细操作，请参见[通过使用自动创建SLB的服务公开应用](#)。

1. 使用以下示例应用的YAML内容，创建名为`ack-cube.yaml`的文件。

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: ack-cube #应用名称。
  labels:
    app: ack-cube
spec:
  replicas: 2 #设置副本数量。
  selector:
    matchLabels:
      app: ack-cube #对应服务中Selector的值需要与其一致，才可以通过服务公开此应用。
  template:
    metadata:
      labels:
        app: ack-cube
    spec:
      containers:
        - name: ack-cube
          image: registry.cn-hangzhou.aliyuncs.com/acr-toolkit/ack-cube:1.0 #替换为您实际的
          镜像地址，格式为：<image_name:tags>。
          ports:
            - containerPort: 80 #需要在服务中暴露该端口。
          resources:
            limits: #设置资源限制。
              cpu: '1'
              memory: 1Gi
            requests: #设置所需资源
              cpu: 500m
              memory: 512Mi
```

2. 执行以下命令，部署Demo应用ack-cube。

```
kubectl apply -f ack-cube.yaml
```

3. 执行以下命令，确认示例应用状态正常。

```
kubectl get deployment ack-cube
```

预期输出：

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
ack-cube	2/2	2	2	96s

4. 使用以下示例服务的YAML内容，创建名为`ack-cube-svc.yaml`的文件。将 `selector` 修改为 `ack-cube`。`yaml`示例应用文件中 `matchLabels` 的值（本示例为 `app: ack-cube`），从而将该服务关联至后端应用。

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: ack-cube
    name: ack-cube-svc
    namespace: default
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: ack-cube # 需要与Deployment YAML文件中的matchLabels的值一致。
  type: LoadBalancer
```

5. 执行以下命令创建名为ack-cube-svc的服务，并通过其公开应用。  
ACK会自动创建一个公网SLB，并将其绑定至该服务。

```
kubectl apply -f ack-cube-svc.yaml
```

6. 执行以下命令确认LoadBalancer类型的服务创建成功。  
示例应用将通过EXTERNAL-IP字段的IP地址向公网公开。

```
kubectl get svc ack-cube-svc
```

预期输出：

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
ack-cube-svc	LoadBalancer	172.16.72.161	47.94.xx.xx	80:31547/TCP	32s

## 步骤五：测试应用

在浏览器地址栏输入该服务EXTERNAL-IP字段的IP地址，即可开始魔方游戏。

## 步骤六：监控应用

关于如何监控应用的运行情况，请参见[步骤五：监控应用](#)。

## 相关文档

- 为了保证应用能够动态调整所需容器资源，您可以配置容器水平伸缩（HPA）、定时容器水平伸缩（CronHPA）、容器垂直伸缩（HPA）等，详情请参见[弹性伸缩概述](#)。
- 除了通过服务（Service）公开应用，您还可以通过路由（Ingress）实现对应用的七层网络路由控制，详情请参见[创建Ingress路由](#)。
- 除了观测容器性能，您还可以观测集群基础设施、应用性能和用户业务，详情请参见[可观测性体系概述](#)。
- 为了避免产生不必要的费用，请及时清理不需要的集群，详情请参见[删除集群](#)。

## 4.更多入门教程

### 4.1. 快速创建Kubernetes托管版集群

本文以托管版Kubernetes集群为例，介绍如何使用控制台快速创建Kubernetes集群。

#### 前提条件

- 在使用容器服务Kubernetes版前，您需要完成授权操作。具体操作步骤，请参见[首次使用容器服务Kubernetes版](#)。
- 登录RAM 管理控制台开通相应服务。
- 开通弹性伸缩服务。

#### 背景信息

本示例将演示如何快速创建一个Kubernetes集群，部分配置采用默认或最简配置。

有关集群的使用限制和说明，请参见[创建Kubernetes托管版集群](#)。

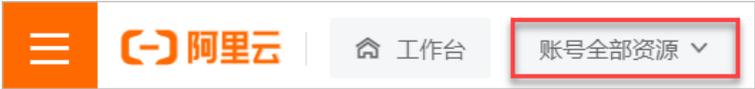
#### 操作步骤

1. 登录[容器服务管理控制台](#)。
2. 在控制台左侧导航栏中，单击[集群](#)。
3. 在[集群列表](#)页面中，单击页面右上角的[创建集群](#)。
4. 配置集群参数。

本文中大多数配置保留默认值，具体的配置如下。

i. 完成集群配置。

参数	描述
集群名称	填写集群的名称。   <b>说明</b> 集群名称应包含1~63个字符，可包含数字、汉字、英文字符或短划线 (-)。
集群规格	选择集群规格，支持标准版和Pro版。
地域	选择集群所在的地域。

参数	描述
账号全部资源组	<p>将鼠标悬浮于页面上方的账号全部资源，选择资源组。在控制台页面顶部选择的资源组可过滤出该资源组内的专有网络及对应的虚拟交换机。在创建集群时，只显示过滤的专有网络实例及专有网络对应的虚拟交换机实例。</p>  <p>本文默认选择账号全部资源。</p>
Kubernetes版本	使用默认的版本。
容器运行时	使用Docker运行时。有关Docker运行时、Containerd运行时、和安全沙箱运行时的区别，请参见 <a href="#">如何选择Docker运行时、Containerd运行时、或者安全沙箱运行时?</a> 。
专有网络	<p>设置集群的网络。</p> <p>设置集群的网络，您可以选择普通VPC和共享VPC。</p> <ul style="list-style-type: none"> <li>共享VPC：VPC的所有者账号（资源所有者）可以将其账号下的VPC内的交换机资源共享给其组织内的其他账号使用。</li> <li>普通VPC：不具备共享功能的VPC。</li> </ul> <p><b>说明</b> Kubernetes集群仅支持专有网络。您可以在已有VPC列表中选择所需的VPC。如果没有您需要的专有网络，可以通过单击创建专有网络进行创建，请参见<a href="#">创建和管理专有网络</a>。</p>
虚拟交换机	<p>设置虚拟交换机。</p> <p>您可以在已有虚拟交换机列表中，根据可用区选择交换机。如果没有您需要的交换机，可以通过单击创建虚拟交换机进行创建，请参见<a href="#">使用交换机</a>。</p>
网络插件	默认选择Flannel。有关Flannel和Terway的详情，请参见 <a href="#">使用Terway网络插件</a> 。

参数	描述
节点IP数量	<p>默认选择64。</p> <p>如果您选择的网络模式为Flannel，您需设置节点IP数量。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> 说明</p> <ul style="list-style-type: none"> <li>■ 节点IP数量是指可分配给一个节点的IP数量，建议保持默认值。</li> <li>■ 根据您所选择的专有网络和节点IP数量，ACK将为您推荐可用的Pod网络CIDR和服务CIDR，并给出相应配置下集群内可允许部署的主机数量以及每台主机可容纳的Pod数量。请您根据集群规模的实际需求，在推荐配置的基础上进行修改。</li> </ul> </div>
Pod网络CIDR	<p>网络插件选择Flannel时，需要配置Pod网络CIDR。</p> <p><b>Pod网络CIDR</b>指定Flannel网络插件需要配置Pod网络CIDR，网段不能和VPC及VPC已有Kubernetes集群使用的网段重复，创建成功后不能修改。而且Service地址段不能和Pod地址段重复，有关Kubernetes网络地址段规划的信息，请参见<a href="#">Kubernetes集群网络规划</a>。</p>
Service CIDR	<p>设置Service CIDR。您需要指定Service CIDR，网段不能与VPC及VPC内已有Kubernetes集群使用的网段重复，创建成功后不能修改。而且Service地址段也不能和Pod地址段重复，有关Kubernetes网络地址段规划的信息，请参见<a href="#">Kubernetes集群网络规划</a>。</p>
配置SNAT	<p>默认选中为专有网络配置SNAT网关。</p> <p>创建集群时，默认不开通公网。如果您选择的VPC不具备公网访问能力，选中为专有网络配置SNAT后，ACK将为您创建NAT网关并自动配置SNAT规则。</p>
API Server访问	<p>ACK默认为API Server创建一个内网SLB实例，您可修改SLB实例规格。更多信息，请参见<a href="#">实例规格</a>。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> 注意 删除默认创建的SLB实例将会导致无法访问API Server。</p> </div> <p>本文默认不选中使用EIP暴露API Server。</p> <p>您可设置是否开放使用EIP暴露API Server。API Server提供了各类资源对象（Pod，Service等）的增删改查及Watch等HTTP Rest接口。</p> <ul style="list-style-type: none"> <li>■ 如果选择开放，ACK会创建一个EIP，并挂载到SLB上。此时，Master节点的6443端口（对应API Server）暴露出来，您可以在外网通过kubconfig连接并操作集群。</li> <li>■ 如果选择不开放，则不会创建EIP，您只能在VPC内部用kubconfig连接并操作集群。</li> </ul>

参数	描述
RDS白名单	<p>设置RDS白名单。将节点IP添加到RDS实例的白名单中。</p> <p><b>说明</b> 允许白名单RDS访问Kubernetes集群，RDS必须在当前集群的VPC内。</p> <p>默认不设置。</p>
安全组	<p>支持选择<b>自动创建普通安全组</b>、<b>自动创建企业级安全组</b>、<b>选择已有安全组</b>。有关安全组的详细内容，请参见<a href="#">安全组概述</a>。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>只有白名单用户可以使用<b>选择已有安全组</b>功能。请<a href="#">提交工单</a>申请。</li> <li>指定已有安全组时，系统默认不会为安全组配置额外的访问规则，可能会导致访问异常，请自行管理安全组规则。关于如何管理安全组规则，请参见<a href="#">最小化集群访问规则</a>。</li> </ul> <p>本文默认选择<b>自动创建企业级安全组</b>。</p>
高级配置项	<p>本文默认不配置集群的高级参数。有关集群的高级配置项详情，请参见<a href="#">集群高级选项配置</a>。</p>

ii. 单击下一步：节点池配置，并完成节点池配置。

参数	描述
Worker实例	<p>设置是否使用已有ECS实例或者购买ECS实例。默认选择<b>新增实例</b>。</p>
节点池名称	<p>填写节点池名称。</p> <p><b>说明</b> 节点池名称为1~63个字符，可包含数字、汉字、英文字符或短划线（-）。</p>
付费类型	<p>默认选择<b>按量付费</b>，支持<b>按量付费</b>和<b>包年包月</b>两种节点付费类型。</p>
实例规格	<p>支持选择多个实例规格。详情请参见<a href="#">实例规格族</a>。</p>
已选规格	<p>呈现选中的规格。</p>
数量	<p>新增Worker实例（ECS实例）的数量。</p>

参数	描述
系统盘	默认选择高效云盘、120 GiB的存储空间。
挂载数据盘	默认不挂载。 支持ESSD云盘、SSD云盘和高效云盘。挂载数据盘时，支持云盘加密和开启云盘备份。节点的数据盘加密时只能使用默认的CMK。
操作系统	默认选择Alibaba Cloud Linux 2.1903。
登录方式	<ul style="list-style-type: none"> <li>■ 设置密钥。 <ul style="list-style-type: none"> <li>■ 密钥对：如您已经创建密钥对，在下拉列表中选择目标密钥对。</li> <li>■ 新建密钥对：此项用于您还未创建密钥对。创建密钥对，请参见<a href="#">创建SSH密钥对</a>。密钥对创建完毕后，设置该密钥对作为登录集群的凭据。</li> </ul> </li> <li>■ 设置密码。 <ul style="list-style-type: none"> <li>■ 登录密码：设置节点的登录密码。</li> <li>■ 确认密码：确认设置的节点登录密码。</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 密码为8~30个字符，且必须同时包含三项（大写字母、小写字母、数字和特殊符号），其中特殊字符不包括下划线（_）。</p> </div>
Worker节点高级配置项	本文默认不配置集群的高级参数。有关集群的高级配置项详情，请参见 <a href="#">集群高级选项配置</a> 。

## iii. 单击下一步：组件配置，并完成组件配置。

参数	描述
Ingress	默认选中安装Ingress组件，负载均衡类型选择公网。
存储插件	默认选择CSI。有关Flexvolume和CSI存储插件的详情，请参见 <a href="#">存储CSI概述</a> 。
监控插件	设置是否启用云监控插件。默认选中在ECS节点上安装云监控插件和使用Prometheus监控服务，前者用于在云监控控制台查看所创建ECS实例的监控信息。 本文默认不安装云监控插件。
日志服务	设置是否启用日志服务，您可使用已有Project或新建一个Project。默认选中使用日志服务。创建应用时，您可通过简单配置，快速使用日志服务，详情参见 <a href="#">通过日志服务采集Kubernetes容器日志</a> 。 默认选中安装node-problem-detector并创建事件中心，您可以选择在日志服务控制台中添加事件中心。更多信息，请参见 <a href="#">创建并使用K8s事件中心</a> 。 本文默认使用日志服务。
控制平面组件日志	选中开启，将从ACK控制层收集托管集群控制平面组件日志到您账号中的SLS日志服务的Log Project中。更多信息，请参见 <a href="#">收集托管集群控制平面组件日志</a> 。

## iv. 单击下一步：确认配置。

## v. 选中并阅读服务协议，然后单击创建集群。

 说明 一个包含多节点的Kubernetes集群的创建时间一般约为十分钟。

## 4.2. 使用镜像快速创建无状态Deployment应用

您可以使用镜像快速创建一个可公网访问的Nginx应用。

### 体验实验室

为了帮助您快速上手本教程，ACK为您提供了可以动手操作的交互式实验环境。如您需要体验，请访问[使用kubectl部署Web应用到k8s集群](#)。

### 前提条件

已创建一个ACK集群。具体操作，请参见[创建Kubernetes托管版集群](#)。

### 操作步骤

1. 登录[容器服务管理控制台](#)。

2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧操作列下的**详情**。
4. 在**集群管理**页左侧导航栏中，选择**工作负载 > 无状态**。
5. 在**无状态**页面中，单击**使用镜像创建**。
6. 在**应用基本信息配置**向导页面中，设置应用的基本信息。

在配置Deployment负载前，您需在页面上方设置应用所在的命名空间。本例设置命名空间为**default**。

参数	描述
应用名称	设置应用的名称。
副本数量	即应用包含的Pod数量，默认数量为2。
类型	定义资源对象的类型，可选择 <b>无状态</b> 、 <b>有状态</b> 、 <b>任务</b> 、 <b>定时任务</b> 、 <b>守护进程集</b> 。
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（Annotation）。
时区同步	容器与节点是否使用相同的时区。

 **说明** 本例中选择**无状态（Deployment）**类型。

7. 单击**下一步**进入**容器配置**页签。
8. 设置容器相关的配置。

 **说明** 在**容器配置**页签上方，单击**添加容器**为应用的Pod设置多个容器。

容器配置的配置项及描述如下。

- **基本配置**

应用基本信息    **容器配置**    高级配置

容器1    + 添加容器

**基本配置**

镜像名称:  [选择镜像](#)

镜像Tag:  [选择镜像Tag](#)

总是拉取镜像

[设置镜像密钥](#) ?

资源限制:

CPU  Core

内存  MiB Ephemeral-Storage

GiB

所需资源:

CPU  Core

内存  MiB Ephemeral-Storage

GiB

? 建议根据实际使用情况设置, 防止因资源不足导致应用不可用

容器启动项:  stdin  tty

特权容器:

Init Container

配置项	描述
镜像名称	您可以单击 <a href="#">选择镜像</a> , 在弹出的对话框中选择所需的镜像并单击确定, 本例中为Nginx。 您还可以填写私有镜像。填写的格式为 <code>domainname/namespace/image name:tag</code> 。

配置项	描述
镜像Tag	<ul style="list-style-type: none"> <li>您可以单击选择<b>镜像Tag</b>选择镜像的版本。若不指定，默认为最新版。</li> <li>ACK支持以下三种镜像拉取策略（imagePullPolicy）： <ul style="list-style-type: none"> <li><b>优先使用本地镜像（IfNotPresent）</b>：如果本地有该镜像（之前拉取过该镜像至宿主机中），则使用本地镜像，本地不存在时拉取镜像。</li> <li><b>总是拉取镜像（Always）</b>：表示每次部署或扩容都会从容器镜像服务重新拉取镜像，而不会从本地拉取镜像。</li> <li><b>仅使用本地镜像（Never）</b>：仅使用本地镜像。</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 10px 0;"> <p> <b>说明</b> 当您选中<b>镜像拉取策略</b>时，默认不设置<b>镜像拉取策略</b>。</p> </div> <ul style="list-style-type: none"> <li>单击设置<b>镜像密钥</b>，您可以实现免密拉取镜像。具体操作，请参见<a href="#">使用免密组件拉取容器镜像</a>。</li> </ul>
资源限制	可指定该应用所能使用的资源上限，包括CPU、内存和Ephemeral-Storage三种资源，防止占用过多资源。其中，CPU资源的单位为millicores，即一个核的千分之一；内存的单位为MiB；Ephemeral-Storage的单位为GiB。
所需资源	即为该应用预留资源额度，包括CPU和内存两种资源，即容器独占该资源，防止因资源不足而被其他服务或进程争夺资源，导致应用不可用。
容器启动项	<ul style="list-style-type: none"> <li>stdin：将控制台输入发送到容器。</li> <li>tty：将标准输入控制台作为容器的控制台输入。</li> </ul>
特权容器	<ul style="list-style-type: none"> <li>选择特权容器，则 <code>privileged=true</code>，开启特权模式。</li> <li>不选择特权容器，则 <code>privileged=false</code>，关闭特权模式。</li> </ul>
Init Container	勾选该项，表示创建一个Init Container。Init Container包含一些实用的工具，详情请参见 <a href="#">Init Containers</a> 。

○（可选）端口设置

设置容器的端口。

- 名称：设置容器端口名称。
- 容器端口：设置暴露的容器访问端口或端口名，端口号必须介于1~65535。
- 协议：支持TCP和UDP。

○（可选）环境变量

支持通过键值对的形式为Pod配置环境变量。用于给Pod添加环境标志或传递配置等，具体请参见[Pod variable](#)。

参数	描述
类型	<p>设置环境变量的类型，支持：</p> <ul style="list-style-type: none"> <li>自定义</li> <li>配置项</li> <li>保密字典</li> <li>变量/变量引用</li> <li>资源引用</li> </ul> <p>配置项、保密字典支持全部文件的引用，以保密字典为例。选择保密字典类型，只选择目标保密字典，则默认引用全部文件。</p>  <p>对应的YAML，也引用了整个密钥（Secret）。</p> <pre>envFrom: - secretRef:   name: test</pre>
变量名称	设置环境变量名称。
变量/变量引用	设置变量引用的值。

○（可选）健康检查

支持存活检查（liveness）、就绪检查（Readiness）和启动探测。存活检查用于检测何时重启容器；就绪检查确定容器是否已经就绪，且可以接受流量；启动探测用于检测容器内应用是否启动。关于健康检查的更多信息，请参见[Configure Liveness, Readiness and Startup Probes](#)。

请求类型	配置说明

请求类型	配置说明
HTTP请求	<p>即向容器发送一个HTTPget请求，支持的参数包括：</p> <ul style="list-style-type: none"><li>■ 协议：HTTP/HTTPS。</li><li>■ 路径：访问HTTP Server的路径。</li><li>■ 端口：容器暴露的访问端口或端口名，端口号必须介于1~65535。</li><li>■ HTTP头：即HTTP Headers，HTTP请求中自定义的请求头，HTTP允许重复的Header。支持键值对的配置方式。</li><li>■ 延迟探测时间（秒）：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为3秒。</li><li>■ 执行探测频率（秒）：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li><li>■ 超时时间（秒）：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li><li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查（liveness）必须是1。</li><li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li></ul>
TCP连接	<p>即向容器发送一个TCP Socket，kubelet将尝试在指定端口上打开容器的套接字。如果可以建立连接，容器被认为是健康的，如果不能就认为是失败的。支持的参数包括：</p> <ul style="list-style-type: none"><li>■ 端口：容器暴露的访问端口或端口名，端口号必须介于1~65535。</li><li>■ 延迟探测时间（秒）：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为15秒。</li><li>■ 执行探测频率（秒）：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li><li>■ 超时时间（秒）：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li><li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查（liveness）必须是1。</li><li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li></ul>

请求类型	配置说明
命令行	<p>通过在容器中执行探针检测命令，来检测容器的健康情况。支持的参数包括：</p> <ul style="list-style-type: none"> <li>■ 命令行：用于检测容器健康情况的探测命令。</li> <li>■ 延迟探测时间（秒）：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为5秒。</li> <li>■ 执行探测频率（秒）：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li> <li>■ 超时时间（秒）：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li> <li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查（liveness）必须是1。</li> <li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li> </ul>

○ 生命周期

您可以为容器的生命周期配置启动执行、启动后处理和停止前处理。具体参见[配置生命周期](#)。

- 启动执行：为容器设置预启动命令和参数。
- 启动后处理：为容器设置启动后的命令。
- 停止前处理：为容器设置预结束命令。



○ （可选）数据卷

支持增加本地存储和云存储声明（PVC）。

- 本地存储：支持主机目录（hostpath）、配置项（configmap）、保密字典（secret）和临时目录，将对应的挂载源挂载到容器路径中。更多信息参见[Volumes](#)。
- 云存储声明（PVC）：支持云存储。

本例中配置了一个云存储类型的数据卷声明disk-ssd，将其挂载到容器的/tmp路径下。



o (可选) 日志配置

配置日志服务，您可进行采集配置和自定义Tag设置。

**注意** 请确保已部署Kubernetes集群，并且在此集群上已安装日志插件。

配置项	描述
采集配置	日志库：即在日志服务中生成一个对应的Logstore，用于存储采集到的日志。
	容器内日志路径：支持stdout和文本日志。 <ul style="list-style-type: none"> <li>■ stdout：表示采集容器的标准输出日志。</li> <li>■ 文本日志：表示收集容器内指定路径的日志，本例中表示收集/var/log/nginx下所有的文本日志，也支持通配符的方式。</li> </ul>
自定义Tag	您还可设置自定义Tag，设置Tag后，会将该Tag一起采集到容器的日志输出中。自定义Tag可帮助您给容器日志打上Tag，方便进行日志统计和过滤等分析操作。

9. 完成容器配置后，单击下一步。

10. (可选) 进行高级配置。

o 访问设置

**说明**

针对应用的通信需求，您可灵活进行访问设置：

- 内部应用：对于只在集群内部工作的应用，您可根据需要创建ClusterIP或NodePort类型的服务，来进行内部通信。
- 外部应用：对于需要暴露到公网的应用，您可以采用两种方式进行访问设置。
  - 创建LoadBalancer类型的服务：使用阿里云提供的负载均衡服务SLB (Server Load Balancer)，该服务提供公网访问能力。
  - 创建路由 (Ingress)：通过路由 (Ingress) 提供公网访问能力，详情请参见[Ingress](#)。

您可以设置暴露后端Pod的方式。本例中选择ClusterIP服务和路由 (Ingress)，构建一个公网可访问的Nginx应用。

配置项	描述
-----	----

配置项	描述
服务 (Service)	在服务 (Service) 右侧，单击 <b>创建</b> 设置创建服务配置项。配置项的详情，请参见 <a href="#">管理服务</a> 。本例中服务类型选择ClusterIP。
路由 (Ingress)	在路由 (Ingress) 右侧，单击 <b>创建</b> 。在弹出的对话框中，为后端Pod配置路由规则。详细的路由配置信息，请参见 <a href="#">路由配置说明</a> 。  <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"><p><b>说明</b> 通过镜像创建应用时，您仅能为一个服务创建路由 (Ingress)。本例中使用一个虚拟主机名称作为测试域名，您需要在hosts中添加一条记录。在实际工作场景中，请使用备案域名。</p><pre>47.100.XX.XX    foo.bar.com    #即ingress的IP</pre></div>

在访问设置区域中，您可看到创建完毕的服务和路由，您可单击**变更**和**删除**进行二次配置。

o 伸缩配置

在伸缩配置区域，配置是否开启**指标伸缩**和**定时伸缩**，从而满足应用在不同负载下的需求。

The screenshot shows a configuration panel for scaling. It is divided into two main sections: '指标伸缩' (Metric Scaling) and '定时伸缩' (Scheduled Scaling).  
Under '指标伸缩', there is a checkbox for '开启' (Enable) which is checked. Below it, a dropdown menu is set to 'CPU使用量' (CPU Usage). The trigger condition is set to '使用量 70 %' (Usage 70%). There are two input fields for replica counts: '最大副本数: 10' (Maximum replicas: 10) with a range of 2-100, and '最小副本数: 1' (Minimum replicas: 1) with a range of 1-100.  
Under '定时伸缩', there is a checkbox for '开启' (Enable) which is checked. There is a '+ 添加任务' (Add task) button. Below it, there are three required fields: '定时任务名称:' (Scheduled task name: 请输入名称), '目标副本数:' (Target replica count: empty), and '调度周期:' (Scheduling cycle: 按天 selected). The scheduling cycle options are '按天' (Daily), '按星期' (Weekly), '按月' (Monthly), and 'Cron表达式' (Cron expression). Below these, there are dropdowns for '每 1' (Every 1) and '分钟' (Minutes), followed by '执行一次' (Execute once).

- 容器服务支持容器组的指标伸缩，即根据容器CPU和内存资源占用情况自动调整容器组数量。

 **说明** 若要启用自动伸缩，您必须为容器设置所需资源，否则容器自动伸缩无法生效。

配置项	描述
指标	支持CPU和内存，需要和设置的所需资源类型相同。
触发条件	资源使用率的百分比，超过该使用量，容器开始扩容。
最大副本数量	该应用可扩容的容器数量上限。
最小副本数量	该应用可缩容的容器数量下限。

- 容器服务支持容器组的定时伸缩，即定时地对容器服务Kubernetes集群进行扩缩容。关于定时伸缩的详细说明，请参见[创建定时伸缩CronHPA任务](#)。

#### ○ 调度设置

您可设置升级方式、节点亲和性、应用亲和性、应用反亲和性和调度容忍，详情请参见[Affinity and anti-affinity](#)。

 **说明** 亲和性调度依赖节点标签和Pod标签。您可使用内置的标签进行调度；也可预先为节点、Pod配置相关的标签。

配置项	描述
升级方式	升级方式包括滚动升级（Rollingupdate）和替换升级（Recreate），详情请参见 <a href="#">Deployments</a> 。
节点亲和性	<p>设置<b>节点亲和性</b>，通过Worker节点的Label标签进行设置。</p> <p>节点调度支持硬约束和软约束（Required/Preferred），以及丰富的匹配表达式（In, NotIn, Exists, DoesNotExist, Gt, and Lt）：</p> <ul style="list-style-type: none"> <li>■ <b>必须满足</b>，即硬约束，一定要满足，对 <code>requiredDuringSchedulingIgnoredDuringExecution</code>，效果与 <code>No</code> <code>deSelector</code> 相同。本例中Pod只能调度到具有对应标签的Worker节点。您可以定义多条硬约束规则，但只需满足其中一条。</li> <li>■ <b>尽量满足</b>，即软约束，不一定满足，对 <code>preferredDuringSchedulingIgnoredDuringExecution</code>。本例中，调度会尽量不调度Pod到具有对应标签的Node节点。您还可为软约束规则设定权重，具体调度时，若存在多个符合条件的节点，权重最大的节点会被优先调度。您可定义多条软约束规则，但必须满足全部约束，才会进行调度。</li> </ul>

配置项	描述
应用亲和性	<p>决定应用的Pod可以和哪些Pod部署在同一拓扑域。例如，对于相互通信的服务，可通过应用亲和性调度，将其部署到同一拓扑域（如同一个主机）中，减少它们之间的网络延迟。</p> <p>根据节点上运行的Pod的标签（Label）来进行调度，支持硬约束和软约束，匹配的表达式有：<code>In, NotIn, Exists, DoesNotExist</code>。</p> <ul style="list-style-type: none"> <li>■ <b>必须满足</b>，即硬约束，一定要满足，对应<code>requiredDuringSchedulingIgnoredDuringExecution</code>，Pod的亲和性调度必须要满足后续定义的约束条件。 <ul style="list-style-type: none"> <li>■ <b>命名空间</b>：该策略是依据Pod的Label进行调度，所以会受到命名空间的约束。</li> <li>■ <b>拓扑域</b>：即<code>topologyKey</code>，指定调度时作用域，这是通过Node节点的标签来实现的，例如指定为<code>kubernetes.io/hostname</code>，那就是以Node节点为区分范围；如果指定为<code>beta.kubernetes.io/os</code>，则以Node节点的操作系统类型来区分。</li> <li>■ <b>选择器</b>：单击选择器右侧的加号按钮，您可添加多条硬约束规则。</li> <li>■ <b>查看应用列表</b>：单击<b>应用列表</b>，弹出对话框，您可在此查看各命名空间下的应用，并可将应用的标签导入到亲和性配置页面。</li> <li>■ <b>硬约束条件</b>：设置已有应用的标签、操作符和标签值。本例中，表示将待创建的应用调度到该主机上，该主机运行的已有应用具有<code>app:nginx</code> 标签。</li> </ul> </li> <li>■ <b>尽量满足</b>，即软约束，不一定满足，对应<code>preferredDuringSchedulingIgnoredDuringExecution</code>。Pod的亲和性调度会尽量满足后续定义的约束条件。对于软约束规则，您可配置每条规则的权重，其他配置规则与硬约束规则相同。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 权重：设置一条软约束规则的权重，介于1~100，通过算法计算满足软约束规则的节点的权重，将Pod调度到权重最大的节点上。</p> </div>
应用反亲和性	<p>决定应用的Pod不与哪些Pod部署在同一拓扑域。应用反亲和性调度的场景包括：</p> <ul style="list-style-type: none"> <li>■ 将一个服务的Pod分散部署到不同的拓扑域（例如不同主机）中，提高服务本身的稳定性。</li> <li>■ 给予Pod一个节点的独占访问权限来保证资源隔离，保证不会有其它Pod来分享节点资源。</li> <li>■ 把可能会相互影响的服务的Pod分散在不同的主机上。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 应用反亲和性调度的设置方式与亲和性调度相同，但是相同的调度规则代表的意义不同，请根据使用场景进行选择。</p> </div>
调度容忍	容忍被应用于Pod，允许这个Pod被调度到相对应的污点上。

配置项	描述
调度到虚拟节点	设置是否调度到虚拟节点。如果您集群中没有虚拟节点，则无法设置该配置项。

- 标签和注释
  - Pod标签：为该Pod添加一个标签，标识该应用。
  - Pod注解：为该Pod添加一个注解（Annotation）。

11. 单击**创建**。

12. 创建成功后，默认进入创建完成页面，会列出应用包含的对象，您可以单击**查看应用详情**进行查看。



默认进入新建的nginx-deployment的详情页面。

**说明** 您也可以通过以下操作创建路由与服务。在访问方式页签：

- 单击**服务（Service）**右侧的**创建**，也可以进行服务创建，请参见**管理服务**。
- 单击**路由（Ingress）**右侧的**创建**，进行路由的创建，请参见**创建路由**。

13. 在集群管理页左侧导航栏中，选择**网络 > 路由**，可以看到路由列表下出现一条规则。



14. 在浏览器中访问路由测试域名，您可访问Nginx欢迎页。



## 4.3. 创建有状态工作负载StatefulSet

阿里云容器服务Kubernetes版ACK（Container Service for Kubernetes）集群支持通过界面创建StatefulSet类型的应用，满足您快速创建有状态应用的需求。本例中将创建一个Nginx的有状态应用，并演示StatefulSet应用的特性。

## 体验实验室

为了帮助您快速上手本教程，ACK为您提供了可以动手操作的交互式实验环境。如您需要体验，请访问[使用kubectI进行StatefulSet应用部署](#)。

## 前提条件

使用镜像创建有状态StatefulSet应用前，请确保您已经进行以下操作：

- [创建Kubernetes托管版集群](#)
- [创建持久化存储卷声明](#)
- [通过kubectI工具连接集群](#)

## 背景信息

StatefulSet包括以下特性：

场景	说明
Pod一致性	包含次序（启动、停止次序）、网络一致性。此一致性与Pod相关，与被调度到哪个Node节点无关。
稳定的持久化存储	通过VolumeClaimTemplate为每个Pod创建一个PV。删除、减少副本，不会删除相关的卷。
稳定的网络标志	Pod的 <code>hostname</code> 模式为： <code>(statefulset名称) - (序号)</code> 。
稳定的次序	对于N个副本的StatefulSet，每个Pod都在 [0, N) 的范围内分配一个数字序号，且是唯一的。

## 操作步骤

1. 登录[容器服务管理控制台](#)。
2. 在控制台左侧导航栏中，单击[集群](#)。
3. 在[集群列表](#)页面中，单击目标集群名称或者目标集群右侧操作列下的[详情](#)。
4. 在[集群管理](#)页左侧导航栏中，选择[工作负载](#) > [有状态](#)。
5. 在[有状态](#)页面中，单击[使用镜像创建](#)。
6. 在[应用基本信息](#)页签中，设置应用的基本信息。

本例中选择[有状态](#)类型，创建StatefulSet类型的应用。

配置项	描述
应用名称	设置应用的名称。
命名空间	设置应用部署所处的命名空间，默认使用default命名空间。
副本数量	应用包含的Pod数量。

配置项	描述
类型	定义资源对象的类型，可选择 <b>无状态</b> 、 <b>有状态</b> 、 <b>任务</b> 、 <b>定时</b> 、 <b>定时任务</b> 、 <b>守护进程集</b> 。
标签	为该应用添加一个标签，标识该应用。
注解	为该应用添加一个注解（Annotation）。
时区同步	容器与节点是否使用相同的时区。

7. 单击**下一步**进入**容器配置**页签。

8. 设置容器相关的配置。

 **说明** 在容器配置页签上方，单击**添加容器**为应用的Pod设置多个容器。

容器配置的配置项及描述如下。

#### o 基本配置

配置项	描述
镜像名称	<ul style="list-style-type: none"> <li>您可以单击<b>选择镜像</b>，选择所需的镜像并单击<b>确定</b>。本例中为官方Nginx镜像，可以在<b>搜索</b>页签中，选择<b>Docker</b>镜像，然后在文本框中输入<code>nginx</code>进行搜索。</li> <li>容器镜像服务：在<b>容器镜像服务</b>页签中，可以选择托管在容器镜像服务ACR中的镜像。需要选择镜像所属地域，以及镜像服务实例。关于ACR的更多信息，请参见<a href="#">什么是容器镜像服务ACR</a>。</li> </ul> <p> <b>说明</b> 您可以在<b>容器镜像服务</b>页签，根据名称搜索托管在容器镜像服务ACR中的镜像。</p> <ul style="list-style-type: none"> <li>Docker官方镜像：在<b>Docker官方镜像</b>页签中，可以选择Docker官方提供的容器镜像。</li> <li>用户收藏：在<b>用户收藏</b>页签中，可以选择您收藏的Docker官方镜像。</li> <li>搜索：在<b>搜索</b>页签中，可以分别在ACR的特定地域和Docker官方镜像中搜索容器镜像。</li> <li>您还可以填写私有镜像。填写的格式为 <code>domainname/namespace/imageName:tag</code>。</li> </ul>

配置项	描述
镜像Tag	<ul style="list-style-type: none"> <li>您可以单击选择<b>镜像Tag</b>选择镜像的版本。若不指定，默认为最新版。</li> <li>ACK支持以下三种镜像拉取策略（imagePullPolicy）：                             <ul style="list-style-type: none"> <li><b>优先使用本地镜像（IfNotPresent）</b>：如果本地有该镜像（之前拉取过该镜像至宿主机中），则使用本地镜像，本地不存在时拉取镜像。</li> <li><b>总是拉取镜像（Always）</b>：表示每次部署或扩容都会从容器镜像服务重新拉取镜像，而不会从本地拉取镜像。</li> <li><b>仅使用本地镜像（Never）</b>：仅使用本地镜像。</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 10px 0;"> <p> <b>说明</b> 当您选中<b>镜像拉取策略</b>时，默认不设置<b>镜像拉取策略</b>。</p> </div> <ul style="list-style-type: none"> <li>单击设置<b>镜像密钥</b>，您可以实现免密拉取镜像。具体操作，请参见<a href="#">使用免密组件拉取容器镜像</a>。</li> </ul>
资源限制	可指定该应用所能使用的资源上限，包括CPU、内存和Ephemeral-storage三种资源，防止占用过多资源。
所需资源	即为该应用预留资源额度，包括CPU和内存两种资源，即容器独占该资源，防止因资源不足而被其他服务或进程争夺资源，导致应用不可用。
容器启动项	<ul style="list-style-type: none"> <li>stdin：表示为该容器开启标准输入。</li> <li>tty：表示为该容器分配一个虚拟终端，以便于向容器发送信号。</li> </ul> <p>通常这两个选项是一起使用的，表示将终端（tty）绑定到容器的标准输入（stdin）上。例如，一个交互式的程序从用户获取标准输入，并显示到终端中。</p>
特权容器	<ul style="list-style-type: none"> <li>选择特权容器，则 <code>privileged=true</code>，开启特权模式。</li> <li>不选择特权容器，则 <code>privileged=false</code>，关闭特权模式。</li> </ul>
Init Container	选中该项，表示创建一个Init Container。Init Container包含一些实用的工具，详情请参见 <a href="#">Init Containers</a> 。

○ （可选）端口设置

设置容器的端口。

- 名称：设置容器端口名称。
- 容器端口：设置暴露的容器访问端口或端口名，端口号必须介于1~65535。
- 协议：支持TCP和UDP。

○ （可选）环境变量

支持通过键值对的形式为Pod配置环境变量。用于给Pod添加环境标志或传递配置等，具体请参见[Pod variable](#)。

- 类型：设置环境变量的类型，支持自定义、配置项、保密字典、变量/变量引用及资源引用。配置项、保密字典支持全部文件的引用，以保密字典为例。

选择保密字典，只选择目标保密字典，则默认全部文件引用。

对应的yaml，则引用了整个Secret。

```
envFrom:
- secretRef:
  name: test
```

- 变量名称：设置环境变量名称。
  - 变量/变量引用：设置变量引用的值。
- (可选) 健康检查

支持存活检查 (Liveness)、就绪检查 (Readiness) 和启动探测。存活检查用于检测何时重启容器；就绪检查确定容器是否已经就绪，且可以接受流量；启动探测用于检测何时启动容器。

说明 仅Kubernetes集群1.18及之后版本支持启动探测。

关于健康检查的更多信息，请参见[Configure Liveness, Readiness and Startup Probes](#)。

请求类型	配置说明
HTTP请求	<p>即向容器发送一个HTTPget请求，支持的参数包括：</p> <ul style="list-style-type: none"> <li>■ 协议：HTTP/HTTPS。</li> <li>■ 路径：访问HTTP server的路径。</li> <li>■ 端口：容器暴露的访问端口或端口名，端口号必须介于1~65535。</li> <li>■ HTTP头：即HTTP Headers，HTTP请求中自定义的请求头，HTTP允许重复的Header。支持键值对的配置方式。</li> <li>■ 延迟探测时间 (秒)：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为3秒。</li> <li>■ 执行探测频率 (秒)：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li> <li>■ 超时时间 (秒)：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li> <li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查 (Liveness) 必须是1。</li> <li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li> </ul>

请求类型	配置说明
TCP连接	<p>即向容器发送一个TCP Socket，kubelet将尝试在指定端口上打开容器的套接字。如果可以建立连接，容器被认为是健康的，如果不能就认为是失败的。支持的参数包括：</p> <ul style="list-style-type: none"> <li>■ 端口：容器暴露的访问端口或端口名，端口号必须介于1~65535。</li> <li>■ 延迟探测时间（秒）：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为15秒。</li> <li>■ 执行探测频率（秒）：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li> <li>■ 超时时间（秒）：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li> <li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查（Liveness）必须是1。</li> <li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li> </ul>
命令行	<p>通过在容器中执行探针检测命令，来检测容器的健康情况。支持的参数包括：</p> <ul style="list-style-type: none"> <li>■ 命令行：用于检测容器健康情况的探测命令。</li> <li>■ 延迟探测时间（秒）：即initialDelaySeconds，容器启动后第一次执行探测时需要等待多少秒，默认为5秒。</li> <li>■ 执行探测频率（秒）：即periodSeconds，指执行探测的时间间隔，默认为10秒，最小为1秒。</li> <li>■ 超时时间（秒）：即timeoutSeconds，探测超时时间。默认1秒，最小1秒。</li> <li>■ 健康阈值：探测失败后，最少连续探测成功多少次才被认定为成功。默认是1，最小值是1。对于存活检查（Liveness）必须是1。</li> <li>■ 不健康阈值：探测成功后，最少连续探测失败多少次才被认定为失败。默认是3，最小值是1。</li> </ul>

○ 生命周期

您可以为容器的生命周期配置启动执行、启动后处理和停止前处理。具体参见[配置生命周期](#)。

- 启动执行：为容器设置预启动命令和参数。
- 启动后处理：为容器设置启动后的命令。
- 停止前处理：为容器设置预结束命令。



o (可选) 数据卷

支持增加本地存储、云存储声明 (PVC) 或增加存储声明模板。

- **本地存储:** 支持主机目录 (Hostpath)、配置项 (ConfigMap)、保密字典 (Secret) 和临时目录, 将对应的挂载源挂载到容器路径中。更多信息, 请参见[Volumes](#)。
- **增加云存储声明 (PersistentVolumeClaim):** 支持云存储。
- **增加存储声明模板:** 支持通过VolumeClaimTemplate为每个Pod创建PVC及PV。关于如何通过VolumeClaimTemplates自动创建PVC及PV, 请参见[创建StatefulSet服务](#)。

**? 说明**

- 由于配置存储声明模板后不可编辑, 因此不推荐使用。
- 单击**增加存储声明模板**后, 在编辑对话框的模板中输入存储声明模板, 例如:

```

metadata:
  name: www
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
  
```

本例中配置了一个云存储类型的数据卷声明disk-ssd，将其挂载到容器的/tmp路径下。



o (可选) 日志配置

配置日志服务，您可进行采集配置和自定义Tag设置。

**注意** 请确保已部署Kubernetes集群，并且在此集群上已安装日志插件。

配置项	描述
采集配置	日志库：即在日志服务中生成一个对应的Logstore，用于存储采集到的日志。
	容器内日志路径：支持stdout和文本日志。 <ul style="list-style-type: none"> <li>■ stdout：表示采集容器的标准输出日志。</li> <li>■ 文本日志：表示收集容器内指定路径的日志，本例中表示收集/var/log/nginx下所有的文本日志，也支持通配符的方式。</li> </ul>
自定义Tag	您还可设置自定义Tag，设置Tag后，会将该Tag一起采集到容器的日志输出中。自定义Tag可帮助您给容器日志打上Tag，方便进行日志统计和过滤等分析操作。

9. 完成容器配置后，单击下一步。

10. (可选) 进行高级配置。

o 访问设置

说明

针对应用的通信需求，您可灵活进行访问设置：

- 内部应用：对于只在集群内部工作的应用，您可以根据需要创建ClusterIP或NodePort类型的服务，来进行内部通信。
- 外部应用：对于需要暴露到公网的应用，您可以采用两种方式进行访问设置。
  - 创建LoadBalancer类型的服务：使用阿里云提供的负载均衡服务SLB（Server Load Balancer），该服务提供公网访问能力。
  - 创建路由（Ingress）：通过路由（Ingress）提供公网访问能力，详情请参见[Ingress](#)。

您可以设置暴露后端Pod的方式。本例中选择ClusterIP服务和路由（Ingress），构建一个公网可访问的Nginx应用。

配置项	描述
服务（Service）	在服务（Service）右侧，单击创建设置创建服务的配置项。配置项的详情，请参见 <a href="#">管理服务</a> 。本例中服务类型选择ClusterIP。
路由（Ingress）	<p>在路由（Ingress）右侧，单击创建。在弹出的对话框中，为后端Pod配置路由规则。详细的路由配置信息，请参见<a href="#">创建Ingress</a>。</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>说明 通过镜像创建应用时，您仅能为一个服务创建路由（Ingress）。本例中使用一个虚拟主机名称作为测试域名，您需要在Hosts中添加一条记录。在实际工作场景中，请使用备案域名。</p> <pre>101.37.XX.XX    foo.bar.com    #即ingress的IP</pre> </div>

在访问设置区域中，您可看到创建完毕的服务和路由，您可单击变更和删除进行二次配置。

伸缩配置

在伸缩配置区域，配置是否开启指标伸缩和定时伸缩，从而满足应用在不同负载下的需求。

- 容器服务支持容器组的指标伸缩，即根据容器CPU和内存资源占用情况自动调整容器组数量。

**说明** 若要启用自动伸缩，您必须为容器设置所需资源，否则容器自动伸缩无法生效。

配置项	描述
指标	支持CPU和内存，需要和设置的所需资源类型相同。
触发条件	资源使用率的百分比，超过该使用量，容器开始扩容。
最大副本数量	该应用可扩容的容器数量上限。
最小副本数量	该应用可缩容的容器数量下限。

- 容器服务支持容器组的定时伸缩，即定时地对容器服务Kubernetes集群进行扩缩容。关于定时伸缩的详细说明，请参见[创建定时伸缩CronHPA任务](#)。

○ 调度设置

您可设置升级方式、节点亲和性、应用亲和性和应用非亲和性，详情请参见[Affinity and anti-affinity](#)。

**说明** 亲和性调度依赖节点标签和Pod标签。您可使用内置的标签进行调度；也可预先为节点、Pod配置相关的标签。

配置项	描述
升级方式	升级方式包括滚动升级（Rollingupdate）和替换升级（Recreate），详细请参见 <a href="#">Deployments</a> 。

配置项	描述
节点亲和性	<p>设置<b>节点亲和性</b>，通过Worker节点的Label标签进行设置。</p> <p>节点调度支持硬约束和软约束（Required/Preferred），以及丰富的匹配表达式（In, NotIn, Exists, DoesNotExist, Gt, and Lt）：</p> <ul style="list-style-type: none"> <li>▪ <b>必须满足</b>，即硬约束，一定要满足，对应requiredDuringSchedulingIgnoredDuringExecution，效果与 <code>NodeSelector</code> 相同。本例中Pod只能调度到具有对应标签的Worker节点。您可以定义多条硬约束规则，但只需满足其中一条。</li> <li>▪ <b>尽量满足</b>，即软约束，不一定满足，对应preferredDuringSchedulingIgnoredDuringExecution。本例中，调度会尽量不调度Pod到具有对应标签的Node节点。您还可为软约束规则设定权重，具体调度时，若存在多个符合条件的节点，权重最大的节点会被优先调度。您可定义多条软约束规则，但必须满足全部约束，才会进行调度。</li> </ul>
应用亲和性	<p>决定应用的Pod可以和哪些Pod部署在同一拓扑域。例如，对于相互通信的服务，可通过应用亲和性调度，将其部署到同一拓扑域（如同一个主机）中，减少它们之间的网络延迟。</p> <p>根据节点上运行的Pod的标签（Label）来进行调度，支持硬约束和软约束，匹配的表达式有：<code>In, NotIn, Exists, DoesNotExist</code>。</p> <ul style="list-style-type: none"> <li>▪ <b>必须满足</b>，即硬约束，一定要满足，对应requiredDuringSchedulingIgnoredDuringExecution，Pod的亲和性调度必须要满足后续定义的约束条件。 <ul style="list-style-type: none"> <li>▪ <b>命名空间</b>：该策略是依据Pod的Label进行调度，所以会受到命名空间的约束。</li> <li>▪ <b>拓扑域</b>：即TopologyKey，指定调度时作用域，这是通过Node节点的标签来实现的，例如指定为 <code>kubernetes.io/hostname</code>，那就是以Node节点为区分范围；如果指定为 <code>beta.kubernetes.io/os</code>，则以Node节点的操作系统类型来区分。</li> <li>▪ <b>选择器</b>：单击选择器右侧的加号按钮，您可添加多条硬约束规则。</li> <li>▪ <b>查看应用列表</b>：单击应用列表，弹出对话框，您可在此查看各命名空间下的应用，并可应用的标签导入到亲和性配置页面。</li> <li>▪ <b>硬约束条件</b>：设置已有应用的标签、操作符和标签值。本例中，表示将待创建的应用调度到该主机上，该主机运行的已有应用具有 <code>app:nginx</code> 标签。</li> </ul> </li> <li>▪ <b>尽量满足</b>，即软约束，不一定满足，对应preferredDuringSchedulingIgnoredDuringExecution。Pod的亲和性调度会尽量满足后续定义的约束条件。对于软约束规则，您可配置每条规则的权重，其他配置规则与硬约束规则相同。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 权重：设置一条软约束规则的权重，介于1~100，通过算法计算满足软约束规则的节点的权重，将Pod调度到权重最大的节点上。</p> </div>

配置项	描述
应用非亲和性	<p>决定应用的Pod不与哪些Pod部署在同一拓扑域。应用非亲和性调度的场景包括：</p> <ul style="list-style-type: none"> <li>■ 将一个服务的Pod分散部署到不同的拓扑域（如不同主机）中，提高服务本身的稳定性。</li> <li>■ 给予Pod一个节点的独占访问权限来保证资源隔离，保证不会有其它Pod来分享节点资源。</li> <li>■ 把可能会相互影响的服务的Pod分散在不同的主机上。</li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 应用非亲和性调度的设置方式与亲和性调度相同，但是相同的调度规则代表的意义不同，请根据使用场景进行选择。</p> </div>
调度容忍	容忍被应用于Pod，允许这个Pod被调度到相对应的污点上。
调度到虚拟节点	设置是否调度到虚拟节点。如果您集群中没有虚拟节点，则无法设置该配置项。

o 标签和注释

- Pod标签：为该Pod添加一个标签，标识该应用。
- Pod注解：为该Pod添加一个注解（Annotation）。

11. 单击创建。

12. 查看创建的应用。

- 创建成功后，默认进入创建完成页面，会列出应用包含的对象，您可以单击[查看应用详情](#)进行查看。  
默认进入有状态副本集详情页面。
- 然后单击左上角[返回列表](#)，进入有状态副本集列表页面，查看创建的StatefulSet应用。

## 相关操作

在有状态页面单击目标应用名称或目标应用操作列下的[详情](#)。

 **说明** 在有状态列表中单击标签字段，输入对应应用标签的key与 `value`，单击确定后可过滤列表中的应用。

在应用详情页面您可以[编辑](#)、[伸缩](#)、[查看YAML](#)、[重新部署](#)、[刷新应用](#)。

- **编辑**：在应用详情页面单击[编辑](#)，您可以修改应用信息。
- **伸缩**：在应用详情页面单击[伸缩](#)，您可以修改所需容器组数量。

以创建的Nginx应用为例，验证服务伸缩性。

- 选择目标Nginx应用，单击右侧[伸缩](#)。
- 在弹出的对话框中，将容器组数量设置为3，您可发现扩容时，扩容容器组的排序依次递增；反之，进行缩容时，先按Pod次序从高到低进行缩容。这体现StatefulSet中Pod的次序稳定性。
- 单击左侧导航栏中的[存储卷 > 存储声明](#)，您可发现，随着应用扩容，会随着Pod创建新的云存储卷。缩容后，已创建的PV和PVC不会删除。

- 查看YAML：在应用详情页面单击查看YAML，您可以更新、下载、另存为YAML文件。
- 重新部署：在应用详情页面单击重新部署，您可以重新部署应用。
- 刷新：在应用详情页面单击刷新，您可以刷新应用。

## 后续步骤

连接到Master节点，执行以下命令，验证持久化存储特性。

1. 执行以下命令在云盘中创建临时文件。

```
kubectl exec nginx-1 ls /tmp          #列出该目录下的文件lost+found。
kubectl exec nginx-1 touch /tmp/statefulset  #增加一个临时文件statefulset。
kubectl exec nginx-1 ls /tmp
```

预期输出：

```
lost+found
statefulset
```

2. 执行以下命令删除Pod，验证数据持久性。

```
kubectl delete pod nginx-1
```

预期输出：

```
pod"nginx-1" deleted
```

3. 过一段时间，待Pod自动重启后，验证数据持久性，证明StatefulSet应用的高可用性。

```
kubectl exec nginx-1 ls /tmp  #数据持久化存储lost+found。
```

预期输出：

```
statefulset
```

想要了解更多信息，参见[Kubernetes有状态服务-StatefulSet使用最佳实践](#)。

## 4.4. 通过Helm CLI一键部署WordPress

本文介绍如何通过Helm CLI部署WordPress应用。

### 体验实验室

为了帮助您快速上手本教程，ACK为您提供可以动手操作的交互式实验环境。如您需要体验，请访问[通过HelmCLI部署wordpress](#)。

### 前提条件

- 已创建一个ACK集群。具体操作，请参见[创建Kubernetes托管版集群](#)。
- 已通过Kubectl连接到Kubernetes托管版集群。具体操作，请参见[通过Kubectl连接Kubernetes集群](#)。

### 安装配置阿里云CLI和kubectl

- 安装Helm。您可以选择以下三种方式：
  - 通过源码或二进制方式安装，请参见[Installing Helm](#)。

- 通过执行以下脚本安装。

```
curl -sSL https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
```

- 通过包管理器工具安装。
  - OS X安装：请参见[Homebrew](#)，安装包管理工具后，执行以下命令安装Helm。

```
brew install helm
```

- Windows 安装：请参考[Chocolatey](#)，安装包管理工具后，执行以下命令安装Helm。

```
choco install kubernetes-helm
```

- 配置Helm Repo。阿里云为中国的开发者提供了Helm Repo的镜像，开发者可以更加方便地使用Helm。
  - 执行以下命令配置阿里云发布的稳定版Repo。

```
helm repo add stable https://apphub.aliyuncs.com/stable/
```

 **说明** 如果您是海外用户，可以继续使用官方版Repo。执行以下命令进行配置。

```
helm repo add stable https://kubernetes-charts.storage.googleapis.com/
```

- 执行以下命令搜索Repo。

```
helm search repo stable
```

- 安装和设置kubectl客户端，详情请参见[Install and Set Up kubectl](#)。

## 部署WordPress

1. 执行以下命令，添加WordPress官方Repo地址。

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. 执行以下命令部署WordPress。

```
helm install wordpress bitnami/wordpress \
  --set mariadb.master.persistence.enabled=true \
  --set mariadb.master.persistence.storageClass=alicloud-disk-ssd \
  --set mariadb.master.persistence.size=20Gi \
  --set persistence.enabled=false
```

 **说明**

- 为了实现数据的持久化，mariadb使用持久化存储卷保存数据。
- 持久化存储选用阿里云SSD云盘，大小为20 GiB，并通过配置StorageClassName自动化创建。
- WordPress不需要进行数据持久化，数据卷persistence.enabled可配置为false。

系统输出类似以下结果，表示已经部署完成。

```
NAME: wordpress
LAST DEPLOYED: Tue Sep  8 10:37:05 2020
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
** Please be patient while the chart is being deployed **
Your WordPress site can be accessed through the following DNS name from within your cluster:
    wordpress.default.svc.cluster.local (port 80)
To access your WordPress site from outside the cluster follow the steps below:
1. Get the WordPress URL by running these commands:
    NOTE: It may take a few minutes for the LoadBalancer IP to be available.
        Watch the status with: 'kubectl get svc --namespace default -w wordpress'
    export SERVICE_IP=$(kubectl get svc --namespace default wordpress --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}" )
    echo "WordPress URL: http://$SERVICE_IP/"
    echo "WordPress Admin URL: http://$SERVICE_IP/admin"
2. Open a browser and access WordPress using the obtained URL.
3. Login with the following credentials below to see your blog:
    echo Username: user
    echo Password: $(kubectl get secret --namespace default wordpress -o jsonpath="{.data.wordpress-password}" | base64 --decode)
```

## 访问WordPress

1. 安装Helm成功后，根据部署提示，执行以下命令获取访问IP地址。

```
kubectl get svc --namespace default wordpress-acs-sample --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}"
```

系统输出类似以下结果：

```
192.168.171.110
```

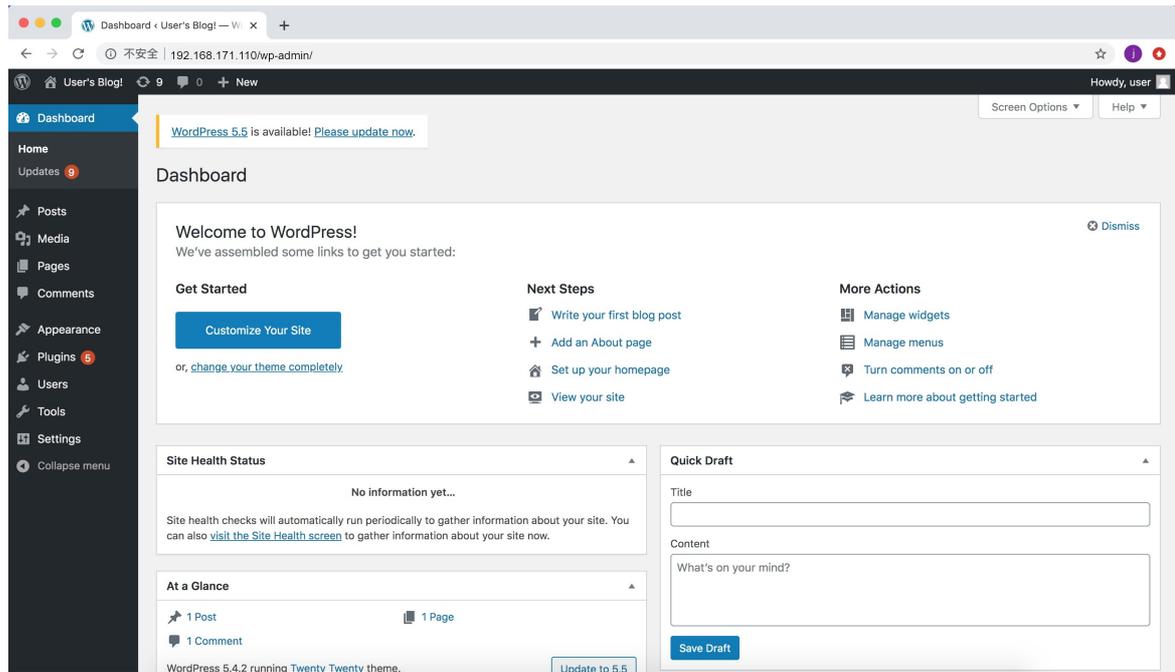
2. 执行以下命令获取admin访问密码。

```
kubectl get secret --namespace default wordpress-ack-wordpress-sample -o jsonpath="{.data.wordpress-password}" | base64 --decode
```

预期输出：

```
K*****X**7
```

3. 在地址栏中输入 `<IP地址>/login` 访问WordPress应用，然后使用用户名admin和访问密码登录admin界面。



## 删除应用

您可以执行以下命令删除部署的WordPress。

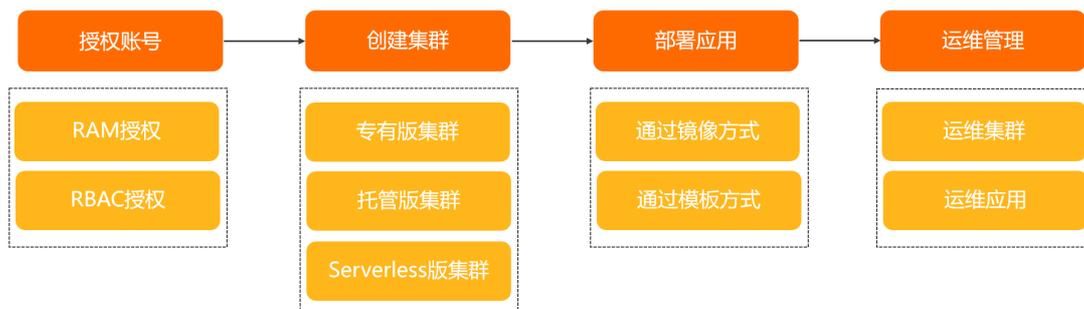
```
helm delete wordpress  
release "wordpress" uninstalled
```

# 5.使用指引

本文介绍容器服务ACK的整体操作流程和相关文档，帮助您快速上手容器服务ACK。

## 整体操作流程

容器服务ACK的整体操作流程如下图所示。



## 文档使用指引

1. 授权账号：授予账号相应的角色。具体操作，请参见[容器服务默认角色](#)。  
关于ACK集群的RAM授权和RBAC授权，请参见[自定义RAM授权策略](#)和[配置RAM用户或RAM角色RBAC权限](#)。
2. 创建集群：创建标准托管版集群。具体操作，请参见[创建Kubernetes托管版集群](#)。  
如果您需要使用更多类型的集群，请参见以下文档：
  - [创建Serverless Kubernetes集群](#)
  - [创建Kubernetes专有版集群](#)
  - [创建ACK Pro版集群](#)
  - [创建边缘托管版集群](#)
  - [创建托管GPU集群](#)
  - [创建专有GPU集群](#)
  - [创建安全沙箱托管版集群](#)
  - [创建安全沙箱专有版集群](#)
  - [创建加密计算托管集群](#)
3. 部署应用：通过镜像或编排模板部署应用。  
具体操作，请参见[创建无状态工作负载Deployment](#)。

**说明** 如果您的应用由多个镜像承载的服务组成，建议选择使用YAML文件部署应用。

4. 运维管理：分别对集群和应用进行运维。

集群运维	应用运维
<ul style="list-style-type: none"> <li>○ 集群管理                             <ul style="list-style-type: none"> <li>■ 升级ACK集群K8s版本</li> <li>■ 扩容ACK集群的节点</li> <li>■ 管理组件</li> </ul> </li> <li>○ 节点维护                             <ul style="list-style-type: none"> <li>■ 添加已有节点</li> <li>■ 设置节点调度</li> <li>■ 批量运维节点</li> <li>■ 集群节点挂载数据盘</li> </ul> </li> <li>○ 节点池管理                             <ul style="list-style-type: none"> <li>■ 管理节点池</li> <li>■ 调度应用至指定节点池</li> <li>■ 配置节点池内抢占式实例和存量实例的比例</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ 应用部署                             <ul style="list-style-type: none"> <li>■ 创建有状态工作负载StatefulSet</li> <li>■ 创建任务工作负载Job</li> <li>■ 使用私有镜像仓库创建应用</li> </ul> </li> <li>○ 应用与镜像更新                             <ul style="list-style-type: none"> <li>■ 使用应用触发器重新部署应用</li> <li>■ 使用免密组件拉取容器镜像</li> <li>■ 使用kritis-validation-hook组件实现自动验证容器镜像签名</li> </ul> </li> <li>○ 应用扩缩容                             <ul style="list-style-type: none"> <li>■ 手动伸缩容器组</li> <li>■ 容器水平伸缩 (HPA)</li> <li>■ 容器垂直伸缩 (VPA)</li> </ul> </li> <li>○ 应用备份                             <ul style="list-style-type: none"> <li>■ 开启集群备份</li> <li>■ 备份和恢复应用</li> <li>■ 跨集群迁移应用</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>○ 监控管理                             <p>应用性能监控、架构感知监控、事件监控、ARMS Prometheus监控、开源Prometheus监控</p> </li> <li>○ 日志管理                             <p>集群审计、通过日志服务采集Kubernetes容器日志、为Kubernetes和日志服务配置Log4JAppender、收集托管集群控制平面组件日志、分析和监控CoreDNS日志</p> </li> <li>○ 成本分析与报警管理                             <p>开启成本分析功能、容器服务报警管理</p> </li> </ul>	