

# SOFAStack

## 单元化应用服务 LHC 操作指南

产品版本：AntStack Plus 1.11.0


文档版本：20220928

# 法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，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。

## 商标声明

 蚂蚁集团 ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

## 免责声明

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。蚂蚁集团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，并在蚂蚁集团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蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

# 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。
粗体	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> <code>Instance_ID</code>
[] 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

1.什么是单元化应用服务	09
1.1. 单元化介绍	09
1.2. 概述	12
1.3. 产品优势	12
1.4. 产品架构	13
1.5. 功能特性	14
1.6. 应用场景	15
1.7. 使用限制	15
1.8. 基础术语	16
2.快速入门	19
3.准备环境	27
3.1. 创建单元化工作空间	27
3.2. 创建工作空间组	30
3.3. 创建逻辑单元	31
3.4. 创建部署单元	34
3.5. 删除工作空间	36
4.集群	40
4.1. 导入集群	40
4.2. 扩容集群	40
4.3. 查看集群详情	43
4.4. 解绑集群	44
4.5. 节点管理	45
4.5.1. 节点调度	45
4.5.2. 节点规划	46
4.5.3. 管理节点标签	48
4.5.4. 管理 Taint	49

4.5.5. 查看节点详情	49
4.5.6. 删除节点	50
4.6. 命名空间	52
4.6.1. 创建命名空间	52
4.6.2. 查看命名空间详情	53
4.6.3. 删除命名空间	54
4.6.4. 设置资源配额	55
4.7. 管理组件	57
5.应用服务	60
5.1. 创建应用服务	60
5.2. 高级配置	71
5.3. 发布模板	75
5.4. 发布应用服务	78
5.4.1. 立即发布	79
5.4.2. 发布单	81
5.4.3. 蓝绿发布	84
5.4.4. 全链路灰度发布	88
5.4.5. 多集群联邦发布容灾	92
5.5. 更新应用服务	92
5.6. 导入与导出应用服务	93
5.7. 查看应用服务详情	94
5.8. 删除应用服务	97
5.9. 系统环境变量	98
5.10. 容器水平伸缩 HPA	100
5.11. 运维单	103
6.流量管理	108
6.1. 应用层流量管理	108
7.网络	112

7.1. 打通多集群容器网络	112
7.2. 创建统一接入集群	114
7.3. 统一接入节点性能压测	117
7.4. 网关集群状态说明	123
7.5. 创建统一接入实例	125
7.6. 联邦负载均衡	126
7.6.1. 创建联邦负载均衡	126
7.6.2. 配置联邦负载均衡	128
7.6.3. 删除联邦负载均衡	128
7.7. 服务 (Services)	129
7.7.1. 概述	129
7.7.2. 创建服务 (Service)	130
7.8. 路由 (Ingress)	132
7.8.1. 概述	132
7.8.2. 管理路由 (Ingress)	132
7.9. NetworkPolicy	133
7.9.1. 概述	133
7.9.2. 创建 NetworkPolicy	134
8.配置管理	136
8.1. 管理配置项	136
8.2. 使用配置项	137
8.3. 管理保密字典	139
8.4. 使用保密字典	140
9.工作负载	143
9.1. 部署 (Deployments)	143
9.2. 有状态副本集 (Statefulsets)	144
9.3. 守护进程集 (Daemonsets)	145
9.4. 容器组 (Pods)	147

9.5. 定时任务 (CronJobs)	149
9.6. 一次性任务 (Jobs)	151
9.7. 自定义资源对象	153
9.7.1. CafeDeployment	153
9.7.2. InPlaceSet	155
10. 存储	157
10.1. 存储概述	157
10.2. 管理存储类型 (StorageClass)	158
10.3. 管理存储卷 (PersistentVolume)	163
10.4. 管理存储卷声明 (PVC)	164
10.5. 使用存储卷声明 (PVC)	165
10.6. 使用 PVC 挂载动态存储卷	167
10.7. 使用 PVC 挂载静态存储卷	169
10.8. StatefulSet 挂载动态存储卷	171
10.9. PVC 与 PV 的绑定	172
11. 镜像中心	174
11.1. 镜像仓库	174
11.2. 构建镜像	175
11.3. 镜像技术栈	181
11.4. 镜像安全扫描	183
11.4.1. 概述	183
11.4.2. 接入镜像仓库	183
11.4.3. 执行扫描任务	185
11.4.4. 查看镜像安全扫描结果	186
11.4.5. 管理镜像仓库	188
12. 应用商店	190
12.1. 应用商店概述	190
12.2. 产品模板	190

---

12.3. 产品安装	193
13.基础元数据	196
13.1. 逻辑单元	196
13.2. 部署单元	197
14.最佳实践	199
14.1. LHC Pod 域名配置	199

# 1. 什么是单元化应用服务

## 1.1. 单元化介绍

在当今的互联网业内，很多大型互联网系统，比如淘宝、支付宝、网商银行等，都已经实现了单元化架构，并从中受益匪浅，更多企业正加入其中。为什么要做单元化，单元化架构能给系统带来什么样的能力。本文结合蚂蚁集团支付宝系统的单元化架构建设实践，阐释单元化的原理与实现。

### 单点瓶颈

任何一个互联网系统，不论是支付宝、淘宝，还是 Google、Facebook，当发展到一定规模时，都会不可避免地触及到单点瓶颈。这里所说的“单点”，在系统的不同发展阶段表现不同。



### 服务器和应用单点

在系统发展初期，服务器和应用单点最先成为瓶颈，解决的方法也很简单，加机器、拆应用。

### 数据库单点

紧接着数据库单点，解决起来就开始不那么容易了，典型的做法是先垂直拆分，再水平拆分，在这个过程中需要解决多数据源、数据分片、透明访问等问题。

### 机房单点

当应用越来越多、服务器越来越多、数据库越来越多，单个机房的容量开始捉襟见肘，装不下这么多服务器。而且业务量的增长也让系统单机房运行的风险激增，一旦发生机房断电或是其他灾害导致机房故障，就会让整个系统完全瘫痪。机房不能放在一个篮子里，必须让系统在两个或更多 IDC 内运行。

多机房部署通常有以下两种模式：

- **垂直模式**：将全站应用、数据库等划分为几个部分，分别放在不同的机房中，完成一个业务可能需要不同机房中的不同应用提供服务。这就相当于在逻辑层面将一个物理机房放大了，机房容量不足的问题得以解决。
- **水平模式**：每个机房中部署的应用都是相同的，每个机房都有完成全站所有业务的能力，在运行时每个机房都只承担整站的一部分业务流量。

从具体实现角度来看，垂直模式更容易些，能够突破机房容量瓶颈，但不具备容灾能力。而容灾，是一个发展到一定规模的互联网系统的重要诉求，更是金融场景和金融业务必不可少的基础能力，因此在具体实践中，大多数大型系统都采用多机房水平扩展的模式。

### 单地部署

上面所说的容灾问题，对于一个有着亿级用户的系统，或一个数据系统来说尤为重要，仅仅是机房级容灾还不够，必须要考虑部署地容灾，也就是说不能把所有机房部署在地理上临近的地区，以防发生地震、海啸、核爆等剧烈灾害而导致系统被毁灭性破坏。这类系统的典型代表是银行、第三方支付等金融类系统，比如银行就对机房部署有着经典的“两地三中心”要求。于是，多地部署开始成为制约业务发展的瓶颈。

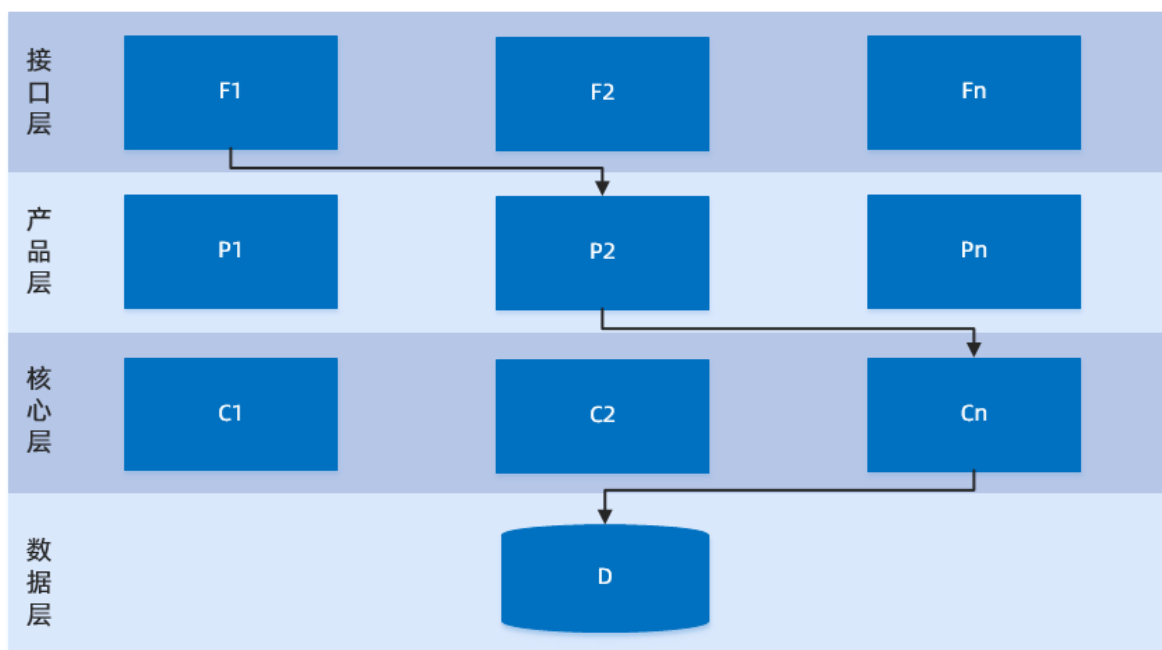
将系统的某个部分部署到距离较远的另外一个地区（城市）的能力，是大型互联网系统走向成熟的标志。多地部署本质上和多机房部署是一样的，面临的问题也几乎一样，除了距离问题。表象上是距离，深层隐藏着的是延时，更远的距离意味着更长的延时，个位数毫秒的延时不会给系统带来什么麻烦，但一旦这个数值变成几十毫秒，量变就引发了质变，很多业务开始不能忍受和忽略延时带来的影响。

## 单元化

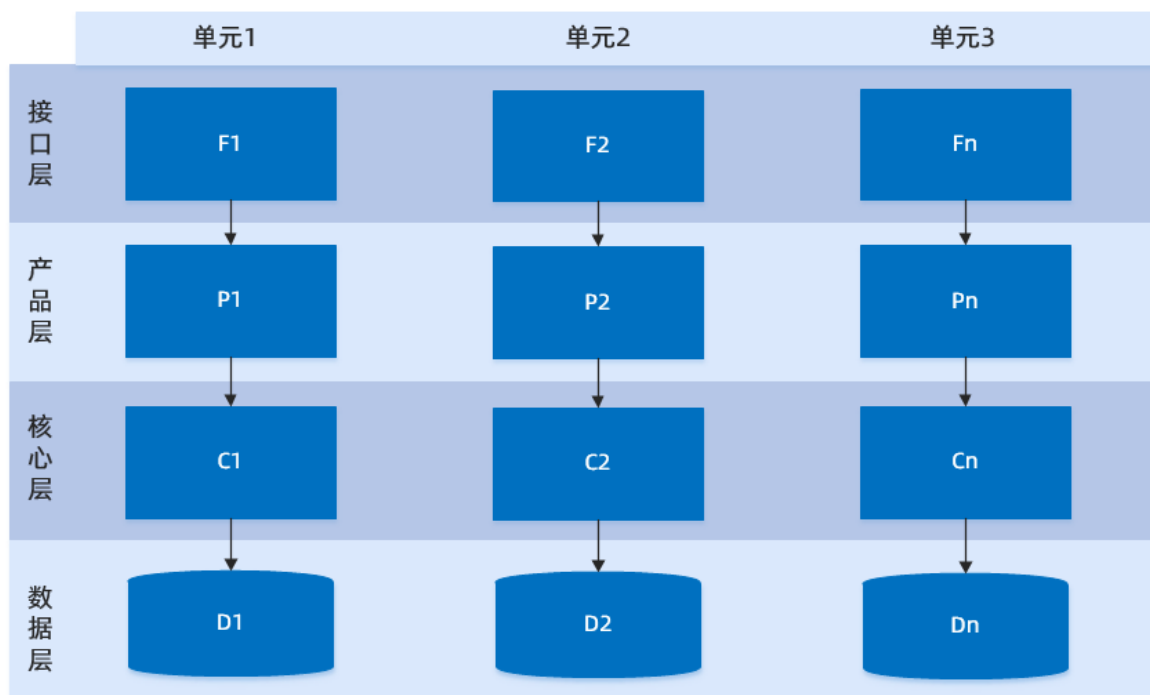
多地多机房部署，是互联网系统的必然发展方向，一个系统要走到这一步，也就必然要解决上面提到的问题：流量调配、数据拆分、延时等。业界有很多技术方案可以用来解决这些问题，而承载这些方案的，是一个部署架构。尽管可采用的部署架构不止一个，但不论是纯理论研究，还是一些先行系统的架构实践，都把“单元化部署”推崇为最佳方案。

单元（即单元化应用服务产品层的部署单元），是指一个能完成所有业务操作的自包含集合，在这个集合中包含了所有业务所需的所有服务，以及分配给这个单元的数据。单元化架构就是将单元作为部署的基本单位，在全站所有机房中部署多个单元，每个机房内单元数目不固定，任一单元均部署系统所需的全部应用，数据则是全量数据按照某种维度划分后的一部分。

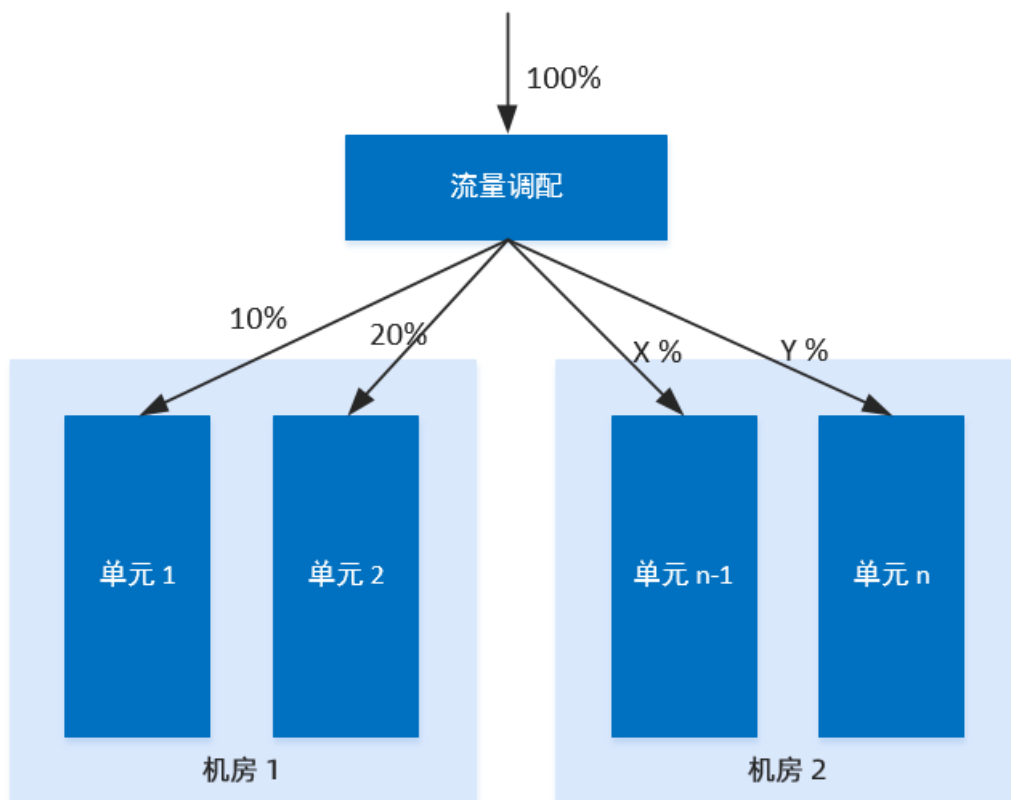
传统意义上的 SOA 化（服务化）架构，服务是分层的，每层的节点数量不尽相同，上层调用下层时，随机选择节点。



单元化架构下，服务仍然是分层的，不同的是每一层中的任意一个节点都属于且仅属于某一个单元，上层调用下层时，仅会选择本单元内的节点。



一个单元，是一个五脏俱全的缩小版整站，它是全能的，因为部署了所有应用；但它不是全量的，因为只能操作一部分数据。能够单元化的系统，很容易在多机房中部署，因为可以轻松将多个单元部署在一个机房内，而将另外几个单元部署在其他机房内。通过在业务入口处设置一个流量调配器，可以调整业务流量在单元之间的比例。



从上述对单元的定义和特性描述中，可以推导出单元化架构要求系统必须具备的一项能力：数据分区，实际上正是数据分区决定了各个单元可承担的业务流量比例。数据分区（shard），即是将全局数据按照某一个维度水平划分开来，每个分区的数据内容互不重叠，这也就是数据库水平拆分所做的事情。

仅把数据分区了还不够，单元化的另外一个必要条件是，全站所有业务数据分区所用的拆分维度和拆分规则都必须一样。若是以用户分区数据，那交易、收单、微贷、支付、账务等全链路业务都应该基于用户维度拆分数据，并且采用一样的规则拆分出同样的分区数。比如，以用户 ID 末 2 位作为标识，将每个业务的全量数据都划分为 100 个分区（00~99）。

有了以上两个基础，单元化才可能成为现实。把一个或几个数据分区，部署在某个单元内，这些数据分区占总量数据的比例，就是这个单元能够承担的业务流量比例。执行数据分区时一个很重要的问题是分区维度的选择，一个好的维度，应该：

- **粒度合适**：粒度过大，会丧失流量调配的灵活性和精细度；粒度过小，会给数据的支撑资源，访问逻辑带来负担。
- **足够平均**：按这个分区维度划分后，每个部署单元的数据量应该是几乎一致的。

以用户为服务主体的系统（很多面向用户的系统，比如支付宝）通常可以按用户维度对数据分区，这是一个最佳实践。

## 逻辑单元

逻辑单元是单元化架构的基础，一个单元被称为一个 Zone。根据业务特点不同，您可以将系统部署在不同类型的逻辑单元中。更多信息，请参考 [创建逻辑单元](#)。

## 1.2. 概述

随着云原生技术在业界的持续升温，越来越多的金融客户希望将能把云搭建在拥有云原生能力的平台之上，利用各类云原生的红利迅速实现技术转型，促进业务的敏态发展和持续创新。

单元化应用服务（LDC Hybrid Cloud，简称 LHC）定位于在云原生基础设施之上，在多机房、多地域的 Kubernetes 多集群场景，提供应用管理、发布运维、流量调拨、配置同步等能力。

作为开发运维人员日常接触的 PaaS 管控层产品，帮助解决应用和逻辑单元管理、按单元的配置变配、网络流量调拨、监控元数据配置等能力。不仅满足金融场景下同城多活和跨地域容灾的业务需求，同时能够让基础设施享受到容器化基础设施、云原生架构的技术红利。

LHC 旨在提供从单 Kubernetes 集群向多活联邦集群演进的能力，提供具备容灾能力的同城双活、两地三中心及更多机房级多活容灾场景。并可以配合 SOFASoft 各中间件产品、OceanBase 分布式数据库，形成单元化异地多活架构解决方案。

## 1.3. 产品优势

### 高可用与容灾

金融行业（尤其是银行）出于监管需求，其生产环境通常是要求在自建 IDC 里，即专有云环境。这对云原生底盘 Kubernetes 的高可用和稳定性提出了极高的要求，比如在双机房情况下如何做到双活，在有异地机房时，如何保证通讯的延迟不导致上层业务的大量超时而影响到最终用户的资金、账户安全。在灾难发生时，如何减少影响面，尽可能减少受影响的用户，并缩短影响的时间。

### 变更管控与保障

针对金融级场景下大规模分布式系统的特点，提供了丰富的发布策略以满足不同的场景，如：分组发布、Beta 发布、灰度发布等，帮助传统架构平滑过渡，适应金融科技风险保障需求，实现大规模金融级运维场景下的容器服务落地。

- 提供发布单追溯功能，记录和升级容器应用变更和版本管理。

- 支持多应用批量发布、设置应用间发布依赖顺序。
- 提供分组分批、可灰度、可暂停、可回滚的发布策略和控制能力。
- 容器支持按需、按资源水位、按计划弹性伸缩。

## 无限弹性扩展

随着金融行业思路的转变，将金融业务内嵌到个人生活服务中，必然会产生诸多高频场景。

采用分布式架构，可以让业务在出现热点后，进行拆分、扩容，以应对流量激增。但如果是由于物理资源不够，或者数据层瓶颈，仅采用应用层的分布式就不够了。如何做到从上至下有一套完整的体系，从接入层到应用层最后到数据层进行规整，可以作为一个整体单元进行快弹、快缩。

## 多地域应用服务统一发布

多地域机房架构下，同一个应用服务的配置中支持多地域差异化配置，保证一次发布可以在多个机房中生效，从而有效降低多机房应用服务发布层面的运维成本。

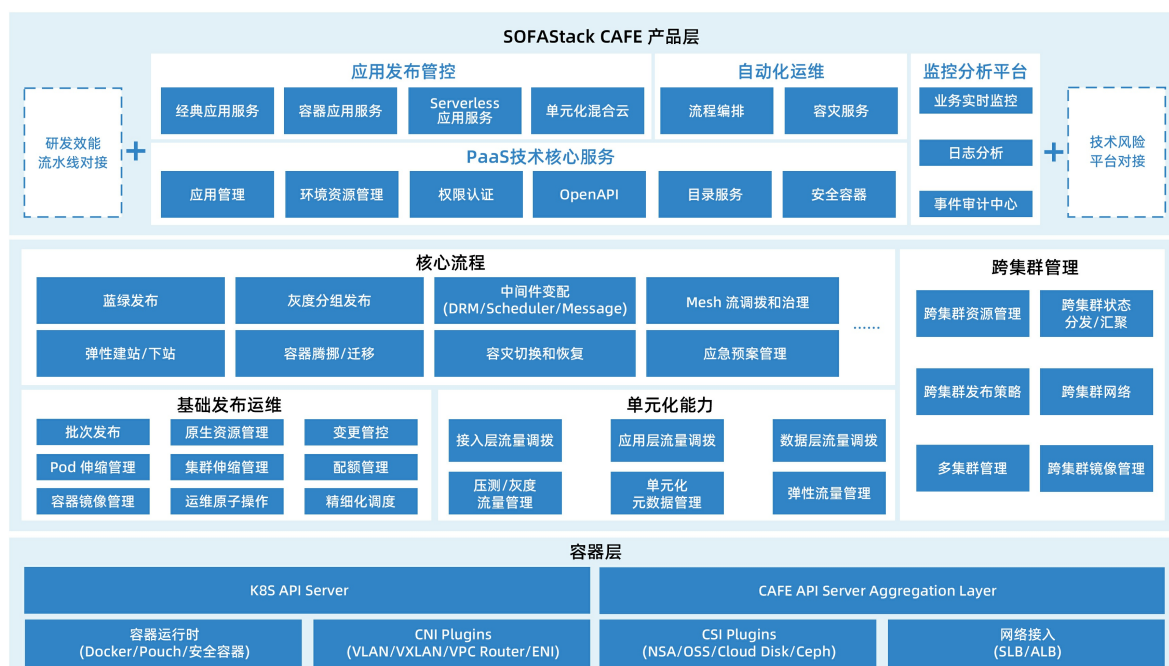
# 1.4. 产品架构

单元化应用服务，基于 Kubernetes 基础设施提供跨集群资源管理、发布运维及单元化管控能力。其纳管的每个集群，均为完整、标准的 Kubernetes 集群及相关扩展。

在集群之上，通过联邦管控平面，协调各集群资源、应用和配置，以提供应用变更管控、分组发布、镜像管理、流量调拨、元数据管理、集群资源管理等功能。用户可以通过控制台、命令行或 SDK 以标准方式对单集群及联邦管控层进行交互。

作为分布式架构的平台管控层，单元化应用服务还承担与各相关系统的对接能力。

- 研发效能产品，可提供持续集成与交付功能，生成 LHC 产品的发布单。
- 与各 PaaS 基础产品对接，提供权限、资源、应用元数据、工作空间、部署单元等领域模型的管理能力。
- 与监控分析平台对接，向用户提供完整集群资源和业务应用的可观测性。



## 1.5. 功能特性

### 跨集群应用管理

- 应用服务 (FedAppService/AplInstanceGroup)
  - 创建应用服务模板，根据部署单元 (Cell) 做相应的属性同步或变配。
  - 应用配置版本管理。
- 发布单管理
  - 有效管控单个或一组应用的发布状态和策略，检查发布变更。
  - 在发布准备时，可以配置应用依赖信息，自动实现串行和并行发布策略。
  - 在发布过程中，确保前置、发布、后置任务事件完整记录，方便追踪发布历程。
- 应用服务级别监控、日志和事件配置和展现
  - 配置跨集群的监控，并且可以聚合、显示。
  - 配置跨集群的日志收集，并且可以聚合、搜索、显示。
  - 展示应用服务级别事件。

### 跨集群资源管理

- Service：是 Kubernetes 自带的核心资源，用于抽象向多个 Pods 访问的接口，类似于负载均衡的概念。目前支持 Cluster IP、Loadbalancer、Node Port、Headless、ExternalDNS 类型。支持通过 YAML 和 UI 方式进行 Service 的创建和修改。
- ConfigMap：存储应用所需配置信息的资源类型，用于保存配置数据的键值对。ConfigMap 可通过 YAML 和 UI 方式进行创建和修改。
- Secret：一种用于存储工作负载所需要认证信息、密钥的敏感信息等的资源类型。Secret 可通过 YAML 和 UI 方式进行创建和修改。
- 工作负载 (Kubernetes NativeWorkloads)：工作负载包括 Kubernetes 原生的 Deployment、StatefulSet、DaemonSet、Job、Pod，支持工作负载配置、监控、更新、删除等生命周期管理。

### 多集群管理

- 集群创建与运维
- 节点扩容与展现
- 命名空间管理
- 提供对工作空间与工作空间组的管理

### 镜像中心

单元化应用服务所包含的镜像中心，提供和用户 Gitlab 代码库关联的镜像构建和版本管理能力，满足从代码到镜像的一键构建需求。同 IAM（身份认证管理系统）对接满足安全保障。

- 镜像构建和版本管理
- 镜像技术栈创建

### 单元化配置

- 逻辑单元配置

- 部署单元配置

## 接入层管理

- 接入网关与流量管理
- 应用层流量配置

## 应用商店

支持通过 HelmChart 应用商店一键编排复杂容器应用。

# 1.6. 应用场景

LHC 服务于云原生模式下，通过一套应用 PaaS 平台，提供统一的应用、资源管理，以及发布运维视图，实现多集群管理、跨集群应用运维发布、资源管理、流量管理。

具体来说，可细分为以下场景：

- **同城双活（active-active）**  
在同一个地域 Region，建立两个或更多可用区下的多个 Kubernetes 集群。
- **两地三中心**
  - 在同城双活的基础上，增加一个异地机房，做数据和应用备份。根据网络延时和带宽情况，可选择异地热备、温备和冷备三种方案。
  - 在异地存在延迟的情况下、业务可接受的范围内，可以选择异地热备。正常情况下访问同城双活测，在容灾期间切至异地灾备机房，提供业务访问。
- **异地多活（Multi-region active-active）**  
数据层做分片（Sharding），不同的 AZ 可以划分为更多的逻辑单元（Logic Data Center），处理不同的数据分片。尽量保证数据访问的链路从接入层到应用层再到数据层不会出现跨可用区的调用。这种架构下，可以做到任意数量地域的多活。
- **异构基础设施下的混合云**  
通过 Kubernetes 屏蔽掉底层 IaaS 的差异性，可充分利用公有云上的资源，将业务同时在专有云和公有云上进行部署，并进行统一运维管控。在该场景下，可以帮助金融客户达到以下目的：
  - 减少开发、测试资源的投入：专有云部署生产应用，公有云按需部署开发测试应用。
  - 线下快速容灾需求：应国家监管需求，需要在线下部署一套环境，以应对公有云上的突发情况（例如天弘余额宝）。
  - 弹性扩容：结合异地多活架构，使业务能够按需进行机房级的无限水平扩展。

# 1.7. 使用限制

专有云场景单元化应用服务（LHC）的功能限制如下。

## 应用发布

- 同一个应用服务，不能同时进行发布和运维操作。
- 以下操作都会在发布应用服务时删除并重建 Pod，而非原地升级：
  - 调整应用服务中 Pod 的 CPU 和 memory 的值。
  - 改变 Pod 中容器的数量。

- 在应用服务版本中更改所关联的负载均衡（LB）或修改统一接入模式下的转发规则都会导致业务中断。
- 应用服务不支持同时配置多个访问配置。
- 负载均衡（LB）在不同的应用服务间不能复用。

## 集群

- 由于网络模式限制，当前暂不提供 cluster IPservice，集群内通信只能走 Pod IP 或 LBSERVICE。
- 存储：飞天版本依赖 ACK 支持存储类型。物理机版本依赖 YODA 本地存储。

## 网络

统一接入集群支持节点扩缩容。

## 镜像

- 镜像构建：限制只能同时进行 3 个构建任务。
- 镜像中心：用户只能往所在租户的 namespace 下 push 镜像，或者从公共的 namespace 拉取镜像。镜像中心只能在 region 内部访问。

# 1.8. 基础术语

中文	英文	说明
混合云	Hybrid Cloud	混合云狭义上指“公有云 + 私有部署”的混合形态，通过平台能力抽象统一、自动化部署、配置管理等方面的技术和产品，淡化开发和运维人员对底层基础设施的关注，使应用和数据能够在混合数据中心环境中进行部署和运维。
单元化工作空间	LDC workspace	提供单元化能力，可用于同城双活及异地容灾场景。您可以通过单元化工作空间组对用户资源进行隔离，不同工作空间组下的集群彼此隔离。
工作空间组	WorkspaceGroup	工作空间（Workspace）在多地域的扩展，在多地域内对资源进行分组隔离管理。多个地域的网络可以通过专线高速通道实现互通。
逻辑单元	Zone	<p>一个单元被称为一个 Zone，有 3 种不同类型：RZone、GZone、CZone。单元的特点如下：</p> <ul style="list-style-type: none"><li>• 同一个应用在每个单元中拥有独立使用的资源。</li><li>• 同一个应用的业务在不同单元中按水平方向拆分。</li><li>• 不同单元处理的业务分片不重叠。</li></ul>

中文	英文	说明
单元化架构	LDC architecture	<p>应用层按照数据层相同的拆片维度，将整个请求链路收敛在一组服务器中，从应用层到数据层就可以组成一个封闭的单元。</p> <p>数据库只需要承载本单元的应用节点的请求，大大节省了连接数。“单元”可以作为一个相对独立整体来挪动，甚至可以将部分单元部署至异地。</p>
部署单元	Cell	<p>部署单元（Cell），是指一个能完成所有业务操作的自包含集合，在这个集合中包含了所有业务所需的所有服务，以及分配给这个单元的数据。</p> <p>单元化架构就是将单元作为部署的基本单位，在全站所有机房中部署数个单元，每个机房里的单元数目不定，任意一个单元都部署了系统所需的所有应用，数据则是全量数据按照某种维度划分后的一部分。</p>
应用服务	Application service	<p>该概念和 <a href="#">经典应用服务</a> 中的应用服务概念一致。但由于容器有其特殊性，LHC 中的应用服务会包含一些额外的元数据信息，比如容器规格配置、镜像、调度策略、日志配置等。</p>
镜像	Image	<p>镜像是应用包，将配置和相关软件等打在一起的二进制包，并且符合 Docker Image 规范。镜像可以来自任何可被 LHC 网络访问到的镜像中心，对于私有镜像中心，需要在 LHC 中配置相应的访问信息。</p>
构建	Build	<p>构建用于描述从应用源代码到制作出镜像过程的配置信息，包括源代码地址、分支信息、源镜像访问信息、目标镜像信息、Dockerfile 位置信息等。</p>
集群	Cluster	<p>LHC 中集群用于描述您所创建的一个工作负载集群，由多个节点组成。</p>
节点	Node	<p>节点表示一台装了 Docker 和 Kubelet，用以运行应用负载的物理机或者虚拟机。</p>
容器组	Pod	<p>Kubernetes 中最小的部署及管理单元。一个 Pod 由多个相关的并且共享磁盘的容器组成。</p>
命名空间	Namespace	<p>命名空间和 Kubernetes 中相应的概念保持一致，用于表示一个逻辑隔离的空间，会将 Pod、Service、ReplicaSet 等元素隔离，但通常来说，网络不隔离。</p>

中文	英文	说明
原地升级	Inplace upgrade	原地升级是指应用服务中 Pod 的更新方式。发布后 Pod 的 IP 通常和发布前无法保持一致，所在的节点也可能发生变化。该更新方式在镜像替换时不会导致 Pod 删除。
标签	Label	Kubernetes 的原生概念，用于给相应的资源打上标签，做聚合或者匹配。
污点	Taint	Kubernetes 的原生概念，用于给节点做污点标记，通常用于 Pods 的调度策略。 与之相对应的概念为：容忍（tolerance），若 Pods 上有相对应的 tolerance 标记，则可以容忍节点上的污点，并调度到该节点。
保密字典	Secret	Kubernetes 的原生概念，用于存储用户的加密内容。
应用容器	Container	应用程序所运行的隔离工作空间，通常是 Docker 容器或者 Pouch 等兼容 CRI 接口的具有隔离能力的沙箱工作空间。
工作负载	Workload	应用程序运行态的载体及其上层聚合。通常包括：Pod、Deployment、StatefulSet、DaemonSet、Job 等。
配置项	Configmap	Kubernetes 的原生概念，用于存储用户的配置信息。
存储类型	Storage Class	Kubernetes 的原生概念，通常由系统管理员定义，用于指定所支持的存储类别，不同的类别会有不同的存储 SLA、备份策略等差异性。
存储卷	Persistent Volume	Kubernetes 的原生概念，表示一个由系统管理员创建好的存储资源。
存储卷声明	Persistent Volume Claim	Kubernetes 的原生概念，一个存储卷声明绑定一个存储卷。

## 2.快速入门

本文以一个同城双活架构下的双集群发布模式为例，引导您快速熟悉 LHC 的操作链路。

LHC 也支持在单元化架构下发布应用，具体内容参见 [单元化入门指南](#)。

### 说明

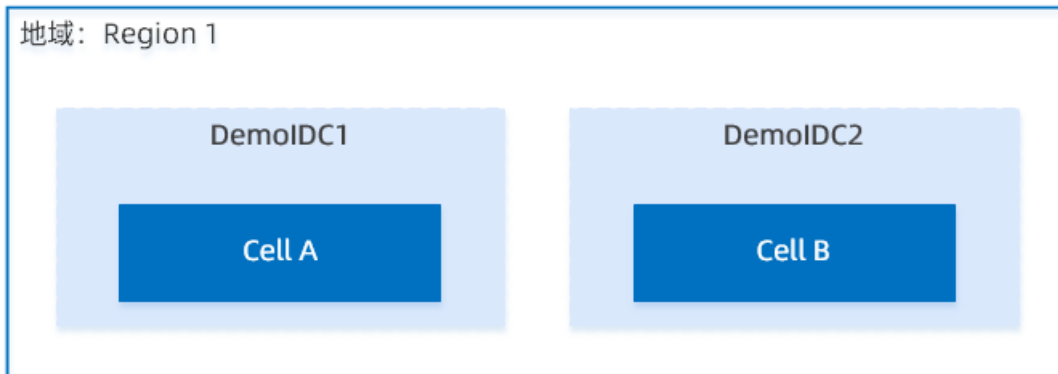
双机房场景下，推荐单元化工作空间与可用区（机房）是 1：2 的关系。

### 背景信息

部署架构如下图：在 SOFAShark 租户下，在同一地域内，创建一个工作空间组，关联两个工作空间，分别对应两个不同的可用区（机房），每个机房对应一个部署单元（Cell）。

#### 【租户】SOFAShark 租户

#### 【单元化工作空间】DemoWSG



### 步骤一：创建单元化工作空间

该步骤中您将创建一个包含双可用区的单元化工作空间。

1. 登录 LHC 控制台。
2. 在左侧导航栏，单击下方的 **全局设置**，进入工作空间列表页面。
3. 单击 **创建工作空间**，选择 **单元化工作空间** 类型，单击 **创建**。
4. 在 **创建工作空间** 页面，输入以下基本信息。
  - **工作空间标识**：2~45 个字符，以大小写字母开头，只能包含字母和数字。工作空间标识全局唯一，一经确定无法修改，例如：dev、test、prod 等。本例中输入：DemoWSG。
  - **工作空间名称**：1~64 个字符，工作空间的显示名称，例如：开发工作空间、测试工作空间、生产工作空间。本例中输入：DemoWSG。
  - **地域**：工作空间所在的地域（Region），一个工作空间一定属于某个地域。
  - **网络类型**：选择 **VPC 网络**。
  - **导入 VPC**：保持默认关闭状态。

5. 单击 **下一步**，在 **创建单元架构** 页面单击 **添加可用区**。每个工作空间配置的可用区（Available Zone）个数无限制。本例中为工作空间配置两个可用区，为支持双机房高可用等架构做准备。

? 说明

系统会根据可用区配置自动为您生成单元化架构拓扑，划分好逻辑单元和部署单元。



6. 单击 **下一步**，在 **创建 VPC** 页面，输入以下配置信息：

- **专有网络名称**：由 2~128 个英文或中文字符组成，必须以大小字母或中文开头，可包含数字、下划线（\_）或连字符（-）。可以使用系统自动生成的名称 `单元化工作空间标识-vpc`。
- **专有网络网段**：专有网络的网段，一旦选择便无法更改，专有网络内的所有资源，如 ECS、RDS、SLB 的私网 IP 都在该网段内。可选网段如下：
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- **描述**：非必填，填写专有网络描述内容。不能以 `http://` 和 `https://` 开头。
- **交换机**：单击 **添加交换机**，在弹出的 **创建交换机** 窗口，填写以下信息，单击 **提交**。
  - **名称**：交换机名称。长度为 2~128 个字符，以英文字母或中文开头，可包含数字、下划线（\_）和短横线（-）。系统自动生成 `单元化工作空间标识-vsw` 组成的默认名称。

- **可用区**：交换机的可用区。同一 VPC 内不同可用区的交换机内网互通。您需要为每个可用区创建一台交换机。
  - **自定义网段**：默认关闭。开启后，需填写网段地址。交换机的网段可以和其所属的 VPC 网段相同或者是其 VPC 网段的子网。
  - **子网掩码**：自定义网段 关闭时，需分别选择子网掩码和网段地址。默认专有的网段掩码是 24 位，例如 172.31.0.0/24。范围为 16~29 位之间，可提供 65532 个地址。
  - **描述**：输入交换机的描述信息。可包含 2~256 个中英文字符，不能以 `http://` 和 `https://` 开头。
7. 单击 **下一步**，在 **创建安全组** 页面，单击 **添加安全组**，在弹出的 **添加安全组** 窗口，填写以下信息，单击 **确定**。
- **安全组名称**：长度为 2~128 个英文或中文字符，只能包含字母、数字、短横线以及 `.`，必须以字母开头。系统自动生成 `单元化工作空间标识-sg` 组成的默认名称。
  - **描述**：可包含 2~256 个中英文字符，不能以 `http://` 和 `https://` 开头。
  - **规则**：保持默认设置，接受所有的出入访问。
8. 单击 **确定** 返回 **创建安全组** 页面，单击 **提交**。

## 步骤二：导入集群

集群是运行工作负载的逻辑分组，包含一组云服务器资源，每台云服务器即集群中的一个节点。首次使用 LHC 时，您需要先导入一个集群。

### 前提条件

- 您拥有 LHC 管理员权限。
- 已创建单元化工作空间。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在 **集群详情** 页，单击 **导入集群**。
3. 在 **导入集群** 页面，完成以下配置。

配置项	说明
集群类型	阿里云
专有网络	从已有 VPC 列表中，选择所需的 VPC。
选择已有集群	从已有集群列表中，选择您需要导入的 ACK 集群。

配置项	说明
部署单元	可以为可用区设置不同的部署单元。 如需修改部署单元，可前往 <a href="#">全局设置</a> > <a href="#">工作空间详情</a> 进行设置。

4. 单击 **提交**。

### 步骤三：创建转账应用

在 **应用管理** 控制台创建名为 **transfer-money** 的应用。

#### 操作步骤

1. 登录应用管理控制台，在左侧导航栏单击 **应用列表**。
2. 在 **应用列表** 页面，单击右上角的 **创建应用**。
3. 在 **创建应用** 页面输入以下信息，单击 **确定**。

- **应用名称**：输入 `transfer-money`。

#### 说明

应用名称在同一租户内必须是唯一的。

- **技术栈**：选择 **SOFABoot**。
- **应用分组**：选择已创建的应用分组。更多信息参见 [管理应用分组](#)。
- **应用标签**：应用分级标签，便于进行应用快速检索，支持添加系统标签和自定义标签。
- **应用描述**：选填，应用描述的长度不能超过 500 个字符，如 `这是一个示例应用`。更多信息参见 [管理应用](#)。

### 步骤四：准备镜像

LHC 以镜像的方式部署应用服务，创建应用服务前需先准备好镜像。为方便您测试，LHC 提供以下可以公开访问的示例镜像：

镜像名称	镜像地址
aks-transfermoney-server	registry-vpc.cn-shanghai.aliyuncs.com/sofa-samples/aks-transfermoney-server:20190726095608708-master.40215e87
aks-transfermoney-client	registry-vpc.cn-shanghai.aliyuncs.com/sofa-samples/aks-transfermoney-client:20190815134744553-master.5ea6278b

### 步骤五：创建应用服务

创建 2 个有依赖关系的应用服务：aks-transfermoney-server、aks-transfermoney-client。

## 前提条件

已创建联邦命名空间，例如 `antcloud-demo`。具体步骤参见 [创建联邦命名空间](#)。

## 创建转账服务端的应用服务

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击 **创建应用服务**。
3. 在 **创建应用服务** 页面，填写以下基本信息，单击 **下一步**。
  - **命名空间**：选择一个集群中的命名空间（namespace），默认为列表第一个。
  - **应用服务名称**：容器服务的名称，输入 `aks-transfermoney-server`。应用服务名称允许包含（小写）字母、数字、连字符，且必须以字母开头，以字母或数字结尾。同一个命名空间下不允许同名。
  - **所属应用**：选择 `transfer-money`。
  - **描述**：选填。容器服务的描述。
4. 在 **Pod 模板配置** 页面，填写以下信息，单击 **下一步**。
  - **容器名称**：输入名称。容器名称允许包含（小写）字母、数字、连字符、且必须以字母开头、以字母或者数字结尾，最多允许输入 63 个字符。
  - **访问类型**：选择 **镜像仓库**，使用示例镜像 `aks-transfermoney-server`，镜像地址参见 [准备镜像](#)。
  - **CPU 配置**：请求核数为 200 millicores，最大核数为 500 cores。
  - **内存配置**：请求内存为 512 MiB，最大内存为 1 GiB。
  - 在 **高级配置 > 环境变量配置** 中，配置 `APPSVC_VERSION = V1`，  
`ANTCLOUD_SOFA_PROFILE = prod`。
  - 在 **高级配置 > 健康检查配置** 中，配置 Readiness Probe 检查方法为 `Http GET` 方法、访问路径为 `/health/readiness`、访问端口 `8080`。其余项按照输入框提示配置。
  - **配置覆盖**：保持关闭。
5. 在 **弹性配置** 页面，填写以下信息，单击 **下一步**。

**副本伸缩策略配置**：目前仅支持 **固定副本数**，默认为 0，勾选部署单元（`RZ01A`、`RZ02A`）并修改为期望副本数 1，即应用服务运行时保持固定数目的 Pod 副本。
6. 在 **访问配置** 页面，填写以下信息，单击 **下一步**。
  - i. 在 **访问配置** 页面，单击 **添加负载均衡**。

ii. 在 **负载均衡** 窗口，填写以下信息后，单击 **确定**。

- **负载均衡名称**：填写服务名称。系统默认生成服务名称前缀为 `应用服务名称-`。

② 说明

需记录下这个 Service Name，需要以环境变量形式配置到客户端应用。

- **访问方式**：选择 **内网**。创建一个内部的 Service，并将流量转发到容器的相应端口上。
- **端口映射**：单击 **添加端口映射**，填写以下信息。其余保持默认设置。
  - **协议**：选择 **TCP**。
  - **转发规则**：选择 **RR 轮询**。
  - **前端端口**：容器镜像中工作负载程序实际监听的端口，输入 `80`。
  - **后端端口**：容器端口映射到集群虚拟 IP 上的端口，输入 `8341`。

7. 在 **部署和调度配置** 页面，为应用服务保持系统默认配置，单击 **下一步**。

8. 在应用服务 **预览** 页面，确认信息无误，单击 **提交**。

③ 说明

该应用服务配置编辑完成提交后，进行版本备注，方便区分版本。

## 创建转账客户端的应用服务

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。

2. 在应用服务列表页，单击 **创建应用服务**。

3. 在 **创建应用服务** 页面，填写以下基本信息，单击 **下一步**。

- **命名空间**：选择一个集群中的命名空间（namespace），默认为列表第一个。
- **应用服务名称**：容器服务的名称，输入 `aks-transfermoney-client`。服务实例名称允许包含（小写）字母、数字、连字符，且必须以字母开头，以字母或数字结尾。同一个命名空间下不允许同名。
- **所属应用**：选择 `transfer-money`。
- **负责人**：选择该应用服务的负责人。
- **描述**：选填。容器服务的描述。

4. 在 **Pod 模板配置** 页面，填写以下信息，单击 **下一步**。

- **容器名称**：输入名称。
- **访问类型**：选择 **镜像仓库**，使用示例镜像 `aks-transfermoney-client`，镜像地址参见 [准备镜像](#)。
- **CPU 配置**：请求核数为 200 millicores，最大核数为 500 core。
- **内存配置**：请求内存为 512 MiB，最大内存为 1 GiB。
- 在 **高级配置 > 环境变量配置** 中，配置变量名称为 `SERVER_TM_ADDRESS`，Server 端值为 Service Name。

5. 在 **弹性配置** 页面，填写以下信息，单击 **下一步**。

**副本伸缩策略配置**：目前仅支持固定副本数，默认为 0，勾选部署单元（ `RZ01A` 、 `RZ02A` ）并修改为期望副本数 1，即应用服务运行时保持固定数目的 Pod 副本。每个 client 能够压测并发上限为 1000。

6. 在 **访问配置** 页面，单击 **添加负载均衡**，填写以下信息，单击 **下一步**。

i. 在 **负载均衡** 窗口，填写以下信息后，单击 **确定**。

- **负载均衡名称**：填写服务名称。系统默认生成服务名称前缀为 `应用服务名称-`。
- **访问方式**：选择 **外网**。创建一个内部的 Service，并将流量转发到容器的相应端口上。

② 说明

选取两个应用服务并进行串行发布。配置外部 Service 会创建一个公网的 LoadBalancer，并将流量转发到容器的相应端口上。访问方式由公网负载均衡服务地址以及设置的访问端口组成，例如 `10.117.117.117:80`。

- **访问方式**：选择 **外网**。创建一个内部的 Service，并将流量转发到容器的相应端口上。
- **端口映射**：单击 **添加端口映射**，填写以下信息。其余保持默认设置。
  - **协议**：选择 **TCP**。
  - **转发规则**：选择 **RR 轮询**。
  - **前端端口**：容器镜像中工作负载程序实际监听的端口，输入 `80`。
  - **后端端口**：容器端口映射到集群虚拟 IP 上的端口，输入 `8341`。

7. 在 **部署和调度配置** 页面，为应用服务保持系统默认配置，单击 **下一步**。

8. 在应用服务 **预览** 页面，确认信息无误，单击 **提交**。

② 说明

该应用服务配置编辑完成提交后，进行版本备注，方便区分版本。

9. 在 **部署和调度配置** 页面，为应用服务保持系统默认配置，单击 **下一步**。

10. 在应用服务 **预览** 页面，确认信息无误，单击 **提交**。

② 说明

该应用服务配置编辑完成提交后，进行版本备注，方便区分版本。

## 步骤六：创建发布单

通过发布单同时发布上一步中创建的两个有依赖关系的应用服务。

 说明

选取两个应用服务并进行串行发布。

## 操作步骤

1. 登录控制台，在左侧导航栏单击 **发布运维** > **发布单** 进入发布部署大盘。
2. 在发布部署大盘，单击 **创建发布单**。
3. 在 **创建发布单** 页面，填写以下发布信息后，单击 **下一步**。

### 基本信息

- **标题**：发布标题。例如 `Demodeploy`。
- **类型**：仅支持分组发布。
- **命名空间**：选择待发布的应用服务所属的命名空间。
- **应用服务发布列表**：在 **待选应用服务列表** 中单击选择需要的应用服务：aks-transfermoney-client、aks-transfermoney-server。单击 > 图标将应用服务添加到 **已选应用服务列表** 中。

 说明

若应用服务存在多个提交版本，需要选择要发布的版本。默认选择最新的版本。

### 高级配置

设置应用服务依赖关系：aks-transfermoney-client 依赖于 aks-transfermoney-server。

4. 在 **预览** 页面确认信息无误后，单击 **创建**。系统会自动跳转到 **发布单详情** 页面，单击 **整体发布** 即可开始发布。

## 步骤七：验证结果

验证应用服务 aks-transfermoney-client 及 aks-transfermoney-server 是否在双集群中发布成功。

## 操作步骤

1. 登录控制台，在左侧导航栏单击 **发布运维** > **应用服务**。
2. 在应用服务列表中，分别单击 aks-transfermoney-client 及 aks-transfermoney-server 进入实例详情页。
3. 在 Pod 页签中，查看两个应用服务在两个部署单元中均有一个应用容器运行中。

更多应用服务内容，参见 [查看应用服务详情](#)。

## 3. 准备环境

### 3.1. 创建单元化工作空间

您可以在 **控制台 > 全局设置** 中进行当前租户的所有工作空间的管理，包括进行查看、添加、编辑、删除工作空间等操作。

#### 背景介绍

一个工作空间的本质，是指“网络互通、安全策略一致、访问延时极小”的一组资源。您可以通过工作空间方便地将资源进行分组管理，例如：根据不同的研发交付需求，将工作空间划分为开发工作空间、测试工作空间、生产工作空间等。同时，不同工作空间中的资源互相隔离，可以为每个工作空间分配单独的操作员权限进行管理。为支持高可用（High Availability），工作空间可以跨可用区（Zone），但不能跨地域（Region）。

每个工作空间都需要创建相应的基础资源（包括：专有网络、交换机、安全组、ECS 等），并基于这些资源部署集群，从而为应用服务提供服务。

#### 前提条件

了解 [单元化架构](#)。

#### 操作步骤

1. 登录控制台，在左侧导航栏单击左下角的 **全局设置**，进入工作空间列表页面。
2. 单击 **创建工作空间**，选择 **单元化工作空间** 类型，单击 **创建**。

##### ② 说明

**单元化工作空间**：提供单元化能力，可用于同城双活及异地容灾场景。您可以通过单元化工作空间组对用户资源进行隔离，不同工作空间组下的集群彼此隔离。

3. 在 **创建工作空间** 页面，输入以下基本信息。
  - **工作空间标识**：2~45 个字符，工作空间的英文标识，全局唯一，一经确定无法修改，例如：dev、test、prod 等。
  - **工作空间名称**：1~64 个字符，工作空间的显示名称，例如：开发工作空间、测试工作空间、生产工作空间。
  - **地域**：工作空间所在的地域（Region），一个工作空间一定属于某个地域。
  - **网络类型**：选择 **VPC 网络**。
  - **导入 VPC**：默认关闭。开启后，您可以复用已有的 VPC 资源，将工作空间与已创建的 VPC 绑定。
4. 单击 **下一步**，在 **创建单元架构** 页面单击 **添加可用区**。每个工作空间配置的可用区（Available Zone）个数不做限制。本例中为工作空间配置两个可用区，为支持双机房高可用等架构做准备。

##### ② 说明

系统会根据可用区配置自动为您生成单元化架构拓扑，划分好逻辑单元和部署单元。更多信息，参考 [创建逻辑单元](#)。



5. 单击 **下一步**，在 **创建 VPC** 页面，输入以下配置信息：

- **专有网络名称**：由 2~128 个英文或中文字符组成，必须以大小字母或中文开头，可包含数字、下划线 ( \_ ) 或连字符 ( - )，不能以 `http://` 和 `https://` 开头。系统自动生成

单元化工作空间标识-vpc 组成的默认名称。

- **专有网络网段**：专有网络的网段，一旦选择便无法更改，专有网络内的所有资源，如 ECS、RDS、SLB 的私网 IP 都在该网段内。可选网段如下：

- 10.0.0.0/9
- 172.16.0.0/12
- 192.168.0.0/16

- **描述**：选填，可包含 2~256 个中英文字符，不能以 `http://` 和 `https://` 开头。

- **交换机**：单击 **添加交换机**，在弹出的 **添加交换机** 窗口，填写以下信息，单击 **提交**。

- **名称**：交换机名称。长度为 2~128 个字符，以英文字母或中文开头，可包含数字、下划线 ( \_ ) 和短横线 ( - )。系统自动生成 单元化工作空间标识-vsw 组成的默认名称。
- **可用区**：交换机的可用区。同一 VPC 内不同可用区的交换机内网互通。您需要为每个可用区创建一台交换机。

- **自定义网段**：默认关闭。开启后，需填写网段地址。交换机的网段可以和其所属的 VPC 网段相同或者是其 VPC 网段的子网。
  - **子网掩码**：自定义网段 关闭时，需分别选择子网掩码和网段地址。默认专有的网段掩码是 24 位，例如 172.31.0.0/24，最多可提供 65536 个私网 IP 地址。范围为 16~29 位之间，可提供 4~65532 个地址。
  - **描述**：输入交换机的描述信息。可包含 2~256 个中英文字符，不能以 `http://` 和 `https://` 开头。
6. 单击 **下一步**，在 **创建安全组** 页面，单击 **添加安全组**，在弹出的 **添加安全组** 窗口，填写以下信息，单击 **确定**。
- **安全组名称**：长度为 2~128 个英文或中文字符，必须以大小字母或中文开头，不能以 `http://` 和 `https://` 开头。可以包含数字、半角冒号（:）、下划线（\_）或者连字符（-）。系统自动生成 `单元化工作空间标识-sg` 组成的默认名称。
  - **描述**：可包含 2~256 个中英文字符，不能以 `http://` 和 `https://` 开头。
  - **规则**：单击 **添加规则**，在弹出的 **创建规则** 窗口，填写以下信息，单击 **确定**。
  - **网卡类型**：选择 **内网**。VPC 类型安全组的网卡类型（NicType）只可设置内网（intranet）。
  - **规则方向**：
    - **内网出方向**：是指 ECS 实例访问内网中其他 ECS 实例或者公网上的资源。
    - **内网入方向**：是指内网中的其他 ECS 实例或公网上的资源访问 ECS 实例。
  - **协议类型**：传输层协议。取值大小写敏感。可选项如下：
    - **TCP**：支持 TCP 协议。
    - **UDP**：支持 UDP 协议。
    - **ICMP**：ICMP 协议仅支持 IPv4 地址。
    - **GRE**：支持 GRE 协议。
    - **ALL**：支持所有协议。
  - **端口范围**：目的端安全组开放的传输层协议相关的端口范围，取值范围为 1~65535，设置格式例如 “1/200”、“80/80”，其中“-1/-1”不能单独设置，代表不限制端口。取值范围：
    - **TCP/UDP 协议**：取值范围为 1~65535。使用斜线（/）隔开起始端口和终止端口。正确示范：1/200；错误示范：200/1。
    - **ICMP 协议**：-1/-1。
    - **GRE 协议**：-1/-1。
    - **协议类型取值为 ALL**：-1/-1。
  - **访问权限**：选择 **接受访问** 或 **拒绝访问**。
  - **优先级**：数字越大代表的优先级越高。选择范围为 1~100。
  - **源端 IP 地址**：填写单一 IP 地址或者 CIDR 网段格式，如：12.1.1.1 或 13.1.1.1/25。如果填写 0.0.0.0/0 表示允许或拒绝所有 IP 地址的访问，设置时请务必谨慎。目前仅支持 IPv4 格式的 IP 地址范围。默认值：0.0.0.0/0。

- **目标端 IP 地址**：填写单一 IP 地址或者 CIDR 网段格式，如：12.1.1.1 或 13.1.1.1/25。如果填写 0.0.0.0/0 表示允许或拒绝所有 IP 地址的访问，设置时请务必谨慎。目前仅支持 IPv4 格式的 IP 地址范围。
- **规则描述信息**：输入安全组规则的描述信息。长度为 1~512 个字符。

7. 单击 **提交**。

## 3.2. 创建工作空间组

通过创建工作空间组，实现应用的跨地域部署。一个租户下可以创建多个工作空间组，一个工作空间组可以跨地域关联多个工作空间。本文介绍如何通过控制台创建工作空间组。

### 背景信息

您可以通过创建工作空间组对多个工作空间进行管理。

- 同城双活架构下，工作空间组采用单地域双机房模式，建议工作空间与地域之间的关系是一对一的关系，而该地域下两个机房可以使用一个工作空间。
- 异地多活架构下，工作空间组可以跨地域，从而实现应用的跨地域部署。

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **全局设置 > 工作空间组** 进入工作空间组列表页。
2. 单击 **创建工作空间组**，输入以下基本信息后，单击 **下一步**。

- **工作空间组名称**：当前环境（工作空间组）的显示名称，例如开发环境。长度为 1~64 个字符。
- **工作空间组标识**：工作空间组的英文标识。租户下唯一标识，会作为环境变量传给容器，例如 dev。

#### ② 说明

工作空间组标识命名约束如下：

- 仅支持字母、数字。
- 必须以大小字母开头。
- 长度为 2~45 个字符。

- **地域**：工作空间组所在的地域（Region），可选择多个以实现跨地域部署。

#### ② 说明

**地域** 选定后，会在所选地域下创建对应工作空间，后续仍支持扩展操作。

- **网络类型**：选择 **VPC 网络**。
  - **导入 VPC**：选择 VPC。选定后，将工作空间组与已创建的 VPC 绑定，复用已有的 VPC 资源，无需新建 VPC。
  - **高级配置**：默认关闭。开启后，可自定义工作空间标识。工作空间唯一标识符，默认工作空间标识会由工作空间组标识加上地域组合而成，也可进行自定义设置。
3. 在 **创建单元架构** 页单击 **添加机房**，选择可用区/机房。本例中为两个地域配置机房。

← 创建工作空间组

✓ 填写基本信息

2 创建单元架构

3 预览

regionA

cn-n~~g~~eng-121e-ar~~g~~est121001-a

GZone

+ 添加机房

regionB

cn-n~~g~~eng-121e-ar~~g~~est121001-b

GZone

+ 添加机房

下一步

上一步

取消

4. 单击下一步，在 预览 页面，确认信息无误，单击 提交。

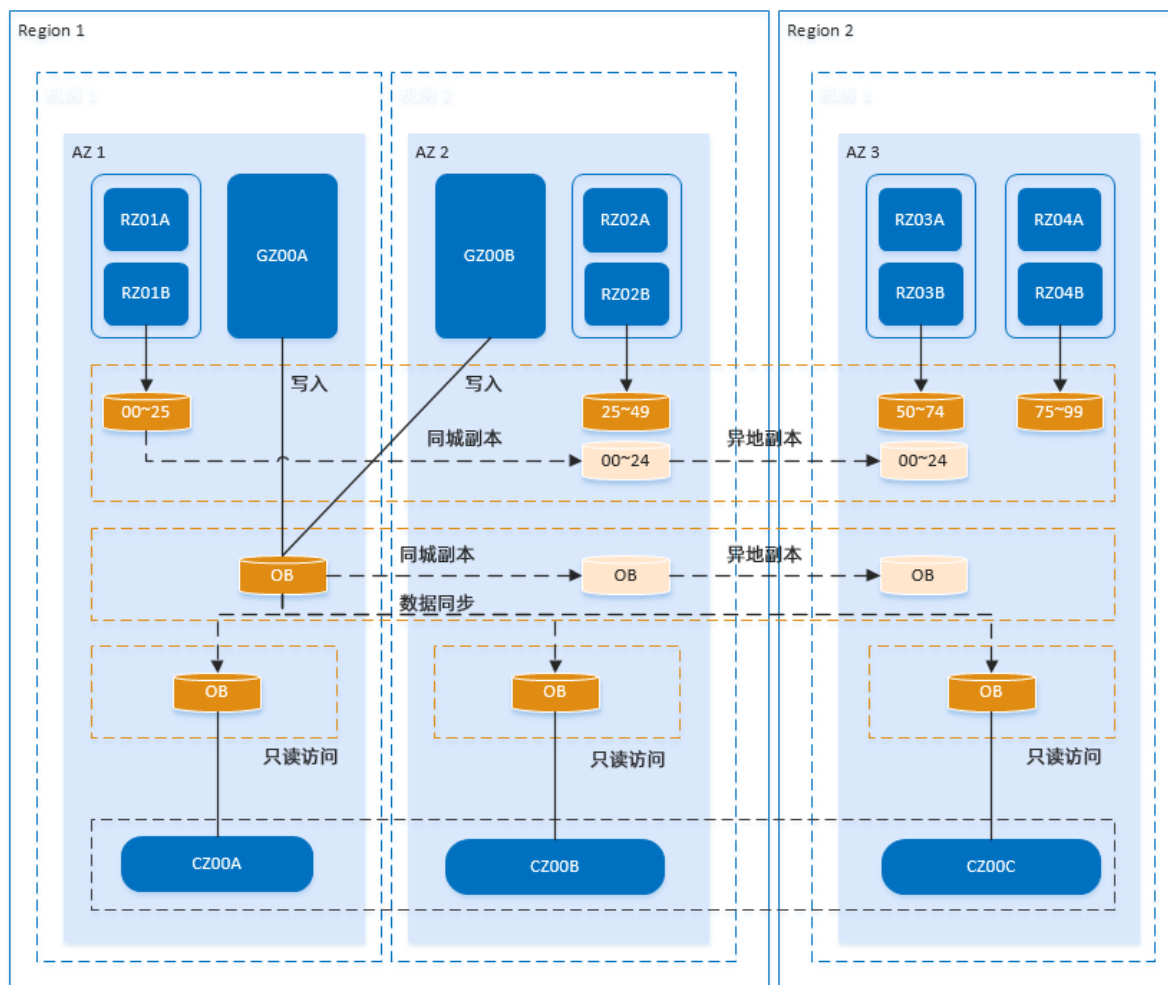
## 3.3. 创建逻辑单元

使用 LHC 部署应用服务前，您需要先定义系统的逻辑单元。逻辑单元是单元化架构的基础，一个单元被称为一个 Zone。根据业务特点不同，您可以将系统部署在不同类型的逻辑单元中。

- RZone（Region Zone）：部署按用户维度拆分的关键业务系统。核心业务和数据单元化拆分，拆分后分片均衡，单元内尽量自包含（调用封闭），拥有自己的数据，能完成所有业务。一个可用区可以有多个 RZone。
- GZone（Global Zone）：部署未按用户维度拆分的系统，被 RZone 依赖，提供不可拆分的数据和服务，如配置型的业务。数据库可以和 RZone 共享，多租户隔离，全局只有一组，可以配置流量权重。
- CZone（City Zone）：部署未按用户维度拆分的系统，被 RZone 高频访问，解决跨域通信延时问题。为了解决异地延迟问题而特别设计，适合读多写少且不可拆分的业务。一般每个城市一套应用和数据，是 GZone 的快照。

> 文档版本：20220928

31



## 操作步骤

### 说明

在创建单元化工作空间时，系统会自动进行逻辑单元和部署单元的规划，参考 [创建单元化工作空间](#)。您可以根据实际需求，在发布应用服务前，对逻辑单元进行修改或增删。

1. 登录控制台，在左侧导航栏单击 **全局设置** 进入工作空间卡片页。
2. 单击需要修改逻辑单元的工作空间卡片进入工作空间详情页。
3. 选择 **单元架构** 页签，单击 **编辑**，架构拓扑进入可编辑的状态。



4. 您可以根据需要对逻辑单元进行以下增删操作：

- **增加**：单击 **添加** 可以选择已有的可用区并增加新的逻辑单元。

② 说明

仅支持增加 RZone 类型的逻辑单元。逻辑单元名称请以 **RZ** 开头，名称长度限制为 3-8 个字符，建议按照命名规范为 **逻辑单元类型简称+数字** 的形式，如 **RZ03**。

- **删除**：单击删除图标并在弹出对话框中单击 **确定**，可以将已有的逻辑单元删除。

添加单元架构

逻辑单元 部署单元

\* 可用区  
上海可用区-D

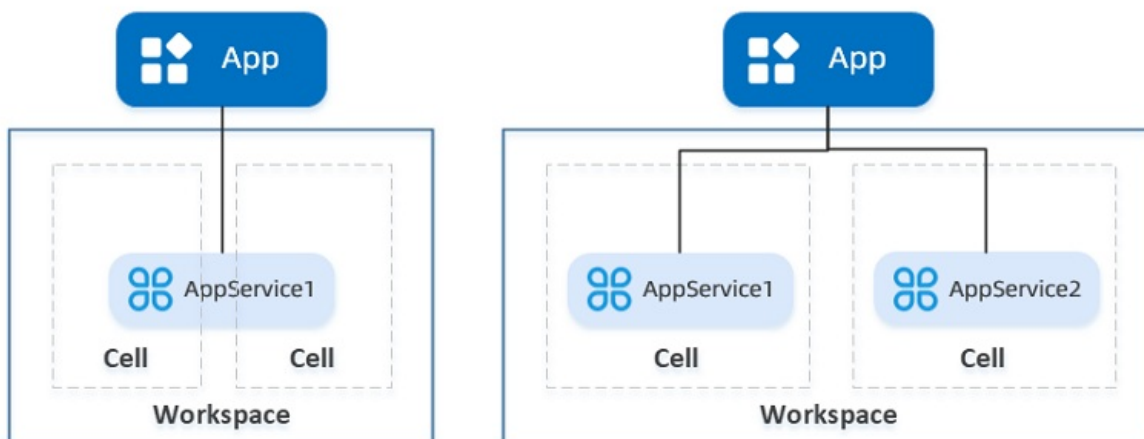
\* 逻辑单元名称  
RZ03

取消 确定

## 3.4. 创建部署单元

部署单元 Cell 是实现单元化的基础。

- 双机房双活场景下，一个应用的应用服务可以跨部署单元（Cell）部署。
- 单元化场景下，一个应用可以创建多个应用服务在不同部署单元（Cell）部署。



### 前提条件

已 [创建逻辑单元](#)。

### 操作步骤

#### ② 说明

在创建单元化工作空间时，系统会自动进行逻辑单元和部署单元的规划，参考 [创建单元化工作空间](#)。您可以根据实际需求，在发布应用服务前，对逻辑单元进行修改或增删。

1. 登录控制台，在左侧导航栏单击 **全局设置** 进入工作空间卡片页。

2. 单击需要修改逻辑单元的工作空间卡片进入工作空间详情页。
3. 选择 **单元架构** 页签，单击 **编辑**，架构拓扑进入可编辑的状态。



4. 您可以根据需要对部署单元进行以下增删操作：
  - **增加**：单击 **添加** 填写如下配置信息：
    - **逻辑单元**：选择所属的逻辑单元，支持 RZone 与 GZone 类型。
    - **可用区**：选择所属的可用区。若选择的逻辑单元为 RZone 类型，会自动选中关联的可用区。

- **部署单元名称**：自动生成逻辑单元名称作为前缀，填入字符长度限制为 1-10 个字符，建议使用 **逻辑单元+字母** 方式命名，如 `GZ00A`。



The image shows a configuration dialog box for a deployment unit. It has a title bar with a close button (X). Inside, there are two tabs: '逻辑单元' (Logical Unit) and '部署单元' (Deployment Unit), with the latter being selected and highlighted with a red box. Below the tabs, there are three required fields marked with an asterisk (\*):  
1. '逻辑单元' (Logical Unit): A dropdown menu showing 'RZ00'.  
2. '可用区' (Availability Zone): A dropdown menu showing '上海可用区-D'.  
3. '部署单元名称' (Deployment Unit Name): A text input field containing 'RZ00' and 'C', with a green checkmark icon to its right. This field is also highlighted with a red box.  
At the bottom right, there are two buttons: '取消' (Cancel) and '确定' (Confirm).

- **删除**：单击删除图标并在弹出对话框中单击 **确定**，可以将已有的部署单元删除。

## 3.5. 删除工作空间

您可以在 **控制台 > 全局设置** 中删除当前租户的工作空间。

### 前提条件

删除工作空间前，需要先清理当前工作空间的所有资源，包括 ECS、VPC、VSwitch、Security Group 资源，当所有资源都被回收之后，才可以删除工作空间，否则会报错。

### 操作步骤

1. 清理 ECS 资源。

在 **集群管理 > 节点列表** 页面选择所有节点，单击 **操作 > 删除**，删除所有节点后，保证当前工作空间的所有 ECS 都被回收。

#### ❓ 说明

清理工作空间前要先删除当前工作空间的集群，这样不会造成闲置集群资源的浪费，因为删除工作空间之后，产品页面上将无法查看到当前工作空间的集群信息，只能在数据库中的进行查找。

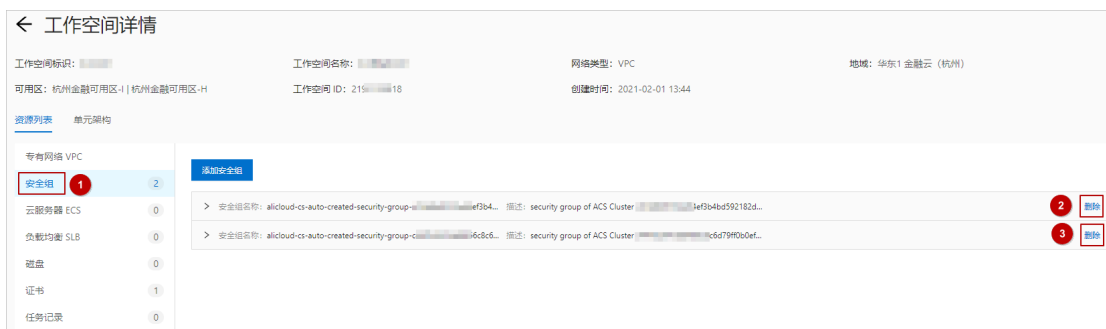
2. 删除交换机。

- i. 在 **控制台 > 全局设置** 中单击工作空间卡片，进入 **工作空间详情** 页。
- ii. 在 **资源列表** 页面，单击 **专有网络 VPC**，在 **交换机** 页签中删除所有交换机。



### 3. 删除安全组。

- 在 控制台 > 全局设置 中单击 工作空间，进入 工作空间详情 页。
- 单击 安全组 页面右侧的 删除。



### 4. 删除 VPC。

- 在 控制台 > 全局设置 中单击 工作空间，进入 工作空间详情 页。
- 单击 专有网络 VPC，删除当前工作空间关联的 VPC。



### 5. 删除单元架构。

- i. 在 控制台 > 全局设置 中单击 工作空间，进入 工作空间详情 页。
- ii. 选择 单元架构 页签，单击 编辑，架构拓扑进入可编辑的状态。
- iii. 单击删除图标，将已有的单元全部删除。





#### 6. 删除工作空间。


当 ECS、交换机、安全组、VPC、单元架构都删除成功之后，当前工作空间才可被删除，返回工作空间列表，单击要删除的工作空间卡片上的删除按钮即可。

#### ② 说明

删除工作空间会造成当前环境不可用，请谨慎操作。



工作空间类型：单元化

工作空间标识：

网络类型：VPC网络

地域：华东1 金融云（杭州）

可用区：杭州金融可用区-I | 杭州金融可用区-H

编辑

删除

## 4. 集群

### 4.1. 导入集群

LHC 对接 ACK 托管版集群后，您可以将 ACK 集群导入 LHC。本文介绍如何通过 LHC 控制台界面快速导入已经创建好的 ACK 集群。

#### 说明

- 目前，仅支持导入 ACK 标准托管版，且容器运行时必须为 Docker。
- 待导入的 ACK 集群的 VPC 必须与当前工作空间所属的 VPC 一致。
- 基于 ACK 集群的部署环境下，每个可用区（机房）允许导入两个 ACK 集群。

#### 前提条件

- 您拥有 LHC 管理员权限。
- 您已在 ACK 控制台创建了一个标准托管版集群，并且该集群所属的 VPC 与将导入的工作空间所属 VPC 相同。

#### 操作步骤

1. 登录 LHC 控制台。
2. 在左侧导航栏中选择 **集群管理 > 集群**，在集群列表页面，单击 **导入集群**。
3. 在 **导入集群** 页面，完成以下配置。

配置项	说明
集群类型	阿里云
专有网络	从已有 VPC 列表中，选择所需的 VPC。
选择已有集群	从已有集群列表中，选择您需要导入的 ACK 集群。
部署单元	可以为可用区设置不同的部署单元。 如需修改部署单元，可前往 <b>全局设置 &gt; 工作空间详情</b> 进行设置。

4. 单击 **提交**，即可导入该集群。

### 4.2. 扩容集群

集群扩容即扩容集群内的节点资源。您可以根据实际业务需要对集群的节点进行扩容。

若节点资源不足，会导致创建工作负载等操作失败。您可以通过以下两种方式进行扩容：

- [创建新节点](#)
- [添加已有节点](#)

## 前提条件

[已导入集群](#)

## 创建新节点

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理** > **集群**，进入集群列表页。
2. 单击目标集群卡片中的 **扩容集群**。



3. 在 **集群扩容** 页面，配置以下节点信息：
  - **虚拟交换机**：选择 ECS 机器所关联的交换机。
  - **付费类型**：支持 **按量付费** 和 **包年包月** 两种节点付费类型。选择 **包年包月** 时，需设置以下参数。
    - **购买时长**：目前支持选择 1、2、3、6 个月和 1~5 年。
    - **自动续费**：设置是否自动续费。
  - **伸缩数量（扩容）**：设置要扩容的机器数量，默认为 1。
  - **实例规格**：支持选择多个实例规格。详情请参见 [实例规格族](#)。
  - **系统盘**：支持 SSD 云盘和高效云盘。至少 40 GB。
  - **挂载磁盘**：支持 SSD 云盘和高效云盘。至少 40 GB。
  - **密码**：设置节点的登录密码。8~30 个字符，且同时包含三项（大写字母、小写字母、数字和特殊符号）。若遗忘可登录 ECS 控制台重置密码。
  - **云监控插件**：选择是否在 ECS 节点上安装云监控插件，安装后可以在云监控控制台查看所创建 ECS 实例的监控信息。
  - **标签**：为节点绑定标签，可通过标签来筛选节点。输入键和对应的值，单击 **添加一行数据**，在弹出框中输入 key 和 value。
  - **污点（Taints）**：为 Worker 节点添加污点。单击 **添加一行数据**，在弹出框中输入变量名、值，选择 Effect。更多详情参见 [官方文档](#)。
  - **部署单元**：保持默认配置。

- **统一接入节点**：勾选后，扩容的节点将用于网络统一接入集群节点资源。
4. 完成配置后，单击 **提交**。

在集群详情页面，如果集群状态显示扩容中，则说明节点正在扩容中。扩容完成后，集群状态显示为运行中。

## 添加已有节点

您可以将工作空间中已有的处于 **运行中** 状态的 ECS 加入到集群。系统会从当前工作空间中筛选出之前以 **清除机器环境** 方式删除的 ECS，以及从当前工作空间中筛选出符合条件的 ECS。

### 说明

您需要将待添加的 ECS 实例添加到集群自动创建的节点安全组内。

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理**。
2. 在 **集群详情** 页，单击 **节点** 页签，进入节点列表页面。
3. 选择目标节点，单击 **添加已有节点**，完成以下 **集群扩容** 配置。

实例 ID	实例名称	IP 地址 (ID)	可用区	实例规格	Terway兼容性 (可支持Pod数量)
i-bj...	v5fr3	10.0.0.180	杭州金融可用区-I	ecs.c5.large	不支持
i-be...	6hopld	10.0.0.179	杭州金融可用区-I	ecs.c5.large	不支持
i-bj...	v72us	10.0.0.178	杭州金融可用区-I	ecs.c5.large	不支持

- **添加方式**：默认为自动添加。自动添加节点方式会列出当前账号下可用的 ECS 云服务器，在 Web 界面进行安装部署，并自动添加到集群。
  - **选择已有云服务器**：勾选要添加的 ECS。
4. 单击 **下一步**，填写以下实例信息。
- **数据盘挂载**：设置是否将容器和镜像存储在数据盘。

### 说明

- 如果 ECS 已挂载数据盘，且最后一块数据盘的文件系统未初始化，系统会自动将该数据盘格式化为 ext4，用来存放内容 /var/lib/docker、/var/lib/kubelet。数据盘内原有数据将丢失，请注意备份数据。
- 如果 ECS 未挂载数据盘，则不会挂载新的数据盘。

- **CPU Policy**：设置 CPU Policy。
  - **none**：默认策略，表示启用现有的默认 CPU 亲和方案。

- **static**：允许为节点上具有某些资源特征的 Pod 赋予增强的 CPU 亲和性和独占性。
  - **密码**：设置节点的登录密码。
  - **确认密码**：确认密码。
  - **部署单元**：使用默认配置。
  - **标签**：为节点绑定标签。输入键和对应的值，单击 **添加**，在弹出框中输入 **变量名**、**值**。
5. 完成配置后，单击 **提交**。

## 4.3. 查看集群详情

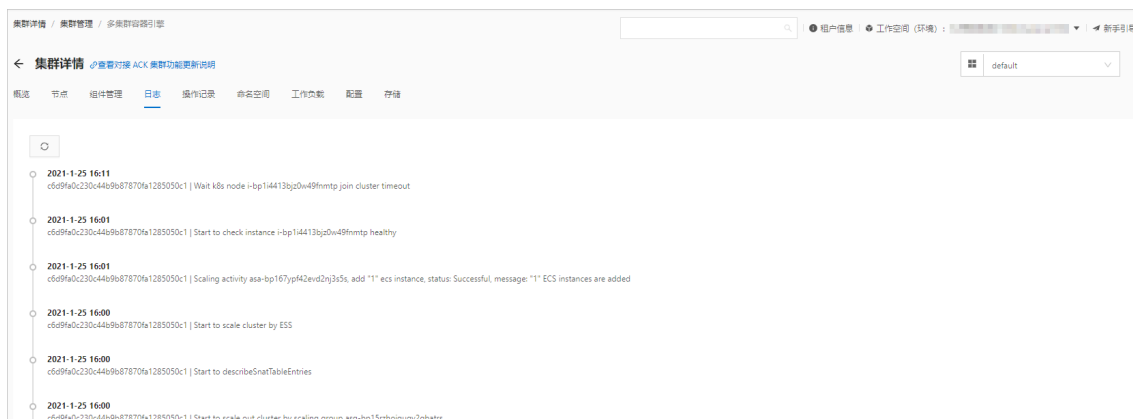
您可以在 LHC 控制台上查看租户下所有的集群信息。

### 前提条件

已导入集群

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，查看以下信息：
  - **概览**：
    - **容器集群**：包含集群的基本信息，例如：集群名称、集群 ID、集群版本、网络类型、容器运行时、服务网格开关状态。日志服务、节点总数（健康数、不健康数）、CPU 总核数（已用、未用）、内存总量（已用、未用）。
    - **集群资源**：CPU 使用率、内存使用率。
    - **配置信息**：网络配置包含：网络类型、网络插件、Service/Pod 网段、公网访问地址、配置 SNAT 状态、删除保护状态。
    - **操作记录**：集群操作记录概览。
  - **节点**：包含节点名称/IP、可用区/部署单元、节点状态、调度状态、Pod 数、配置（CPU/内存）、创建时间等信息。
  - **组件管理**：集群的组件管理，主要分为系统组件和可选组件。详情参见 [管理组件](#)。
  - **日志**：查看集群日志记录。



- **操作记录**：查看集群的历史操作记录。可根据操作类型、操作状态、日期字段进行筛选查找。

- 命名空间：可以查看命名空间名称、状态等信息。

#### 说明

命名空间相关操作，请参见 [创建命名空间](#)。

- 工作负载：即 Kubernetes 对一组 Pod 的抽象模型，包括 Deployments、Statefulsets、Daemonsets 等多种类型。在该页面可以查看工作负载名称、镜像、Pod 实际数目等信息。

#### 说明

工作负载相关操作，请分别参见 [部署（Deployments）](#)、[有状态副本集（Statefulsets）](#)、[守护进程集（Daemonsets）](#)、[容器组（Pods）](#)。

- 配置：可以查看名称、命名空间、创建时间等信息。

#### 说明

配置相关操作，请分别参见 [管理配置项](#)、[管理保密字典](#)。

- 存储：为工作负载提供存储的组件。可以查看 Pod 信息、事件类型、事件原因、事件内容等信息。

集群详情

查看对接 ACK 集群功能更新说明

default

概览 节点 组件管理 操作记录 命名空间 工作负载 配置 存储

存储类型(StorageClass)

存储卷(PersistentVolume)

存储卷声明(PersistentVolumeClaim)

创建存储类型

如何创建存储类型

搜索 StorageClass 名称

刷新

名称	存储驱动	回收策略	绑定模式	创建时间
sc0 device	yodaplugin baba...	Delete	Immediate	2021-6-01
sc0 lvm	yodaplugin baba...	Delete	Immediate	2021-5-31
sc0 device	yodaplugin baba...	Delete	Immediate	2021-5-31
sc0 003	yodaplugin baba...	Delete	WaitForFirstConsumer	2021-5-31
sc0 002	yodaplugin baba...	Delete	WaitForFirstConsumer	2021-5-31

#### 说明

存储相关操作，请分别参见 [存储类型（StorageClass）](#)、[存储卷（PersistentVolume）](#)、[存储卷声明（PersistentVolumeClaim）](#)。

## 4.4. 解绑集群

本文介绍如何将集群与 LHC 控制台解绑。解绑集群后，您将无法在 LHC 控制台上管理该集群，但该集群不会被删除。

#### ② 说明

- 解绑集群是高危操作，会导致当前环境不可用，请谨慎操作。
- 集群解绑后如想继续使用，可通过集群导入操作将集群加入当前工作空间，详情请参见 [导入集群](#)。

### 前提条件

- [已导入集群](#)。
- 您拥有 LHC 管理员权限。

### 操作步骤

1. 登录 LHC 控制台。
2. 在左侧导航栏中选择 **集群管理 > 集群**，进入集群列表页面。
3. 单击某个集群卡片中的 **解绑集群**。
4. 在 **解绑集群** 弹框中，输入 **解绑集群** 完成校验。
5. 单击 **确认**。

## 4.5. 节点管理

### 4.5.1. 节点调度

您可以通过控制台设置节点调度状态，从而合理分配各节点的负载。

#### 前提条件

[已导入集群](#)

#### 暂停调度

节点暂停调度后，在后续进行应用部署时，Pod 不会再调度到该节点。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入该集群的 **集群详情** 页。
3. 单击 **节点** 页签，进入节点列表页面。
4. 勾选目标节点，单击 **更多操作 > 暂停调度**，在弹出框中单击 **确定**。

返回到节点详情页，可以看到节点状态将变为 **暂停调度**。



## 恢复调度

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入该集群的 **集群详情** 页。
3. 单击 **节点** 页签，进入节点列表页面。
4. 勾选目标节点，单击 **更多操作 > 恢复调度**，在弹出框中单击 **确定**。

返回到节点详情页，可以看到节点状态将变为 **可调度**。



## 4.5.2. 节点规划

随着集群种类的多样化以及应用场景的多元化，您可能需要对导入的原生 K8s 集群或其内节点进行部署单元打标。本文介绍如何通过控制台将节点与部署单元进行关联。

当导入 K8s 集群或扩容集群内的节点资源时，您可以通过以下方式对节点进行部署单元关联：

- [关联集群部署单元](#)
- [关联节点部署单元](#)

### 前提条件

[已导入集群](#)。

### 关联集群部署单元

由于集群在导入过程中不直接进行部署单元打标，您可以通过以下步骤建立或解除集群与部署单元间的绑定关系。

1. 登录 LHC 控制台。
2. 在左侧导航栏中选择 **集群管理 > 集群**，在集群列表页面，单击 **解绑/绑定**。



3. 在 **解绑/绑定** 页面，完成以下配置后，单击 **提交**。
  - **选择集群**：选择需要解绑的集群。
  - **部署单元列表**：根据需要勾选或取消勾选部署单元。
    - **勾选**：如需将集群与部署单元关联，可勾选一个或多个要绑定的部署单元。

#### 说明

如果部署单元无法勾选，说明该部署单元可能已被其他集群绑定，建议选择其他部署单元。

- **取消勾选**：如需将集群与部署单元解绑，可取消勾选一个或多个要解绑的部署单元。

#### 说明

解绑前，须确保该部署单元下无任何节点，否则可能会出现报错信息。

## 关联节点部署单元

当集群进行了扩容节点时，您可以通过以下步骤对节点进行部署单元打标。

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群卡片，进入 **集群详情** 页。
3. 单击 **节点** 页签，进入节点列表页面。
4. 选择节点，单击 **更多操作 > 节点规划**。



5. 在节点规划 页，完成以下配置后，单击 提交。

- 部署单元：选择需要关联的部署单元。更换节点部署单元后，对于运行在当前节点的业务容器，会在下次发布时重新调度到其他节点，请谨慎操作。

#### 说明

节点部署单元设置范围与集群部署单元设置范围保持一致。例如，如果某一集群的部署单元是 Cell A 和 Cell B，则节点部署单元只能选择 Cell A 或 Cell B。

- spanner 节点：默认不勾选。勾选后，会打上 spanner 统一接入集群的标记。

## 4.5.3. 管理节点标签

节点标签可以给节点定义不同的属性，通过这些标签可以快速的了解各个节点的特点，便于节点的管理和工作负载的调度。

### 前提条件

已导入集群

### 添加标签

- 登录控制台，在左侧导航栏单击 集群管理 > 集群，进入集群列表页。
- 单击某个集群卡片，进入 集群详情 页。
- 单击 节点 页签，进入节点列表页面。
- 选择节点，单击 更多操作 > 编辑标签。
- 在 添加标签 页面，单击 添加一行数据，输入变量名和值，单击 提交。

### 删除标签

删除标签只能够删除用户自定义的标签，节点固有的标签不能删除。

- 登录控制台，在左侧导航栏单击 集群管理 > 集群，进入集群列表页。
- 单击某个集群卡片，进入 集群详情 页。
- 单击 节点 页签，进入节点列表页面。
- 选择节点，单击 更多操作 > 编辑标签。

5. 在 **添加标签** 页面，单击标签右侧的 **删除**。

## 4.5.4. 管理 Taint

Taint（污点）用于给节点做污点标记，使 Pod 排斥一类特定的节点，每个节点上都可以应用一个或多个污点。本部分介绍如何添加和删除 Taint。

### 前提条件

[已导入集群](#)

### 添加 Taint

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群卡片，进入 **集群详情** 页。
3. 单击 **节点** 页签，进入节点列表页面。
4. 选择节点，单击 **更多操作 > 编辑 Taint**。
5. 在 Taint 列表中，单击 **添加一行数据**，在弹出框中输入 **键**、**值**，选择 **Effect**，单击 **确定**。Effect 包含以下选项：
  - **NoSchedule**：不允许无匹配 toleration 的 Pod 调度到该节点。
  - **NoExecute**：若无匹配 toleration 的 Pod 已经在节点上运行，则将 Pod 驱逐，若无匹配 toleration 的 Pod 尚未在节点上运行，则不会将 Pod 调度到该节点上。
  - **PreferNoSchedule**：尽量不将无匹配 toleration 的 Pod 调度到该节点。

更多详情参见 [官方文档](#)。

### 删除 Taint

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群卡片，进入 **集群详情** 页。
3. 单击 **节点** 页签，进入节点列表页面。
4. 选择节点，单击 **更多操作 > 编辑 Taint**。
5. 在 Taint 列表中，单击 Taint 右侧的 **删除**。

## 4.5.5. 查看节点详情

集群导入完成后，如需查看集群中节点的详细信息，您可以通过 LHC 控制台查看节点信息，包括节点服务器 IP、节点创建时间、Pod 列表以及节点事件等信息。

#### ② 说明

从对接 ACK 集群的功能上线时起（2020 年 8 月 15 号），阿里云底座上新建集群默认为 ACK 托管版集群。

### 前提条件

[已导入集群](#)

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页。
3. 单击 **节点** 页签，进入 **节点列表** 页面。
4. 在 **节点列表** 中，单击 **节点名称** 进入节点详情页面。您可以查看以下信息。

信息模块	具体信息
基本信息	节点服务器 IP
	节点的创建时间
	调度状态： <ul style="list-style-type: none"><li>◦ 可调度状态</li><li>◦ 不可调度状态</li></ul>
	机器详情：单击详情链接跳转至工作空间资源管理页面，可查看 ECS 详情。
	注释：系统注释信息
	系统定义的节点标签，可用于节点分组。
Pod 列表	节点所包含的容器列表，包括 Pod 名称、IP、命名空间、状态、创建时间。单击 Pod 名称跳转到 Pod 详情页。
节点事件	反映节点生命周期和运行状况的事件日志，包含事件类型、事件原因、事件内容。 事件类型分为 Normal、Warning、Error。您可以查看 Warning 和 Error 类型事件进行故障排查。
YAML	节点的 YAML 文件，可以一键复制。

## 4.5.6. 删除节点

当您不再需要某个节点继续工作时，可以将其移除。本文介绍如何通过 LHC 控制台删除节点。

### ② 说明

从对接 ACK 集群的功能上线时起（2020 年 8 月 15 号），阿里云底座上新建集群默认为 ACK 托管版集群。

### 前提条件

## 已导入集群

 注意

- 删除节点会自动驱逐 Pod 引起 Pod 迁移，可能会影响业务，建议在业务低峰期操作。
- 操作过程中可能存在非预期风险，请提前做好相关的数据备份。

## 操作步骤

1. 在左侧导航栏，单击 **集群管理 > 集群**，进入集群列表页面。
2. 单击集群名称进入该集群的详情页，单击 **节点** 页签，进入节点列表页面。
3. 勾选待删除节点，单击 **更多操作 > 删除**。
4. 在弹出窗口中，选择以下删除方式，单击 **确定**。
  - **自动排空节点 (drain)**：把待移除节点上的 Pod 转移到其他节点。请确保集群其他节点的资源充足。您还可以通过执行命令 `kubect1 drain node-name` 的方式把待移除节点上的 Pod 转移到其他节点。

 说明

node-name 格式为 `your-region-name.node-id`。例如 `cn-hangzhou.i-xxx`。

`your-region-name` 为您集群所在的地域名称。node-id 为待移除节点所在的 ECS 实例 ID。

- **同时释放 ECS**：彻底删除服务器。
  - 释放 ECS 实例仅释放按量付费的 ECS 实例。

 说明

- 如果节点上存在 Pod，请选择 **自动排空节点**，并确保集群资源充足，Pod 将自动迁移到其他节点。
- 释放 ECS 仅释放按量付费类型的 ECS 实例，不被释放的 ECS 会继续收费。
- 移除节点是异步操作，请在节点列表中查看移除进度。

- 对于预付费 ECS 实例，计费周期到期后，ECS 实例会自动释放。

- 您也可以在 ECS 实例到期前：
  - 申请退款，提前释放实例，请参见 [退款规则及退款流程](#)。
  - 将计费方式转为按量付费后释放实例，请参见 [包年包月转按量付费](#)。
  - 若不选择同时释放 ECS，该节点所在的 ECS 实例会继续计费。

#### 说明

- 如果节点上存在 Pod，请选择 **自动排空节点**，并确保集群资源充足，Pod 将自动迁移到其他节点。
- 释放 ECS 仅释放按量付费类型的 ECS 实例，不被释放的 ECS 会继续收费。
- 移除节点是异步操作，请在节点列表中查看移除进度。

## 4.6. 命名空间

### 4.6.1. 创建命名空间

命名空间（Namespace）是对一组资源和对象的抽象整合。您可以跨集群或者在同一个集群内创建不同的命名空间，不同命名空间中的数据彼此隔离，使得它们既可以共享服务，也能够互不干扰。例如，将开发环境、测试环境的业务分别放在不同的命名空间。

#### 命名空间类别

LHC 的命名空间分为以下两类：

- 联邦命名空间（Federated Namespace）：跨集群的命名空间，需要自定义。
- 单集群命名空间（Namespace）：在单个集群中，您可以按照需要创建命名空间，例如开发环境、联调环境和测试环境分别创建对应的命名空间。或者按照不同的业务创建对应的命名空间，例如，系统分为登录和游戏，可以分别创建对应命名空间。LHC 默认会创建以下集群内的系统命名空间。
  - **default**：在不指定命名空间时，默认使用 default。
  - **kube-public**：用来部署公共插件、容器模板等。
  - **kube-system**：部署系统组件。
  - **sigma-operator-machine-conditions**：节点运维所需的系统命名空间。
  - **operator-extensions**：节点运维所需的系统名称空间。
  - **antstack**：系统应用所在的命名空间。仅存在于 AntStack 底座的专有云上。

#### 前提条件

[已导入集群](#)

#### 创建联邦命名空间（Federated Namespace）

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 命名空间**，进入命名空间列表页。
2. 单击 **创建** 进入 **创建命名空间** 页面，输入以下信息，单击 **创建**。
  - **名称**：输入命名空间名称。默认名称的前缀为 `租户名-工作空间组标识-`。

- **添加标签**：为命名空间添加自定义标签。

## 创建单集群命名空间（Namespace）

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **命名空间** 页签。
3. 单击 **创建命名空间**，在 **创建命名空间** 页面，输入以下配置信息：
  - **名称**：命名空间名称。
  - **标签**：为命名空间添加自定义标签。
4. 完成配置后，单击 **提交**。

## 4.6.2. 查看命名空间详情

您可以通过控制台查看命名空间详情。

### 查看联邦命名空间

1. 登录控制台，在左侧导航栏单击 **集群管理 > 命名空间**，进入命名空间列表页。

命名空间(Namespace)				
<div>创建</div> <div>命名空间名称</div>				
名称	标签	状态	创建时间	操作
tfh-9-default	sofastack.io/tenant:TF sofastack.io/workspace-group:LHC	c061cd1b...47e2914f3... c91b16e...3ce1b05f9... 4f3ac8a:create success c91b16e...3ce1b05f9...:create success	2021/02/19 17:09:08	删除
tfh-9-lit	sofastack.io/tenant:TF sofastack.io/workspace-group:LHC	c061cd1b...47e2914f3... c91b16e...3ce1b05f9... 3ac8a:create success c91b16e...3ce1b05f9...:create success	2021/02/20 11:23:04	删除

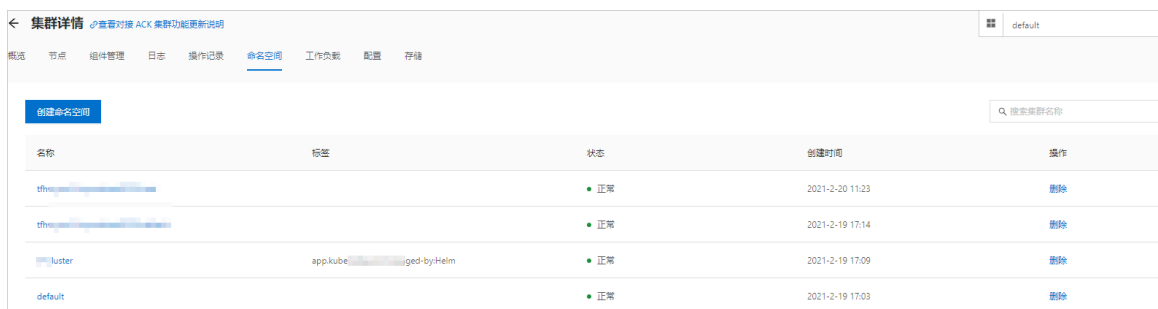
2. 单击命名空间名称，进入 **命名空间详情** 页。您可以查看如下信息：

命名空间详情-tfh-9-default		
<div>基本信息</div> <div>命名空间名称 tfh-9-default</div> <div>集群 c061cd1b...47e2914f3... c91b16e...3ce1b05f9...</div> <div>状态 c061cd1b...47e2914f3... c91b16e...3ce1b05f9...</div> <div>创建时间 2021/02/19 17:09:08</div>		
变量名	值	操作
sofastack	TF	
sofastack-ace-group	LHC-219	

- **基本信息**
  - 命名空间名称
  - 集群
  - 状态
  - 创建时间
- **标签信息**
  - 变量名
  - 值

## 查看单集群命名空间

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **命名空间** 页签。



3. 单击命名空间名称，进入命名空间详情页。您可以查看如下信息：



- 基本信息
  - 命名空间名称
  - 状态
  - 创建时间
- 标签信息
  - 变量名
  - 值

## 4.6.3. 删除命名空间

### 注意

- 删除命名空间将会释放该命名空间下的所有资源。
- 请勿删除系统命名空间。

## 删除联邦命名空间

1. 登录控制台，在左侧导航栏单击 **集群管理 > 命名空间**，进入命名空间列表页。
2. 单击命名空间右侧 **操作** 栏中的 **删除**，在弹出的对话框中单击 **确认**。系统会弹出删除命名空间成功的提示。

## 删除单集群命名空间

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **命名空间** 页签。
3. 单击命名空间右侧 **操作** 栏中的 **删除**，在弹出的对话框中单击 **确认**。

系统会弹出删除命名空间成功的提示。

## 4.6.4. 设置资源配额

通过设置命名空间级资源配额，实现多团队或多用户在共享集群资源的情况下限制团队或用户可以使用的资源总量，包括限制命名空间下创建某一类型对象的数量以及对象消耗计算资源（CPU、内存）的总量。本文介绍如何通过控制台设置资源配额。

### 说明

目前，仅单集群命名空间支持设置资源配额。

## 背景信息

默认情况下，运行中的 Pod 可无限制使用节点（Node）上的 CPU 和内存，即任何一个 Pod 都可以无节制地使用集群的计算资源，导致某个命名空间的 Pod 可能会耗尽集群的资源。

命名空间的一个重要的作用是充当一个虚拟的集群，用于多种工作用途，满足多用户的使用需求。因此，为命名空间配置资源额度是一种最佳实践。您可以为命名空间配置包括 CPU、内存、Pod 数量等资源的额度。

更多信息请参见 [Resource Quotas](#)。

## 工作原理

资源配额（Resource Quotas）是对命名空间进行资源配额，限制资源使用的一种策略。

- 资源配额应用在命名空间上，可以为每个命名空间创建一个或多个资源配额对象。
- 开启计算机资源限制（比如 CPU 或内存限制）后，创建容器组时必须配置计算资源请求或限制。
- 若使用的资源用量超过设定的资源配额，将禁止创建新的资源。

## 前提条件

- [已导入集群](#)
- [已创建单集群命名空间](#)

## 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **命名空间** 页签。
3. 单击目标命名空间右侧 **操作** 栏中的 **资源配额**。
4. 在 **资源配额** 对话框中，根据需要完成资源配额配置。如果不勾选，则代表不限制该资源的使用。

## i. 完成计算机资源限制配置。

## ② 说明

对命名空间设置 CPU 或内存配额后，创建容器组时，必须指定 CPU 或内存资源限制。详情请参见 [Resource Quotas](#)。

配置项	说明	单位
CPU 限制	用于限制命名空间下 Pod 能申请到的 CPU 资源的最大值。	核
内存限制	用于限制命名空间下 Pod 能申请到的内存资源的最大值。	Mi、Gi、Ti、M、G、T

## ii. 完成存储资源限制配置。

配置项	说明	单位
存储空间	用于限制命名空间下允许的存储空间上限。	Mi、Gi、Ti、M、G、T
存储声明数量	用于限制命名空间下允许的存储声明的最大数量。	个

## iii. 完成其他资源限制配置。

配置项	说明	单位
配置文件数量	该命名空间下允许存在的配置文件数量上限。	个
容器组数量	该命名空间下允许存在的容器组数量上限。	个
服务数量	该命名空间下允许存在的服务数量上限。	个
负载均衡型服务数量	该命名空间下允许存在的负载均衡型服务数量上限。	个
保密字典数量	该命名空间下允许存在的保密字典数量上限。	个

5. 完成资源配额配置后，单击 **确定**。

资源配额

计算机资源限制

☒ CPU限制

最大使用量:

4

核

☒ 内存限制

最大使用量:

4T

②

存储资源限制

☒ 存储空间

最大使用量:

5G

②

☒ 存储声明数量

最大使用量:

60

个

其他资源限制

☒ 配置文件数量

最大使用量:

200

个

☒ 容器组数量

最大使用量:

60

个

☒ 服务数量

最大使用量:

38

个

☒ 负载均衡型服务数量

最大使用量:

10

个

☒ 保密字典数量

最大使用量:

20

个

取消

确定

## 4.7. 管理组件

集群 AddOn 组件是单元化应用服务提供的扩展功能包，用于管理集群的扩展功能。您可以根据业务需求选择所需的组件。

### 安装组件

#### 前提条件

已导入集群

#### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群卡片，进入 **集群详情** 页。
3. 单击 **组件管理** 页签，进入组件列表页面。
4. 找到需要安装的组件，单击所需组件卡片右下角的 **安装**。关于组件说明可参考下表。

组件名称	功能	是否必选	是否推荐安装
是			
cafeextcontroller	CAFE 发布部署核心组件，用于应用发布部署。	是	导入集群过程中会自动安装。
cloud-controller-manager	使用 Cloud Controller Manager 为 K8s 应用创建负载均衡及统一接入路由管理。	是	导入集群过程中会默认安装。
pontus-pontusagent	监控 Agent 组件，Daemonset 方式部署。	否	使用 <a href="#">监控产品 RMS</a> 时必须安装，否则无需安装。
chaosblade-operator	面向云原生领域的混沌实验注入工具，配合金融云高可用管理平台使用，可以实现针对基础资源、用户应用、蚂蚁产品的故障演练。	否	使用 <a href="#">高可用管理产品 HAS</a> 时必须安装，否则无需安装。

## 执行结果

组件安装完成后，卡片右上角的状态会变为 **已安装**。



## 重装组件

当组件运行异常时，您可以使用重装功能。

## 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群卡片，进入 **集群详情** 页。
3. 单击 **组件管理** 页签，进入组件列表页面。
4. 找到需要重装的组件，单击组件卡片的 **重装 > 确定**。

## 升级组件

 注意

- 请务必通过 **组件管理 > 升级** 对集群组件进行升级，否则可能会带来不可预知的问题。
- 升级组件前，请确保集群内所有节点均处于运行中状态，否则可能会导致组件升级失败。
- 组件升级过程中，请勿发布应用，可能出现非预期问题。

## 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群卡片，进入 **集群详情** 页。
3. 单击 **组件管理** 页签，进入组件列表页面。
4. 找到待升级的组件，单击组件卡片的 **升级**。

# 5.应用服务

## 5.1. 创建应用服务

LHC 应用服务对 Kubernetes 原生的 Deployment 做了能力增强，您可以通过创建应用服务定义容器服务的基本信息、访问策略、发布及调度策略等信息，为后续容器服务的部署做准备。

### 前提条件

- [已创建应用](#)
- [已导入集群](#)

创建应用服务的过程分为以下 6 个步骤：

1. [填写基本信息](#)
2. [Pod 模板配置](#)
3. [弹性配置](#)
4. [访问配置](#)
5. [部署和调度配置](#)
6. [预览并提交](#)

### 一、填写基本信息

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击 **创建应用服务**。
3. 在 **创建应用服务** 页面，填写以下基本信息，单击 **下一步**。
  - **命名空间**：选择命名空间，默认为列表第一个。
  - **应用服务名称**：容器服务的名称。服务实例名称允许包含（小写）字母、数字、连字符，且必须以字母开头，以字母或数字结尾。同一个工作空间组下不允许同名。
  - **所属应用**：选择一个该容器服务所关联的应用。
  - **有状态模式**：默认关闭。开启后，Pod 名称会加上 `-0`、`-1` ... 数字后缀，重建过程中不会生成新 Pod 名称。相关操作，请参见 [LHC Pod 域名配置](#)。
  - **数据卷模版配置**：可选，用于挂载存储。单击 **添加配置** 可设置多个。具体配置项如下：
    - **名称**：输入模板名称。允许包含字母、数字、连字符，且必须以字母开头，以字母或数字结尾。
    - **存储类型**：用于指定支持的存储类别。单击 **创建存储类型** 将跳转至存储页。

#### ② 说明

- 对于存储类型，通常由系统管理员定义，具体类别因存储 SLA、备份策略等而异。
- 对于多集群下的存储类型，需要多集群下都存在此存储类型才可以被调用。

- 容量：设置数据卷模板的容量。单位为 GiB。

② 说明

数据卷容量大小受设置的 PV 容量的限制。关于 PV 容量设置，请参见 [创建存储卷](#)。

- 描述：选填。容器服务的描述。

## 二、Pod 模板配置

对 Pod 里的容器做详细的配置。Pod 与容器之间的比例并非 1:1，可通过单击 [添加应用容器](#) 为单 Pod 配置多个容器。

② 说明

对于可添加的容器数量，暂无限制。

2

Pod 模板配置

3

弹性配置

4

访问配置 (选填)

pod-ter

+ 添加应用容器

\* 容器名称

pod-ter

镜像选择

镜像仓库

构建记录

\* 镜像地址

cr.registry.inter.ter:20210901211441-nfb8a

使用保密字典

☐

[创建保密字典](#)

\* CPU 配置 ①

请求核数

1

core(s)

最大核数

2

core(s)

\* 内存配置 ①

请求内存

1

GiB

最大内存

2

GiB

启动命令

/home/admin/int.sh

### 基本选项

- 容器名称：容器名称。

- **镜像来源**：支持 **镜像仓库**、**构建记录** 两种方式。

- **镜像仓库**：直接输入镜像仓库地址，比如：`registry-cnzh.cloud.alipay.com/aks/nginx:1.8`。

#### ② 说明

如果配置多个容器，输入的镜像地址不得相同。

- **构建记录**：选择该类型来源时，会自动关联应用服务所属应用的构建，具体参见 [镜像构建](#)。

#### ② 说明

目前暂不支持 **构建记录**。如有需要，请提交工单进行处理。

- **使用保密字典**：若该镜像属于私有镜像，需要访问认证，您可以开启 **使用保密字典** 拉取镜像。若尚未创建保密字典，可单击创建保密字典，详情请参见 [创建保密字典](#)。
- **CPU 配置**：设置容器使用的 CPU 的数量。**请求核数** 为能保证的最小核数，**最大核数** 为能使用的最大核数。换算方式：1 core = 1000 millicores。
- **内存配置**：设置容器使用的内存的数量。**请求内存** 为能保证的最小内存数量，**最大内存** 为能使用的最大内存数量。换算方式：1024 Bytes = 1 KiB；1024 KiB = 1 MiB；1024 MiB = 1 GiB；1024 GiB = 1 TiB。
- **启动命令**：选填。用于指定容器启动时执行的命令。

## 高级配置

默认关闭。如果开启，需要完成以下选项配置。

- **hostNetwork 配置**：默认关闭。开启后，Pod IP 地址与节点 IP 地址保持一致。在 host 模式下，一个节点最多只能挂载 1 个 Pod。
- **环境变量配置**：设置在容器启动时传入应用进程的键值对，例如：`USER=tester`。
- **数据卷配置**：配置容器使用的数据卷，目前仅支持挂在当前容器所在宿主机的目录。
- **健康检查配置**：包含两种检查机制：Readiness 和 Liveness。具体参见 [Kubernetes Probe](#)。
- **生命周期事件回调配置**：为容器添加生命周期事件回调，分别在容器启动后和容器停止前执行。
- **日志服务配置**：配置日志服务（SLS），可选择已有日志库或创建新的日志库。

#### ② 说明

首次使用日志服务配置功能，需前往 [日志服务控制台](#) 开启配置索引。详情请参见 [配置索引](#)。

更多信息参见 [高级配置](#)。

## 配置覆盖

默认关闭。如果开启，需要完成以下选项配置。

- **环境变量配置**：您可以为每个部署单元单独设置环境变量，覆盖应用服务中的默认配置，但仅针对于手动输入类型的环境变量。
- **镜像地址**：您可以为每个部署单元单独配置不同的镜像地址，覆盖应用服务中的默认配置。

### 三、弹性配置

副本伸缩策略配置：支持以下两种弹性策略。

- 固定副本数：默认为 0，可勾选部署单元并修改为期望副本数，即应用服务运行时保持固定数目的 Pod 副本。
- 弹性扩缩容：可对应用服务自动伸缩进行如下弹性规则配置。

\* 副本伸缩策略配置

固定副本数

弹性扩缩容(体验)

\* 固定副本数 ①

☒

GZ00B

2

☐

RZ00A

0

☒

GZ00A

3

☐

RZ01A

0

\* 最小副本数 ①

1

\* 最大副本数 ①

10

\* 规则 ①

+ 添加配置

高级配置

☐ 隐藏

下一步

上一步

预览并提交

- 固定副本数：默认为 0，可勾选部署单元并修改为期望副本数。
- 最小副本数：弹性扩缩容时 Pod 的最小副本数。
- 最大副本数：弹性扩缩容时 Pod 的最大副本数。
- 规则：当前伸缩规则基于的指标类型。单击 + 添加配置，配置以下选项。

\* 规则 ①

\* 规则类型

CPU

▼

\* 目标类型

使用率

▼

\* 使用率/平均值

输入使用率/平...

\* 单位

%

操作

删除

+ 添加配置

- 规则类型：当前伸缩规则基于的规则类型，目前支持 CPU、Memory、QPS、响应时间。
- 目标类型：当前伸缩规则基于的指标类型所期望达到的值的类型，目前支持 使用率 与 平均资源使用。
- 使用率/平均值：输入使用率/平均值。
- 单位：无需输入，根据所选的规则类型自动匹配。

- 高级配置：默认关闭。如果开启，需要完成以下选项配置。

高级配置 ☒

扩容规则 缩容规则

幅度: 百分比 100

间隔时间: 15 秒

冷却时间: 500 秒

- **幅度**：可以通过设置按百分比或指定个数来调整扩缩容的速率，单位为百分比或个数。
- **间隔时间**：上次扩缩容操作与本次扩缩容操作之间的时间间隔，单位为秒。
- **冷却时间**：冷却扩缩容操作的时间，即完成一次扩缩容操作之后不再次触发扩缩容操作的时间窗口，单位为秒。

#### ? 说明

一个扩缩容活动执行完成后，在设定的冷却时间内不执行其他扩缩容活动。

## 四、访问配置（选填）

应用服务支持统一接入及负载均衡两种访问方式。负载均衡是基于端口的请求负载均衡，统一接入是基于规则的请求负载均衡。您可以根据业务需要做好规划。

### 负载均衡

您可以在创建应用服务时设置访问方式，也可以应用服务创建完成后添加访问方式。

#### 创建应用服务时设置访问方式

- 在 **访问配置** 页面，单击 **添加负载均衡**。
- 在 **负载均衡** 配置中，填写以下信息后，单击 **下一步**。
  - **负载均衡名称**：填写服务名称。系统默认生成服务名称前缀为 `应用服务名称-`。服务名称允许包含（小写）字母、数字、连字符且必须以字母开头、以字母或者数字结尾。
  - **访问方式**：选择负载均衡服务的访问方式。支持以下选项：
    - **内网**：创建一个内网的 LoadBalancer，并将流量转发到容器的相应端口上。
    - **外网**：创建一个公网的 LoadBalancer，并将流量转发到容器的相应端口上。访问方式由公网负载均衡服务地址以及设置的访问端口组成，例如 `10.117.117.117:80`。

- **自定义**：在自定义模式下，会在选定的部署单元内分别创建一个 LoadBalancer 类型的 Service，通过用户填入的 ID 复用已有的负载均衡实例。

 注意

- 当前仅支持填入阿里云负载均衡实例 ID。更多合作伙伴型号支持，敬请期待。
- 阿里云场景下，被复用的 SLB 必须要满足如下条件：
  - 确保被复用的 SLB 实例是在 SLB 控制台手动创建的。
  - 确保期望使用的监听器端口尚未被占用。
  - 如果是内网地址类型的 SLB 实例，须确保与部署单元对应的集群同属于同一个 VPC。

- **部署单元 ID**：仅当访问方式选择 **自定义** 时显示该选项。为各部署单元填写对应的负载均衡实例 ID。

 说明

- 对于同一应用服务下的不同部署单元，支持设置相同的负载均衡实例 ID。
- 确保应用服务端口映射中配置的前端端口在已有负载均衡实例上尚未被其他应用服务占用，否则会因端口冲突而导致应用服务发布失败。

负载均衡

iso0815test-unit01- customlbtest 删除

\* 访问方式: ☐ 内网 ☐ 外网 ☒ 自定义

\* 部署单元ID:

部署单元	负载均衡实例ID
GZ00A	lb-7zjg...w3rrh402w
RZ00A	lb-7zj...lx81sts57i

当前仅支持填写阿里云负载均衡实例 ID

\* 端口映射

协议	前端端口	后端端口	转发规则	带宽	http健康检查	会话保持	操作
+ 添加端口映射							

- **联邦负载均衡实例**：仅当访问方式选择 **内网** 或 **外网** 且开启 **使用联邦负载均衡** 时显示该选项。从下拉列表中选择对应联邦负载均衡实例，可选择到的联邦负载均衡的部署单元必须能够覆盖应用服务 [弹性配置](#) 步骤中选择的所有部署单元。

### ? 说明

- 如需创建新的联邦负载均衡实例，请参见 [创建联邦负载均衡](#)。
- 如需修改联邦负载均衡配置的部署单元，请参见 [配置联邦负载均衡](#)。

负载均衡

test1-

fedslib

删除

\* 访问方式: ☒ 内网 ☐ 外网 ☐ 自定义

使用联邦负载均衡 ☒

\* 联邦负载均衡实例

○ 端口映射：单击 添加端口映射，填写以下信息。

- 协议：支持 TCP、HTTP、HTTPS 协议。
- 转发规则：支持 RR 轮询、WLC 最小连接数。
- 前端端口：容器镜像中工作负载实际监听的端口，端口范围为 1~65535。
- 后端端口：容器端口映射到负载均衡实例上的端口，用负载均衡 IP 访问工作负载时使用，端口范围为 1~65535。
- 带宽：单位为 Mbps，默认值为 -1，表示无限制。支持 -1 或 1~5120 范围内的整数。
- HTTP 健康检查：HTTP 模式的健康检查是通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查后端 Pod 是否健康。若开启，需填写以下的配置项。

### ? 说明

TCP 端口默认开启 TCP 健康检查，此处可按需开启额外 HTTP 健康检查。

- URL：指定具体的检查路径。
- 域名：非必填。如果您的应用 Pod 需要校验请求的 host 字段，则需要配置相关域名，确保健康检查正常工作。若在健康检查中配置了域名，SLB 会将域名配置到 host 字段中去；反之，如果没有配置域名，SLB 则不会在请求中附带 host 字段，因此健康检查请求就会被服务器拒绝，可能导致健康检查失败。
- 端口：健康检查服务访问后端 Pod 时的探测端口。端口范围 1~65535。
- 超时时间 (s)：接收响应需要等待的时间。如果后端 Pod 在指定的时间内没有正确响应，则判定为健康检查失败。默认为 3 秒。
- 时间间隔 (s)：负载均衡的健康检查请求进行健康检查的时间间隔。默认值为 5 秒。
- http 健康状态码：选择健康检查正常的 HTTP 状态码。默认值为 http\_2xx。

- **会话保持**：开启会话保持后，负载均衡监听会把来自同一客户端的访问请求分发到同一台后端 Pod 上。

HTTP 协议会话保持基于 Cookie。负载均衡提供了两种 Cookie 处理方式：

- **植入 Cookie**：您只需要指定 Cookie 的过期时间。客户端第一次访问时，负载均衡会在返回请求中植入 Cookie（即在 HTTP 或 HTTPS 响应报文中插入 SERVERID），下次客户端携带此 Cookie 访问，负载均衡服务会将请求定向转发给之前记录到的后端服务器上。
- **重写 Cookie**：可以根据需要指定 HTTPS 或 HTTP 响应中插入的 Cookie。您需要在后端 Pod 上维护该 Cookie 的过期时间和生存时间。

负载均衡服务发现用户自定义了 Cookie，将会对原来的 Cookie 进行重写，下次客户端携带新的 Cookie 访问，负载均衡服务会将请求定向转发给之前记录到的后端 Pod。详情请参见 [会话保持规则配置](#)。

## 统一接入

统一接入 支持配置转发规则，将流量转发到容器的相应端口上。您可以在创建应用服务时设置访问方式，也可以应用服务创建完成后添加访问方式。

### 创建应用服务时添加访问方式

1. 在 **访问配置** 页面，单击 **添加统一接入**。
2. 在 **统一接入** 配置中，填写以下信息后，单击 **下一步**。
  - **服务名称**：填写服务名称。系统默认生成服务名称前缀为 `应用服务名称-`。服务名称允许包含（小写）字母、数字、连字符、且必须以字母开头、以字母或者数字结尾。
  - **所属统一接入实例**：选择已创建的统一接入实例。详情参见 [创建统一接入实例](#)。
  - **接入协议**：选择统一接入实例上支持的协议，一般有 HTTP 或 HTTPS。
  - **路由模式**：选择 **单元化** 或 **非单元化**。
    - **单元化** 代表为这个统一接入开启单元化转发的能力，即会根据请求 Cookie 内的特定 UID 进行单元化切片，每个 Zone 处理固定区间的用户请求。
    - **非单元化** 代表随机分发。
  - **统一接入规则**：单击 **添加统一接入规则** 转发规则，填写以下信息。
    - **域名**：填写需要转发的请求域名。
    - **后端端口**：选择容器镜像中工作负载程序实际监听的端口，端口范围为 1~65535。
    - **转发路径**：请求转发到的具体容器路径地址。默认为根目录。

#### 注意

添加或删除已有转发规则都会导致发布时线上服务不可用。

## Headless Service

Headless Service 对应的每一个 Endpoint，即每一个 Pod，都会有对应的 DNS 域名。配置 Headless Service 名称，这样各 Pod 之间便可以通过域名互相访问。Headless Service 不会分配 Cluster IP，kube-proxy 不会处理此类 Service，但可以通过域名解析访问 Pod。相关操作，请参见 [LHC Pod 域名配置](#)。

您可以在创建应用服务时添加 Headless Service 名称，也可以应用服务创建完成后重新编辑应用服务进行添加。添加之后需要发布应用服务才会生效。

## 创建应用服务时添加 Headless Service

### ? 说明

一个应用服务仅支持添加一个 Headless Service。

1. 在 **访问配置** 页面，单击 **添加 Headless Service**。
2. 在 **Headless Service** 配置中，填写 Headless Service 名称。名称由服务名称前缀、Headless Service 名称组成。
  - 前缀：系统默认生成服务名称前缀 **应用服务名称-**。
  - Headless Service 名称：输入名称，允许包含小写字母、数字、连字符（-）。最大长度不超过 20 个字符。
3. 单击 **下一步**。

## 五、部署和调度配置（选填）

您可以自定义部署和调度配置，若不修改，发布应用服务时，系统会使用默认配置。

该配置项用于配置容器服务在部署时需要用到的信息，包括以下内容：

- **是否使用 Beta 验证**：发布时选取部分 Pod 先行发布，待确认无异常后继续发布。默认开启。

开启 Beta 分组后，发布时会给应用服务设置一个特殊的 Beta 分组，在该组中，系统会在每个部署单元选择一个 Pod，Beta 分组会在第一组发布。

  - Beta 分组发布完成后系统会自动暂停应用服务发布，等待系统负责人或者运维工程师对应用服务的发布情况进行确认。若容器服务发布正常，则单击 **Beta 分组确认**，使应用服务继续分组发布。
  - Beta 分组可以与所有分组策略共同决定分组。创建新的发布申请时，默认开启 **添加 Beta 分组**，此时同一个发布单上的所有应用服务都设置 Beta 分组。
- **部署分组策略**：指定发布容器服务时 Pod 的分组策略，支持以下几种策略，具体可参考下表。

策略类型	说明
按部署单元分组	默认为 <b>按部署单元分组</b> 。 按部署单元维度发布，尽可能让 Pod 均匀分布在各部署单元中。
每个 Pod 一组	每组一个 Pod，有几个 Pod 便分几组。
共分一组	所有 Pod 在一组中进行发布。
快速分组	在发布时按照组的维度进行发布。

策略类型	说明
共分一组（Beta 单元模式）	<p>在发布时按设定的每次最大同时变更 Pod 数量进行发布。选定后，可以根据需要配置以下选项。</p> <ul style="list-style-type: none"><li>最大步长：每个普通分组最大变更 Pod 数（不包含 Beta 分组）。如果将最大步长设为 2，则普通分组的 Pod 数不超过 2。</li><li>部署单元发布顺序：<ul style="list-style-type: none"><li>如果不设置部署单元发布顺序，按默认的部署单元顺序对各部署单元的 Pod 进行发布或变更。</li><li>如果设置了部署单元发布顺序，按指定顺序对各部署单元的 Pod 进行发布或变更。</li></ul></li><li>是否开启 Beta 验证：<ul style="list-style-type: none"><li>如果未开启，所有 Pod 会依次发布，无 Beta 分组，无分组暂停，即用户不需要确认任何分组，执行完一个普通分组后自动执行下一个分组。</li><li>如果开启，部署顺序中的第一个单元会整体作为 Beta 分组进行发布，需要用户确认。其余单元的所有 Pod 会依次发布，这些分组无分组暂停。</li></ul></li></ul> <div><p> 说明</p><p>选择 共分一组（Beta 单元模式）后，在某一时刻可能出现某一部署单元内的 Pod 同时在发布，以致 Pod 不可用的情形。如需规避此现象，即在加快发布进度的同时保证一定 Pod 可用，建议选择 按部署单元百分比并发模式。</p></div>
按部署单元百分比并发模式	<p>如需加快按部署单元发布进度，可选择该分组策略。选定后，需指定一个百分比值。发布变更时，按每个部署单元该百分比数量的 Pod 为一批。</p> <p>如果将百分比设为 25%，则第一批发布所有部署单元的前 25% 的 Pod，第二批发布所有部署单元的 25%-50% 的 Pod，以此类推直至 100%，可参考以下示例图。</p>

- **部署单元发布顺序**：仅当分组策略选择 **按部署单元分组** 或 **共分一组（Beta 单元模式）** 时显示该选项。默认关闭。用于重新编排原有部署单元发布顺序。开启后，支持通过拖拽操作对部署单元进行排序，便于根据实际业务需要灵活调整发布顺序。

部署分组策略

按部署单元分组 ⓘ

部署单元发布顺序 ⓘ

☒

拖拽操作	单元名称	排序
≡	RZ00A	1
≡	GZ00A	2

- **最小分组数**：分组策略选择 **按部署单元分组** 或 **快速分组** 时需设置。

#### ❓ 说明

最小分组数是发布时分组的参考值，实际分组数受发布时 Pod 在部署单元的分布情况影响。

- **分组暂停**：支持仅第一个分组或每组暂停发布，待确认发布后继续发布。
  - **全部暂停**：默认选中。发布单会在每个分组暂停发布。
  - **首批暂停**：选择 **首批暂停** 后，发布单只会在第一个分组暂停发布。
- **自定义标签**：发布应用服务时，将会以 label 的形式发布在 Pod 上。以 `key=value` 的格式输入，多个 label 间用英文逗号分隔。
- **应用服务与节点亲和性配置**：添加应用服务在节点级别的亲和性配置。可以通过 node 标签来限定应用服务可以调度的节点范围。

#### ❓ 说明

亲和性需要 Pod 重新创建才会生效。首次发布时，亲和性配置一定会生效，后续发布时，必须选择替换升级方式，才能让亲和性配置生效。

- **应用服务间亲和性配置**：添加应用服务在 Pod 级别的亲和性配置。通过选择与某些应用服务在相同或不同的节点来限定应用服务可以调度的节点范围。

#### ❓ 说明

亲和性需要 Pod 重新创建才会生效。首次发布时，亲和性配置一定会生效，后续发布时，必须选择替换升级方式，才能让亲和性配置生效。

- **应用服务污点容忍配置**：配置完成后应用服务将部署到对应污点的节点上。在弹出框中输入 **键**、**值**，选择 **Effect**。Effect 包含以下选项：

- **NoSchedule**：不允许无匹配 toleration 的 Pod 调度到该节点。
- **NoExecute**：若无匹配 toleration 的 Pod 已经在节点上运行，则将 Pod 驱逐，若无匹配 toleration 的 Pod 尚未在节点上运行，则不会将 Pod 调度到该节点上。
- **PreferNoSchedule**：尽量不将无匹配 toleration 的 Pod 调度到该节点。
- **注入 SOFAMesh**：需集群开启 Mesh 功能。开启引流配置时，每次发布会按所设置的引流规则转发流量到新版本，也可在发布过程中配置。开启后，会为应用服务开启服务网格功能，可以在服务网格控制台对应用服务进行管控和治理。

## 六、预览并提交

在应用服务 预览 页面，确认信息无误，单击 提交。

### 说明

- 提交前您可以单击修改图标对应用服务信息进行修改。
- 应用服务提交后处于 待部署 状态，需要单击 发布 才会将应用服务部署到集群中。

## 5.2. 高级配置

在创建应用服务过程中，您可以在配置 Pod 模板时选择是否启用高级配置，以实现对应用服务性能优化和灵活性提升。本文主要对配置 Pod 模板时的高级配置项进行说明。

### hostNetwork 配置

#### 说明

使用 hostNetwork 时，为避免端口监听冲突，1 个节点最多只能挂载 1 个 Pod。

在某些场景下，需要将 Pod 的网络从容器网络修改为宿主机网络。比如，要解决网络的连通性问题等。

如果使用 hostNetwork 配置网络，在应用服务发布成功后，Pod IP 与节点 IP 将保持一致。

### 环境变量

通过键值对的形式为 Pod 配置环境变量，从而给 Pod 添加环境标志或传递配置等，在容器启动时传入应用进程，例如：`USER=tester`。更多信息，请参见 [Pod variable](#)。

参数说明如下表所示。

参数名	说明
来源	环境变量配置来源。取值如下： <ul style="list-style-type: none"><li>● 手动输入：自行指定环境变量的名称和值。</li><li>● 配置项：选择已配置的 ConfigMap 来指定环境变量的值。</li><li>● 保密字典：选择已配置的 Secret 来指定环境变量的值。</li></ul>

参数名	说明
变量名	<p>环境变量的名称。</p> <div><p> 说明</p><p>变量名必须由字母、数字、下划线（_）、短横线（-）或英文句号（.）组成，且不能以数字开头。</p></div>
变量值/变量引用	<p>根据选择的环境变量来源，配置不同的变量值或引用。</p> <ul style="list-style-type: none"><li>当选择 <b>手动输入</b> 时，输入环境变量的值。</li><li>当选择 <b>配置项</b> 或 <b>保密字典</b> 时，输入 ConfigMap 和 Secret 的引用。</li></ul>

## 数据卷配置

配置容器使用的数据卷，目前仅支持挂载在当前容器所在宿主机的目录。

参数说明如下表所示。

参数名	说明
类型	<p>支持的类型如下：</p> <ul style="list-style-type: none"><li>本地存储</li><li>配置项</li><li>保密字典</li><li>PVC 模板</li><li>PVC</li></ul> <p>更多信息，请参见 <a href="#">Volumes</a>。</p>
权限	支持 读写、只读。
容器目录	容器中的数据卷路径。

参数名	说明
主机目录/配置项引用/PVC 名称	<p>根据选择的数据卷配置类型，设置对应的主机目录或配置项引用。</p> <ul style="list-style-type: none"><li>当选择 <b>本地存储</b> 时，输入 Pod 所在宿主机的目录。</li><li>当选择 <b>配置项</b> 时，从下拉列表中选择对应的配置项。</li><li>当选择 <b>保密字典</b> 时，从下拉列表中选择对应的保密字典。</li><li>当选择 <b>PVC 模板</b> 时，从下拉列表中选择对应的 PVC 模板。</li></ul> <div><p> <b>说明</b></p><p>如果尚未配置 PVC 模板，可在创建或编辑应用服务时进行配置。更多信息，请参见 <a href="#">数据卷模板配置</a>。</p></div> <ul style="list-style-type: none"><li>当选择 <b>PVC</b> 时，从下拉列表中选择对应的 PVC。</li></ul>
subPath/subPathExpr	输入 subPath 或 subPathExpr 名称，用于指定所引用卷内的子路径，而不是其根路径。配置完成后，容器目录下已有的文件不会被覆盖。

## 健康检查配置

健康检查是指容器运行过程中，基于用户配置的健康检查，定时检查容器健康状况。若不配置健康检查，如果出现业务异常，容器将无法感知，也不会自动重启去恢复业务，进而可能影响业务的正常运作。

目前提供两种检查机制：Readiness Probe 和 Liveness Probe。更多信息，请参见 [Kubernetes Probe](#)。

### 说明

- Readiness Probe**：用于检查容器是否准备好接收流量。
  - 如果检查失败，集群会屏蔽请求访问该容器。
  - 如果检查成功，会开放对该容器的访问。
- Liveness Probe**：检测容器是否正常，如果检查失败，集群会对该容器执行重启操作。

配置 Readiness Probe、Liveness Probe 参数说明如下表所示。

参数名	说明
检查方式	<ul style="list-style-type: none"><li><b>TCP Socket 端口</b>：通过发送 TCP 请求来进行判断，若能成功建立连接，则检查通过。</li><li><b>HTTP Get 方法</b>：通过发送 HTTP Get 方法的请求来进行判断，若返回的状态码大于等于 200 且小于 400，则检查通过。</li><li><b>在容器中执行命令</b>：通过在容器中执行命令来进行判断，若命令成功执行，并返回 0，则检查通过。</li></ul>

参数名	说明
执行命令	当 检查方式 选择 HTTP Get 方法 或 在容器中执行命令 时需设置。 <ul style="list-style-type: none"><li>选择 HTTP Get 方法 时，指定 HTTP 请求的执行命令。</li><li>选择 在容器中执行命令 时，指定在容器中执行的命令。</li></ul>
访问端口	当 检查方式 选择 TCP Socket 端口 或 HTTP Get 方法 时需设置。 <ul style="list-style-type: none"><li>选择 TCP Socket 端口 时，填写 TCP 访问的容器端口。</li><li>选择 HTTP Get 方法 时，填写 HTTP 请求的访问容器端口。</li></ul>
首次检查延时	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。
检查超时	即 timeoutSeconds，探测超时时间。默认 1 秒，最小 1 秒。
检查间隔	即 periodSeconds，指执行探测的时间间隔。默认为 10 秒，最小为 1 秒。
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。
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。

## 生命周期事件回调配置

为容器添加生命周期事件回调，分别在容器启动后和容器停止前执行。

配置 Post Start、Pre Stop 参数说明如下表所示。

参数名	说明
回调方式	支持 在容器中执行命令、HTTP GET 方法。
访问路径	当 回调方式 选择 HTTP Get 方法 时需设置。 输入访问路径，如 <code>/healthy/readiness</code> 。
访问端口	当 回调方式 选择 HTTP Get 方法 时需设置。 输入访问端口，范围为 1~65535。

参数名	说明
执行命令	当 回调方式 选择 在容器中执行命令 时需设置。 输入执行命令，如 <code>cat tmp/healthy</code> 。

## 日志服务配置

配置日志服务，可选择已有日志库或创建新的日志库。

### 说明

首次使用日志服务配置功能，需前往 [日志服务控制台](#) 开启配置索引。详情请参见 [配置索引](#)。

参数说明如下表所示。

参数名	说明
日志库	在日志服务中生成一个对应的 logstore，用于存储采集到的日志。 从已有的日志库中选择。如果没有可用日志库，可单击 <a href="#">创建日志库</a> 来创建新的日志库。
日志类型	容器文本日志。
路径	收集容器内指定路径的日志。例如： <code>/home/admin/*.log</code> 或 <code>/root/*.log</code> 。

## 5.3. 发布模板

LHC 应用服务发布支持卡点设置，您可以根据不同业务场景需求通过创建发布模板来实现。本文介绍如何通过控制台创建发布模板及后续操作。

### 适用场景

如需在应用服务的发布过程中增加一些卡点设置，且有借助这些卡点对接部分服务调用方面的诉求，可通过创建发布模板来实现，进而高效便捷地实现对整个发布进程的执行能力扩展。

### 前提条件

登录高可用管理控制台 [创建应急预案](#)。

### 创建发布模板

1. 登录 SOFAShield 控制台。
2. 在左侧导航栏单击 [发布运维](#) > [发布模板](#)。


3. 在发布模板列表页，单击 **创建发布模板**。
4. 在 **创建发布模板** 页，完成发布模板配置。
  - i. 完成基础信息配置。

配置项	说明
模板名称	必填。输入发布模板的英文名称。长度为 1~64 个字符。允许包含小写字母、数字、连字符且必须以小写字母开头、以小写字母或者数字结尾。
发布模板说明	选填。自定义模板说明。

- ii. （可选）完成高级设置。

- **是否覆盖发布版本策略**：默认关闭，代表以应用服务设置的版本策略作为生效版本发布策略。

开启后，此处配置的版本发布策略将全局覆盖关联的应用服务设置的版本策略，具体配置项说明请参见 [部署和调度配置](#)。

是否覆盖发布版本策略 

是否使用 Beta 验证 ⓘ 

部署分组策略

按部署单元分组 ⓘ

最小分组数 ⓘ

10

☐ 分组暂停 ⓘ

- **是否开启变更卡点**：默认关闭。开启后，即可按需增加卡点设置来对接部分服务调用。

是否开启变更卡点 ☒

卡点类型

☐ API ☒ 守夜人

编辑卡点

卡点规则ID

删除

分组类型设置

☐ 第一个分组 ☐ 最后一个分组 ☒ 全部分组 ☐ 自定义分组

前置卡点

预案  ☐ 是否暂停 ☐ 参数覆盖

后置卡点

预案  ☐ 是否暂停 ☐ 参数覆盖

+ 添加规则

提交

具体步骤如下表所示。

序号	操作
1	单击 <b>添加规则</b> 。
2	设置分组类型，支持如下类型： <ul style="list-style-type: none"><li>■ 第一个分组</li><li>■ 最后一个分组</li><li>■ 全部分组</li><li>■ 自定义分组：选定后，需输入分组编号，支持整型。比如输入 1，代表第一个分组；输入 -1，代表倒数第一个分组。</li></ul>
3	设置前置卡点。如果选择 <b>预案</b> 后下拉列表内无数据，请前往高可用管理控制台 <a href="#">创建应急预案</a> 。
4	设置后置卡点。如果选择 <b>预案</b> 后下拉列表内无数据，请前往高可用管理控制台 <a href="#">创建应急预案</a> 。

#### 说明

- LHC 支持卡点暂停操作，在页面右侧选中 **是否暂停** 后，当前卡点在实际的应用发布过程中将不会生效。
- 右侧 **参数覆盖** 按钮默认关闭，开启后可以对预案内的数据进行修改。

#### 5. 单击 **提交**。

创建完成后，新创建的发布模板将显示在发布模板列表页。

## 后续操作

### 创建发布单

1. 进入发布单创建页面。
  - i. 登录 SOFAShark 控制台。
  - ii. 在左侧导航栏单击 **发布运维** > **发布单**。
  - iii. 在发布单页，单击 **创建发布单**。
2. 在 **创建发布单** 页面，**填写基本信息**。
3. 进行高级配置。设置应用服务依赖关系并选择已创建的发布模板。

4. 在 **预览** 页面确认信息无误后，单击 **创建**。系统会自动跳转到 **发布单详情** 页面，单击 **整体发布** 即可开始发布。
5. (可选) 发布单创建完成后，查看发布单的执行详情。

## 其他操作

- 如需搜索发布模板，请在发布模板列表页的搜索框中，输入发布模板名称关键字即可进行搜索。
- 如需修改发布模板，请在发布模板列表页单击目标发布模板 **操作** 列中的 **编辑**，并在 **编辑发布模板** 页按需进行修改，然后单击 **提交**。
- 如需删除发布模板，请在发布模板列表页单击目标发布模板 **操作** 列中的 **删除**，然后在弹出的提示对话框中单击 **确定**。

## 5.4. 发布应用服务

## 5.4.1. 立即发布

应用服务创建完成后，您可以通过 LHC 控制台发布该应用服务。

可以通过如下两种方式发布应用服务：

- [方式一：应用服务创建成功后立即发布](#)
- [方式二：通过应用服务详情页发布](#)

### 前提条件

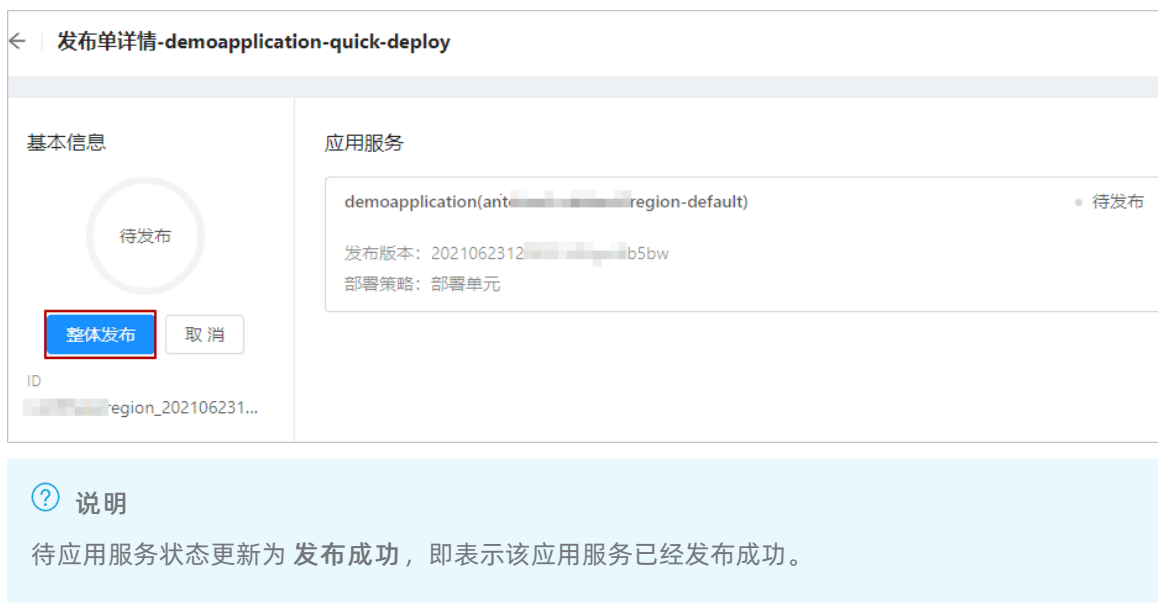
发布应用服务前，需确保集群内有充足的节点资源（如 CPU、Memory），否则 Pod 将无法正常工作到节点。

### 方式一：应用服务创建成功后立即发布

1. 完成应用服务创建后，单击 **立即发布**。具体创建步骤可参考 [创建应用服务](#)。



2. 在 **发布单详情** 页，单击 **整体发布**。



### 方式二：通过应用服务详情页发布

1. 登录 LHC 控制台。
2. 在左侧导航栏上，单击 **发布运维 > 应用服务**。

3. 在应用服务列表页，单击应用服务名称进入 应用服务详情 页。
4. 单击右上角的 发布。



#### ? 说明

- 若应用服务存在多个版本，会提示 是否发布该容器服务最近一次变更的版本。
- 如需发布以前的版本，选择 版本 页签，单击版本右侧的 发布。

5. 在 发布预览 页面，确认信息无误后单击 开始发布。
6. 发布单创建成功后，在 发布单详情 页面，单击 整体发布。

#### ? 说明

待应用服务状态更新为 发布成功，即表示该应用服务已经发布成功。

## 后续操作

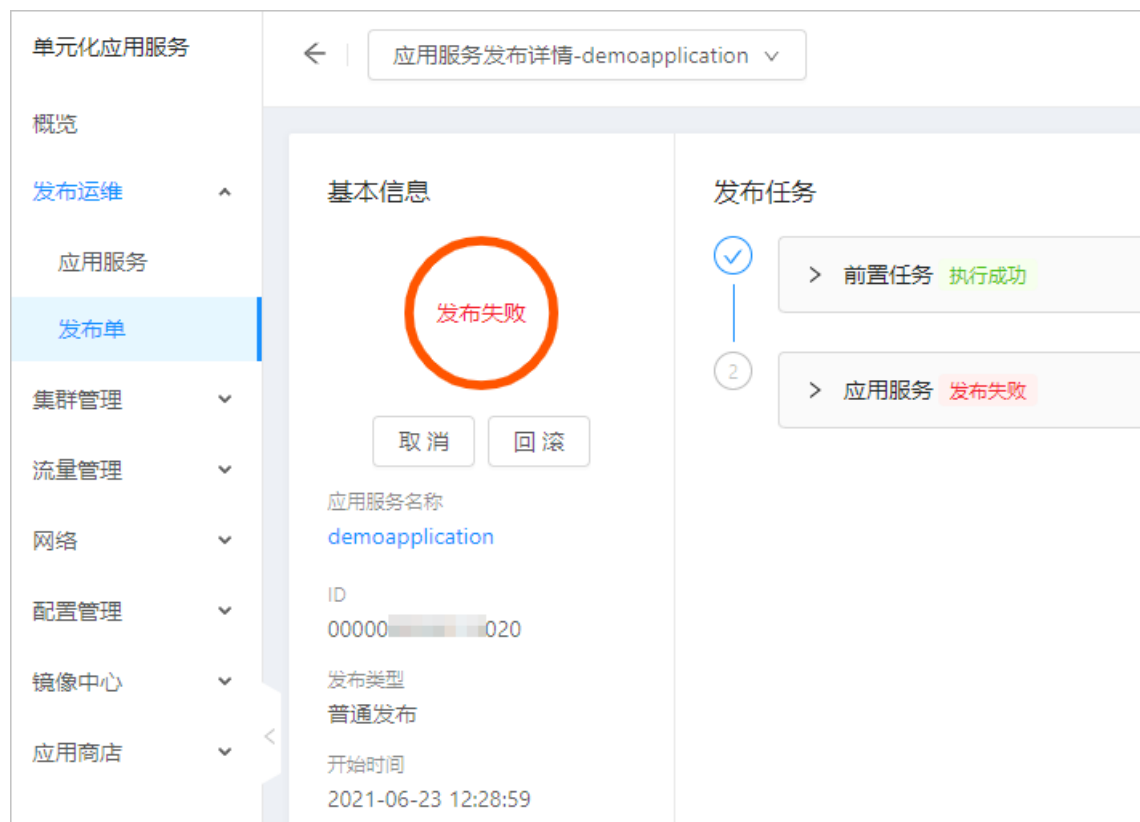
### 查看发布详情

单击 发布 后自动跳转到 发布单详情 页面。您可以展开右侧的 发布任务 查看详细的发布状态。



### 重试回滚

若出现发布失败，您可以单击 **重试**，继续进行发布，或单击 **回滚** 恢复到上一个已发布的版本。



#### 说明

仅当前应用有其它成功发布的发布包（除当前版本外）时，您才可以回滚应用。若无任何成功发布记录，回滚将释放所有资源。

## 5.4.2. 发布单

通过发布单可以同时发布多个应用服务。本文介绍如何通过创建发布单发布应用服务，以及查看发布单详情等。

### 创建发布单

1. 登录 LHC 控制台。
2. 在左侧导航栏单击 **发布运维** > **发布单**。
3. 在发布部署大盘，单击 **创建发布单**。

#### 说明

目前发布部署大盘页面的搜索框仅支持搜索发布单名称的前缀。

4. 在 **创建发布单** 页面，填写以下发布信息后，单击 **下一步**。

#### 基本信息

- **标题**：输入发布标题。

- **类型**：发布类型。支持 **分组发布**、**蓝绿发布**、**全链路灰度发布**。本例中选择 **分组发布**。蓝绿发布相关操作，请参见 [蓝绿发布](#)。全链路灰度发布相关操作，请参见 [全链路灰度发布](#)。
- **命名空间**：选择待发布的应用服务所属的命名空间。
- **应用服务发布列表**：您可以在 **待选应用服务列表** 中单击选择需要的应用服务，单击 **将应用服务添加到已选应用服务列表** 中。

#### 说明

若应用服务存在多个提交版本，需要选择要发布的版本。默认选择最新的版本。

- **发布指定部署单元**：默认关闭。如果开启，需根据实际业务诉求勾选部署单元。应用服务将仅发布至已选择且当前版本配置中的部署单元，不会对未选中部署单元的已有 Pod 做任何变更。

#### 警告

此功能仅适用于机房容灾场景，即通过勾选部署单元，实现容灾场景下应用服务的正常发布。关于容灾更多信息，请参见 [多集群联邦发布容灾](#)。



#### 高级配置

- **设置应用服务依赖关系**：当选择了 2 个及以上应用服务时，可按需填写应用服务之间的依赖关系。若应用服务之间不存在依赖关系，则直接跳过。您可通过逐条添加的方式添加依赖关系。在 **依赖于** 列中输入某个应用服务所依赖的服务名称（可以同时选择多个）。例如，若 appservice1 依赖 appservice2，则在 appservice1 对应的 **依赖于** 列中输入 appservice2 即可，依赖于多个应用服务时同理。

#### 注意

应用服务之间不可循环依赖，即不可出现应用服务 A 依赖于应用服务 B，同时应用服务 B 又依赖于应用服务 A 这种情况。

- **引入发布模板**：可选。如需在应用服务发布过程中增加卡点设置，可根据需要选择发布模板。

#### 说明

如果选择的发布模板开启了参数覆盖，此处可以对卡点参数进行修改，相关操作请参见 [发布模板](#)。

5. 在 **预览** 页面确认信息无误后，单击 **创建**。系统会自动跳转到 **发布单详情** 页面，单击 **整体发布** 即可开始发布。

## 查看发布单详情

发布单创建完成后，系统会自动跳转到 **发布单详情** 页面，您可以查看发布单的执行详情。

#### ? 说明

- 若发布单初始化失败，则无法进行发布，可以将鼠标悬停到 **失败原因** 上，查看具体原因，或单击 **重新初始化**。
- 在详情页左下角单击 **查看操作日志**，可以查看该发布单具体的事件类型，以及对应的操作人和操作时间。

## 前置任务

- **执行服务配置变更**：若容器服务存在添加或修改 Service 服务配置，执行 Service 服务配置任务，若无则直接跳过。
- **发布日志配置**：若容器服务首次配置日志配置，则执行日志配置前置任务：创建或更新，若无则直接跳过。

## 应用服务

分组发布应用服务，Pod 状态说明如下：

- **未更新**：Pod 配置未更新为最新版本。
- **更新中**：Pod 配置已更新为最新版本，但状态还未 ready 或保持 ready 状态少于 30 秒。
- **更新成功**：Pod 配置已更新为最新版本，并保持 ready 状态 30 秒以上。
- **更新失败**：上线或更新 Pod 异常。

## 后置任务

- **清理无效服务配置**：若容器服务存在删除 Service 服务配置，执行 Service 清理任务，若无则直接跳过。
- **发布日志配置**：若容器服务未配置日志配置或取消日志配置，则执行日志配置后置任务—删除，若无则直接跳过。

## 后置任务（回滚）

**回退日志配置**：若应用服务需回退至无资源状态，且应用服务的日志配置已生效，则执行日志配置的回退任务，若无则直接跳过。

## 取消发布

发布单创建完成后，系统会自动跳转到 **发布单详情** 页面，您可以在正式发布应用服务前选择 **取消** 发布，终止发布操作。

在 **发布单详情** 页面，单击 **取消**。

若发布单包含多个应用，单击右侧待发布的应用服务，进入单个应用发布页面，单击 **取消**，会取消当前应用发布，其他未取消的应用可以继续发布。

## Beta 确认

配置应用服务的部署和调度策略时，若开通了 Beta 确认功能，在应用服务发布时，系统会给服务器分组设置一个特殊的 Beta 分组，即系统从每个部署单元获取一台机器，加入 Beta 分组，并作为第一组发布。

Beta 分组发布完成后系统会自动暂停应用发布，此时系统负责人或运维人员需要对应用的发布情况进行确认。若应用发布正常，则单击 **Beta 确认**，使应用继续分组发布。

#### ② 说明

- 若每个部署单元只有一个 Pod，则 Beta 确认无效。
- Beta 确认功能目前仅在发布应用服务新版本时生效。
- 若分组策略中没有勾选 **分组暂停**，Beta 组发布后，剩余的分组会自动分批发布直至发布完成。

## 重试回滚

若发布失败，您可以单击 **重试** 或 **回滚** 继续进行操作。

### 重试

若单击 **重试**，系统继续进行发布。

#### ② 说明

若应用发布失败，无法创建新的发布单进行发布。

## 回滚

若单击 **回滚**，根据不同的回滚场景，您需要进行二次确认：

- **取消发布**：找不到可回滚版本，继续回滚将终止此次发布。

#### ② 说明

该场景仅在发布成功过一次，重新发布同一个版本失败时出现。

- **回退至无资源状态**：无任何成功发布记录，继续回滚将释放所有资源。
- **回滚至上一版本**：即将回滚到上一个成功发布版本。

回滚完成后，单击 **确认回滚**。您可以单击右上角的 **原应用发布详情** 查看原发布单发布详情。

若回滚失败，可以选择 **重试回滚** 或 **取消回滚**。

#### 🔔 注意

取消回滚将终止此次发布，可能导致线上存在多个版本。

## 查看 Pod 事件

发布过程中或者结束后，您可以查看 Pod 事件，了解 Pod 的调度与运行状态。

- 在 **发布单详情** 页面，单击 Pod 名称查看 Pod 事件。
- 在 **应用服务详情** 页的 Pod 列表中，单击 Pod 名称进入 **Pod 详情** 页，选择 **事件** 页签，查看 Pod 事件。

## 5.4.3. 蓝绿发布

蓝绿发布是迭代产品在生产环境安全上线的一种重要手段，提供零宕机部署方式。本文介绍如何通过创建蓝绿发布单发布应用服务。

### ? 说明

蓝绿发布单在应用服务维度互斥，即不能通过两个发布单同时发布一个应用服务，且与普通发布单互斥。

## 前提条件

已创建应用服务，且配置应用服务的 [部署和调度策略](#) 时，将部署分组策略设为 [共分一组](#) 或 [快速分组](#)。

关于分组策略限制说明可参考下表。

部署分组策略	蓝绿发布是否支持	说明
每个 Pod 一组	×	由于蓝绿发布的灰度验证是通过 spanner 来控制，而不是通过应用服务内百分比升级的 Pod 来控制的。
按部署单元分组	×	假设流量配比为 100 : 0，对于流量为 0 的区域，若按部署单元分组，流量无法调配至该区域。
共分一组	√	/
快速分组	√	/
共分一组（Beta 单元模式）	×	/
按部署单元百分比并发模式	×	/

## 创建蓝绿发布单

1. 登录 LHC 控制台。
2. 在左侧导航栏单击 [发布运维](#) > [发布单](#)。
3. 在发布部署大盘，单击 [创建发布单](#)。

### ? 说明

目前发布部署大盘页面的搜索框仅支持搜索发布单名称的前缀。

4. 在 [创建发布单](#) 页面，填写以下发布信息后，单击 [下一步](#)。

#### ○ 基本信息

- **标题**：发布标题。
- **类型**：选择 [蓝绿发布](#)。
- **命名空间**：选择待发布的应用服务所属的命名空间。

- **应用服务发布列表**：您可以在 **待选应用服务列表** 中单击选择需要的应用服务，单击 **添加** 将应用服务添加到 **已选应用服务列表** 中。

#### ② 说明

- 若应用服务存在多个提交版本，需要选择要发布的版本。默认选择最新的版本。
- 蓝绿发布单选择的应用服务上一次部署状态必须是成功的，保证有一个完整的蓝版本在线。
- 蓝绿发布单选择的版本与当前运行的版本对比，不能有统一接入的配置变更。
- 蓝绿发布单选择的两个版本不能有拓扑变更，比如从 Cell A 变为 Cell B。对于副本数变更，不属于拓扑变更。

#### ○ 高级配置

**设置应用服务依赖关系**：当选择了 2 个及以上应用服务时，可能需要填写应用服务之间的依赖关系。如果应用服务之间不存在依赖关系，则直接跳过。

在 **依赖于** 列中输入某个应用服务所依赖的服务名称（可以同时选择多个）。您可通过逐条添加的方式添加依赖关系。

例如，如果 appservice1 依赖 appservice2，则在 appservice1 对应的 **依赖于** 列中输入 appservice2 即可，依赖于多个应用服务时同理。

#### 🔔 注意

应用服务之间不可循环依赖，即不可出现应用服务 A 依赖于应用服务 B，同时应用服务 B 又依赖于应用服务 A 这种情况。

5. 在 **预览** 页面确认信息无误后，单击 **创建**。系统会自动跳转到 **发布单详情** 页面，单击 **整体发布** 即可开始发布。

## 查看蓝绿发布单详情

发布单创建完成后，系统会自动跳转到 **发布单详情** 页面，您可以查看发布单的执行详情。

#### ② 说明

- 若发布单初始化失败，则无法进行发布，可以将鼠标悬停到 **失败原因** 上，查看具体原因，或单击 **重新初始化**。
- 蓝绿发布单的蓝版本为当前在线版本。

## 前置任务

- **执行服务配置变更**：若容器服务存在添加或修改 Service 服务配置，执行 Service 服务配置任务，若无则直接跳过。
- **发布日志配置**：若容器服务首次配置日志配置，则执行日志配置前置任务：创建或更新，若无则直接跳过。

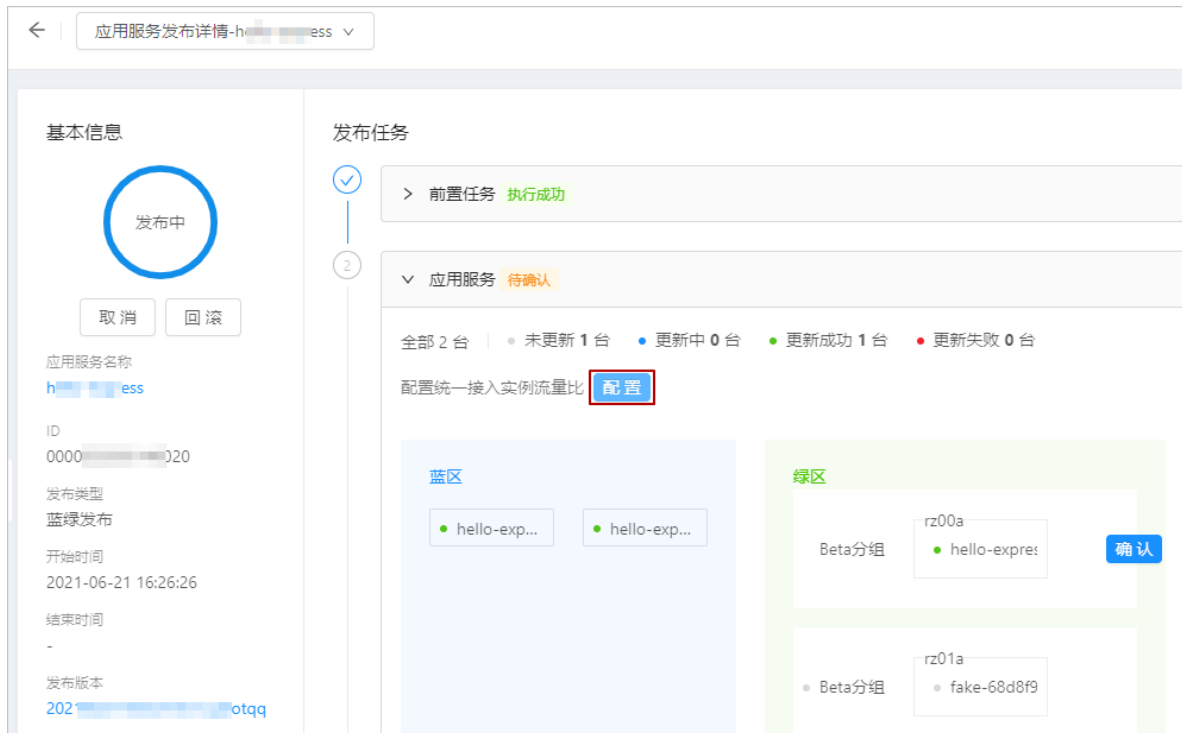
## 应用服务

若应用服务带有统一接入，可配置蓝绿区流量配比。

### 说明

仅应用服务配置了统一接入，才可以为蓝绿区设置流量配比，即在绿版本发布的过程中进行流量调拨。关于如何为应用服务配置统一接入的信息，请参见 [配置统一接入](#)。

- 如果需要自定义蓝绿区流量配比，在应用服务区域单击 **配置**，在弹框中输入蓝区、绿区流量配比即可。



- 如果未自定义蓝绿区流量配比，流量配比将按 **蓝区：绿区 = 100：0 (%)** 来确定流量分发的比重。随后，将在执行后置任务时，全部流量将流向绿区。

## 后置任务

- **流量切换到绿区**：若蓝绿发布应用服务，执行流量切换到绿区后置任务。
- **清理无效服务配置**：若容器服务存在删除 Service 服务配置，执行 Service 清理任务，若无则直接跳过。
- **发布日志配置**：若容器服务未配置日志配置或取消日志配置，则执行日志配置后置任务—删除，若无则直接跳过。

## 后置任务（回滚）

**回滚日志配置**：若应用服务需回退至无资源状态，且应用服务的日志配置已生效，则执行日志配置的回退任务，若无则直接跳过。

## 其他操作

- 蓝绿发布单创建完成后，如需终止蓝绿发布，可单击 **取消**。具体操作步骤与普通发布的相同，请参见 [取消发布](#)。

### 说明

蓝绿发布单取消之后，可能会同时出现多个运行版本，需等到下一次用户主动发布才会自动清除多余版本。

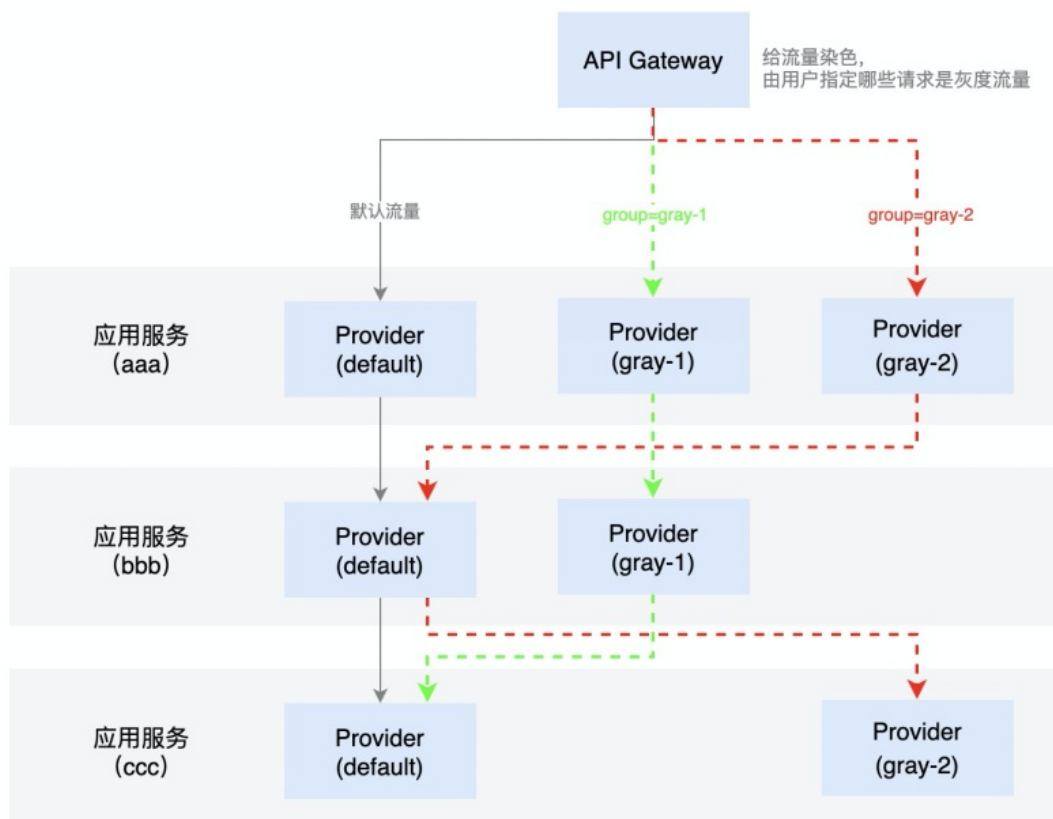
- 配置应用服务的部署和调度策略时，若开通了 Beta 确认功能，在应用服务发布时，单击 **Beta 确认**，使应用继续分组发布。具体操作步骤与普通发布的相同，请参见 [Beta 确认](#)。
- 若蓝绿发布执行失败，可单击 **重试** 或 **回滚** 继续进行操作。具体操作步骤与普通发布的相同，请参见 [重试回滚](#)。
- 发布过程中或完成后，您可以查看 Pod 事件，了解 Pod 的调度与运行状态。具体操作步骤与普通发布的相同，请参见 [查看 Pod 事件](#)。

## 5.4.4. 全链路灰度发布

全链路灰度发布主要专注于整个调用链，关注重心不再是链路上经过具体哪些服务，流量控制视角由服务移至请求链路，仅需要少量治理规则即可构建从网关到整个后端服务的多个流量隔离环境，从而有效保证了多个亲密关系的服务顺利安全发布以及服务多版本并行开发，进一步促进业务的快速发展。

### 灰度链路图

全链路灰度发布配合 SOFA 微服务体系与网关，打造一条独立 RPC 调用通道，通过精细化流量切分保证用户在影响可控的情况下，对于多微服务组成的链路进行验证，同时提供链路观测确保流量的可观测性。



### 前提条件

已创建应用服务。

## 创建全链路灰度发布单

1. 登录控制台，在左侧导航栏单击 **发布运维** > **发布单**。
2. 在发布部署大盘，单击 **创建发布单**。
3. 在 **创建发布单** 页，填写以下发布信息。
  - i. 完成基础信息配置后，单击 **下一步**。

配置项	说明
标题	必填。填写发布单标题。
类型	选择 <b>全链路灰度发布</b> 。
灰度标签	必填。填写灰度标签时，须遵循以下命名规则： <ul style="list-style-type: none"><li>■ 允许包含小写字母、数字、连字符且必须以小写字母开头、以小写字母或者数字结尾。</li><li>■ 灰度标签不能为 default。</li></ul>
对接网关	目前仅支持 <b>SOFA_APIGATEWAY</b> 。
场景规则	填写 header 参数和值，支持添加多个灰度流量规则。也可以在灰度发布执行期间调整规则，相关操作请参见步骤 5。 设置灰度流量规则后，灰度链路会精确匹配这个网关的规则，只有符合这个规则的流量才会路由到灰度链路上。
命名空间	选择待发布的应用服务所属的命名空间。
应用服务发布列表	在 <b>待选应用服务列表</b> 中选择需要的应用服务，单击 <b>添加</b> 将应用服务添加到 <b>已选应用服务列表</b> 中。 <div><p> <b>说明</b></p><p>如果应用服务存在多个提交版本，需要选择要发布的版本。默认选择最新的版本。</p></div>
副本伸缩策略配置	按需勾选部署单元并将对应的副本数修改为期望数值。

- ii. （可选）完成高级设置。设置应用服务依赖关系，并且可以按需选择已创建的发布模板。
4. 在 **预览** 页面确认信息无误后，单击 **创建**。系统会自动跳转到 **发布单详情** 页面，单击 **整体发布** 开始发布。

← 发布单详情-gray-test-demo

图形化发布单

基本信息

待发布

整体发布

取消

ID  
lhctest0402\_2022052714435...

执行人  
-

发布类型  
全链路灰度发布

开始时间

应用服务发布顺序

调整规则

分组A

▼ A 1 待执行

test0426-server01(antcloud-lhctest0402-def...  
发布版本: 20220510174156994.fmvzg28h  
部署策略: 按部署单元分组

待执行

test0426-server02(antcloud-lhctest0402-def...  
发布版本: 20220507165749379.yluggej4  
部署策略: 按部署单元分组

待执行

test0426-server03(antcloud-lhctest0402-def...  
发布版本: 20220507165726763.de74kack  
部署策略: 按部署单元分组

待执行

5. (可选) 发布执行期间可根据需要调整灰度流量规则。

- 在 **发布单详情** 页，单击 **调整规则**。
- 在 **调整规则** 面板，根据需要调整灰度规则流量规则，具体操作可参考下表。

操作项	说明
修改规则	在 <b>场景规则</b> 区域，修改 header 参数和值。 修改后，带有修改后的 header 可以访问到灰度 Pod。
添加规则	在 <b>场景规则</b> 区域，单击 <b>添加 1 行数据</b> ，完成 header 参数和值的填写。 添加后，带有新增 header 访问到灰度 Pod，其他 header 也可以访问到灰度 Pod。
删除规则	在 <b>场景规则</b> 区域，找到需要删除的规则，完成 header 参数和值的填写。 带有删除的 header 会访问到 default Pod，带有其他已存 header 访问到灰度 Pod。

← 发布单详情-gray-test-demo

调整规则

基本信息

发布中

取消

ID  
lhctest0402\_2022052714435...

执行人  
admin

发布类型  
全链路灰度发布

应用服务发布顺序

调整规则

分组A

▼ A 1 执行中

test0426-server01(antcloud-lhctest0402-def...  
发布版本: 20220510174156994.fmvzg28h  
部署策略: 按部署单元分组

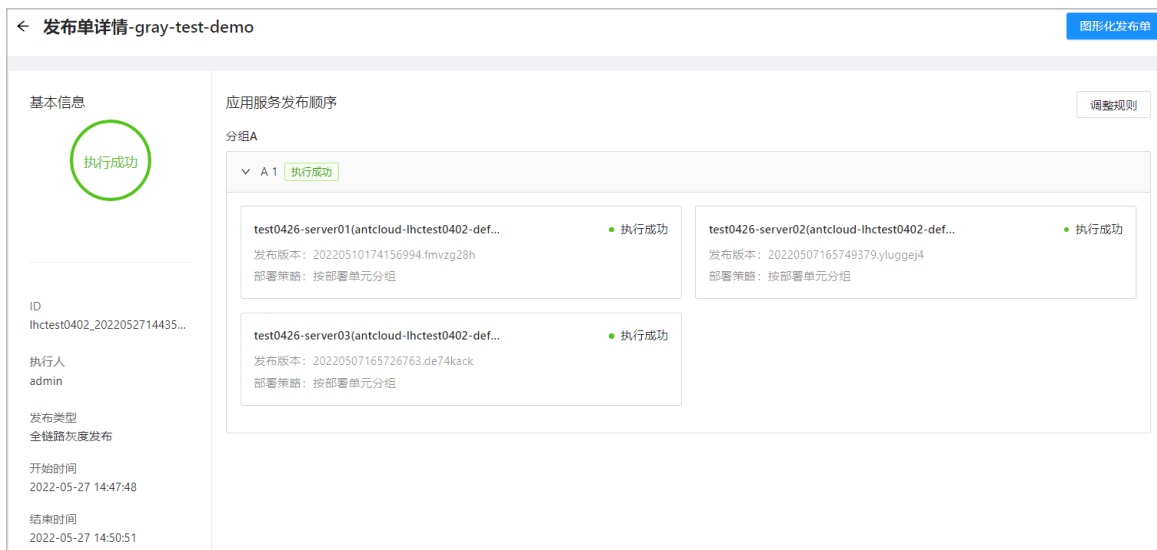
test0426-server03(antcloud-lhctest0402-def...  
发布版本: 20220507165726763.de74kack  
部署策略: 按部署单元分组

场景规则

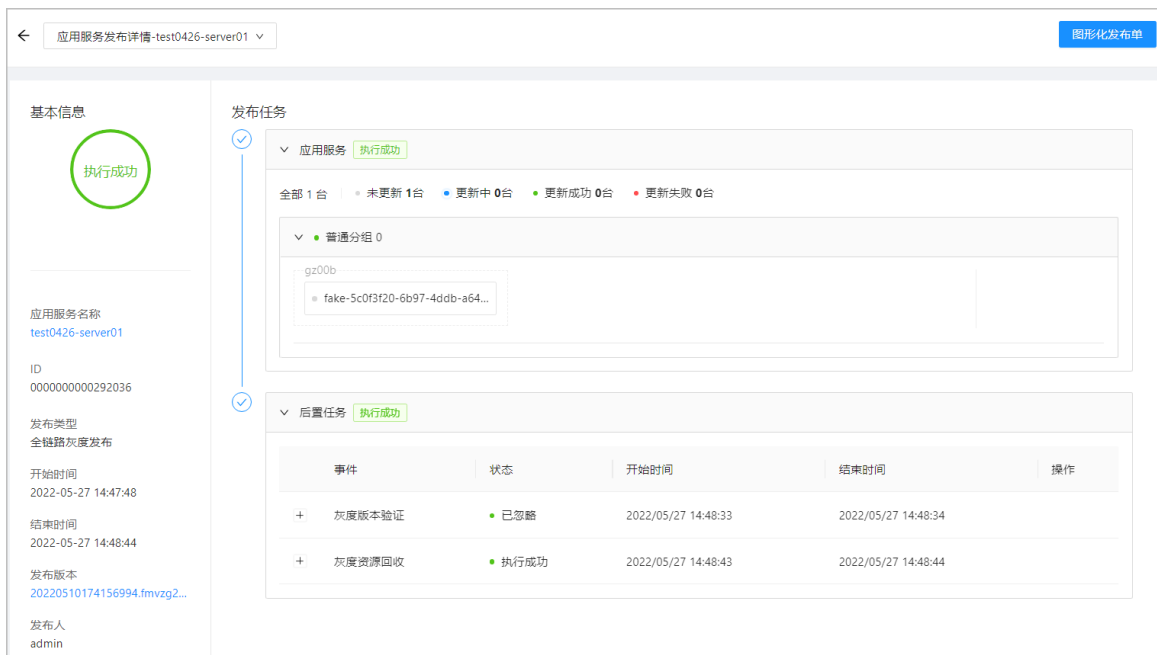
header参数	条件	值	操作
header0527	EQUALS	value0527	删除
+ 添加1行数据			

iii. 单击 **确定**。

6. (可选) 全链路灰度发布单执行完成后, 查看发布单详情, 如下图所示。



在 **发布单详情** 页, 单击应用服务卡片, 可查看对应的应用服务执行详情, 如下图所示。



## 其他操作

- 全链路灰度发布单创建完成后, 如需中止灰度发布, 可单击 **取消**。具体操作步骤与普通发布的相同, 请参见 [取消发布](#)。
- 配置应用服务的部署和调度策略时, 若开通了 Beta 确认功能, 在应用服务发布时, 单击 **Beta 确认**, 使应用继续分组发布。具体操作步骤与普通发布的相同, 请参见 [Beta 确认](#)。
- 如果全链路灰度发布失败, 可单击 **回滚** 继续进行操作。具体操作步骤与普通发布的相同, 请参见 [重试回滚](#)。
- 发布过程中或完成后, 您可以查看 Pod 事件, 了解 Pod 的调度与运行状态。具体操作步骤与普通发布的相同, 请参见 [查看 Pod 事件](#)。

## 5.4.5. 多集群联邦发布容灾

本文介绍 LHC 多集群联邦发布容灾相关信息，包括基本概念、灾难感知等。

### 背景信息

当站点因为不可抗力因素或者设备故障导致应用在短时间内无法恢复时，LHC 可提供多集群联邦发布容灾保护。当站点故障时，通过简单的配置，即可在容灾站点迅速恢复业务。

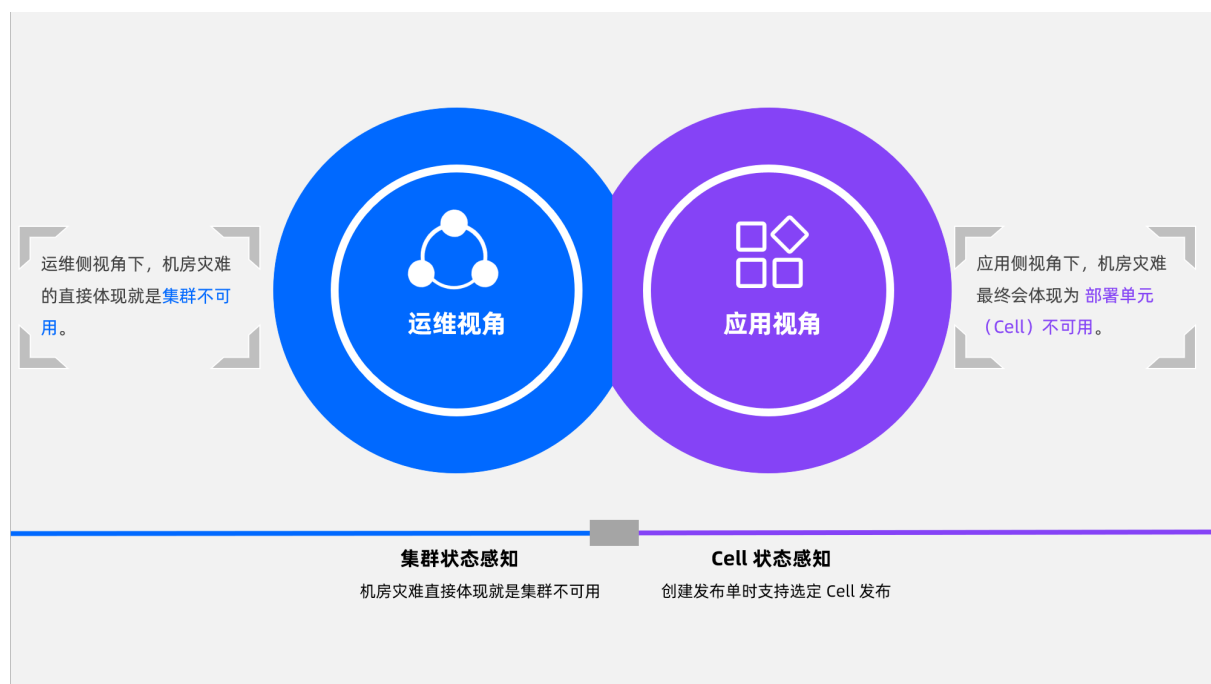
容灾是一个范畴比较广泛的概念。广义上，容灾是一个系统工程，包括所有与业务连续性相关的内容。狭义的容灾是指建立两套或多套功能相同的 IT 系统，互相之间可以进行状态监视和功能切换，当主要站点因意外停止工作时，整个应用系统可以利用辅助站点快速恢复，并继续工作。

容灾的主要目的是当自然或人为的原因导致生产系统发生灾难时，能够尽可能地保证业务的连续性。

#### 机房灾难感知

为实现容灾场景下 LHC 多集群的正常发布，可通过以下方式来判断机房灾难。

- **集群状态**：运维侧视角下，机房灾难的直接体现就是集群不可用。
- **部署单元状态**：应用侧视角下，机房灾难最终会体现为部署单元（Cell）不可用，导致用户通过发布单发布应用服务时出现不符合预期的发布。



### 机房灾难下的多集群联邦发布

对于应用发布而言，容灾场景下存在多种可选项。在发布单执行前，用户可通过仅指定可用的 Cell 执行发布。

相关操作请参见：[创建发布单](#)。

## 5.5. 更新应用服务

创建并提交应用服务后会生成对应的版本。每个版本都是无法进行修改的，若更新配置，则会生成新的版本。新的版本需要通过发布才会在线上生效。对于日常更新版本，只需选择镜像而不用其他配置，可在应用服务详情页选择快捷操作。

## 标准更新流程

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击应用服务的名称进入 **应用服务详情** 页。
3. 单击右上角的 **编辑配置信息**，进入 **编辑版本** 页面。

### 说明

若当前有未提交的草稿，在弹出的对话框中选择是否恢复打开草稿，若选择 **否**，会删除当前草稿。

4. 修改以下配置信息。参数说明参考 [创建应用服务](#)。

- **基本信息** 中的 **描述**。
- **Pod 模板配置** 中除 **容器名称** 以外的所有信息。
- **弹性配置** 中的副本数信息。
- **访问配置** 中除服务名称以外的所有信息。

### 说明

在编辑应用服务页对访问配置 **已创建的自定义 LB** 执行删除时，只会删除配置的元数据，而自定义的负载均衡实例本身不会被删除。

- **部署调度** 中所有信息。
5. 修改完成后，预览并提交。  
配置信息编辑成功后，您可以在应用服务详情页的版本页签中查看更新的应用服务版本信息。
  6. 在应用服务详情页的版本页签，选择新版本，并单击 **发布**。

## 快捷更新流程

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击应用服务的名称进入 **应用服务详情** 页。
3. 单击 **快捷操作**，选择 **更换镜像** 或 **扩缩容**，选择目标版本镜像或修改副本数，单击 **提交**。
4. 提交后会自动生成发布单，单击 **整体发布** 进行发布。

## 5.6. 导入与导出应用服务

应用服务导入功能支持将本地应用服务数据导入到 LHC 控制台的应用服务创建页。本文介绍如何通过 LHC 控制台导入与导出应用服务。

### 背景介绍

企业级应用研发一般会有多个环境，比如开发测试环境、生产环境等，在不同的环境中需要配置相同或相近的应用服务。如果在开发测试环境创建和发布了应用服务 A，进行测试验证后，需要在生产环境重新发布和发布应用服务 A，可能会带来如下问题：



- **重复配置**：每个应用服务在新的环境都需要手动配置，导致很大的重复工作量。
- **测试生产不一致**：手动填写应用服务创建页会出现遗漏现象，可能导致生产和测试环境配置不一样，无法达到开发测试验证的预期效果。

基于以上两个问题，LHC 提供应用服务导入与导出功能，支持将应用服务以 JSON 格式导出，然后导入其他环境，极大简化了应用服务的创建和发布操作，进而有效提升运维效率，保障用户体验。

## 导入应用服务

### 注意

应用服务导入功能，并非实际创建应用服务，而是基于导入的应用服务数据预先填充应用服务创建页。

您可以通过 LHC 控制台将 JSON 格式的应用服务配置信息导入当前环境。具体操作如下：

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击 **导入**。
3. 在弹窗中选择本地保存的 JSON 格式的应用服务数据，单击 **打开**。

导入后，可根据实际业务需要在自动填充的应用服务创建页修改具体的配置项信息。

## 导出应用服务

将当前环境下的目标应用服务导出为 JSON 格式文件。具体操作如下：

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击应用服务名称进入 **应用服务详情** 页。
3. 选择 **版本** 页签进入版本列表页。
4. 单击目标应用服务版本右侧 **操作** 栏中的 **导出**，以 JSON 格式将应用服务数据保存至本地。

# 5.7. 查看应用服务详情

创建完应用服务后，您可以查看应用服务详情，包括应用服务的基本信息、配置信息，方便您快速了解应用服务详细信息，例如 Pod 详情、负载均衡信息、统一接入信息等。同时，应用服务详情页还提供了丰富的功能菜单，能够让您更加快捷高效地进行应用服务的更新操作，包括镜像更换、扩缩容等。

## 操作步骤

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击应用服务实例名称，进入 **应用服务详情** 页面。
3. 查看应用服务配置信息。

应用服务详情 页包括以下内容。

模块	说明
基本信息	应用服务的基本信息，例如应用服务名称、所属应用、创建人信息、技术栈、镜像来源、请求 CPU、请求内存、最大 CPU、最大内存、命名空间、副本个数等。
Pod	显示部署单元维度容器服务所包含的 Pod。可使用 web shell 登录进入 Pod 内的容器进行操作。详情参考 <a href="#">Pod 详情</a> 。 在 Pod 列表页，单击 <b>操作</b> 列中的 <b>监控</b> ，可监控 Pod 详情。
负载均衡	显示服务名称、网络类型、负载均衡地址、NodePort、部署单元、创建时间。
统一接入	显示统一接入名称、统一接入实例名称、部署单元等信息。 单击统一接入实例名称可跳转至 <b>实例详情</b> 页。
版本	显示应用服务的版本号、发布单、发布状态等信息。详情参考 <a href="#">查看应用服务版本</a> 。
事件	显示事件数量、事件内容、创建时间。 支持按自定义的时间段筛选，只需在日历中选择开始和结束时间，或者在文本框内手动输入即可。
Webshell日志	显示具体的日志内容，包括 Pod 名称、事件内容、操作人和事件创建时间。

## Pod 详情

### 容器控制台

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击应用服务名称进入 **应用服务详情** 页。
3. 在 Pod 列表中，单击 Pod 名称进入 **Pod 详情** 页。
4. 在容器列表中，单击容器右侧的 **前往控制台**。

## 5. 输入命令管理容器。

命令样例: `exec [-cl] [-a name] [command [arguments ...]] [redirection ...]`

## 查看容器详情

在容器列表中，单击容器名称进入容器详情页。您可以查看容器的基本信息。单击下方页签可查看事件、标准日志、环境变量、数据卷配置详情。例如，您可以查看容器在某一时间段的标准日志记录，如下图所示。

容器详情 - 运行中 控制台

最大 CPU: 700m 最大内存: 700Mi  
创建时间: 2021/03/17 21:23:02 启动命令:

事件 标准日志 环境变量 数据卷配置

起始时间: 2021-03-01 11:44:41 15 分钟 1 小时 2 小时 3 小时 查询 复制到剪贴板

```
RUN_MODE 运行模式未指定, 默认使用 GLOBAL
spring.application.name: ha- app
cafe.pod.name: sofa-cell-rz- m-c58s4
cafe.pod.namespace: -01-default
cafe.pod.ip: 100. .18
cafe.pod.tenant.name: 
cafe.pod.workspace.name: test0301
cafe.pod.workspace.group.name: 
cafe.pod.cluster.name: -01-import-cluster-05- /35
cafe.pod.cell.name: RZ00A
cafe.pod.dc.name: cn-hangzhou-a
cafe.pod.region.name: cn-hangzhou
cafe.pod ldc.server.url: 17. .71
java: no process found
```

## 查看 Pod 事件

参考 [查看 Pod 事件](#)。

## 查看应用服务版本

1. 登录控制台，在左侧导航栏单击 **发布运维** > 应用服务。
2. 在应用服务列表页，单击应用服务名称进入 **应用服务详情** 页。
3. 选择 **版本** 页签进入版本列表页。默认会将当前版本显示在最上面。您可以查看以下版本信息：
  - **应用服务版本**：应用服务的所有提交过的版本 ID，单击版本链接会跳转到查看应用服务的详细配置页面。
  - **发布单**：若服务已经发布，会展示关联的发布单 ID，单击发布单链接会跳转到发布单详情页。
  - **发布状态**：发布完成、发布失败、待部署、已取消、已回滚等。

- **版本状态**：版本的提交状态，如已提交。
- **创建人**：显示应用服务版本的创建人。
- **创建时间**：显示应用服务版本的创建时间。
- **操作**：支持版本对比、导出、删除操作。
  - **版本对比**：显示当前版本与其他版本间的差异。
  - **导出**：将应用服务数据导出至本地。详情参考 [导出应用服务](#)。
  - **删除**：删除不需要的应用服务版本。

 **注意**

- 删除版本后，将无法恢复，请谨慎操作。
- 版本删除功能约束与限制如下：
  - 当仅存在单个应用服务版本时，不允许删除。
  - 当存在多个应用服务版本时，发布状态为 **发布中**、**运行中** 或 **回滚中** 的版本不允许删除。

## 其他操作

在应用服务详情页，您还可以根据需要进行更多快捷操作，具体可参考下表。

操作	说明
更新应用服务	在应用服务详情页，单击 <b>编辑配置信息</b> 。 详情参考 <a href="#">更新应用服务</a> 。
仅更换镜像	单击 <b>快捷操作</b> ，选择 <b>更换镜像</b> ，选择目标版本镜像即可完成镜像更新。 详情参考 <a href="#">更新应用服务</a> 。
仅更新伸缩配置	单击 <b>快捷操作</b> ，选择 <b>扩缩容</b> ，然后按需修改副本数即可完成伸缩配置更新。 详情参考 <a href="#">更新应用服务</a> 。
立即发布应用服务	在应用服务详情页，单击右上角 <b>发布</b> 按钮即可立即发布应用服务。 详情参考 <a href="#">立即发布应用服务</a> 。

## 5.8. 删除应用服务

在应用服务列表页，您可以删除不再需要的应用服务。删除应用服务后，服务下的资源将全部被释放。本文主要介绍如何删除应用服务。

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 可以通过如下方式之一删除应用服务：
  - 找到需要删除的应用服务，单击其卡片上的更多图标 **...** > **删除**。



- 单击右上角列表图标进入应用服务列表页，单击目标应用服务右侧 **操作** 栏中的 **删除**。

应用服务							
<a href="#">创建应用服务</a> <a href="#">导入</a>		ant-...-est0823... <input type="text" value="搜索应用服务名称"/>					
应用服务	所属应用	镜像	Pods	发布状态	负责人	创建	操作
mei-...-center	me-...-enter	镜像版本: m-...-enter2021090713...-enter-test-0907.64	运行: 1 / 更新: 1 / 期望: 1	发布完成	admin	202	<a href="#">发布</a> <a href="#">删除</a>
bff-...-ant	bff-...-t	镜像版本: b-...-t2021090713513...-ur-test-0907.4157a4	运行: 2 / 更新: 2 / 期望: 2	发布完成	admin	202	<a href="#">发布</a> <a href="#">删除</a>
acc-...-enter	acc-...-enter	镜像版本: a-...-enter20210907135...-enter-test-0907.67d...	运行: 2 / 更新: 2 / 期望: 2	发布完成	admin	202	<a href="#">发布</a> <a href="#">删除</a>

3. 在弹出的删除验证对话框中，输入应用服务名称后，单击 **确定**。

删除时，该应用服务下所有 Pod 也会随之一并删除。

#### 注意

正在发布中的应用服务是无法删除的。

## 相关文档

您可以参考以下文档，了解应用服务更多功能及操作：

- [创建应用服务](#)
- [发布应用服务](#)
- [更新应用服务](#)
- [导入与导出应用服务](#)
- [查看应用服务详情](#)

## 5.9. 系统环境变量

作为基于 Kubernetes 的容器服务平台，LHC 默认会在容器里注入一系列平台相关的环境变量，这样通过单元化应用服务部署的应用可以很方便地通过环境变量获取许多基础信息。平台默认注入的环境变量列表如下：

#### 说明

系统环境变量为只读属性，用户不可以自行修改。

环境变量名称	环境变量解释	示例
CAFE_ENV_TENANT_NAME	租户名称	ANTCLOUD
CAFE_ENV_APPNAME	应用名称	transform
CAFE_ENV_DC_NAME	机房名称	cn-hangzhou
CAFE_ENV_REGION_NAME	Region 名称	杭州金区
CAFE_ENV_CELL_NAME	部署单元名称	CellA/RZ01A（单元化架构）
CAFE_ENV_WSG_NAME	工作空间组名称	dev
CAFE_ENV_WS_NAME	工作空间（单元化工作空间）名称	dev
CAFE_ENV_POD_NAME	Pod 名称	transform-01
CAFE_ENV_POD_NAMESPACE	Pod 所属 namespace	default
CAFE_ENV_POD_IP	Pod IP	127.0.0.1
CAFE_ENV_HOST_NAME	Pod 所属机器的名称	node01
CAFE_ENV_HOST_IP	Pod 所属机器的 IP	127.0.0.1
app.env.zmode	是否是单元化模式（中间件依赖）	true、false
app.env.dbmode	数据库 mode（中间件依赖）	test、pre、prod

## 5.10. 容器水平伸缩 HPA

容器水平伸缩（Horizontal Pod Autoscaler，简称 HPA）是 Kubernetes 中实现 Pod 水平自动伸缩的功能。对于 Kubernetes 中的 Pod 而言，HPA 可以实现很多自动化功能，比如当 Pod 中工作负载上升时，会基于设定的扩容规则，动态调整副本数，从而保证业务系统稳定运行；当 Pod 中工作负载下降时，会基于设定的缩容规则，动态调整副本数，从而提高资源利用率。

### HPA 基本原理

HPA 是用来控制 Pod 水平伸缩的控制器，基于设定的扩缩容规则，实时采集监控指标数据，根据用户设定的指标阈值计算副本数，进而调整目标资源的副本数量，完成扩缩容操作。

HPA 工作流程如下：



### 通过控制台创建 HPA

LHC 支持在控制台界面上快速创建支持 HPA 的应用服务，您可以轻松通过单元化应用服务控制台在创建应用服务时进行创建。

#### 操作步骤

1. 登录控制台，在左侧导航栏单击 **发布运维 > 应用服务**。
2. 在应用服务列表页，单击 **创建应用服务**。
3. 在 **创建应用服务** 页面，完成基本信息、Pod 模板配置后，单击 **下一步**。
4. 在 **弹性配置** 页面，完成以下配置后，单击 **下一步**。

\* 副本伸缩策略配置

固定副本数

弹性扩缩容(体验)

\* 固定副本数 ①

☒ GZ00B

2

☐ RZ00A

0

☒ GZ00A

3

☐ RZ01A

0

\* 最小副本数 ①

1

\* 最大副本数 ①

10

\* 规则 ①

*规则类型	*目标类型	*使用率/平均值	*单位	操作
CPU	使用率	80	%	删除

+ 添加配置

高级配置 ☒ 显示

扩容规则

缩容规则

幅度: 百分比 45

间隔时间: 500 秒

冷却时间: 300 秒

- 副本伸缩策略配置：选择 弹性扩缩容。
- 固定副本数：勾选部署单元并修改为期望副本数。
- 最小副本数：弹性扩缩容时 Pod 的最小副本数。默认值为 1。
- 最大副本数：弹性扩缩容时 Pod 的最大副本数。默认值为 10。
- 规则：当前伸缩规则基于的指标类型。单击 + 添加配置，配置以下选项：
  - 规则类型：当前伸缩规则基于的规则类型。目前支持 CPU、Memory、QPS、响应时间。
  - 目标类型：当前伸缩规则基于的指标类型所期望达到的值的类型，目前支持 使用率 与 平均资源使用。
  - 使用率/平均值：输入使用率/平均值。
  - 单位：无需输入，根据所选的规则类型自动匹配。
- 高级配置：默认关闭。如果开启 ☒ 显示，需要为扩缩容规则配置幅度、间隔时间、冷却时间。
  - 幅度：可以通过设置按百分比或指定个数来调整扩缩容的速率，单位为百分比或个数。
  - 间隔时间：上次扩缩容操作与本次扩缩容操作之间的时间间隔，单位为秒。

> 文档版本：20220928

101

- **冷却时间**：冷却扩缩容操作的时间，即完成一次扩缩容操作之后不再次触发扩缩容操作的时间窗口，单位为秒。

#### ❓ 说明

一个扩缩容活动执行完成后，在设定的冷却时间内不执行其他扩缩容活动。

5. （可选）完成访问配置、部署和调度配置后，单击 **提交**。

## 通过 kubectl 命令创建

您还可以通过编排模板来手动创建 HPA，并将其绑定到要伸缩的 Deployment 对象上，通过 kubectl 命令实现容器自动伸缩配置。

下面针对一个 Nginx 应用进行举例。

1. 创建并复制以下内容到 nginx.yml 中。Deployment 的编排模板如下。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
          ports:
            - containerPort: 80
          resources:
            requests: ##必须设置，否则 HPA 无法运行。
              cpu: 500m
```

#### ❓ 说明

需要为 Pod 设置 request 资源，如果不设置，HPA 不会运行。

2. 执行以下命令，创建 Nginx 应用。

```
kubectl create -f nginx.yml
```

3. 创建 HPA。通过 scaleTargetRef 设置当前 HPA 绑定的对象。在本例中，绑定对象是名为 nginx 的 Deployment。

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-hpa
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      targetAverageUtilization: 50
```

##绑定名为 nginx 的 Deployment。

4. 执行 `kubectl describe hpaname` 会发现类似以下信息的警告信息。

```
Warning FailedGetResourceMetric 2m (x6 over 4m) horizontal-pod-autoscaler miss
ing request for cpu on container nginx in pod default/nginx-deployment-basic-75675f5897
-mqzs7
Warning FailedComputeMetricsReplicas 2m (x6 over 4m) horizontal-pod-autoscaler fail
ed to get cpu utilization: missing request for cpu on container nginx in pod default/ng
inx-deployment-basic-75675f5创建好HPA后，再次执行kubectl describe hpa
```

5. 创建好 HPA 后，再次执行 `kubectl describe hpaname` 命令。如果看到以下信息，则表示 HPA 已正常运行。

```
Normal SuccessfulRescale 39s horizontal-pod-autoscaler New size: 1; reason: All metrics
below target
```

此时，当 Nginx 的 Pod 的利用率超过本例中设置的 50% 利用率时，则会进行水平扩容，低于 50% 时会进行缩容。

## 5.11. 运维单

借助运维单功能，您可以重启或替换单个或多个应用服务的 Pod，还可以指定 Pod 进行缩容，不再受 Pod 编号顺序约束。本文介绍如何通过控制台创建运维单，以及查看运维单详情等。

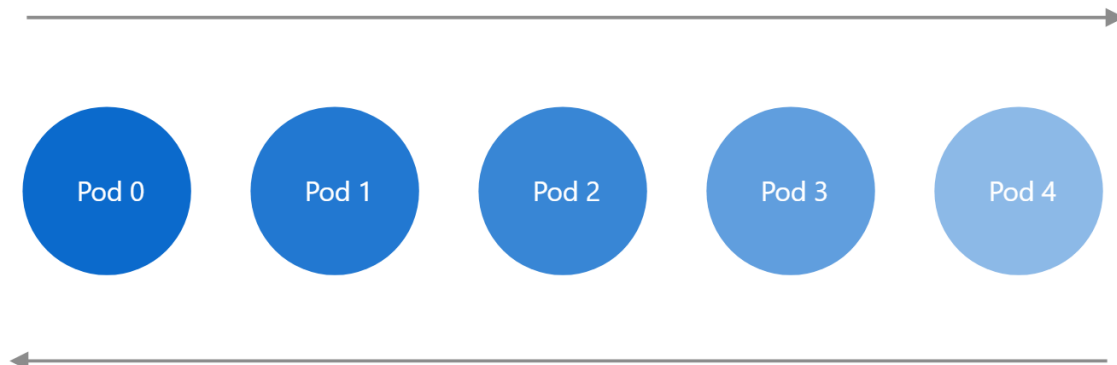
### 背景信息

运维单发布提供如下运维功能：

- **重启**：通过指定 Pod 进行重启操作。
- **替换**：通过直接删除 Pod，[CafeDeployment](#) 会自动生成新的 Pod，从而实现 Pod 的替换。
- **指定缩容**：指定 Pod 进行缩容，通过为 Pod 设置标记来实现。

若不使用指定缩容功能，扩缩容链路会根据 [CafeDeployment](#) 为每个 Pod 维护的 Pod 编号，按从小到大（或从大到小）编号顺序实现 Pod 的扩缩容操作，可参考下图。

## 扩容顺序

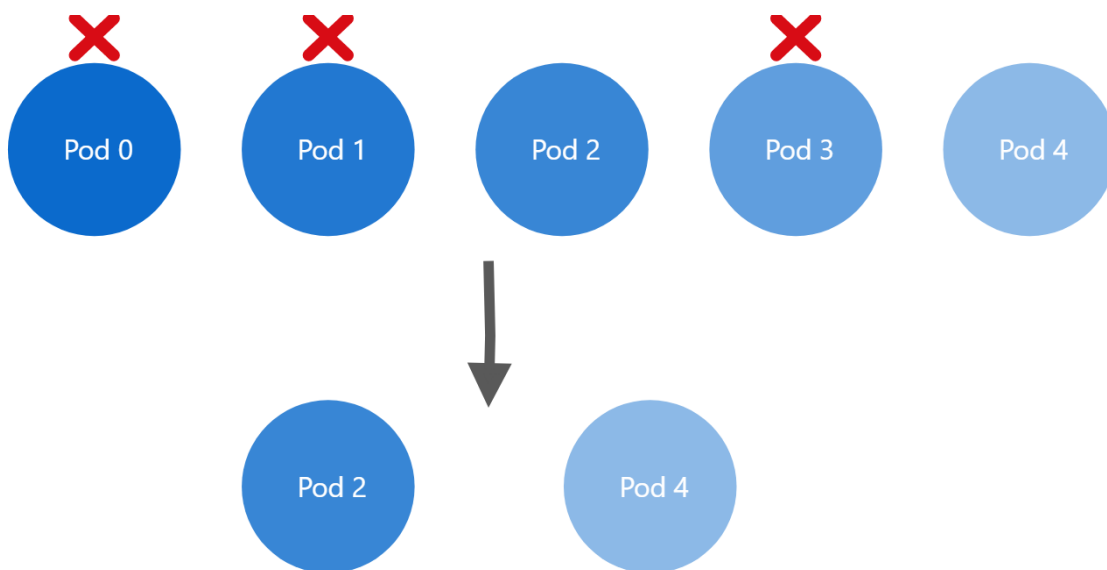


## 缩容顺序

InPlaceSet 提供指定缩容能力，只需先在目标 Pod 上设置标记

`cafe.sofastack.io/deletion-priority`，再减少 CafeDeployment 或 InPlaceSet 的

`spec.replicas` 来实现缩容。此时 CafeDeployment 会控制其下的 InPlaceSet 进行缩容，对于被设置标记的 Pod 则会优先被缩容，从而实现指定 Pod 缩容能力。



## 创建运维单

1. 登录控制台。
2. 在左侧导航栏单击 发布运维 > 运维单。
3. 在运维单列表页，单击 创建运维单。
4. 在 创建运维单 页，填写以下信息后，单击 提交。
  - 标题：输入运维单标题。
  - 运维操作类别：详细说明见下表：

参数	说明
重启	重启单个或多个应用服务的 Pod。
替换	替换单个或多个应用服务的 Pod。
指定缩容	<p>指定 Pod 进行缩容，不再受 Pod 编号顺序约束。</p> <div> <b>注意</b> 集群组件 cafeextcontroller 版本必须大于或等于 0.7.2，否则将随机缩容。</div>
指定摘流	<p>指定 Pod 进行摘流。</p> <div> <b>注意</b> 如果要选择 <b>指定摘流</b> 类别，需满足以下两个条件：<ul style="list-style-type: none"><li>■ 集群组件 cafeextcontroller 版本必须大于或等于 0.7.2。</li><li>■ cloud-controller-mananager 版本需大于或等于 0.8.0。</li></ul></div>
负载均衡流量调拨	<p>选择后可以对部署单元 Pod 进行流量权重设置。</p> <div> <b>注意</b> 如果要选择 <b>负载均衡流量调拨</b> 类别，需满足以下三个条件：<ul style="list-style-type: none"><li>■ 应用服务需配置自定义负载均衡，如何配置请参见 <a href="#">创建应用服务</a>。</li><li>■ 集群组件 cloud-controller-manager 版本需大于或等于 1.2.1。</li><li>■ 集群网络插件为 terway-eni。</li></ul></div>

- **命名空间**：从下拉列表选择对应命名空间。
- **应用服务列表**：运维操作类别选择 **负载均衡流量调拨** 后显示，选择对应的应用服务，支持多选。
- **部署单元Pod权重设置**：运维操作类别选择 **负载均衡流量调拨** 后显示，配置单元 Pod 的流量权重。

 注意

- 不能存在权重为 0 的情况，只要有一个部署单元的权重设置为 0，应用服务将发布失败，需要重新创建一个运维单将权重设置为非 0 才能发布成功。
- 如果部署单元中的 Pod 数量不一致，那么整个部署单元的权重不等于此处设置的部署单元 Pod 权重。举例：GZ00A 权重 50、副本数 1，GZ00B 权重 50 副本数 2，那么部署单元的权重比例应该是  $50 * 1 : 50 * 2 == 1 : 2$ 。

- 发布策略：默认关闭，开启后，您可以设置发布策略，包括按部署单元分组、按部署单元百分比等策略。
- Pod 列表：单击 添加，选择应用服务、Pod。

← 创建运维单

\* 标题

O&M\_Order\_Test

\* 运维操作类别

重启

\* 命名空间

antcloud-cafe2021-default

\* pod列表

应用服务	Pod	操作
clcore-mosn2	clcore-mosn2-cell-gz00a-z6vt2-fgjtx ×	删除
clcustomerlayotto	clcustomerlayotto-cell-gz00a-44w8c-7nks2 ×	删除
+ 添加		

提交

取消

5. 单击 提交。

在运维单列表页，单击创建的运维单会跳转到 **运维单详情** 页面，单击 **整体发布** 即可开始发布。

## 查看运维单详情

在 **运维单详情** 页面，您可以查看运维单的基本信息和执行详情。

### 说明

如果运维单类别是 **负载均衡流量调拨**，在此页面可以进行流量调拨操作以及查看对应的流量调拨信息。

- **基本信息**：运维单 ID、执行人、发布类型等。
- **应用服务**：分组发布应用服务，Pod 状态说明如下：
  - **未更新**：Pod 配置未更新为最新版本。
  - **更新中**：Pod 配置已更新为最新版本，但状态还未 ready 或保持 ready 状态少于 30 秒。
  - **更新成功**：Pod 配置已更新为最新版本，并保持 ready 状态 30 秒以上。
  - **更新失败**：上线或更新 Pod 异常。

← 重启运维单详情-O&M\_Order\_Test

基本信息

执行成功

ID  
cafe2021\_202112011738421...

执行人  
admin

发布类型  
运维单发布

开始时间  
2021-12-01 17:38:49

结束时间  
2021-12-01 17:40:08

应用服务发布顺序

分组A

▼ A1 执行成功

clcore-mosn2(antcloud-cafe2... 执行成功

部署策略：每个Pod一组

clcustomerlayotto(antcloud-... 执行成功

部署策略：每个Pod一组

## 其他操作

- 如需在发布运维单过程中终止发布操作，在 **运维单详情** 页，单击 **取消** 即可取消当前运维单发布。
- 若运维单初始化失败，则无法进行发布，可以将鼠标悬停到 **失败原因** 上，查看具体原因，或单击 **重新初始化**。

## 6. 流量管理

### 6.1. 应用层流量管理

#### 应用层流控

单元化架构下应用层流控规则主要涉及两块内容：UID 分配和部署单元的流量比。UID 分配指各个本地域逻辑单元组（RZoneGroup）所对应的 UID 分片（00-99）；部署单元流量比指各个部署单元所占对应的逻辑单元组（GZoneGroup、CZoneGroup、RZoneGroup）的流量权重。



上图中流量分配情况表示两个 RZone（RZ01 和 RZ02）都承载流量，UID 分片在 0-19 的请求会访问到 RZ01，20-99 UID 分片的请求会访问到 RZ02。在 RZ01 和 RZ02 这两个逻辑单元内，只存在一个部署单元，分别是 RZ01A 和 RZ02A，所以每个部署单元就承载了 100% 的逻辑单元流量。逻辑单元 GZ01 包含两个部署单元 GZ01A 和 GZ01B，各占 50% 的逻辑单元流量。

#### 正式流量

##### 说明

- 压测流量与正式流量的功能使用一致，唯一区别在于压测流量规则仅适用于压测场景。
- 压测流量仅在专有云环境中支持。

在左侧导航栏单击 **流量管理** > **应用层**，在 **正式流量** 页签中，系统会先展示最近一次推送成功的全局流量规则，即当前生效的全局流量快照。当环境初始化时，在没有任何流量推送记录的情况下，或者推送的不是全局流量规则的情况下，不会展示任何内容。



## 规则配置

1. 单击 **规则配置** 进入 **正式流量规则配置** 页面，该页面展示的是当前已配置的规则（只是配置，不代表已生效）。
2. 单击逻辑单元卡片或分片数据即可进入编辑页面，填写以下配置信息：
  - 对于 RZone 类型的部署单元，配置对应部署单元的 UID 分片配置和流量权重配置。
  - 对于 GZone 类型的部署单元，配置对应部署单元的流量权重配置。

### 说明

请确保调整后的各个部署单元的流量比总和为 100%。UID 分片的取值范围为 00~99。

权重管理

分片管理

⚠ 请确保调整后的各个部署单元的流量比总和为 100%

RZ02A 流量比

100

权重管理

分片管理

分片: 20 ~ 99 ⊖

+ 添加 UID 分片

3. 调整权重或 UID 分片后，单击 **保存** 以确保配置生效。

## 流量规则推送

1. 在左侧导航栏单击 **流量管理 > 应用层**。
2. 单击 **规则配置** 进入 **正式流量规则配置** 页面。
3. 单击 **流量规则推送**，进入 **流量规则推送** 页面。
4. 设置以下推送配置：
  - **规则详情**：系统会根据当前配置的流量规则自动生成 JSON 格式的规则文本，该文本会作为实际推送到应用或中枢服务的参数，让全局应用或指定应用感知到流量变化。
  - **部署单元**：选择 **全部部署单元** 或 **指定部署单元**。可指定部署单元进行规则推送，针对局部部署单元内的应用生效。
  - **目标组件**：选择 **中间件** 或 **统一接入网关**。一般情况下流量规则的变更都需要让这两个组件感知，除非一些灰度测试场景。
  - **目标应用**：选择需要感知规则变化的目标应用，一般来说会选择全局应用推送，但在灰度或测试场景也可以只让部分应用感知新的流量规则做一些流量验证。
  - **是否覆盖其他应用**：选择推送到全局应用时，若开启覆盖其他应用，会用新配置的流量规则覆盖掉应用当前生效的规则。不覆盖，则会保留之前应用单独推送的规则。

流量规则推送

\* 规则详情

```
[
  {
    "domain": {
      "GZ00A": "GZ00A.alipay.net",
      "RZ00A": "RZ00A.alipay.net",
      "RZ01A": "RZ01A.alipay.net"
    },
    "drRule": [
      {
        "idcName": "cn-shanghai-a",

```

\* 部署单元

全部部署单元

指定部署单元

\* 目标组件

☒ 中间件 ☒ 统一接入网关

\* 目标应用

全部应用

指定应用

是否覆盖其他应用 ☐ 否

5. 单击 **推送**，开始推送当前配置完的流量规则。

## 推送历史

在 **推送历史** 页签中，您可以查看最近推送的记录及状态。

- **关联应用**：routeClient 表示推送全局应用规则，否则会展示实际推送勾选的目标应用。
- **目标部署单元/机房**：展示推送的目标机房或部署单元。
- **规则类型**：目前仅支持路由规则。
- **推送结果**：展示成功或失败。错误信息会在推送失败后展示一些可供排查的失败原因。

关联应用	目标部署单元/机房	规则类型	推送结果	错误信息	操作人	推送时间
+ routeClient	全部	路由规则	成功		zonemng	2020/03/17 23:19:37
+ routeClient	全部	路由规则	成功		zonemng	2020/03/17 22:41:00
+ routeClient	全部	路由规则	成功		zonemng	2020/03/17 15:50:11
+ routeClient	全部	路由规则	成功		zonemng	2020/03/17 14:20:34

## 7. 网络

### 7.1. 打通多集群容器网络

目前在单元化架构场景中，多个 ACK 集群的容器之间网络互通需要通过为集群安全组添加访问授权的方式实现。

#### 说明

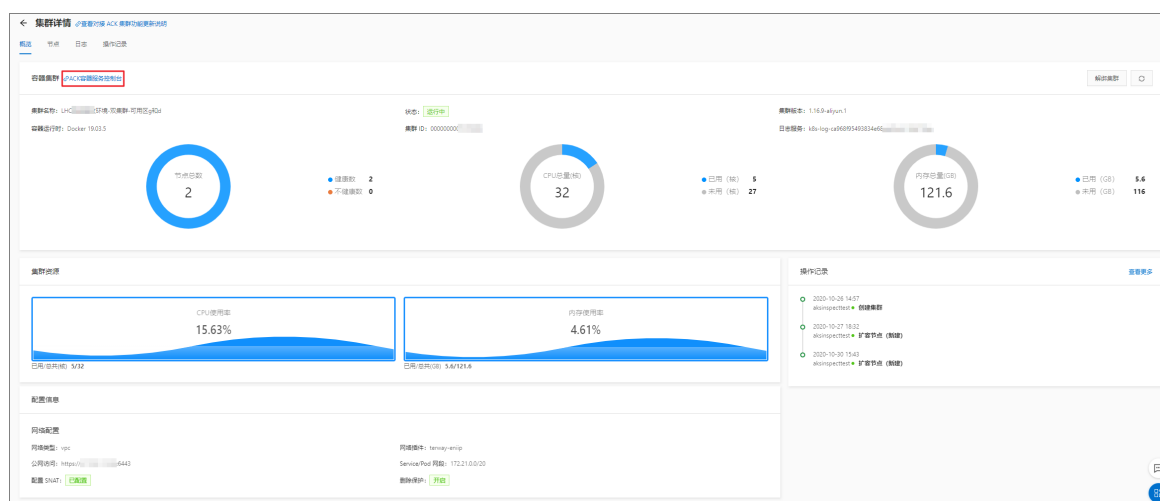
某个集群的安全组必须和其他所有集群的安全组打通，若总集群个数为  $N$ ，则每个集群的打通条目为  $N-1$ ，整体需要做的打通次数是  $N*(N-1)$ 。

本文以双集群环境为例，打通集群 1 和集群 2 的安全组。

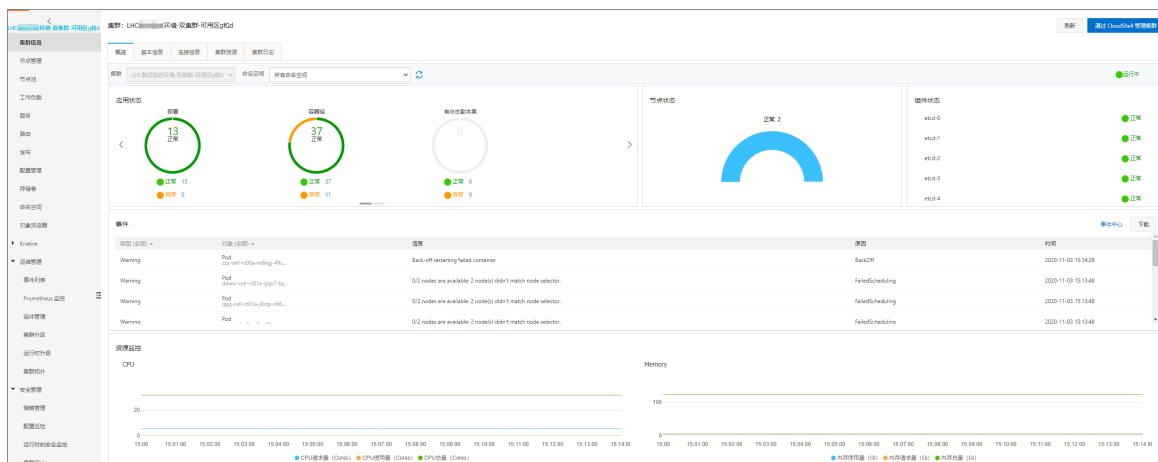


#### 操作步骤

1. 登录控制台。在左侧导航栏选择 **集群管理** > **集群**。
2. 单击目标集群卡片进入 **集群详情** 页。



3. 单击左上角的 **ACK 容器服务控制台** 跳转到 ACK 的集群信息概览页。



4. 单击 **集群资源** 页签，在集群管理的资源列表中，单击安全组链接，进入 **安全组规则** 页面。

集群: LHC-环境: 双集群-可用区g和d

概览	基本信息	连接信息	集群资源	集群日志
这些资源是容器服务集群管理的资源，请不要删除或自行修改，以避免导致集群异常，影响集群内应用的正常运行。				
资源编排 ROS			k8s-for-	
虚拟专有网络 VPC			vpc-	
节点虚拟交换机			vsw-	
Pod 虚拟交换机			vsw-	
安全组			sg-	
Worker RAM 角色			KubernetesWorkerRole-6727ee17-99fc-4268-be2e	
伸缩组			asg-uf615h	
Nginx Ingress SLB			lb-uf60tl	
日志服务 Project			k8s-log-	
负载均衡 (SLB)			lb-	

5. 在 **入方向** 页签中，单击 **手动添加**，填入以下信息后，单击 **保存**。

- 授权策略：保持默认 **允许**。
- 优先级：优先级可选范围为 1~100，默认值为 1，即最高优先级。保持默认。
- 协议类型：选择 **全部**。
- 端口范围：保持默认 **-1/-1**，代表不限制端口。
- 授权对象：选择待被授权集群所在的安全组 ID。

访问规则 入方向 出方向

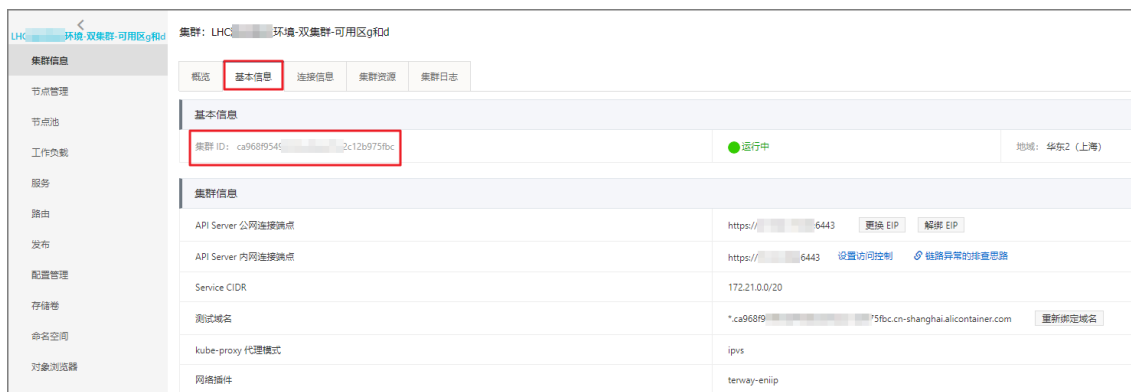
手动添加 批量编辑 输入端口范围和对端IP地址

授权策略	优先级	协议类型	端口范围	授权对象	描述	操作
允许	1	全部	-1/-1	sg-uf615h	输入安全组ID，多个请用逗号隔开	保存 删除 刷新

安全组 ID 后半部分是集群 ID，可以在 ACK 控制台 > 集群信息 > 基本信息 页签获取。

### 说明

安全组 ID 后半部分是集群 ID，可以在 ACK 控制台 > 集群信息 > 基本信息 页签获取。



## 7.2. 创建统一接入集群

统一接入集群承担统一接入和单元化网关的角色，在每个 K8s 集群内统一接入集群平行发布运维，多个平行的统一接入集群联合成一个联邦接入集群，是一个逻辑的集群实体。统一接入网关以集群的方式来承载业务负载均衡的流量，您需要根据流量的类型来规划集群，比如公网、内网等。

### 说明

集群上线后，您需要创建统一接入实例，并为应用服务创建负载均衡/统一接入类型的访问配置，才能进行各类业务流量的接入及路由，具体的转发及监听规则需在 [应用服务访问配置](#) 中设定。

### 组件作用

统一接入网关的作用主要是请求转发，即根据用户请求中的路由类型、域名和路径信息，识别出用户请求的目标应用并转发。

### 路由类型

路由类型主要分为 Gzone 路由（非单元化路由）和 Rzone 路由（单元化路由），详情如下

- Gzone 路由：根据目标应用和目标单元类型，随机找出其路由的目标服务器组，如果存在多个目标服务器组，则按照目标服务器组的权重来进行流量分发。
- Rzone 路由：当识别到用户请求最终目标为 Rzone 服务器时，通过用户请求的 Cookie 对锁进行解析，并计算出确定的 Rzone 路由目标。解析步骤如下：
  - i. 锁的 Key 为 zone\_op，通过 Key 计算出其 Value 值。
  - ii. 根据该 Value 值进行目标单元计算和转发。

### 路由示例

根据 zone\_op 对应 Value 值的范围，假设我们实现设置了如下路由规则：

- [00,49]，路由到 rz00 单元。
- [50,99]，路由到 rz01 单元。

则路由结果如下：

- 当 zone\_op 的 Value 值不在 [00,99] 范围内：会将流量路由到当前 spanner 所在机房的目标应用对应的服务器组中，如果存在多个目标服务器组，则按照目标服务器的权重进行流量分发。
- 当 zone\_op 的 Value 值位于 [00,49] 范围内：会将流量路由到目标应用的 rz00 单元所在目标服务器中，如果存在多个目标服务器组，则按照目标服务器的权重进行流量分发。

- 当 zone\_op 的 Value 值位于 [50,99] 范围内：会将流量路由到目标应用的 rz01 单元所在目标服务器中，如果存在多个目标服务器组，则按照目标服务器的权重进行流量分发。

## 使用场景

主要使用场景如下：

- 在异地多活、弹性流量和容灾切换场景下，需要对机房流量进行分配和再分配（划拨、切入和切出）。
- 动态修改路由规则之后，接入路由组件会根据新的路由规则，将业务流量路由至新的目标机房，从而实现流量分配、弹性容量和容灾切换。
- 部署方式：按机房部署，即各个机房各部署一套接入路由集群。

## 注意事项

- 统一接入集群的容器只能调度到专有的统一接入类型的节点上，因此，创建统一接入集群时需要保证有足够的资源供调度。
- 在创建集群后报资源不足调度失败错误时，您需要在集群管理页面进行节点扩容，并选择 **统一接入节点** 类型。详情请参考 [扩容集群](#)。
- 统一接入集群横跨各可用区，所以需要同时保证各可用区资源充足。

## 操作步骤

1. 登录控制台。
2. 在左侧导航栏中选择 **网络 > 统一接入集群**，进入网管集群列表页面。
3. 单击 **创建网关集群**，完成以下配置后，单击 **提交**。

## i. 完成基本信息配置。

配置项	说明
集群名称	只允许包含小写字母、数字、中划线，且必须以字母开头、以字母或数字结尾。长度为 1-29 个字符。
网络类型	可指定的网络类型包括内网、公网。指定集群的网络类型后，该集群上仅能创建与之相同网络类型的统一接入实例。
初始集群状态	<p>可指定 <b>在线</b> 和 <b>维护</b>。</p> <ul style="list-style-type: none"><li>■ <b>在线</b>：网关集群对外提供流量转发服务，同时可以接受转发规则的更新。</li><li>■ <b>维护</b>：网关集群仅对外提供流量转发服务，但不接受转发规则的更新请求。</li></ul>
容器规格	<p>统一接入集群内单个节点的总资源等于控制平面固定资源和数据平面自定义资源之和。容器的规格和节点的数量决定了该集群的请求处理能力。具体数据请参考 <a href="#">压测结果</a>。</p> <ul style="list-style-type: none"><li>■ <b>控制平面固定资源</b>：固定规格为 1C/1G/10G（CPU/内存/磁盘）。</li><li>■ <b>数据平面自定义资源</b>：可自定义，推荐使用的最小规格为 4C/8G/40G，CPU：内存：磁盘容量的配比为 1：2：10。通常情况下，每 1C/2G/10G 的数据平面规格可以支持 3200 QPS 的 HTTP 短连接。</li></ul>
host 网络模式	<p>网关集群所采用的网络模型。</p> <ul style="list-style-type: none"><li>■ 若选择 <b>是</b>，Pod 会复用容器所在宿主机的 IP。</li><li>■ 若选择 <b>否</b>，每个 Pod 会拥有独立 IP。</li></ul> <div><p>❓ 说明</p><p>高密度部署模式下，仅可以选择 <b>否</b>，即选择非 host 模式。</p></div>

## ii. 完成容器版本配置。

配置项	说明
名称	网关容器的版本名称。长度为 1-5 个字符。只包含小写字母、数字、中划线，必须以字母开头，以字母或数字结尾。
容器镜像	网关容器的镜像地址，输入 <code>registry.cn-hangzhou.aliyuncs.com/sofastack/spannerplus:1.1.0</code> 。
副本数	网关节点的数量。出于容灾需要，建议一个集群至少包含 3 个以上节点。可以按照对容量的需求，合理规划容器规格与节点数量。 <div>❓ 说明 创建副本数为 0 的网关集群会导致集群不可用，需要重新扩容。</div>
标签	输入 key、value，为该版本的网关容器添加对应的 label。

## 7.3. 统一接入节点性能压测

本文介绍 LHC 统一接入节点性能压测相关内容，包括压测环境、压测方案和压测结果等信息。

### 压测概述

产品名称	部署拓扑/规格	性能指标概述
单元化应用服务 LHC	4C8G，4 个统一网关集群节点可以支持的最大 QPS 数量。	<ul style="list-style-type: none"><li>• HTTP 短连接：9000 QPS</li><li>• HTTP 长连接：11000 QPS</li><li>• HTTPS 短连接（不复用 session）：1500 QPS</li><li>• HTTPS 短连接（复用 session）：4000 QPS</li><li>• HTTPS 长连接：9000 QPS</li></ul>

### 压测环境

本次压测所使用的机器配置信息如下：

- ECS 类型：ecs.sn2ne.2xlarge 系列，规格为 8 vCPU、32 GiB 内存。

- 操作系统：CentOS Linux 7 (Core)
- 内核版本：3.10.0-1062.18.1.el7.x86\_64

## 压测方案

### ? 说明

本次 Spanner 跟应用混部，节点上没有特殊的网络配置。

- 配置：
  - upstream 关闭长连接。
  - 采用 rule-tt-vsg 精细化引流配置。
- 压测 endpoint：直接压测 Spanner IP。
- 压测工具：wrk
- 后端服务：nginx
- 返回 body 大小：本次压测没有考虑不同 body 大小的场景。
- Spanner 配置：单 worker 节点配置为 4C8G，4 个 worker 节点。
- HTTPS 请求协议为 TLSv1.2、加密套件：ECDHE-RSA-AES256-GCM-SHA384

- 统计方式：

```
tail -f access.log | grep " 200 " | awk '{print $3}' | uniq -c
```

## 测试详情

### HTTP 短连接

```
wrk -t4 -c 60 -d 200 -H 'Connection: Close' -H 'Host: XXX' -http://IP:PORT
```

9000 QPS 左右

```
8820 [20/Nov/2020:17:10:16
8601 [20/Nov/2020:17:10:17
9232 [20/Nov/2020:17:10:18
9243 [20/Nov/2020:17:10:19
9291 [20/Nov/2020:17:10:20
8903 [20/Nov/2020:17:10:21
9446 [20/Nov/2020:17:10:22
9337 [20/Nov/2020:17:10:23
9346 [20/Nov/2020:17:10:24
9244 [20/Nov/2020:17:10:25
9454 [20/Nov/2020:17:10:26
9086 [20/Nov/2020:17:10:27
9474 [20/Nov/2020:17:10:28
9016 [20/Nov/2020:17:10:29
9315 [20/Nov/2020:17:10:30
9315 [20/Nov/2020:17:10:31
9317 [20/Nov/2020:17:10:32
9062 [20/Nov/2020:17:10:33
9088 [20/Nov/2020:17:10:34
9252 [20/Nov/2020:17:10:35
9073 [20/Nov/2020:17:10:36
8057 [20/Nov/2020:17:10:37
9244 [20/Nov/2020:17:10:38
9194 [20/Nov/2020:17:10:39
9154 [20/Nov/2020:17:10:40
9451 [20/Nov/2020:17:10:41
9316 [20/Nov/2020:17:10:42
9252 [20/Nov/2020:17:10:43
9271 [20/Nov/2020:17:10:44
9246 [20/Nov/2020:17:10:45
8004 [20/Nov/2020:17:10:46
```

## HTTP 长连接

```
wrk -t4 -c 60 -d 200 -H 'Host: XXX' http://IP:PORT
```

11000 QPS 左右

```
6121 [20/Nov/2020:17:18:58
11572 [20/Nov/2020:17:18:59
11854 [20/Nov/2020:17:19:00
11576 [20/Nov/2020:17:19:01
11645 [20/Nov/2020:17:19:02
11885 [20/Nov/2020:17:19:03
11679 [20/Nov/2020:17:19:04
11613 [20/Nov/2020:17:19:05
11805 [20/Nov/2020:17:19:06
11655 [20/Nov/2020:17:19:07
11733 [20/Nov/2020:17:19:08
11543 [20/Nov/2020:17:19:09
11673 [20/Nov/2020:17:19:10
11638 [20/Nov/2020:17:19:11
11366 [20/Nov/2020:17:19:12
11496 [20/Nov/2020:17:19:13
11686 [20/Nov/2020:17:19:14
11576 [20/Nov/2020:17:19:15
11578 [20/Nov/2020:17:19:16
11508 [20/Nov/2020:17:19:17
11477 [20/Nov/2020:17:19:18
11641 [20/Nov/2020:17:19:19
11464 [20/Nov/2020:17:19:20
11484 [20/Nov/2020:17:19:21
11503 [20/Nov/2020:17:19:22
11481 [20/Nov/2020:17:19:23
3748 [20/Nov/2020:17:19:24
```

## HTTPS 短连接，不复用 session

```
wrk -t4 -c 60 -d 200 -H 'Host: XXX' -H 'Connection: Close' -m 0 https://IP:PORT
```

1500 QPS

```
1510 [20/Nov/2020:17:20:25
1609 [20/Nov/2020:17:20:26
1561 [20/Nov/2020:17:20:27
1624 [20/Nov/2020:17:20:28
1655 [20/Nov/2020:17:20:29
1674 [20/Nov/2020:17:20:30
1561 [20/Nov/2020:17:20:31
1476 [20/Nov/2020:17:20:32
1650 [20/Nov/2020:17:20:33
1654 [20/Nov/2020:17:20:34
1665 [20/Nov/2020:17:20:35
1669 [20/Nov/2020:17:20:36
1570 [20/Nov/2020:17:20:37
1501 [20/Nov/2020:17:20:38
1611 [20/Nov/2020:17:20:39
1665 [20/Nov/2020:17:20:40
1667 [20/Nov/2020:17:20:41
1609 [20/Nov/2020:17:20:42
1655 [20/Nov/2020:17:20:43
1681 [20/Nov/2020:17:20:44
1560 [20/Nov/2020:17:20:45
1667 [20/Nov/2020:17:20:46
1647 [20/Nov/2020:17:20:47
1618 [20/Nov/2020:17:20:48
1631 [20/Nov/2020:17:20:49
1347 [20/Nov/2020:17:20:50
1652 [20/Nov/2020:17:20:51
1659 [20/Nov/2020:17:20:52
1213 [20/Nov/2020:17:20:53
1581 [20/Nov/2020:17:20:54
1332 [20/Nov/2020:17:20:55
1449 [20/Nov/2020:17:20:56
1439 [20/Nov/2020:17:20:57
1207 [20/Nov/2020:17:20:58
1632 [20/Nov/2020:17:20:59
```

## HTTPS 短连接，复用 session

```
wrk -t4 -c 60 -d 200 -H 'Host: XXX' -H 'Connection: Close' https://IP:PORT
```

4000 QPS

```
4923 [20/Nov/2020:17:22:28
5098 [20/Nov/2020:17:22:29
4899 [20/Nov/2020:17:22:30
4989 [20/Nov/2020:17:22:31
4355 [20/Nov/2020:17:22:32
4272 [20/Nov/2020:17:22:33
4372 [20/Nov/2020:17:22:34
4518 [20/Nov/2020:17:22:35
3844 [20/Nov/2020:17:22:36
4650 [20/Nov/2020:17:22:37
4068 [20/Nov/2020:17:22:38
3822 [20/Nov/2020:17:22:39
5085 [20/Nov/2020:17:22:40
4390 [20/Nov/2020:17:22:41
4485 [20/Nov/2020:17:22:42
3971 [20/Nov/2020:17:22:43
4137 [20/Nov/2020:17:22:44
4193 [20/Nov/2020:17:22:45
4411 [20/Nov/2020:17:22:46
3640 [20/Nov/2020:17:22:47
3927 [20/Nov/2020:17:22:48
4002 [20/Nov/2020:17:22:49
```

## HTTPS 长连接

9000 QPS 左右

```
180 [20/Nov/2020:17:23:27
9783 [20/Nov/2020:17:23:28
8667 [20/Nov/2020:17:23:29
9043 [20/Nov/2020:17:23:30
8692 [20/Nov/2020:17:23:31
9046 [20/Nov/2020:17:23:32
9226 [20/Nov/2020:17:23:33
8980 [20/Nov/2020:17:23:34
8691 [20/Nov/2020:17:23:35
9837 [20/Nov/2020:17:23:36
9471 [20/Nov/2020:17:23:37
9286 [20/Nov/2020:17:23:38
9608 [20/Nov/2020:17:23:39
9106 [20/Nov/2020:17:23:40
9538 [20/Nov/2020:17:23:41
8531 [20/Nov/2020:17:23:42
9006 [20/Nov/2020:17:23:43
8830 [20/Nov/2020:17:23:44
9395 [20/Nov/2020:17:23:45
8491 [20/Nov/2020:17:23:46
9671 [20/Nov/2020:17:23:47
8093 [20/Nov/2020:17:23:48
8829 [20/Nov/2020:17:23:49
```

## 压测结果

在上述压测环境下，单个网关节点支持的 QPS 如下：

### ② 说明

不同 ECS 系列、CPU 规格、网卡型号、内核参数、及不同的请求都会影响网关节点的转发性能，具体以业务压测结果为准。

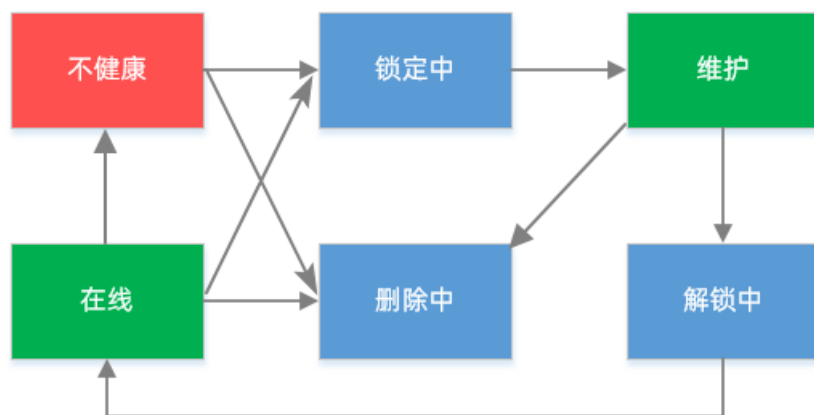
容器规格	HTTP 短连接	HTTP 长连接	HTTPS 短连接（不复用 session）	HTTPS 短连接（复用 session）	HTTPS 长连接
4C8G	9000 QPS	11000 QPS	1500 QPS	4000 QPS	9000 QPS

## 7.4. 网关集群状态说明

本文介绍统一接入集群在不同状态下的含义，便于您更深入地理解统一接入集群处于不同阶段的运行情况。

统一接入集群的状态可参考下表。

状态	状态属性	说明
不健康	稳定状态	统一接入集群处于不健康状态。 如果底层副本数不对或者集群创建出了问题，会显示 <b>不健康</b> 。
健康—在线	稳定状态	正常，网关集群对外提供流量转发服务，同时可以接受转发规则的更新。
健康—维护	稳定状态	锁定，网关集群仅对外提供流量转发服务，但不接受转发规则的更新请求。
锁定中	中间状态	统一接入集群处于正在被锁定的状态。
删除中	中间状态	统一接入集群处于正在被删除的状态。



在创建统一接入集群时，您可以设置初始集群状态为在线或维护，集群创建成功后，可以通过操作栏中的 **锁定**、**解锁**、**删除** 选项来变更集群状态。

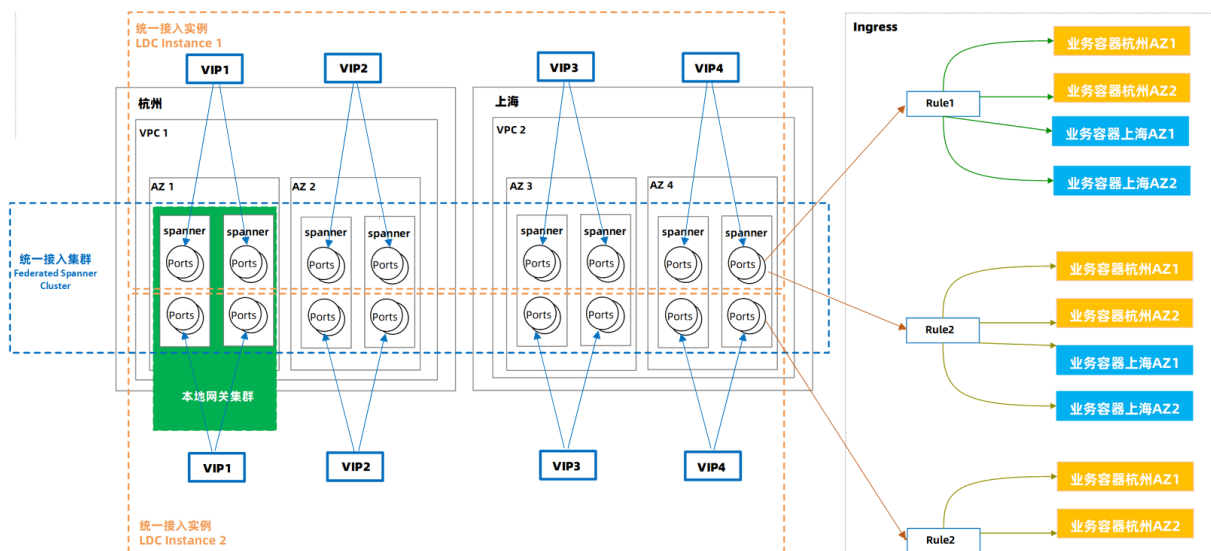
金融分布式架构		统一接入集群							
总览		创建网关集群							
运维管理		搜索集群名称							
应用管理		集群名称	ID	容器规格	host网络模式	网络类型	关联实例数	集群状态	创建时间
资源管理		-dev-test133312...	4000117	1C/1Gi/5Gi	否	内网	0	不健康	2020-10-22 16:06:07
容器应用服务		-dev-test113123...	4000068	1C/1Gi/5Gi	否	内网	0	不健康	2020-10-19 15:03:59
多集群容器引擎		-dev-test1	4000067	1C/1Gi/5Gi	否	公网	0	不健康	2020-10-19 15:02:51
概览		-dev-test788	4000059	1C/1Gi/5Gi	否	公网	0	维护	2020-10-16 16:03:52
发布运维		-dev-test777	4000058	1C/1Gi/5Gi	是	公网	0	不健康	2020-10-16 16:01:20
集群管理		-dev-test666	4000057	1C/1Gi/5Gi	是	内网	0	不健康	2020-10-16 15:59:22
网络		-dev-test123456	4000056	1C/1Gi/5Gi	是	内网	0	不健康	2020-10-16 15:49:49
统一接入实例		-20201010-1	4000023	1C/1Gi/5Gi	否	公网	11	在线	2020-10-10 16:22:13
配置管理		test1	-	1C/1Gi/5Gi	否	内网	0	删除中	2020-10-10 15:39:35
流量管理									
经典应用服务									
实时监控									
全局设置									
使用入门									

## 7.5. 创建统一接入实例

创建完统一接入网关集群后，您需要继续为网关集群节点绑定的统一接入实例，才能为业务应用提供统一的访问入口。

统一接入实例为一个VIP独立、统一接入集群共享的逻辑实例，在多个集群的情况下，该实例应该包含多个VIP，需要前置域名作为统一入口。

统一接入实例本质上是共享某个统一接入集群的协议入口集合，目前统一接入实例支持 HTTP 和 HTTPS 两种协议，同时统一接入集群也会为每个实例中的每种协议生成一个单独的端口。统一接入实例落在联邦网关集群上，一个实例横跨多个用户集群，作为统一入口。



### 前提条件

- 已创建统一接入集群。
- 网关集群的状态是 在线 或 维护中，且网络类型与接入实例的网络类型一致。

### 操作步骤

1. 登录 LHC 控制台。
2. 在左侧导航栏中选择 网络 > 统一接入实例，进入统一接入实例列表页面。

3. 单击 **创建实例**，填写以下配置信息。

○ **基本信息：**

- **实例名称：**必填，允许包含小写字母、数字、中划线、且必须以字母开头、以字母或者数字结尾。允许长度为 1~34 个字符。不可与已有实例名称重复。
- **网络类型：**支持 **内网** 或 **公网**。
- **统一接入集群：**选择可用的接入集群。选择网络类型后，可选集群仅显示网络类型相符合的统一接入集群列表。

○ **配置信息：**

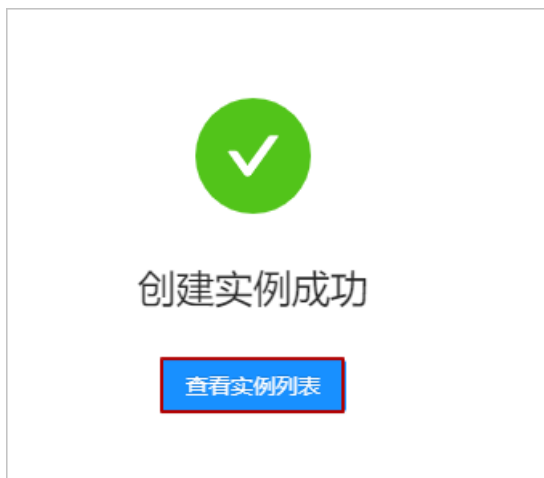
- **HTTP 协议：**必填。输入前端端口号，范围为 1~65535。
- **HTTPS 协议：**选填。单击 **添加 HTTPS 配置**，并输入前端端口号和证书 ID。

② 说明

目前 HTTPS 协议的支持仍在内测中，不建议在生产环境中使用。若需测试，可以提交工单获取证书 ID。

4. 单击 **提交**。

系统将弹出“创建实例成功”对话框，提示已成功创建统一接入实例。单击 **查看实例列表**，即可看到创建的实例出现在统一接入实例列表中。



## 7.6. 联邦负载均衡

### 7.6.1. 创建联邦负载均衡

联邦负载均衡，即不同部署单元负载均衡所引入的抽象层，每次在应用服务内应用时，只需引用了一个联邦的负载均衡 ID 即可，剩下关联不同部署单元的实际负载均衡，便交由联邦负载均衡管控。您可以根据业务场景选择配置对外公开或对内私有的联邦负载均衡，系统会根据您的选择分配公网或内网地址。本文介绍如何通过控制台创建联邦负载均衡。

#### 操作步骤

1. 登录控制台。



联邦负载均衡

添加联邦负载均衡

刷新

antcloud-10823...

搜索名称

名称	地址类型	状态	创建时间	关联应用服务数	操作
test	内网	成功	2021-09-06 19:57:52	1	<a href="#">配置负载均衡</a> <a href="#">删除</a>
cloudbn	内网	成功	2021-09-06 14:11:56	0	<a href="#">配置负载均衡</a> <a href="#">删除</a>

< 1 >

10 条/页

## 7.6.2. 配置联邦负载均衡

本文介绍如何通过控制台配置联邦负载均衡实例。

### 前提条件

已创建联邦负载均衡

### 操作步骤

1. 登录控制台。
2. 在左侧导航栏单击 **网络 > 联邦负载均衡**，进入联邦负载均衡列表页面。
3. 单击目标联邦负载均衡右侧操作列中的 **配置负载均衡**。
4. 在 **配置联邦负载均衡** 页，根据需求选择部署单元。

华东 1 金融云

搜索

多集群容器引擎 / 网络 / 联邦负载均衡

联邦负载均衡

添加联邦负载均衡

名称	地址类型	状态	创建时间
test	内网	创建或更新中	2022-03-17 15:19:59

配置联邦负载均衡

\* 名称

test

\* 命名空间

tthwywof-lhcprod0314-default

\* 地址类型

内网

\* 规格

简约型(slb.s1.small)

\* 部署单元列表

☒ GZ00A

☒ RZ00A

☐ GZ00B

☐ RZ01A

取消

提交

#### 说明

- 仅支持选择部署单元，其他配置项均不支持修改。
- 对于部署单元列表内选中的部署单元，不支持取消选择。

5. 单击 **提交**。

## 7.6.3. 删除联邦负载均衡

在联邦负载均衡列表页，您可以删除不再需要的联邦负载均衡实例。本文主要介绍如何通过控制台删除联邦负载均衡。

## 前提条件

如果已使用联邦负载均衡（即已关联应用服务），需要先删除关联的应用服务（详见 [删除应用服务](#)），否则无法删除联邦负载均衡。

### 说明

后续应用服务版本管理将提供 **废弃历史版本** 能力，届时只需废弃关联了联邦负载均衡的应用服务版本后，即可删除关联的联邦负载均衡实例。

## 操作步骤

1. 登录控制台。
2. 在左侧导航栏单击 **网络 > 联邦负载均衡**，进入联邦负载均衡列表页面。
3. 单击目标联邦负载均衡右侧操作列中的 **删除 > 确定**。

## 相关文档

您可以参考以下文档，了解联邦负载均衡更多功能及操作：

- [创建联邦负载均衡](#)
- [配置联邦负载均衡](#)

# 7.7. 服务（Services）

## 7.7.1. 概述

本文介绍服务（Service）的背景信息以及基本概念信息，以帮助您快速了解服务（Services）。

### 背景信息

由于应用通常具备敏捷迭代、快速弹性等特性，而单一容器及其相关网络资源的生命周期较短暂，因此亟需固定的访问地址，以及自动负载均衡来实现灵活的业务弹性。LHC 采用 Service 方式为一组容器提供固定的访问入口，并对这一组容器做负载均衡。

### Service 访问方式

LHC 提供如下几种 Service 访问方式。

### 说明

对于 LoadBalancer 类型的 Service，通常情况下，您只需在应用服务访问配置中 [添加负载均衡配置](#) 即可，系统会自动生成 LoadBalancer 类型的 Service，实现负载均衡实例的管理。

访问方式	Service 类型	说明
公网	LoadBalancer	使用 Service 的 LoadBalancer，公网 IP 可直接访问到后端 Pod。

访问方式	Service 类型	说明
内网	LoadBalancer	使用 Service 的 LoadBalancer 并设置注解，来创建私网访问的 Service 以及相应的内网 SLB。
端口访问	NodePort	提供主机端口映射到容器的访问方式，通过暴露集群节点上的某一固定端口暴露，以便在集群外部可通过节点 IP 和固定端口进行访问。
仅集群内访问	ClusterIP	使用 Service 的 ClusterIP，自动分配 Service 网段中的 IP，用于集群内访问。
域名访问	Headless Service	当不需要 SLB 以及单独的 Service IP 时，可采用 Headless Service 这种方式。Headless Service 对应的每一个 Endpoint，即每一个 Pod，都会有对应的 DNS 域名。配置 Headless Service 名称，这样各 Pod 之间便可以通过域名互相访问。

## 相关文档

- 关于 Kubernetes Service 的更多信息，请参见 [Kubernetes 官方文档](#)。
- Service 具体操作可参考 [创建服务（Service）](#)。
- Headless Service 相关操作请参见 [添加 Headless Service](#)、[LHC Pod 域名配置](#)。

## 7.7.2. 创建服务（Service）

本文介绍如何通过控制台创建服务（Service）以及其他常用操作。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **网络** 页签。
3. 在左侧菜单选项，选择 **服务（Service）**，然后单击 **创建 Service**。
4. 在弹出的输入框中输入 YAML 内容，单击 **提交**。

YAML 样例如下：

```
apiVersion: v1
kind: Service
metadata:
  annotations:
  labels:
    name: service-demo
    namespace: default
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

参数说明可参考下表。

参数	说明
kind	标识 Service 资源类型。
metadata	Service 的基本信息，比如 Service 的名称、Label 和命名空间等。
spec.selector	标识 Service 的选择器。
spec.ports.port	标识 Service 暴露给 ClusterIP 的端口。
spec.ports.targetPort	标识后端 Pod 的端口。
spec.type	标识 Service 访问方式。 <ul style="list-style-type: none"><li>◦ <b>LoadBalancer</b>：使用 SLB 公开服务。当 Service 不指定 SLB 时，默认创建的 SLB 实例类型为公网。</li><li>◦ <b>ClusterIP</b>：在集群内部公开服务，可用于集群内部访问。</li><li>◦ <b>NodePort</b>：使用节点的端口映射到后端 Service。</li></ul>

## 执行结果

创建成功后，名为 **service-demo** 的 Service 将出现在列表页面中。



## 相关操作

### 注意

对于在创建或编辑应用服务 [访问配置](#) 过程中自动生成的 Service 资源（即带有 `cafe.sofastack.io/app-instance-group` 标签的 Service），不支持编辑或删除操作。

- 如需查看 Service 详情，请在 Service 列表页单击目标 Service 名称，进入 Service 详情页。在该页面上，可以查看 Service 基本信息及 YAML 配置信息。
- 如需搜索 Service，请在 Service 列表页右上角的输入栏，输入 Service 名称关键字即可进行搜索。
- 如需更新 Service，请在 Service 列表页单击操作栏中的 [查看/编辑](#)，并在编辑 Service 对话框中编辑 YAML 信息，然后单击 [提交](#)。
- 如需删除 Service，请在 Service 列表页单击操作栏中的 [删除](#)，然后在弹出的提示对话框中单击 [确定](#)。

## 7.8. 路由（Ingress）

### 7.8.1. 概述

路由（Ingress）是集群内服务（Service）对外暴露七层的访问接入点。借助 Ingress 资源可以配置不同的七层转发规则，比如通过访问路径路由到不同的 Service，达到七层负载均衡作用。本文介绍 Ingress 的基本概念信息，以帮助您快速了解 Ingress。

### 背景信息

Ingress 作为集群内服务对外暴露的访问接入点，是 Kubernetes 中的一个资源对象，用来管理集群外部访问集群内部服务的方式。您可以通过 Ingress 资源来配置不同的转发规则，从而达到根据不同的规则设置访问集群内不同的 Service 所对应的后端 Pod。

Service 基于四层 TCP 和 UDP 协议转发，而 Ingress 可以基于七层的 HTTP 和 HTTPS 协议转发，通过域名和路径实现更精细的把控。

### 相关文档

- 关于 Kubernetes Ingress 的更多信息，请参见 [Kubernetes 官方文档](#)。
- Ingress 具体操作可参考 [创建路由（Ingress）](#)。

### 7.8.2. 管理路由（Ingress）

Ingress 转发功能是否生效取决于集群中运行的 Ingress Controller 实例。通常情况下，您在应用服务访问配置中 [添加统一接入配置](#) 即可实现多集群 Ingress 路由能力。

## 查看 Ingress 详情

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **网络** 页签。
3. 在左侧菜单选项，选择 **路由（Ingress）**，即可查看 Ingress 列表信息。
4. 在 Ingress 列表页，单击目标 Ingress 名称，进入 Ingress 详情页。

在该页面上，您可以查看 Ingress 基本信息、规则信息及 YAML 配置信息。

## 相关操作

### 注意

对于在创建或编辑应用服务 [访问配置](#) 过程中自动生成的 Ingress 资源（即带有 `cafe.sofastack.io/app-instance-group` 标签的 Ingress），不支持编辑或删除操作。

- 如需创建 Ingress，请在 Ingress 列表页单击 **创建 Ingress**，然后在弹出的输入框中输入 YAML 样例，单击 **提交** 即可。
- 如需搜索 Ingress，请在 Ingress 列表页右上角的输入栏，输入 Ingress 名称关键字即可进行搜索。
- 如需更新 Ingress，请在 Ingress 列表页单击操作栏中的 **查看/编辑**，并在编辑 Ingress 对话框中编辑 YAML 信息，然后单击 **提交**。
- 如需删除 Ingress，请在 Ingress 列表页单击操作栏中的 **删除**，然后在弹出的提示对话框中单击 **确定**。

# 7.9. NetworkPolicy

## 7.9.1. 概述

网络策略（NetworkPolicy）是 Kubernetes 提供的一种资源，提供基于策略的网络控制。如果您希望在 IP 地址或端口层面（OSI 三层或四层）控制网络流量，您可以针对集群内特定应用采用网络策略。本文介绍 NetworkPolicy 的基本概念信息，以帮助您快速了解 NetworkPolicy。

### 背景信息

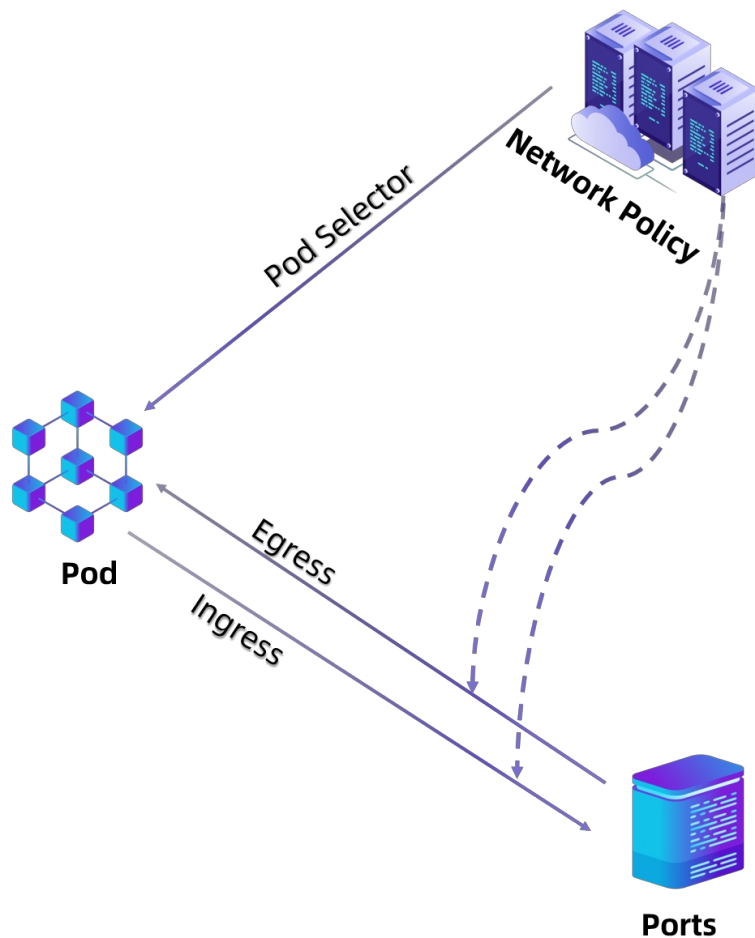
随着业务逻辑的复杂化，传统的单一外部防火墙或依照应用分层的防火墙已逐渐无法满足需求。

LHC 基于 Kubernetes 网络策略功能作了增强，通过配置网络策略，允许在集群内实现网络隔离，也就是可以在多组 Pod 之间架起防火墙。

### 基本原理

NetworkPolicy 会应用在与之相匹配的 PodSelector 的 Pod 上，基于入向（Ingress）或出向（Egress）规则指定允许访问这些 Pod 的源地址或这些 Pod 可以访问的目标地址。

Ingress 和 Egress 这两种规则可匹配由 PodSelector 选出的 Pod 或命名空间内的所有 Pod，或者通过 CIDR 指定的 IP 地址段。



#### ? 说明

- Ingress: 入方向流量规则，用于配置访问服务的源地址或端口白名单。
- Egress: 出方向流量规则，用于配制 Pod 访问外部服务的目的地址或端口白名单。

## 相关文档

- 关于 Kubernetes NetworkPolicy 的更多信息，请参见 [Kubernetes 官方文档](#)。
- NetworkPolicy 具体操作可参考 [创建 NetworkPolicy](#)。

## 7.9.2. 创建 NetworkPolicy

网络策略（NetworkPolicy）是 Kubernetes 提供了一种资源，提供基于策略的网络控制。如果您希望在 IP 地址或端口层面（OSI 三层或四层）控制网络流量，您可以针对集群内特定应用采用网络策略。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **网络** 页签。
3. 在左侧菜单选项，选择 **Networkpolicy**，然后单击 **创建 NetworkPolicy**。
4. 在弹出的输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-all-ingress-example
spec:
  podSelector:
    matchLabels:
      run: nginx
  ingress:
    - from:
      - podSelector:
          matchLabels:
            access: "true"
```

## 执行结果

创建成功后，名为 **allow-all-ingress-example** 的 NetworkPolicy 将出现在列表页面中。



## 相关操作

- 如需查看 NetworkPolicy 详情，请在 NetworkPolicy 列表页单击目标 NetworkPolicy 名称，进入 NetworkPolicy 详情页。在该页面上，可以查看 NetworkPolicy 基本信息及 YAML 配置信息。
- 如需搜索 NetworkPolicy，请在 NetworkPolicy 列表页右上角的输入栏，输入 NetworkPolicy 名称关键字即可进行搜索。
- 如需更新 NetworkPolicy，请在 NetworkPolicy 列表页单击操作栏中的 **查看/编辑**，并在编辑 NetworkPolicy 对话框中编辑 YAML 信息，然后单击 **提交**。
- 如需删除 NetworkPolicy，请在 NetworkPolicy 列表页单击操作栏中的 **删除**，然后在弹出的提示对话框中单击 **确定**。

## 8. 配置管理

### 8.1. 管理配置项

配置项（ConfigMap）是一种用于存储工作负载所需配置信息的资源类型，需要您自定义配置项的内容。

#### 创建配置项

您可以添加联邦配置项，也可以为单个的集群添加配置项。

##### 创建联邦配置项

1. 登录 LHC 控制台，在左侧导航栏单击 **配置管理 > 配置项**。
2. 单击 **创建**，在 **创建配置项** 页面，填写以下信息：
  - **命名空间**：选择所属命名空间。
  - **名称前缀**：配置项名称，同一个命名空间里命名必须唯一，最多 32 个字符。允许包含（小写）字母、数字、连字符，且必须以字母开头、以字母或者数字结尾。
  - **配置源**：添加键值对。变量名代表配置项数据的变量名，值代表配置项数据的值。
  - **自定义配置**：默认关闭。开启后，可以自定义部署单元级别的配置项。
3. 配置完成后，单击 **创建**。

##### 创建单集群配置项

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **配置** 页签。
3. 在左侧导航栏，选择 **配置项 (Configmaps)**。
4. （可选）选择所属命名空间，如果不选择，默认使用 **default**。
5. 单击 **创建**，在 **创建配置项** 页面，填写以下信息：
  - **命名空间**：所属的命名空间（已在上一步中选定，无需填写，默认为 **default**）。
  - **名称**：配置项名称，同一个命名空间里命名必须唯一，最多 32 个字符。允许包含（小写）字母、数字、连字符，且必须以字母开头、以字母或者数字结尾。
  - **配置源**：
    - **键值对**：添加键值对。变量名代表配置项数据的变量名，值代表配置项数据的值。
    - **文件**：可上传文件创建配置项。
6. 配置完成后，单击 **提交**。

#### 修改配置项

修改配置文件会影响使用该配置文件的应用。

1. 在配置项列表页，单击 **名称** 进入配置项详情页。
2. 您可以进行以下修改：
  - 单击 **编辑**，修改已有的配置项数据，单击 **提交**。
  - 单击 **添加**，增加变量名和值，单击 **提交**。

## 8.2. 使用配置项

您可以在 Pod 中使用配置项，主要包括以下使用场景：

- [用配置项定义 Pod 环境变量](#)
- [通过配置项设置命令行参数](#)
- [在数据卷中使用配置项](#)

### 说明

在 Pod 里使用配置项时，需两者处于同一命名空间中。

更多关于配置项的信息，参见 [Configure a Pod to Use a ConfigMap](#)。

### 用配置项定义 Pod 环境变量

您可以使用配置项定义 Pod 中环境变量，使用 `valueFrom` 引用配置项数据。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在集群列表页，单击集群名称进入集群详情页。
3. 在左侧导航栏单击 **工作负载 > 容器组（Pods）**。
4. 单击 **从 YAML 创建**，在输入框中输入以下内容，单击 **创建**。下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-1
  namespace: yournamespace #改成当前的命名空间名称
spec:
  containers:
    - name: test-container
      image: busybox
      command: ["/bin/sh", "-c", "env"]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom: #使用valueFrom来指定env引用配置项的value值
            configMapKeyRef:
              name: special-config #引用的配置文件名称
              key: SPECIAL_LEVEL #引用的配置项key
  restartPolicy: Never
```

### 通过配置项设置命令行参数

您可以使用配置项设置容器中的命令或者参数值，使用环境变量替换语法 `$(VAR_NAME)` 来进行。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在集群列表页，单击集群名称进入集群详情页。
3. 在左侧导航栏单击 **工作负载 > 容器组（Pods）**。

4. 单击 **从 YAML 创建**，在输入框中输入以下内容，单击 **创建**。下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-3
  namespace: yournamespace # 改成当前的命名空间名称
spec:
  containers:
  - name: test-container
    image: busybox
    command: ["/bin/sh", "-c", "echo ${SPECIAL_LEVEL_KEY} ${SPECIAL_TYPE_KEY}"]
    env:
  - name: SPECIAL_LEVEL_KEY
    valueFrom:
      configMapKeyRef:
        name: special-config
        key: SPECIAL_LEVEL
  - name: SPECIAL_TYPE_KEY
    valueFrom:
      configMapKeyRef:
        name: special-config
        key: SPECIAL_TYPE
  restartPolicy: Never
```

运行该 pod 后，会输出如下结果。

```
very charm
```

## 在数据卷中使用配置项

您也可以在数据卷里面使用配置项，在 volumes 下指定配置项名称，会将 key/values 的数据存储到 mountPath 路径下（本例中是 `/etc/config`）。最终生成以 key 为文件名，values 为文件内容的配置文件。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在集群列表页，单击集群名称进入集群详情页。
3. 在左侧导航栏单击 **工作负载 > 容器组 (Pods)**。
4. 单击 **从 YAML 创建**，在输入框中输入以下内容，单击 **创建**。下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-4
  namespace: yournamespace #改成当前的命名空间名称
spec:
  containers:
  - name: test-container
    image: busybox
    command: ["/bin/sh", "-c", "ls /etc/config/"] #列出该目录下的文件名
    volumeMounts:
    - name: config-volume
      mountPath: /etc/config
  volumes:
  - name: config-volume
    configMap:
      name: special-config
  restartPolicy: Never
```

运行 pod 后，会输出配置项的 key。

```
SPECIAL_TYPE    SPECIAL_LEVEL
```

## 8.3. 管理保密字典

保密字典（Secret）是一种用于存储工作负载所需要认证信息、密钥的敏感信息等的资源类型。

### 创建保密字典

您可以创建全局的保密字典，也可以为单个集群创建保密字典。

### 创建联邦保密字典

1. 登录 LHC 控制台，在左侧导航栏单击 **配置管理 > 保密字典**。
2. 在保密字典列表页，单击 **创建**。
3. 在 **创建保密字典** 页，填写以下信息：
  - **命名空间**：选择所属命名空间。
  - **名称**：保密字典名称。
  - **保密字典类型**：
    - **Opaque**：一般密钥类型。输入变量名、值。值必须使用 Base64 编码。
    - **ImagePullSecret**：存放拉取私有仓库镜像所需的认证信息。输入容器注册地址、用户名和登录密码及（可选）邮箱。
  - **自定义配置**：开启后可以自定义部署单元级别的配置项，默认关闭。
4. 配置完成后，单击 **创建**。

### 创建单集群保密字典

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **配置** 页签。

3. 在左侧导航栏，选择 **保密字典 (Secrets)**。
4. (可选) 选择所属命名空间，如果不选择，默认使用 **default**。
5. 单击 **创建**，在 **创建保密字典** 页面，填写以下信息：
  - **命名空间**：所属的命名空间（已在上一步中选定，无需填写，默认为 **default**）。
  - **名称**：保密字典名称。
  - **保密字典类型**：
    - **Opaque**：一般密钥类型。输入变量名、值。值必须使用 Base64 编码。
    - **ImagePullSecret**：存放拉取私有仓库镜像所需的认证信息。输入容器注册地址、用户名及登录密码。
6. 配置完成后，单击 **提交**。

## 修改保密字典

对于 Opaque 类型的保密字典，您可以继续修改保密字典数据。

1. 登录 LHC 控制台，在左侧导航栏单击 **配置管理 > 保密字典**。
2. 在保密字典列表页，单击 **名称** 进入保密字典详情页。
3. 您可以进行以下修改：
  - 单击 **编辑**，修改已有的配置项数据，单击 **提交**。
  - 单击 **添加保密字典数据**，增加变量名和值，单击 **提交**。

## 查看保密字典

1. 登录 LHC 控制台，在左侧导航栏单击 **配置管理 > 保密字典**。
2. 在保密字典列表页，单击 **名称** 进入保密字典详情页。

### ② 说明

对于 Opaque 类型的保密字典，您可以继续添加保密字典数据。

## 删除保密字典

### 🔔 注意

删除保密字典可能会影响引用该保密字典的应用容器。建议您在确保没有引用该保密字典的应用容器后，再进行删除操作。

1. 在保密字典列表页，单击 **删除**。
2. 在弹出框中，单击 **确认**。

# 8.4. 使用保密字典

您可以在工作负载环境变量、命令行参数和数据卷三个场景使用保密字典 (Secrets)，也可以在 [创建应用服务](#) 的 [配置 Pod 模板](#) 步骤中使用保密字典拉取镜像。

## 前提条件

已创建保密字典。

#### ? 说明

在 Pod 中使用保密字典时，Pod 和保密字典需处于同一命名空间中。

本文以如下保密字典为例，介绍保密字典的用法。

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
namespace: yournamespace #改成当前的命名空间名称
type: Opaque
data:
  username: my-username #用户名
  password: ***** #需要用Base64编码
```

## 用保密字典配置 Pod 的数据卷

保密字典可以在 Pod 中作为文件使用。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在集群列表页，单击目标集群名称进入 **集群详情** 页。
3. 在左侧导航栏，单击 **工作负载 > 容器组 (Pods)**。
4. 单击 **从 YAML 创建**，在输入框中输入以下内容，单击 **提交**。

如以下 Pod 示例所示，mysecret 密钥的 username 和 password 以文件方式保存在 `/etc/foo` 目录下。

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
namespace: yournamespace #改成当前的命名空间名称
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
    volumes:
    - name: foo
      secret:
        secretName: mysecret
```

## 用保密字典设置 Pod 的环境变量

保密字典可以在 Pod 中设置为环境变量。

1. 登录控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 在集群列表页，单击目标集群名称进入集群详情页。
3. 在左侧导航栏，单击 **工作负载 > 容器组（Pods）**。
4. 单击 **从 YAML 创建**，在输入框中输入以下内容，单击 **提交**。

如以下 Pod 示例所示，mysecret 密钥的 username 和 password 配置为 Pod 的环境变量。

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
namespace: yournamespace #改成当前的命名空间名称
spec:
  containers:
  - name: mycontainer
    image: redis
    env:
  - name: SECRET_USERNAME
    valueFrom:
      secretKeyRef:
        name: mysecret
        key: username
  - name: SECRET_PASSWORD
    valueFrom:
      secretKeyRef:
        name: mysecret
        key: password
  restartPolicy: Never
```

## 9. 工作负载

### 9.1. 部署（Deployments）

部署（Deployments）即运行中始终不保存任何数据或状态的工作负载，例如 Nginx。您可以在 LHC 控制台上轻松管理部署（Deployments），如执行创建、更新、删除等操作。

#### ? 说明

Deployments 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官网文档](#)。

#### 创建 Deployments

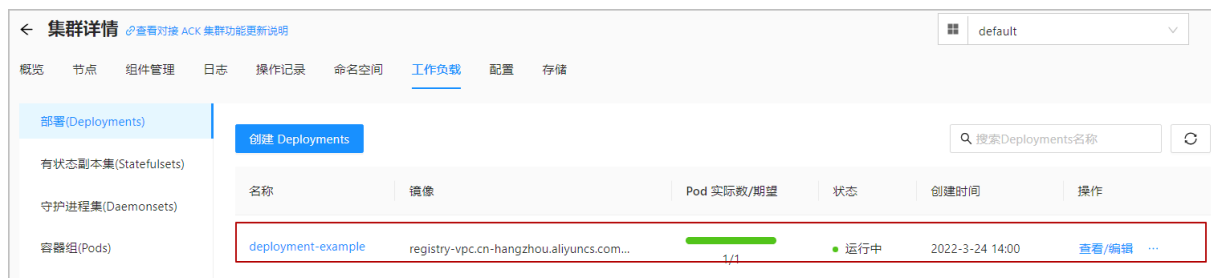
1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **部署（Deployments）**，然后单击 **创建 Deployments**。
4. 在弹出的输入框中输入 YAML 内容，单击 **提交**。

YAML 样例如下：

```
kind:Deployment
metadata:
  name: deployment-example
  labels:
    k8s-app: deployment-example
namespace:default
spec:
  replicas:1
  selector:
    matchLabels:
      k8s-app: deployment-example
  template:
    metadata:
      labels:
        k8s-app: deployment-example
    spec:
      containers:
      - name: nginx
        image: reg-cnsh.cloud.alipay.com/library/nginx:latest # 改成当前 region 的镜像地址
```

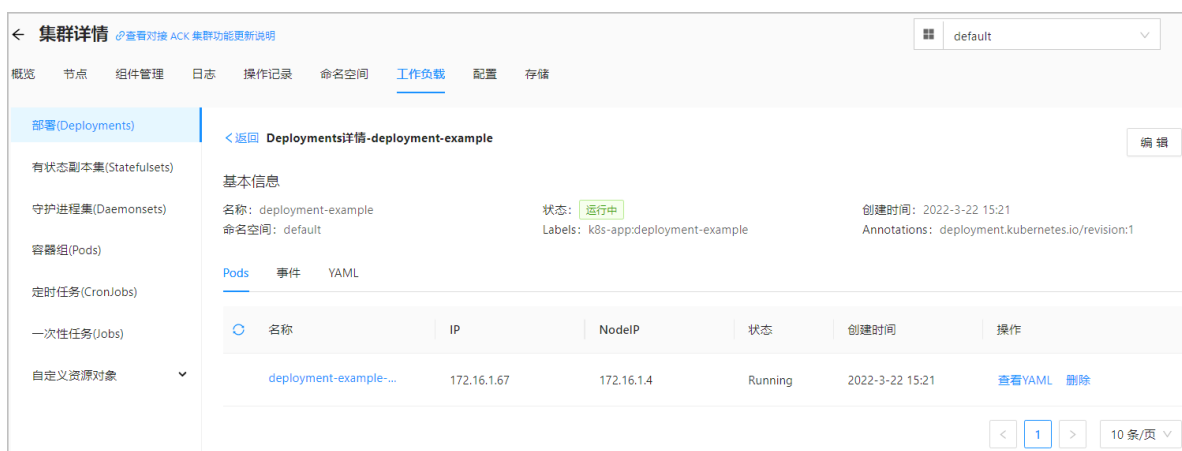
#### 执行结果

创建成功后，名为 **deployment-example** 的 Deployments 将出现在列表页面中。



## 相关操作

- 如需查看 Deployments 详情，请在 Deployments 列表页单击目标 Deployments 名称，进入 Deployments 详情页。在该页面上，可以查看 Deployments 基本信息、Pod 列表信息、事件信息及 YAML 配置信息。



- 如需搜索 Deployments，请在 Deployments 列表页右上角的输入栏，输入 Deployments 名称关键字即可进行搜索。
- 如需更新 Deployments，请在 Deployments 列表页单击操作栏中的 查看/编辑，并在编辑 Deployments 对话框中编辑 YAML 信息，然后单击 提交。
- 如需删除 Deployments，请在 Deployments 列表页单击操作栏中的更多图标 > 删除，然后在弹出的提示对话框中单击 确定。

## 9.2. 有状态副本集 (Statefulsets)

有状态副本集 (Statefulsets) 即运行过程中保存数据或状态的工作负载，例如 MySQL。您可以在 LHC 控制台上轻松管理有状态副本集 (Statefulsets)，如执行创建、更新、删除等操作。

### 说明

Statefulsets 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官网文档](#)。

### 创建 Statefulsets

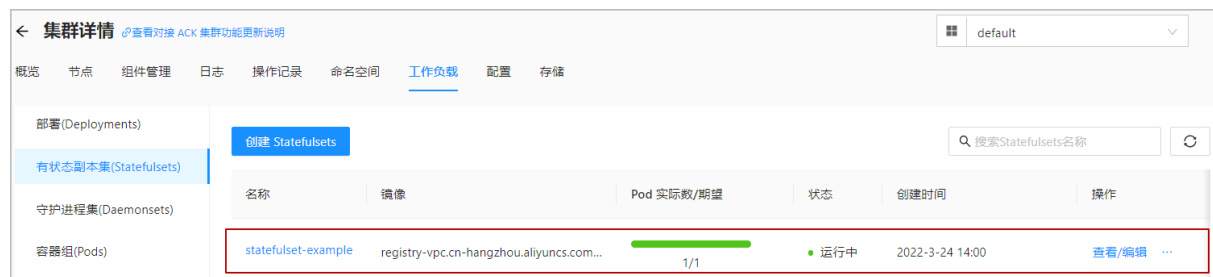
- 登录 LHC 控制台，在左侧导航栏单击 集群管理 > 集群，进入集群列表页。
- 单击目标集群名称，进入 集群详情 页，单击 工作负载 页签。
- 在左侧菜单选项，选择 有状态副本集 (Statefulsets)，然后单击 创建 Statefulsets。
- 在弹出的输入框中输入 YAML 内容，单击 提交。

YAML 样例如下：

```
kind: StatefulSet
metadata:
  name: statefulset-example
  labels:
    k8s-app: statefulset-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: statefulset-example
  template:
    metadata:
      labels:
        k8s-app: statefulset-example
    spec:
      containers:
        - name: nginx
          image: 'registry-vpc.cn-hangzhou.aliyuncs.com/sofa-samples/nginx:latest'
```

## 执行结果

创建成功后，名为 `statefulset-example` 的 Statefulsets 将出现在列表页面中。



## 相关操作

- 如需查看 Statefulsets 详情，请在 Statefulsets 列表页单击目标 Statefulsets 名称，进入 Statefulsets 详情页。在该页面上，可以查看 Statefulsets 基本信息、Pod 列表信息、事件信息及 YAML 配置信息。
- 如需搜索 Statefulsets，请在 Statefulsets 列表页右上角的输入栏，输入 Statefulsets 名称关键字即可进行搜索。
- 如需更新 Statefulsets，请在 Statefulsets 列表页单击操作栏中的 **查看/编辑**，并在编辑 Statefulsets 对话框中编辑 YAML 信息，然后单击 **提交**。
- 如需删除 Statefulsets，请在 Statefulsets 列表页单击操作栏中的更多图标 > **删除**，然后在弹出的提示对话框中单击 **确定**。

## 9.3. 守护进程集（Daemonsets）

守护进程集（Daemonsets）确保全部（或者部分）节点都运行一个 Pod 实例，支持实例动态添加到新节点。您可以在 LHC 控制台上轻松管理守护进程集（Daemonsets），如执行创建、更新、删除等操作。

### ? 说明

Daemonsets 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官方文档](#)。

## 创建 Daemonsets

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **守护进程集 (Daemonsets)**，然后单击 **创建 Daemonsets**。
4. 在弹出的输入框中输入 YAML 内容，单击 **提交**。

YAML 样例如下：

```
kind:DaemonSet
metadata:
  name: daemonset-example
  labels:
    k8s-app: daemonset-example
namespace:default
spec:
  selector:
    matchLabels:
      k8s-app: daemonset-example
  template:
    metadata:
      labels:
        k8s-app: daemonset-example
    spec:
      containers:
      - name: nginx
        image: reg-cnsh.cloud.alipay.com/library/nginx:latest
```

## 执行结果

创建成功后，名为 **daemonset-example** 的 Daemonsets 将出现在列表页面中。

← 集群详情

[查看对接 ACK 集群功能更新说明](#)

概览

节点

组件管理

日志

操作记录

命名空间

工作负载

配置

存储

部署 (Deployments)

有状态副本集 (Statefulsets)

守护进程集 (Daemonsets)

容器组 (Pods)

创建 Daemonsets

名称

镜像

Pod 实际数/期望

状态

创建时间

操作

daemonset-example

registry-vpc.cn-hangzhou.aliyuncs.com...

4/4

● 运行中

2022-3-24 14:01

[查看/编辑](#) ...

搜索 Daemonsets 名称

## 相关操作

- 如需查看 Daemonsets 详情，请在 Daemonsets 列表页单击目标 Daemonsets 名称，进入 Daemonsets 详情页。在该页面上，可以查看 Daemonsets 基本信息、Pod 列表信息、事件信息及 YAML 配置信息。

名称	IP	NodeIP	状态	创建时间	操作
daemonset-example-8...	172.16.1.38	172.16.1.7	Running	2022-3-18 17:04	查看YAML 删除
daemonset-example-...	172.16.1.56	172.16.1.4	Running	2022-3-18 17:04	查看YAML 删除
daemonset-example-...	172.16.1.45	172.16.1.3	Running	2022-3-18 17:04	查看YAML 删除
daemonset-example-...	172.16.1.66	172.16.1.5	Running	2022-3-18 17:04	查看YAML 删除

- 如需搜索 Daemonsets，请在 Daemonsets 列表页右上角的输入栏，输入 Daemonsets 名称关键字即可进行搜索。
- 如需更新 Daemonsets，请在 Daemonsets 列表页单击操作栏中的 查看 / 编辑，并在编辑 Daemonsets 对话框中编辑 YAML 信息，然后单击 提交。
- 如需删除 Daemonsets，请在 Daemonsets 列表页单击操作栏中的更多图标 > 删除，然后在弹出的提示对话框中单击 确定。

## 9.4. 容器组（Pods）

容器组（Pods）是 Kubernetes 中最小可部署单元，代表 Kubernetes 中一个独立的应用程序运行实例，该实例可能由单个容器或者多个紧耦合在一起的容器组成。

### 说明

Pods 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官方文档](#)。

### 创建 Pods

- 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
- 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
- 在左侧菜单选项，选择 **容器组（Pods）**，然后单击 **创建 Pods**。
- 在弹出的输入框中输入 YAML 内容，单击 **提交**。

YAML 样例如下：

```
kind: Pod
metadata:
  name: pod-example
  labels:
    k8s-app: pod-example
  namespace: default
spec:
  containers:
  - name: nginx
    image: 'registry-vpc.cn-hangzhou.aliyuncs.com/sofa-samples/nginx:latest'
```

## 执行结果

创建成功后，名为 **pod-example** 的 Pods 将出现在列表页面中。

<div>集群详情 <small>查看对接 ACK 集群功能更新说明</small></div> <div>default</div>						
概览 节点 组件管理 日志 操作记录 命名空间 工作负载 配置 存储						
部署(Deployments) 有状态副本集(Statefulsets) 守护进程集(Daemonsets) 容器组(Pods) 定时任务(CronJobs) 一次性任务(Jobs) 自定义资源对象						
创建 Pods <input type="text" value="搜索Pods名称"/>						
名称	IP	节点IP	镜像	状态	创建时间	
hello-1648101720-nq...	172.16.0.1	172.16.0.1	registry-vpc.cn-hangzhou.aliyuncs.com...	Succeeded	2022-3-24 14:02	
pod-example	172.16.1.64	172.16.1.4	registry-vpc.cn-hangzhou.aliyuncs.com...	Running	2022-3-24 14:01	
daemonset-example-8...	172.16.0.1	172.16.0.1	registry-vpc.cn-hangzhou.aliyuncs.com...	Running	2022-3-24 14:01	
daemonset-example-...	172.16.0.1	172.16.0.1	registry-vpc.cn-hangzhou.aliyuncs.com...	Running	2022-3-24 14:01	

## 相关操作

- 如需查看 Pods 详情，请在 Pods 列表页单击目标 Pods 名称，进入 Pods 详情页。在该页面上，可以查看 Pods 基本信息、容器及事件信息。

集群详情

查看对接 ACK 集群功能更新说明

default

概览

节点

组件管理

日志

操作记录

命名空间

工作负载

配置

存储

部署 (Deployments)

有状态副本集 (Statefulsets)

守护进程集 (Daemonsets)

容器组 (Pods)

定时任务 (CronJobs)

一次性任务 (Jobs)

自定义资源对象

NetworkPolicy

SafeDeployment

< 返回

容器组详情-pod-example

编辑

基本信息

名称: pod-exampleIP: 172.16.1.64节点IP: 172.16.1.4

创建时间: 2022-3-18 17:03

容器

事件

名称	镜像	状态	重启次数	最后启动时间	操作
nginx	registry-vpc.cn-hangzhou.aliyuncs.com...	运行中	0	2022-3-18 17:03	前往控制台

<

1

>

10 条/页

在 Pods 详情页，您还可以根据需要执行如下操作：

- 查看容器详情：单击想要查看详情的 Pod 名称，即可进入容器详情页，如下图所示。



- 前往容器服务控制台：在 Pods 详情页单击目标 Pod 操作 列的 前往控制台，或者单击容器详情页上的 控制台，将跳转至容器服务控制台。可按需输入命令来管理容器。
- 如需搜索 Pods，请在 Pods 列表页右上角的输入栏，输入 Pods 名称关键字即可进行搜索。
- 如需更新 Pods，请在 Pods 列表页单击操作栏中的 查看/编辑，并在编辑 Pods 对话框中编辑 YAML 信息，然后单击 提交。
- 如需删除 Pods，请在 Pods 列表页单击操作栏中的更多图标 > 删除，然后在弹出的提示对话框中单击 确定。

## 9.5. 定时任务（CronJobs）

定时任务（CronJobs）用于创建周期性以及重复性的任务，例如数据库备份或者发送邮件。Job 负责批量处理短暂的一次性任务，即仅执行一次的任务，而 CronJob 则是基于 Job 增加了时间调度，即在指定时间周期内运行指定 Job。本文介绍如何通过控制台创建 CronJobs 及其他常用操作。

### 说明

CronJob 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官方文档](#)。

### 创建 CronJobs

- 登录 LHC 控制台，在左侧导航栏单击 集群管理 > 集群，进入集群列表页。
- 单击目标集群名称，进入 集群详情 页，单击 工作负载 页签。
- 在左侧菜单选项，选择 定时任务（CronJobs），然后单击 创建 CronJobs。
- 在弹出的输入框中输入 YAML 内容，单击 提交。

YAML 样例如下：

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - command:
                - /bin/sh
                - '-c'
                - date; echo Hello from the Kubernetes cluster
              image: 'registry-vpc.cn-hangzhou.aliyuncs.com/sofa-samples/nginx:latest'
              imagePullPolicy: IfNotPresent
              name: hello
            restartPolicy: Never
          schedule: '* * */1 * * *'
```

参数说明可参考下表。

参数	说明
kind	标识 CronJob 资源类型。
metadata	CronJob 的基本信息，比如 CronJob 的名称。
spec.schedule	<p>CronJob 执行的 Cron 的策略。即指定任务运行周期，格式同 <a href="#">Cron schedule</a>。</p> <div><pre># 格式说明 # —分钟 (0 - 59) #   —小时 (0 - 23) #     —日 (1 - 31) #       —月 (1 - 12) #         —星期 (0 - 6) #           # * * * * *</pre></div>
spec.jobTemplate	Cron 执行的 Job 模板。即指定需要运行的任务，格式同 <a href="#">Job patterns</a> 。

## 执行结果

创建成功后，名为 **hello** 的 CronJobs 将出现在列表页面中。CronJob 执行时会在指定的时间创建出 Job，然后由 Job 创建出 Pod。

**定时任务(CronJobs)**

名称	标签	镜像	创建时间	最近调度	挂起	计划	操作
hello	-	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 11:51	-	false	* * / 1 * * *	查看/编辑 ...

**CronJobs hello 创建成功**

**创建 Jobs**

名称	标签	状态	镜像	创建时间	完成时间	操作
hello-1648094700		成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 12:05	2022-3-24 12:05	查看/编辑 ...
hello-1648094640		成功	registry-vpc.i			
hello-1648094580		成功	registry-vpc.i			

**CronJob 执行时会在指定时间内创建 Job，然后由 Job 创建 Pod**

**容器组(Pods)**

名称	IP	状态	镜像	创建时间	完成时间	操作
hello-1648094700-afw...	172.16.0.1	成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 12:05		
hello-1648094640-brtbf	172.16.0.1	成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 12:04		
hello-1648094580-qnc...	172.16.0.1	成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 12:03		

## 相关操作

- 如需查看 CronJobs 详情，请在 CronJobs 列表页单击目标 CronJobs 名称，进入 CronJobs 详情页。在该页面上，可以查看 CronJobs 基本信息、任务列表信息及事件信息。
- 如需搜索 CronJobs，请在 CronJobs 列表页右上角的输入栏，输入 CronJobs 名称关键字即可进行搜索。
- 如需更新 CronJobs，请在 CronJobs 列表页单击操作栏中的 **查看/编辑**，并在编辑 CronJobs 对话框中编辑 YAML 信息，然后单击 **提交**。
- 如需删除 CronJobs，请在 CronJobs 列表页单击操作栏中的更多图标 > **删除**，然后在弹出的提示对话框中单击 **确定**。

### 说明

对 CronJobs 执行删除时，不会删除 CronJobs 创建的 Jobs，如需删除对应的 Jobs，请前往 Jobs 列表页手动删除。

## 9.6. 一次性任务 (Jobs)

一次性任务 (Jobs) 负责批量处理短暂的一次性任务，即仅执行一次的任务，保证批处理任务的一个或多个 Pod 成功结束。本文介绍如何通过控制台创建 Jobs 及其他常用操作。

### 说明

Jobs 是 kubernetes 原生概念。更多详情请参见 [Kubernetes 官方文档](#)。

## 创建 Jobs

- 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
- 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。

3. 在左侧菜单选项，选择 **部署 (Jobs)**，然后单击 **创建 Jobs**。
4. 在弹出的输入框中输入 YAML 内容，单击 **提交**。

YAML 样例如下：

```
apiVersion: batch/v1
kind: Job
metadata:
  name: hello
spec:
  template:
    spec:
      containers:
        - name: hello
          image: 'registry-vpc.cn-hangzhou.aliyuncs.com/sofa-samples/nginx:latest'
          command:
            - /bin/sh
            - '-c'
            - date; echo Hello from the Kubernetes cluster
      restartPolicy: Never
      backoffLimit: 1
```

参数说明可参考下表。

参数	说明
kind	标识 Job 资源类型。
metadata	Job 的基本信息，比如 Job 的名称。
spec.Template	Job 管理的 Pod 的详细模板配置。
restartPolicy	任务的重启策略。支持 Never、OnFailure。 <ul style="list-style-type: none"><li>◦ Never：表示在任务需要重新运行时执行。</li><li>◦ OnFailure：表示在任务失败时再执行。</li></ul>
backoffLimit	任务的重试次数。用于控制任务重试次数。

## 执行结果

创建成功后，名为 **hello** 的 Jobs 将出现在列表页面中。

<div>集群详情 <a href="#">查看对接 ACK 集群功能更新说明</a></div> <div>默认</div>							
<div>概览 节点 组件管理 日志 操作记录 命名空间 工作负载 配置 存储</div>							
<div>部署 (Deployments) 创建 Jobs 搜索 Job 名称</div>							
<div>有状态副本集 (Statefulsets)</div>							
<div>守护进程集 (Daemonsets)</div>							
<div>容器组 (Pods)</div>							
<div>定时任务 (CronJobs)</div>							
<div>一次性任务 (Jobs)</div>							
<div>自定义资源对象</div>							
名称	标签	状态	镜像	创建时间	完成时间	操作	
hello		运行中	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 13:54	2022-3-24 13:54	查看/编辑 ...	
hello-1648101240		成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 13:54	2022-3-24 13:54	查看/编辑 ...	
hello-1648101180		成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 13:53	2022-3-24 13:53	查看/编辑 ...	
hello-1648101120		成功	registry-vpc.cn-hangzhou.aliyuncs.com...	2022-3-24 13:52	2022-3-24 13:52	查看/编辑 ...	

## 相关操作

- 如需查看 Jobs 详情，请在 Jobs 列表页单击目标 Jobs 名称，进入 Jobs 详情页。在该页面上，可以查看 Jobs 基本信息、Pod 列表信息、事件信息及 YAML 配置信息。

<div>返回 Jobs 详情-hello 编辑</div>					
<div>基本信息</div>					
名称: hello	状态: 成功	创建时间: 2022-3-24 13:54			
命名空间: default	Labels: controller-uid:ca3e0deb-f283-426c-8...	Annotations:			
<div>Pods 事件 YAML</div>					
名称	IP	NodeIP	状态	创建时间	操作
hello-mkcgq	172.17.0.55	172.17.0.18	Succeeded	2022-3-24 13:54	查看 YAML 删除

- 如需搜索 Jobs，请在 Jobs 列表页右上角的输入栏，输入 Jobs 名称关键字即可进行搜索。
- 如需更新 Jobs，请在 Jobs 列表页单击操作栏中的 查看/编辑，并在编辑 Jobs 对话框中编辑 YAML 信息，然后单击 提交。
- 如需删除 Jobs，请在 Jobs 列表页单击操作栏中的更多图标 > 删除，然后在弹出的提示对话框中单击 确定。

## 9.7. 自定义资源对象

### 9.7.1. CafeDeployment

CafeDeployment 是 CAFE 产品设计的高阶工作负载（workload），为 InPlaceSet（原地升级）工作负载提供高阶发布能力，提供 Pod 单元化管理、Beta 和分组发布功能以及版本控制能力，实现 Pod 原地升级。本文介绍如何查询 CafeDeployment。

#### 使用限制

当前该资源类型仅支持查询，无法修改、增加、删除。

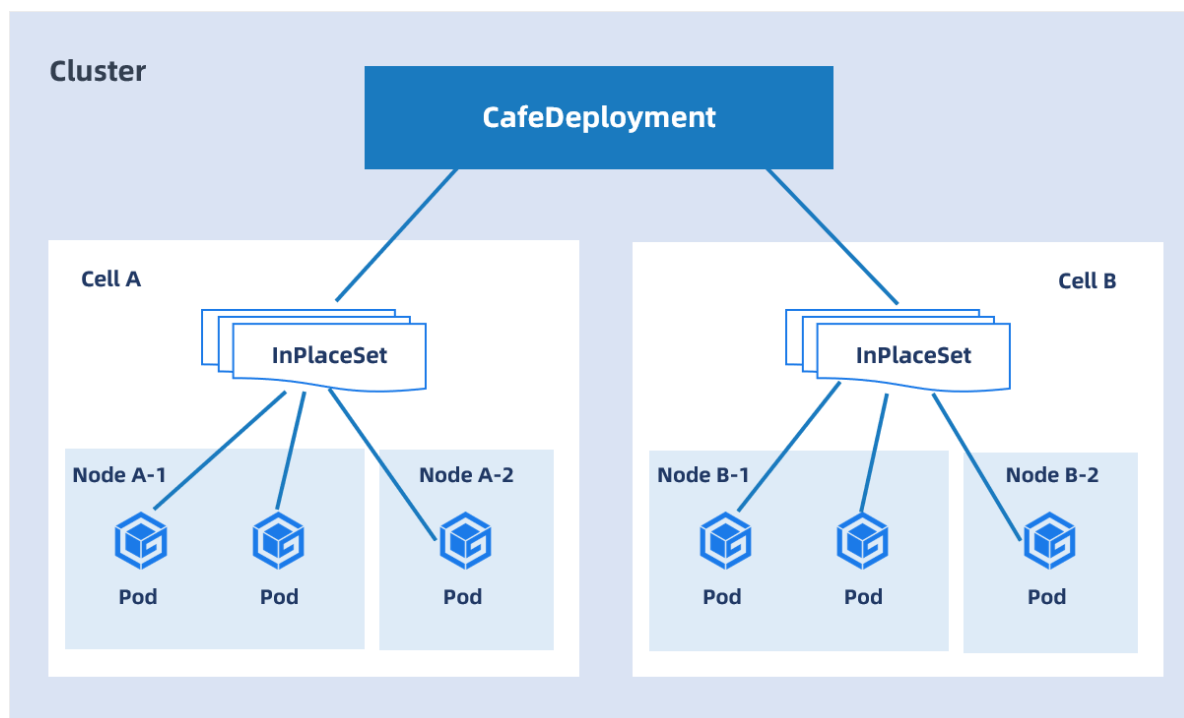
#### 背景信息

- CafeDeployment 与应用服务

在 LHC 中，应用服务是一个中枢的 fedcafedeployment，它会在每一个要下发的部署单元内创建一个 Cafedeployment，而 Cafedeployment 下一层挂载 Inplaceset，再下层是实体 Pod。应用服务创建并发布后，会创建 CafeDeployment。

### ● CafeDeployment 与 InPlaceSet

CafeDeployment 主要提供跨部署单元的管理功能，其下管理多个 InPlaceSet。每个 InPlaceSet 对应一个部署单元。部署单元是逻辑概念，它通过 Node 上的 label 来划分集群中的节点，而 InPlaceSet 则通过 NodeAffinity 能力，将其下的 Pod 部署到同一个部署单元的机器上。由此实现 CafeDeployment 跨部署单元的管理。



## 应用场景

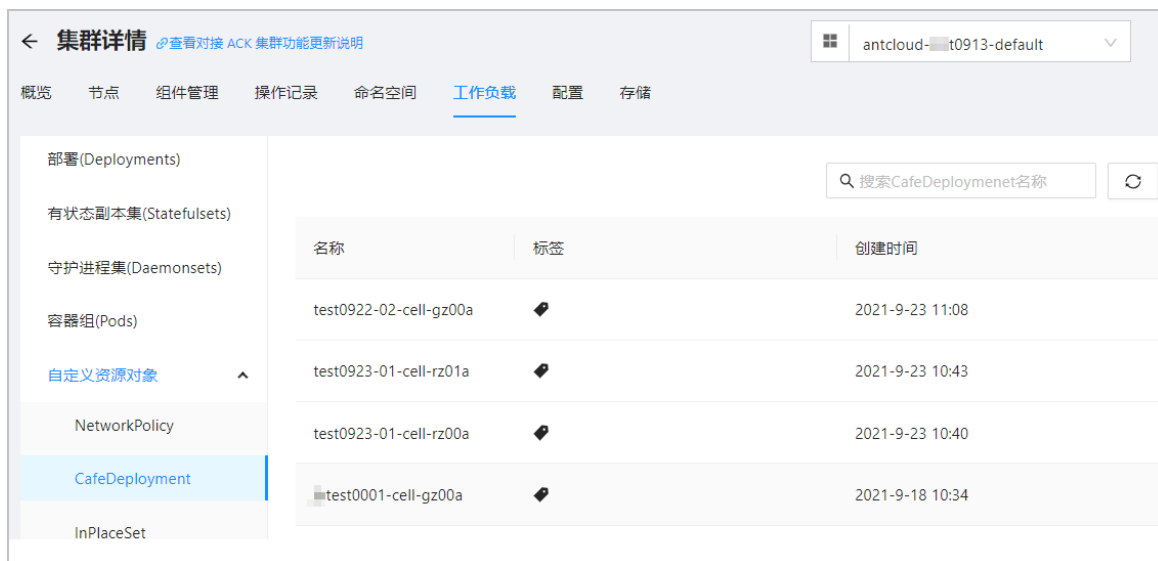
CafeDeployment 的一些典型使用场景如下：

- 单元化：应用实例需要分割到不同的节点进行部署。
- 精细的发布管理：在 Pod 配置的更新过程中，提供了 Beta 发布、分组发布、发布暂停、发布取消等功能。
- 版本控制：在每次 Pod 配置更新后自动保存版本，方便版本回退。
- 原地升级：在保持 Pod IP 不变的前提下实现 Pod 其他配置的变更。

## 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **自定义资源对象 > CafeDeployment**。
4. 在 **CafeDeployment** 列表页，查询 CafeDeployment 数据。

该页面列出了支持的全部 CafeDeployment、标签以及创建时间。您还可以通过右侧搜索框，使用关键词搜索 CafeDeployment。



## 9.7.2. InPlaceSet

InPlaceSet 是 CAFE 产品设计的基础工作负载（workload），负责管理 Pod 的副本和版本，并管理 Pod 生命周期，进而提供高阶的能力。本文介绍如何查询 InPlaceSet。

### 背景信息

InPlaceSet 是一种新的工作负载抽象自定义资源，主要用于 Pod 原地升级，提升整体发布效率。LHC 作为蚂蚁集团全新打造的基于 Kubernetes 的容器产品，发布部署能力的建设是其非常重要的一环，通过操作单一 Pod 对象，目前已具备基础的发布部署能力，所采用的模式也是原地升级。

#### 说明

InPlaceSet、应用服务、CafeDeployment 之间的关系可参考 [CafeDeployment 与应用服务](#)、[CafeDeployment 与 InPlaceSet](#)。

### 使用限制

当前该资源类型仅支持查询，无法修改、增加、删除。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**。
2. 单击目标集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **自定义资源对象 > InPlaceSet**。
4. 在 **InPlaceSet** 列表页，查询 InPlaceSet 数据。

该页面列出了支持的全部 InPlaceSet、标签以及创建时间。您还可以通过右侧搜索框，使用关键词搜索 InPlaceSet。

集群详情

查看对接 ACK 集群功能更新说明

antcloud-0913-ll-test

概览

节点

组件管理

操作记录

命名空间

工作负载

配置

存储

部署(Deployments)
有状态副本集(Statefulsets)
守护进程集(Daemonsets)
容器组(Pods)
自定义资源对象
NetworkPolicy
CafeDeployment
InPlaceSet

搜索InPlaceSet名称

名称	标签	创建时间
test0923-cell-rz00a-2j8		2021-9-23 11:27
test00-cell-rz00a-8t6		2021-9-22 14:09
test0918-01-cell-rz00a-8bz		2021-9-18 15:57

<

1

>

# 10. 存储

## 10.1. 存储概述

本文介绍 LHC 存储相关的基础知识，方便您在使用存储能力时，了解相应模块的基础知识和基本关系。

### 说明

在基于物理机底座的部署环境下，存储功能依赖 Yoda 本地存储。

### 基本概念

#### Yoda

由蚂蚁专有云团队与阿里云团队合力研发的存储系统，由多个组件构成的统一本地磁盘调度和管理系统，目标是解决当前 Kubernetes 社区对于本地磁盘和 Local PV 管理和自动化能力缺失的问题。

#### 存储类型（Storage Class）

Kubernetes 的原生概念，通常由系统管理员定义，用于指定所支持的存储类别，不同的类别会有不同的存储 SLA、备份策略等差异性。

#### 存储卷（Persistent Volume）

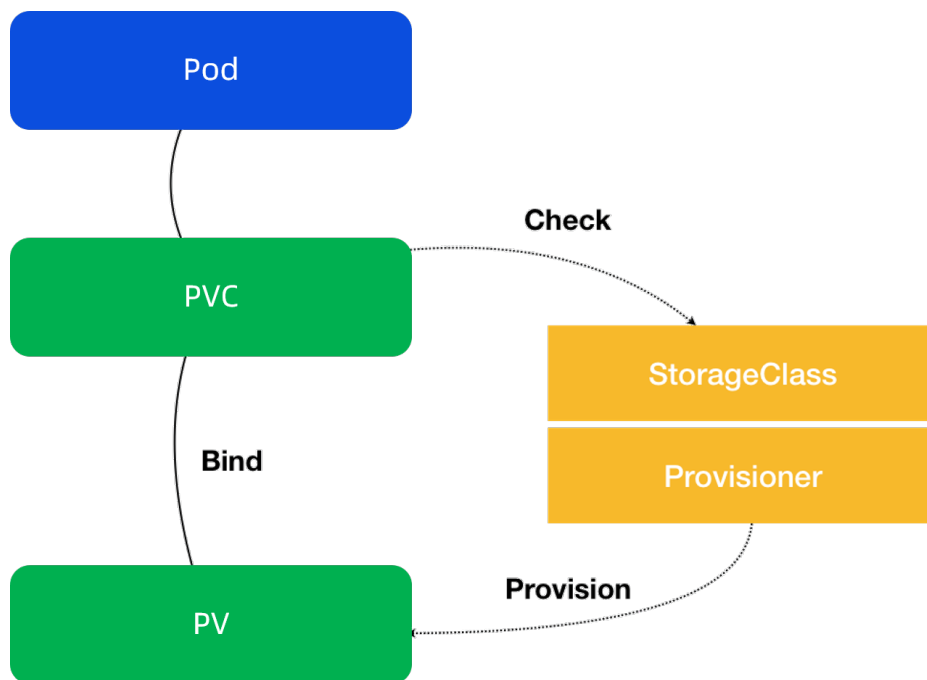
存储卷（Persistent Volume，简称 PV），Kubernetes 的原生概念，PV 在 Kubernetes 中代表一个具体存储类型的卷，其对象中定义了具体存储类型和卷参数。即目标存储服务所有相关的信息都保存在 PV 中，Kubernetes 引用 PV 中的存储信息执行挂载操作。

#### 存储卷声明（Persistent Volume Claim）

存储卷声明（Persistent Volume Claim，简称 PVC），Kubernetes 的原生概念，一个存储卷声明绑定一个存储卷。是 Kubernetes 中一种抽象的存储卷类型，代表了某个具体类型存储的数据卷表达。

### 基本关系

StorageClass、PV、PVC 之间的关系可参考下图：



- PVC：描述 Pod 想要使用的持久化存储的属性，比如存储的大小、读写权限等。
- PV：描述具体的 Volume 属性，比如 Volume 类型、挂载目录等。
- StorageClass：充当 PV 的模板，且只有同属于一个 StorageClass 的 PV 和 PVC 才可以绑定在一起，具体可参考 [PVC 与 PV 的绑定](#)。

## 相关文档

- 关于 Kubernetes 存储的更多信息，请参见 [Storage](#)。
- 具体存储操作可参考如下文章：
  - [管理存储类型 \(StorageClass\)](#)
  - [管理存储卷 \(PersistentVolume\)](#)
  - [管理存储卷声明 \(PVC\)](#)

## 10.2. 管理存储类型 (StorageClass)

您可以在 LHC 控制台上轻松管理存储类型 (StorageClass)，如创建、删除等操作。

目前，LHC 支持以下两种类型的存储类型：

- [NAS 类型](#)
- [云盘 \(cloud\\_disk\) 类型](#)

### 🔍 说明

目前 LHC 未对接阿里云的 NAS 及云盘存储，若要使用存储功能，需要安装第三方存储插件。

## NAS 类型

 注意

NAS 类型的 Storage Class 必须和 LHC 集群节点在相同地域。

LHC 定义的 NAS 类型的 Storage Class 如下：

```
apiVersion: storage.k8s.io/v1

kind:StorageClass
metadata:
  name: csi-nas-capacity
provisioner: antcloud-csi
parameters:
  storage_type:"cloud_nas"# Allowed enum: cloud_disk, cloud_nas
# filesystem parameters
  nasType:"Capacity"# Allowed enum: Capacity, Performance, Extreme *
  protocolType:"NFS"# Allowed enum: NFS, SMB *
  serviceId:"1c7aff75-8b4f-4d2c-8dac-74af0b830d86-aliyun-nas-service-id"
  planId:"44dd37f1-77f9-42c5-91a5-743c007ba484-aliyun-nas-plan-id"
reclaimPolicy:Retain# Allowed enum: Delete, Retain
volumeBindingMode:Immediate# Allowed enum: Immediate, WaitForFirstConsumer
```

## 参数说明

参数名	参数说明	是否必选	备注
provisioner	antcloud-csi	是	LHC动态卷控制器名称，不可更改。
storage_type	后端存储类型（块存储，文件存储）	是	目前仅支持： <ul style="list-style-type: none"><li>cloud_disk，表示阿里云盘。</li><li>cloud_nas，表示阿里云NAS。</li></ul>
nasType	NAS 类型	是	可选值： <ul style="list-style-type: none"><li>Capacity，详情参考 <a href="#">容量型 NAS</a>。</li><li>Performance，详情参考 <a href="#">性能型 NAS</a>。</li><li>Extreme，详情参考 <a href="#">极速型 NAS</a>。</li></ul>
protocolType	文件系统类型	否	目前仅支持 NFS

参数名	参数说明	是否必选	备注
readOnly	是否只读	否	可选值： <code>true</code> ， <code>false</code>
serviceId	Service broker 必选参数	是	必须为：1c7aff75-8b4f-4d2c-8dac-74af0b830d86-aliyun-nas-service-id
planId	Service broker必选参数	否	必须为：44dd37f1-77f9-42c5-91a5-743c007ba484-aliyun-nas-plan-id
reclaimPolicy	回收策略	是	<ul style="list-style-type: none"><li>• <b>Retain</b>：保留后端存储，删除 PV/PVC 不会删除对应的后端存储如云盘。详情参考 <a href="#">Kubernetes 官网文档</a>。</li><li>• <b>Delete</b>：当删除 PVC 时，自动删除 PV 和后端的存储。详情参考 <a href="#">Kubernetes 官网文档</a>。</li></ul>
volumeBindingMode	存储卷绑定策略	是	<p>控制存储卷绑定和动态供应行为</p> <ul style="list-style-type: none"><li>• <b>Immediate</b>：创建 PVC 后立即创建后端存储卷，并且立即绑定新创建的 PV 和 PVC。</li><li>• <b>WaitForFirstConsumer</b>：当 PVC 被 Pod 使用时，才触发 PV 和后端存储的创建，同时实现 PVC/PV 的绑定，启用该配置后，Storage Class 中的 Zone 和 Region 将不再生效，而是使用 Pod 调度所在节点的 zone 和 region 创建文件系统，保证文件系统能被 Pod 挂载。</li></ul>

## 云盘（cloud\_disk）类型

### 注意

云盘类型的 Storage Class 必须和 LHC 集群节点在相同地域及可用区。

LHC 定义的云盘类型的 Storage Class 如下：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-disk-efficiency
provisioner: antcloud-csi
parameters:
  failure-domain.beta.kubernetes.io/zone: <必须与当前集群中节点可用区匹配>
  failure-domain.beta.kubernetes.io/region: <必须与当前集群所在region匹配>
  fsType: "ext4"
  storage_type: "cloud_disk" # Allowed enum: cloud_disk, cloud_nas
  type: "cloud_ssd"
  readOnly: "false"
  serviceId: "8814fddf-5be7-4f8a-862c-f4a57c25a0b4-aliyun-disk-service-id"
  planId: "02b41c80-915f-43cd-a574-e9cd446f13ea-aliyun-disk-plan-id"
reclaimPolicy: Retain # Allowed enum: Delete, Retain
volumeBindingMode: Immediate # Allowed enum: Immediate, WaitForFirstConsumer
```

## 参数说明

参数名	说明	是否必填	备注
provisioner	antcloud-csi	是	动态卷控制器名称，不可更改。
failure-domain.beta.kubernetes.io/region	地域	否	该字段当 volumeBindingMod=Immediate 时为必选，会被用来创建云盘。例如：CN-SHANGHAI，代表上海地域。
failure-domain.beta.kubernetes.io/zone	可用区	否	该字段当 volumeBindingMod=Immediate 时为必选，会被用来创建云盘。例如：CN-SHANGHAI-A，代表上海可用区 A。
fsType	分区格式	否	支持 ext4、ext3，默认 ext4。
storage_type	后端存储类型	是	<ul style="list-style-type: none"><li>cloud_disk，表示阿里云云盘</li><li>cloud_nas，表示阿里云 NAS。</li></ul>

参数名	说明	是否必填	备注
type	磁盘类型	是	可选值：cloud_ssd、cloud_efficiency、cloud。详情参考 <a href="#">云盘类型</a> 。
readOnly	是否只读	否	可选值：true、false。
serviceId	Service broker 必选参数	是	必须为： 8814fddf-5be7-4f8a-862c-f4a57c25a0b4-aliyun-disk-service-id
planId	Service broker 必选参数	否	必须为： 02b41c80-915f-43cd-a574-e9cd446f13ea-aliyun-disk-plan-id
reclaimPolicy	回收策略	是	<ul style="list-style-type: none"> <li>• <b>Retain</b>：保留后端存储，删除 PV/PVC 不会删除对应的后端存储如云盘。详情参考 <a href="#">Kubernetes 官网文档</a>。</li> <li>• <b>Delete</b>：当删除 PVC 时，自动删除 PV 和后端的存储。详情参考 <a href="#">Kubernetes 官网文档</a>。</li> </ul>
volumeBindingMode	存储卷绑定策略	是	<p>控制存储卷绑定和动态供应行为。</p> <ul style="list-style-type: none"> <li>• <b>Immediate</b>：创建 PVC 后立即创建后端存储卷，并且立即绑定新创建的 PV 和 PVC</li> <li>• <b>WaitForFirstConsumer</b>：当 PVC 被 pod 使用时，才触发 PV 和后端存储的创建，同时实现 PVC 和 PV 的绑定，启用该配置后，storage class 中的 zone 和 region 将不再被使用，而是使用 pod 调度所在节点的 zone 和 region 创建云盘，保证云盘能被 Pod 挂载。</li> </ul>

## 创建存储类型

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储类型 (StorageClass)**，然后单击 **创建存储类型**。
4. 在输入框中输入 YAML 内容，单击 **提交**。
  - NAS 类型样例：见 [NAS 类型](#)。

## 更新存储类型

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储类型 (StorageClass)**，然后单击 **操作** 栏的 **查看/编辑**。
4. 在 YAML 编辑框中修改信息，单击 **提交**。

更新成功后，系统会自动根据新的 YAML 文件信息更新存储类型。

## 删除存储类型

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储类型 (StorageClass)**。
4. 单击 **操作** 栏的更多图标 > **删除**。
5. 在弹出的对话框中单击 **确定**。

# 10.3. 管理存储卷 (PersistentVolume)

您可以在 LHC 控制台上轻松管理存储卷 (PersistentVolume)，如创建、删除等操作。

## 创建存储卷

### 前提条件

[已创建存储类型 \(StorageClass\)](#)。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷 (PersistentVolume)**，然后单击 **存储卷**。
4. 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0601-sample
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: sc0601-sample
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
```

创建成功后，名为 **pv0601-sample** 的存储卷将出现在列表页面中。

<

集群详情

查看对接 ACK 集群功能更新说明

默认

>

概述

节点

组件管理

操作记录

命名空间

工作负载

配置

存储

存储类型(StorageClass)

存储卷(PersistentVolume)

存储卷声明(PersistentVolumeClaim)

创建存储卷

如何创建存储卷

搜索 PersistentVolume 名称

名称	存储类型	容量	状态	访问模式	回收策略	存储卷声明	创建时间	操作
pv0601-sample	sc0601-sample	5Gi	未绑定	读写一次	Recycle	-	2021-6-01 19:16	查看/编辑
7d-3e8a...	sc0531-device	4Gi	已绑定	读写一次	Delete	default/html-test-ngin...	2021-6-01 00:10	查看/编辑
61-a155...	sc	4Gi	待回收	读写一次	Delete	default/html-nginxc-m...	2021-5-31 22:16	查看/编辑

## 更新存储卷

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷（PersistentVolume）**，然后单击 **操作** 栏的 **查看/编辑**。
4. 在 YAML 编辑框中修改信息，单击 **提交**。

更新成功后，系统会自动根据新的 YAML 文件信息更新存储卷。

## 删除存储卷

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷（PersistentVolume）**。
4. 单击 **操作** 栏的更多图标 > **删除**。
5. 在弹出的对话框中单击 **确定**。

# 10.4. 管理存储卷声明（PVC）

您可以在 LHC 控制台上轻松管理存储卷声明（PersistentVolumeClaim），如创建、删除等操作。

## 创建存储卷声明

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷声明（PersistentVolumeClaim）**，然后单击 **创建存储卷声明**。
4. 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test-pvc
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 25Gi
  storageClassName: csi-disk-efficiency # 对应 storage class 的名称
```

### 注意

对于磁盘类型设置为 cloud\_efficiency 和 cloud\_ssd 类型的 Storage Class，其对应的最小存储空间是 20 Gi，当 PVC 中指定的大小小于 20 Gi 时，PV 无法创建成功，PVC 会一直处于 pending 状态。

PVC 创建完成后，您可以在 YAML 文件的 status 中查看创建情况。PVC 创建成功，对应的存储卷会展示在 **存储卷** 页面。

## 更新存储卷声明

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷声明**，然后单击 **操作** 栏的 **查看/编辑**。
4. 在 YAML 编辑框中修改信息，单击 **提交**。

更新成功后，系统会自动根据新的 YAML 文件信息更新存储卷声明。

## 删除存储卷声明

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷声明**。
4. 单击 **操作** 栏的更多图标 > **删除**。
5. 在弹出的对话框中单击 **确定**。

# 10.5. 使用存储卷声明（PVC）

PVC 创建成功后，您可以在创建 Pod 时使用 PVC。

#### 注意

- 云盘不能跨可用区（zone）挂载，因此在创建 `volumeBindingMode: Immediate` 类型的 PVC 后，使用该 PVC 的 pod/workload 需要通过 nodeSelector/nodeAffinity 限制 pod 运行在 PV/PVC 同一个可用区。
- 目前 LHC 未对接阿里云的 NAS 及云盘存储，若要使用存储功能，需要安装第三方存储插件。

## 前提条件

已创建 PVC。

## 操作步骤

- 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群详情**，进入集群列表页。
- 单击某个集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
- 在左侧菜单选项，选择 **容器组（Pods）**，单击 **从 YAML 创建**。
- 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-disk
namespace: default
labels:
  app: nginx
spec:
  tolerations:
  - key: node.kubernetes.io/unreachable
    effect: NoSchedule
    operator: Exists
  automountServiceAccountToken: false
  containers:
  - name: busybox
    image: reg-cnsh-nf.cloud.alipay.com/library/busybox
    command: ["sleep", "3000"]
    ports:
    - containerPort: 80
    volumeMounts:
    - name: data
      mountPath: "/data"
    volumes:
    - name: data
      persistentVolumeClaim:
        claimName: test-pvc
```

容器运行后，可以看到 pod 的状态为 running：

```
# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
pod-with-disk  1/1Running0103s
```

## 10.6. 使用 PVC 挂载动态存储卷

挂载动态存储卷是基于 Storage Class 实现的，在挂载动态存储卷前，需要先创建存储类型（StorageClass）。

### 前提条件

[已创建存储类型](#)。

### 步骤一：创建 PVC

您需要在创建 Pod 前创建存储声明，供创建 Pod 时使用。PVC 创建后，会有对应的 PV 自动创建出来。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **存储** 页签。
3. 在左侧菜单选项，选择 **存储卷声明（Persistent VolumeClaim）**，然后单击 **创建存储卷声明**。
4. 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc0602-sample
  namespace: default

spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 4Gi

  storageClassName: sc0601-sample
  volumeMode: Filesystem
status:
  phase: Bound
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 4Gi
```

创建成功后，名为 **pvc0602-sample** 的存储卷声明将出现在列表页面中。

名称	容量	访问模式	状态	存储类型	存储卷	创建时间	操作
pvc0602-sample	4Gi	读写一次	已绑定	sc0601-sample	yoda-673d3759-eb45-...	2021-6-02 15:33	查看/编辑
html-test-nginx-devic...	4Gi	读写一次	已绑定	sc0531-device	yoda-93ac307d-3e8a-...	2021-6-01 00:10	查看/编辑
html-nginx-mountpoi...	4Gi	读写一次	已绑定	sc	yoda-121ae65e-1e5e-...	2021-5-31 22:04	查看/编辑
html-nginx-lvm-0531-0	4Gi	读写一次	已绑定	sc0531-lvm	yoda-615834d2-fa21-...	2021-5-31 22:02	查看/编辑

PVC 创建后，将自动创建对应的 PV，如下图所示。

名称	存储类型	容量	状态	访问模式	回收策略	存储卷声明	创建时间	操作
yoda-673d3759-eb45-...	sc0601-sample	4Gi	已绑定	读写一次	Delete	default/pvc0602-sample	2021-6-02 15:33	查看/编辑
yoda-93ac307d-3e8a-...	sc0531-device	4Gi	已绑定	读写一次	Delete	default/html-test-nginx...	2021-6-01 00:10	查看/编辑
yoda-0549d361-a155-...	sc	4Gi	待回收	读写一次	Delete	default/html-nginx-m...	2021-5-31 22:16	查看/编辑

## 步骤二：使用 PVC

您可以在创建 Pod 时使用 PVC 挂载动态数据卷。

### 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **容器组（Pods）**，单击 **创建 Pods**。
4. 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
kind: Pod
metadata:
  name: pod-0602example
  labels:
    k8s-app: pod-example
  namespace: default
spec:
  containers:
    - name: nginx
      image: 'hub.antstack-plus.net/library/nginx:latest'
  command: ["/bin/sh", "-c"]
  args: [ "tail -f /dev/null" ]
  volumeDevices:
    - name: data
      devicePath: /dev/xvda
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: pvc0601-sample
```

容器运行后，可以看到 Pod 的状态为 running：

```
# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
pod-0602example 1/1Running0103s
```

## 10.7. 使用 PVC 挂载静态存储卷

您可以通过绑定已存在的 PV 为 Pod 挂载静态存储卷。

### 前提条件

[已创建存储类型](#)。

### 步骤一：创建静态 PV

您可以使用 YAML 文件创建静态 PV。

YAML 样例如下：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0601-sample
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: sc0601-sample
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
```

## 步骤二：创建 PVC 并指定其绑定到 PV

您可以使用 YAML 文件创建 PVC。

YAML 样例如下：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc0601-sample
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: pv0601-sample
  resources:
    requests:
      storage: 8Gi
  storageClassName: sc0601-sample
  selector:
    matchLabels:
      release: "stable"
    matchExpressions:
      - {key: environment, operator: In, values: [dev]}
```

## 步骤三：使用 PVC

您可以在创建 Pod 时使用 PVC 挂载静态数据卷。

样例如下：

```
ind: Pod
metadata:
  name: pod-0602example
  labels:
    k8s-app: pod-example
  namespace: default
spec:
  containers:
    - name: nginx
      image: 'hub.antstack-plus.net/library/nginx:latest'
  command: ["/bin/sh", "-c"]
  args: [ "tail -f /dev/null" ]
  volumeDevices:
    - name: data
      devicePath: /dev/xvda
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: pvc0601-sample
```

## 10.8. StatefulSet 挂载动态存储卷

### 前提条件

- [已创建存储类型](#)。
- [已创建应用服务](#)。

### 使用 Yoda 本地存储创建 StatefulSet

1. 登录 LHC 控制台，在左侧导航栏单击 **集群管理 > 集群**，进入集群列表页。
2. 单击某个集群名称，进入 **集群详情** 页，单击 **工作负载** 页签。
3. 在左侧菜单选项，选择 **有状态副本集（Statefulsets）**，单击 **创建 Statefulsets**。
4. 在输入框中输入 YAML 内容，单击 **提交**。

样例如下：

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-lvm
spec:
  serviceName: "nginx"
  replicas: 2
  volumeClaimTemplates:
  - metadata:
      name: pv0601-sample
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: csi-lvm-novg
      resources:
        requests:
          storage: 1Gi
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
          volumeMounts:
            - mountPath: "/data"
              name: html
```

## 10.9. PVC 与 PV 的绑定

本文介绍 PVC 与 PV 之间的绑定关系。

### PV 状态说明

PV 状态说明详见下表。

状态	说明
Available	已创建的 PV 在未与 PVC 绑定时会处于 <b>Available</b> 状态。
Bound	一个 PVC 与 PV 绑定后，PVC 就会进入 <b>Bound</b> 状态。
Released	若 PV 的回收策略为 Retain，当其绑定的 PVC 被删除时，该 PV 状态会从 <b>Bound</b> 状态转变为 <b>Released</b> 状态。

### PVC 状态说明

PV 状态说明详见下表。

状态	说明
Pending	没有满足条件的 PV 能与 PVC 绑定时，PVC 将处于 Pending 状态。
Bound	当一个 PV 与 PVC 绑定之后，PVC 会进入 Bound 的状态。

## 绑定关系

PVC 与 PV 是一一对应关系，不能一个 PVC 挂载多个 PV，也不能一个 PV 挂载多个 PVC。为应用配置存储时，需要声明一个存储卷声明（PVC），而 Kubernetes 会通过最佳匹配的方式选择一个满足 PVC 需求的 PV，并与之绑定。

**VolumeMode** 主要用于定义 volume 是文件系统（Filesystem）类型还是块（Block）类型，PV 与 PVC 的 VolumeMode 值必须相匹配才会建立绑定关系，具体可参考下表。

PV volumeMode	PVC volumeMode	是否绑定
—	—	Y
—	Block	N
—	Filesystem	Y
Block	—	N
Block	Block	Y
Block	Filesystem	N
Filesystem	Filesystem	Y
Filesystem	Block	N
Filesystem	—	Y

# 11. 镜像中心

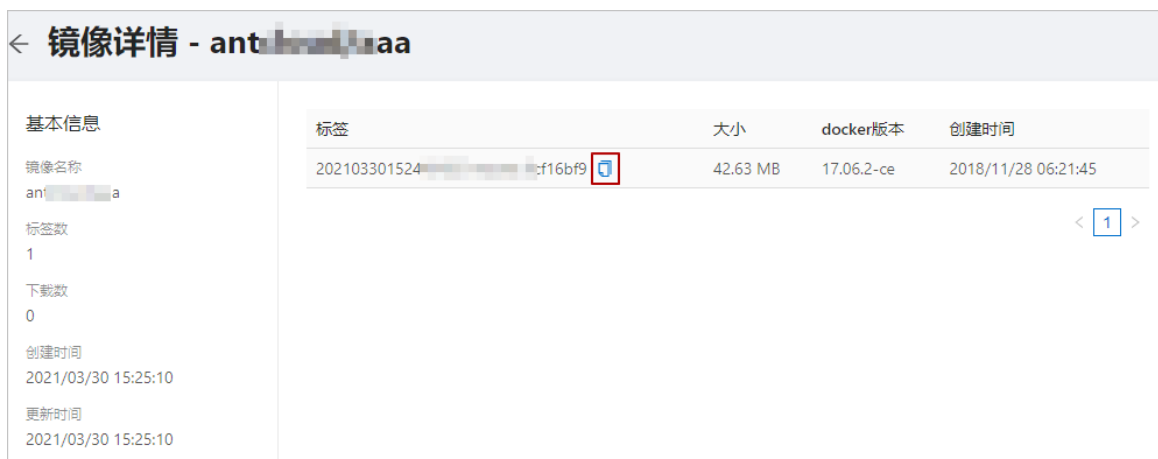
## 11.1. 镜像仓库

镜像仓库是镜像的集合，用于存储、管理容器镜像。您可以在控制台上轻松存储、管理、部署容器镜像。本文主要介绍如何下载平台镜像、上传私有镜像。

### 下载平台镜像

平台镜像中提供常用的样例镜像。

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像仓库**。
2. 选择 **平台镜像** 页签。平台提供的镜像以卡片形式展示。
3. 单击卡片进入镜像信息页面，单击标签右侧的复制图标，复制镜像地址，用 `docker pull` 命令下载镜像。



### 上传私有镜像

#### 前提条件

- 准备一台装有 Docker 的机器。

#### 说明

Docker 版本必须为 1.6.0 及以上。

- 租户初次使用服务账号登录，需先在租户内新建一个用户。

#### 操作步骤

1. 登录装有 Docker 的机器。
2. 获取用户的 accessKey 和 accessSecret，分别用作用户名和密码登录镜像中心。

格式：`docker login 镜像中心域名`，示例：`docker login hub.alipay.net`。

 说明

accessKey 和 accessSecret，可以在 控制台 > 账户信息 > AccessKey 信息 中获取。

3. 将镜像命名为 `镜像中心域名/租户名/应用名称:标签` 的格式。

示例：`hub.alipay.net/tenantname/appname:imagetag`。

4. 执行以下命令上传镜像：

```
docker push hub.alipay.net/tenantname/appname:imagetag
```

上传后的镜像会存储在 **镜像中心 > 镜像仓库 > 私有镜像** 中。单击卡片进入镜像信息页面，单击标签右侧的复制图标复制镜像地址，您可以用 `docker pull` 命令下载镜像。

## 11.2. 构建镜像

镜像构建提供将应用的源代码编译成镜像的服务能力。镜像成功构建后，默认会推送至 LHC 默认的镜像中心。

### 创建构建

本部分介绍如何通过单元化应用服务控制台完成镜像构建操作。

### 前提条件

[已创建应用](#)。

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像构建**。
2. 在构建列表页，单击 **创建镜像构建**。
3. 在 **创建镜像构建** 页面，完成构建配置。
  - i. 完成构建基础选项配置。

配置项	说明
所属应用	选择构建要关联的应用。
构建集名称	默认与所选的应用名称相同，可根据需要进行修改。
镜像名	默认与所选的应用名称相同，无法修改。
从源码构建	支持对接 GitLab 源码仓库拉取源码以进行镜像构建。

配置项	说明
从包构建	支持上传指定技术栈应用（SOFABoot、Java 等）的代码包。系统会根据相关配置信息自动地进行镜像构建，并在后续的容器服务发布中使用系统构建的目标镜像进行发布。
代码库类型	<p>如果选择的是 <b>从源码构建</b>，需要配置 <b>代码库类型</b>。具体如下：</p> <ul style="list-style-type: none"> <li>■ <b>代码库类型</b>：目前支持 <b>外部 GitLab</b>，可配置支持公网访问并且版本在 8.8 及以上的 GitLab（同时支持 GitLab v3 和 v4 版本 API）。</li> <li>■ <b>外部 GitLab 地址</b>：外部 GitLab 的地址，本例中输入 <code>http://git.cloud.alipay.com/test/test.git</code>。</li> <li>■ <b>外部 GitLab Token</b>：输入 GitLab 的访问 Token，需 GitLab 8.8 以上版本，具体可参考 <a href="#">Personal access tokens</a>。</li> </ul> 
技术栈	<p>选择 <b>从包构建</b> 时，需要配置 <b>技术栈</b>。</p> <p>选择镜像技术栈，即构建基于的基础镜像。包括用户自定义的技术栈和官方提供的技术栈。详情参见 <a href="#">镜像技术栈</a>。</p>
技术栈版本	<p>选择 <b>从包构建</b> 时，需要配置 <b>技术栈版本</b>。</p> <p>选择镜像技术栈的版本。</p>
目标镜像中的位置	<p>选择 <b>从包构建</b> 时，需要配置 <b>目标镜像中的位置</b>。</p> <p>根据所选择的技术栈及版本自动生成。代码包在构建镜像中的目录路径，需以 <code>/</code> 开头。系统会将发布包拷贝到基础镜像中用户指定的目录。</p>

配置项	说明
基础镜像	<p>选择 <b>从包构建</b> 时，需要配置 <b>基础镜像</b>。 根据所选择的技术栈及版本自动生成。</p> <div> <div>从源码构建 从包构建</div> <div> <p>* 技术栈</p> <p>z001</p> </div> <div> <p>* 技术栈版本</p> <p>sofaboot:1622100</p> </div> <div> <p>* 目标镜像中的位置 ①</p> <p>/c/c</p> </div> <div> <p>* 基础镜像</p> <p>hub.antstack-plus.net/pu626234612660-master.5031a6a9</p> </div> </div>

## ii. (可选) 完成构建高级选项配置。

配置项	说明
源镜像中心访问配置	<p>Dockerfile 中 FROM 镜像所在的镜像中心访问配置，单击 <b>添加源镜像中心访问配置</b> 可配置多个。如果基础镜像来自外部且需认证的镜像中心，必须填写该配置。</p> 
上传到自定义镜像中心	<p>用于配置构建成功后额外的 Push 镜像的镜像中心访问信息。</p> <p>默认关闭。如果保持关闭，将仅上传至 SOFAShark 平台默认的镜像中心。如果开启，需配置以下选项：</p> <ul style="list-style-type: none"><li>■ <b>自定义镜像名</b>：输入自定义的镜像名称，比如： <pre>registry.cn-hangzhou.aliyuncs.com/mytenant/sayhello</pre></li><li>■ <b>用户名</b>：选填。当所配置的镜像中心需要用户名和密码进行登录时进行配置。</li><li>■ <b>密码</b>：选填。当所配置的镜像中心需要用户名和密码进行登录时进行配置。</li></ul> 
构建路径	<p>选填。执行 <code>docker build</code> 所在的工作目录，相对源代码 checkout 后的路径。默认为源代码根目录。</p>
Dockerfile 路径	<p>选填。执行 <code>docker build</code> 时的 <code>--file</code> 参数。默认为源代码根目录下 Dockerfile 文件。</p>

4. 单击 **预览**，确认信息无误后，单击 **提交**。

## 触发构建

您可以在完成本地代码更新后触发构建镜像。

## 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像构建**，进入构建列表页面。
2. 单击 **构建集名称** 进入构建详情页。
3. 在 **构建历史** 页签中，单击 **触发构建**。

4. 在弹出的 **触发构建** 对话框中，完成以下配置后，单击 **确定**。

配置项	说明
构建分支/标签	选择 <b>从源码构建</b> 时，需配置 <b>构建分支/标签</b> 。 代码的分支或标签，可根据需要选择构建分支或标签。
自定义构建参数	选择 <b>从源码构建</b> 时，需配置此选项。 输入 key 值、value 值。
发布包	选择 <b>从包构建</b> 时，需配置此选项。 上传代码包，无格式限制。
Dockerfile 预览	选择 <b>从包构建</b> 时，此选项才会展现。 构建镜像的 Dockerfile 内容，派生自基础镜像，只作发布包覆盖。
目标镜像仓库地址（自定义）	选择 <b>从包构建</b> 时，需配置此选项。 构建出的镜像名称，此处填写目标镜像标签。系统会将构建出来的目标镜像按照此处定义的地址推送到镜像中心，供后续发布所用。

## 删除构建

您可以在控制台上删除不需要的构建。

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像构建**，进入构建列表页。
2. 单击构建 **操作** 栏中的 **删除**，在弹出的 **删除构建** 对话框中单击 **确定**。

您可以勾选 **同时删除镜像**，将构建关联的镜像一起删除。

## 查看构建历史

完成镜像构建后，就可以进入该构建详情页面查看构建详情。

### 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像构建**，进入构建列表页。
2. 单击 **构建集名称** 进入构建集详情页。
3. 在 **构建历史** 页签中，查看构建历史信息。您可以单击 **镜像版本** 右侧的复制按钮复制镜像地址，或者单击镜像构建记录序号查看详细构建日志，可参考 [查看触发日志](#)。

### 镜像构建详情

#### 基本信息

构建集名称: te-02

所属应用名称: te-02

构建次数: 13

负责人:

更新时间: 2021-05-28 19:50:08

代码仓库: git@git-group/test-demo.git

镜像名: hub.test0526-02 技术栈名称:

技术栈版本名称:

技术栈版本号: 16215742

#### 构建历史

触发构建

记录	提交人	状态	代码分支	代码版本	镜像版本	开始时间	更新时间
#13		✓成功			po	2021-05-28 19:40:52	2021-05-28 19:41:23
#12		✓成功	master	9c9	D210500774-master.9cf16bf9	2021-05-28 19:38:58	2021-05-28 19:39:33
#11		✓成功			po	2021-05-28 19:32:05	2021-05-28 19:32:35

## 查看触发日志

您可以在控制台查看完整的构建触发日志。

## 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像构建**，进入构建列表页。
2. 单击 **构建集名称** 进入构建详情页。
3. 在 **构建历史** 页签中，单击镜像构建记录序号，查看详细的触发日志信息。

触发构建序号: 4

提交人: [REDACTED]

代码分支:

代码版本:

镜像版本: 45 [REDACTED] 646

创建时间: 2021-05-28 14:36:47

更新时间: 2021-05-28 14:37:21

日志

✓成功

信息: The build succeeded.

```
0 pulling build image which may take some time ...
1 pulling build image which may take some time ...
2
3 FROM hub.[REDACTED].net/public/[REDACTED]626234612660-master.[REDACTED]a6a9
4
5 RUN mkdir -p ddd
6
7 COPY --from=BUILDER /tmp/binary2image/resource/ ddd
8 Option is-push-alipaydocker is false
9 Option image-tag is 4[REDACTED]646
10 Build image from binary directly
11 INFO cd to compile dir /home/
12 INFO docker login hub.antstack-plus.net
13 Login Succeeded
14 INFO Preparing binary to image dockerfile
15 FROM hub.a[REDACTED].net/sofastack/busybox-curl AS BUILDER
16
17 RUN tmpdir=/tmp/binary2image/resource && mkdir -p ${tmpdir} && curl -o ${tmpdir}/111.json "http://minio.
18 et:9000/hubstorage/binary2image/binary/ANTCLOUD/00000000[REDACTED]4/111.json?X-Amz-Algorithm=AWS4-HMAC-SHA256
19 tial=minio%2F20210528%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210[REDACTED]3647Z&X-Amz-Expires=600&X-Amz-
20 =host&X-Amz-Signature=1ae8ce79c977[REDACTED]bc513695fcdefe5b110ca903bea08a79f6f5c05f70"
19 FROM hub.[REDACTED].net/public/sofaboot-app:2[REDACTED]4612660-master.5031a6a9
20
```

## 11.3. 镜像技术栈

镜像技术栈为用户提供基础镜像管理的能力，可自定义镜像技术栈。

### 创建镜像技术栈

创建镜像技术栈将会定义镜像技术栈的基本元数据。

#### ② 说明

目前对可以创建的镜像技术栈个数不作限制。

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像技术栈**，进入镜像技术栈列表页面。
2. 单击 **创建镜像技术栈**，在弹出的窗口中完成以下配置。

配置项	说明
技术栈名称	必填。允许包含字母、数字、连字符，且必须以字母开头、以字母或数字结尾。最多允许输入 32 个字符。 创建后技术栈的名称不可修改。

配置项	说明
技术栈描述	必填。填写技术栈的描述。

3. 单击 **提交**。

## 创建镜像版本

创建镜像技术栈后，您可以根据需要创建不同的镜像版本。

### ② 说明

目前对可以创建的镜像版本个数不作限制。

1. 在镜像技术栈列表页，单击需要创建镜像版本的技术栈名称，进入技术栈详情页。

2. 单击 **创建镜像版本**，完成以下配置。

配置项	说明
镜像版本名称	允许包含字母、数字、连字符、且必须以字母开头、以字母或数字结尾。最多允许输入 32 个字符。创建后版本的名称不可修改。
基础镜像信息	选择要使用的镜像仓库和基础镜像版本。  <div>② 说明 如果基础镜像来自外部镜像中心，需提供用户名和密码。</div>
代码包上传路径地址	代码包在构建镜像中的目录路径。需以 <code>/</code> 开头。系统会将发布包拷贝到基础镜像中用户指定的目录。
备注	输入基础镜像的 Git 地址。

3. 单击 **提交**。

创建后系统会提示创建成功，并以列表形式在技术栈详情页展示。在该页面，您可以对自定义的技术栈版本继续进行浏览、下架的操作。

### ② 说明

- 镜像版本号由系统自动生成，目前不支持修改。
- 镜像版本下架后不可恢复。下架的版本无法在创建应用服务时选择使用。

## 11.4. 镜像安全扫描

### 11.4.1. 概述

本文介绍镜像安全扫描的背景信息以及与同类产品的差异对比，以帮助您全面认识 CAFE 镜像安全扫描。

#### 背景信息

随着越来越多的应用程序被容器化，容器镜像安全也随之变得越来越重要。

镜像安全扫描功能支持对镜像中存在的系统漏洞、应用漏洞进行检测，为您提供一站式漏洞管理能力，进而创造安全的镜像运行环境。

#### CAFE 镜像安全扫描与阿里云安全中心

CAFE 镜像安全扫描与 [阿里云安全中心](#) 的具体差异可参考下表。

		
Harbor	第三方镜像服务	Harbor ACR Quay
立即扫描 定期扫描	扫描触发方式	立即扫描 定期扫描
支持同时设置并执行多个扫描任务	多扫描任务	仅支持一种扫描任务配置
支持扫描前设置过滤级别	按漏洞级别过滤	不支持扫描前设置，仅能将扫描结果筛选
系统漏洞 应用漏洞	筛选漏洞类型	系统漏洞 应用漏洞
支持	忽略未修复漏洞	不支持
支持	忽略指定漏洞	不支持

### 11.4.2. 接入镜像仓库

对镜像执行安全扫描之前，您需要先接入镜像仓库。本文介绍如何通过控制台接入镜像仓库。

#### 操作步骤

1. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **接入**。



3. 在 接入镜像仓 面板，完成以下接入配置。

接入镜像仓

\* 镜像仓库类型

harbor

ACR

\* 版本

1.x

\* 协议

http

\* 镜像仓库接入地址

hu...b.net

\* 账号

admin

\* 密码

.....

配置项	说明
镜像仓库类型	支持接入的容器镜像仓库类型为 harbor。
版本	镜像仓库的版本。支持的版本为 1.x。
协议	镜像仓库的通信协议。可选择 HTTP 或 HTTPS。
镜像仓库接入地址	输入镜像仓库接入的地址。
账号	输入访问镜像仓库时使用的账号。
密码	输入访问镜像仓库的密码。

#### 4. 单击 确定。

成功接入镜像仓库后，您可以在 **镜像安全扫描** 页面的接入镜像仓库列表中查看已接入的镜像仓库信息。



### 后续步骤

完成镜像仓库的接入后，您还需要执行镜像安全扫描操作，才能通过镜像安全扫描检测您的镜像是否存在风险。更多信息，请参见 [执行扫描任务](#)。

### 其他操作

您还可以在接入镜像仓库列表中执行更多操作，包括修改、删除等操作。更多信息，请参见 [管理镜像仓库](#)。

## 11.4.3. 执行扫描任务

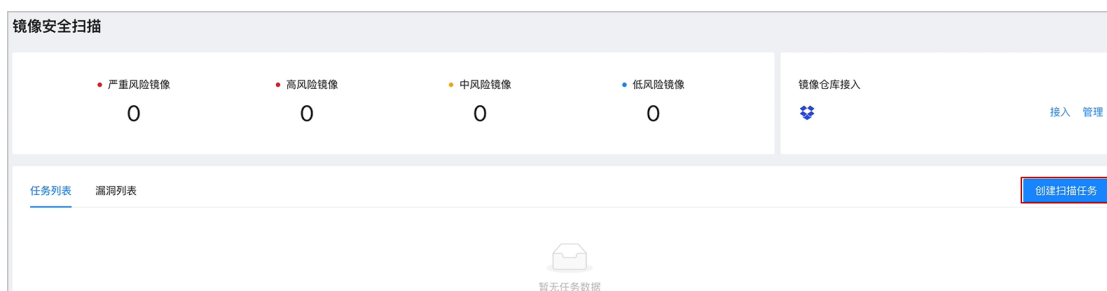
通过镜像安全扫描，检测目标镜像中是否存在镜像漏洞和恶意样本，从而创造安全的镜像运行环境。本文介绍如何执行扫描任务。

### 前提条件

[已接入镜像仓库](#)。

### 操作步骤

1. 创建扫描任务。
  - i. 登录控制台，在左侧导航栏单击 **镜像中心 > 镜像安全扫描**。
  - ii. 在 **镜像安全扫描** 页，单击 **任务列表** 页签 > **创建扫描任务**。

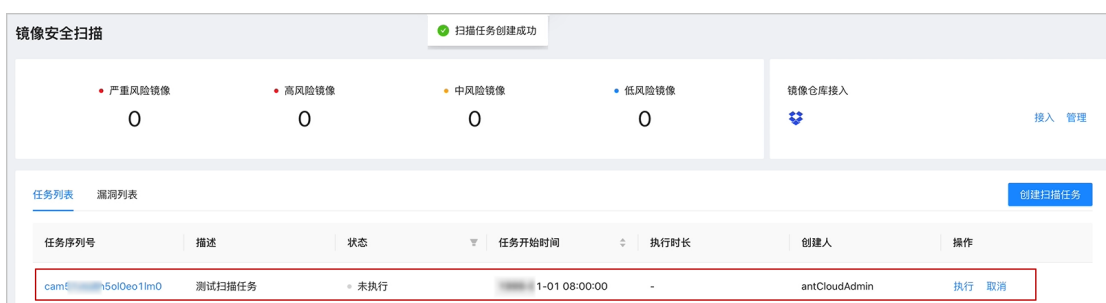


iii. 在 **创建扫描任务** 页，完成以下扫描配置。

配置项	说明
选择镜像范围	根据需要选择镜像仓库。展开镜像仓库后，将显示该镜像仓库下的镜像列表，支持按需选择具体镜像。
描述	输入关于扫描任务的说明。

iv. 单击 **创建**。

创建完成后，扫描任务将显示在任务列表中。



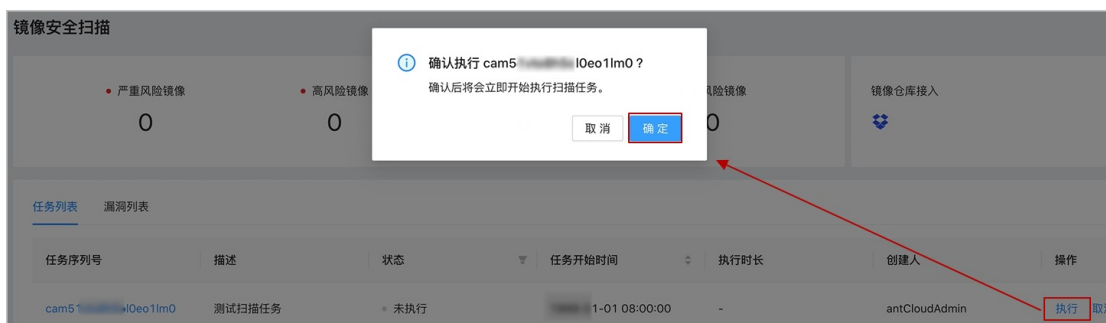
2. 执行扫描任务。

i. 在任务列表中，找到刚创建的扫描任务，单击 **操作** 列中的 **执行**。

#### ? 说明

如需取消执行过程，单击 **操作** 列中的 **取消** 即可。

ii. 在弹出的对话框中，单击 **确定**。



## 后续步骤

创建完镜像扫描任务后，您可以查看镜像安全扫描结果。更多信息，请参见 [查看镜像安全扫描结果](#)。

## 11.4.4. 查看镜像安全扫描结果

镜像安全扫描可以检测镜像资产中存在的系统漏洞和应用漏洞，并进行分类展示，帮助您全面了解您的镜像资产中存在的安全风险。本文介绍如何查看镜像安全扫描结果。

### 前提条件

已创建扫描任务。

## 查看扫描结果

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，查看所有任务扫描结果的统计信息。

扫描结果按照严重风险、高风险、中风险、低风险四个等级进行展示。



在 **镜像安全扫描** 页，还可以查看扫描任务信息和漏洞列表信息。

- 单击 **任务列表** 页签，查看任务列表信息。更多信息，请参见 [查看任务详情](#)。
- 单击 **漏洞列表** 页签，查看漏洞列表信息。更多信息，请参见 [查看漏洞数据详情](#)。

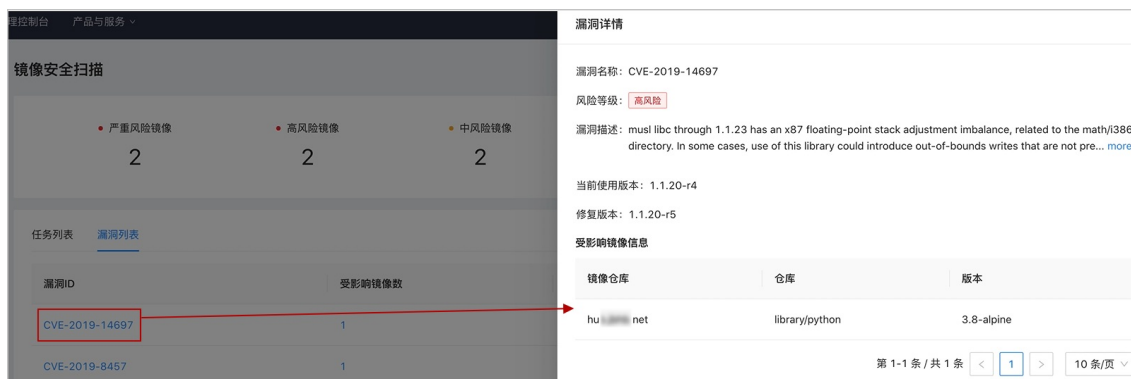
## 查看任务详情

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **任务列表** 页签，查看任务列表信息。
3. 单击目标任务序列号进入任务详情页面，查看如下信息：
  - **基本信息**：显示任务序列号、状态、创建人、执行时长等。
  - **漏洞列表**：显示该任务的漏洞数据信息，比如漏洞 ID、漏洞级别、受影响镜像数、最新扫描时间。

任务详情			
基本信息			
任务序列号: cam5-0eo1m0	状态: 已完成	任务开始时间: 2022-06-17 18:01:43	执行时长: 164.208523ms
创建人: an-Admin	任务描述: 测试扫描任务		
漏洞列表			
漏洞ID	受影响镜像数	风险等级	最新扫描时间
CVE-2019-14697	1	高风险	2022-06-17 18:01:43
CVE-2019-8457	1	高风险	2022-06-17 18:01:43
CVE-2019-15903	1	中风险	2022-06-17 18:01:43

## 查看漏洞数据详情

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **漏洞列表** 页签，查看漏洞列表信息。
3. 单击目标漏洞名称进入漏洞详情页面，查看如下信息：
  - **漏洞基本信息**：显示漏洞名称、风险等级、漏洞描述信息等。
  - **受影响镜像信息**：显示受影响的镜像信息，比如镜像仓库地址信息、版本等。



## 11.4.5. 管理镜像仓库

本文介绍如何通过控制台管理镜像仓库，包括镜像仓库查看、修改和删除。

### 前提条件

已接入镜像仓库。

### 查看镜像仓库列表

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **管理**。
3. 在 **接入镜像仓库设置** 页，查看接入镜像仓库列表。
  - **镜像仓库类型**：支持接入的容器镜像仓库类型。
  - **镜像仓库接入地址**：镜像仓库接入的地址。
  - **镜像仓库 ID**：镜像仓库的 ID。
  - **协议**：镜像仓库的通信协议。
  - **账号**：访问镜像仓库时使用的账号。
  - **操作**：支持 **编辑**、**删除**。相关操作，请参见 [修改镜像仓库](#)、[删除镜像仓库](#)。

### 修改镜像仓库

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **管理**。
3. 在 **接入镜像仓库设置** 页，找到需要修改的镜像仓库，单击 **操作** 列中的 **编辑**。



4. 在 **编辑镜像仓库** 页，根据需要进行修改后，单击 **确定**。当前仅允许修改账户和密码。

### 删除镜像仓库

 注意

删除用户镜像仓库会导致已存在的扫描任务中相关仓库的镜像被删除，请谨慎操作。

1. 登录控制台，在左侧导航栏单击 **镜像中心** > **镜像安全扫描**。
2. 在 **镜像安全扫描** 页，单击 **管理**。
3. 在 **接入镜像仓库设置** 页，找到需要删除的镜像仓库，单击 **操作** 列中的 **删除**。
4. 在弹出的对话框中，单击 **确定**。

# 12.应用商店

## 12.1. 应用商店概述

LHC 应用商店功能集成了 Helm 能力，提供 Helm 相关功能，并进行了相应扩展，例如提供图形化界面、官方仓库、版本回滚等。

### 背景信息

Helm 是管理 Kubernetes 应用程序的打包工具，主要用来管理 Charts。Helm Chart 是用来封装 Kubernetes 原生应用的一系列 YAML 文件。可在您部署应用时自定义应用的一些 Metadata，以便于应用的分发。

对于应用发布者而言，可通过 Helm 打包应用、管理应用依赖关系、管理应用版本等。对于使用者而言，使用 Helm 后无需编写复杂的应用部署文件，即可轻松查找、安装、升级、回滚及卸载应用。

LHC 应用商店基于 Helm 能力作了增强，通过产品模版简化部署复杂应用的过程，并根据行业特性构建行业交付标准、交付流程和交付路径的基础，旨在打造行业通用的应用商店，可根据不同需求应对不同的业务场景。

### 功能说明

LHC 应用商店支持如下功能：

#### ● 产品模板

- 私有平台：支持上传用户自定义的应用模板、应用模板包、新版本。
- 公有平台：支持引入并管理第三方应用模板仓库，以及将第三方模板复制到私有平台。
- 版本控制：针对模板变更自动做版本号的增加，而不是单纯覆盖。
- 下载模板：支持下载私有平台中的应用模板包。

#### ② 说明

您可以对私有平台中的产品模板进行增加、删除、编辑内容的操作，公有平台的产品模板仅支持增删操作。若要对公有平台的产品模板进行定制，可以先将第三方模板复制至私有平台。

#### ● 产品发布

- 快速部署：选择私有或公有平台的应用模板，自定义参数并快速地一键部署常用的应用到 Kubernetes 中。
- 自定义参数：模板安装时用户自定义的参数，会在安装详情页里作持久化保留，而不是覆盖原有商店中的配置。
- 版本回滚：当前模板有其它成功发布的历史版本（除当前版本外）时，可以进行一键回滚。

### 相关文档

- 关于 Helm 的更多信息，可参考 [Helm 官网文档](#)。
- 关于产品模板的更多信息，可参考 [产品模板](#)。
- 关于产品发布的更多信息，可参考 [产品安装](#)。

## 12.2. 产品模板

产品模板为基于 Kubernetes Helm 标准的应用模板提供了统一高效的管理机制。本文介绍如何在私有平台上传私有模板、如何在公有平台添加外部源，以及如何将公有平台模板导入私有平台等。

## 私有平台

私有平台相当于用户的私有仓库，存储以下两种类型的模板。

- 平台模板

由 SOFAShark 平台提供和维护的官方示例模板，当前支持 redis、etcd、mysql、mongodb、elasticsearch-exporter、kibana 等示例模板，让您可以即开即用。

### 说明

此类模板仅支持查看和安装操作，用户无法修改模板信息。

- 私有模板

满足用户管理和维护自定义模板的需求，私有平台的模板的来源和用途局限于当前集群。您可以对模板内容进行增删改查。

## 上传私有模板

### 说明

目前没有限制模板的上传个数。

## 前提条件

已准备好符合 Helm 规范的产品模板。

## 操作步骤

1. 登录 LHC 控制台，在左侧导航栏单击 **应用商店 > 产品模板**，进入私有平台页面。
2. 单击 **上传产品模板**，在弹出框中选择准备好的自定义产品模板。

### 说明

仅允许上传 tgz/zip 格式的文件，推荐使用 `helm package` 打包。

上传成功后，模板会以卡片的形式展示在列表页。单击卡片进入 **模板详情** 页，您可以继续进行如下操作：

- 在 **模板版本** 页签中查看、下载、删除历史版本。
- 在 **内容详情** 页签中查看、修改、编辑模板内容，修改后单击 **保存**。

## 公有平台

公有平台作为全局仓库，可以引入第三方模板源。公有平台的模板不支持直接修改内容。如需修改，可安装和复制模板到私有平台。

## 添加外部源

您可以通过 LHC 控制台将第三方库加入到源列表，此操作等效于 `helm repo add [NAME] [URL]` 命令。

## ② 说明

最多可添加 5 个外部源。

1. 登录 LHC 控制台，在左侧导航栏单击 **应用商店** > **产品模板**，单击 **公有平台** 页签进入公有平台页面。
2. 单击 **添加外部源**，在弹出框中完成以下配置后，单击 **确定**。



- **外部源名称**：Helm Repo 的标记名称，可包含英文字母、数字、下划线或中划线。本例中输入 `stable`。
- **外部源 url 地址**：Helm Repo 的路径地址，必须以 `http` 或 `https` 开头，且需符合 URL 规范。本例中输入 `https://apphub.aliyuncs.com/stable`。

添加成功后，外部仓库中的模板会自动导入。

## 导入私有平台

若对公有平台模板有定制化需求，您可以先将公有模板导入私有平台。

1. 登录 LHC 控制台，在左侧导航栏单击 **应用商店** > **产品模板**，单击 **公有平台** 页签进入公有平台页面。
2. 单击目标模板卡片上的 **导入私有平台**，在弹出框中单击 **确定**。



导入成功后，您可以在私有平台搜索到该模板，并对模板内容进行编辑和版本管理。

## 12.3. 产品安装

通过安装产品模板，您可以将标准的 Helm 应用模板发布为相应的 Kubernetes workloads。

### 前提条件

已 [上传产品模板](#) 或 [添加外部源](#)。

### 安装私有模板

1. 登录 LHC 控制台。
2. 在左侧导航栏上，单击 [应用商店](#) > [产品模板](#)。
3. 单击 [私有平台](#) > [私有模板](#)，进入模板列表页面。
4. 单击模板卡片进入模板详情页，单击右上角的 [产品安装](#)。
5. 在弹出框中填入以下配置。
  - **发布名称**：输入自定义的发布产品名称。支持英文小写字母、数字、中划线，必须以英文小写字母开头，最大长度为 32 个字符。
  - **集群**：选择对应集群。
  - **命名空间**：选择相应的命名空间。
  - **版本**：选择模板的版本。
  - **描述**：选填。最大长度 100 个字符。
6. 单击 [提交](#)，进入 [产品安装详情](#) 页。
7. 单击 [开始/重新安装](#)，在弹出框中单击 [确定](#)。

安装成功后，可以单击发布详情中的工作负载或者应用服务链接进入对应的详情页面。

### ? 说明

- 如果安装失败，可以单击 **发布错误日志** 查看错误日志排查问题。
- 如果要中止安装，可以单击 **取消安装**，在弹出框中单击 **确定**。

## 发布公有模板

1. 登录 LHC 控制台。
2. 在左侧导航栏上，单击 **应用商店 > 产品模板**，进入私有模板列表页面。
3. 选择 **公有平台** 页签，进入公有模板列表页面。
4. 单击公有模板卡片进入模板内容详情。
5. 单击右上角的 **产品安装**，在弹出框中完成以下配置。
  - **发布名称**：输入自定义的发布产品名称。支持英文小写字母、数字、中划线，必须以英文小写字母开头，最大长度为 32 个字符。
  - **命名空间**：选择相应的命名空间。
  - **版本**：选择模板的版本。
  - **描述**：选填。最大长度 100 个字符。
6. 单击 **提交**，进入 **产品安装详情** 页。
7. 单击 **开始/重新安装**，在弹出框中单击 **确定**。

安装成功后，可以单击发布详情中的工作负载或者应用服务链接进入对应的详情页面。

### ? 说明

- 如果安装失败，可以单击 **发布错误日志** 查看错误日志排查问题。
- 如果要中止安装，可以单击 **取消安装**，在弹出框中单击 **确定**。

## 更多操作

模板安装完成后，在 **产品安装详情** 页您还可以根据实际业务需要执行更多操作。

单元化应用服务

← 产品安装详情-test1

开始/重新安装 更新版本 删除实例

概览

发布运维

集群管理

流量管理

网络

配置管理

镜像中心

应用商店

产品模板

产品发布

成功

基本信息

产品名称: test1

产品模板名称: nginx-demo

应用版本: 0.1.0

命名空间: region-ddd

创建时间: 2021-06-17 16:45:32

更新时间: 2021-06-17 16:47:22

发布详情 历史版本

nginx-demo 成功

Deployment

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	UNAVAILABLE
test1-nginx-demo	1	1	1	1	1	-

Service

NAME	TYPE	CLUSTER_IP	EXTERNAL_IPS	PORTS
test1-nginx-demo	ClusterIP	10.1.1.60.34	-	[80/TCP]

具体操作可参考下表。

操作项	操作场景	操作步骤
更新模板版本	在产品模板安装前后，您可以快速更新要安装的产品模板版本。	<ol style="list-style-type: none"> <li>1. 单击 <b>更新版本</b>。</li> <li>2. 在弹出窗中选择要更新到的版本，单击 <b>确定</b>。</li> </ol>
删除实例	如果不需要继续使用某个应用，您可以删除实例，删除后会释放所有后端资源。	<p>单击 <b>删除实例</b> &gt; <b>确定</b>。</p> <div>  <b>注意</b> 删除后无法恢复，请谨慎操作。         </div>
回滚历史快照	如果需要将应用回滚到之前的版本，您可以执行回滚操作。	<ol style="list-style-type: none"> <li>1. 选择 <b>历史版本</b> 页签。</li> <li>2. 单击目标版本操作栏中的 <b>回滚</b>，然后单击 <b>确定</b>。</li> </ol> <div>  <b>说明</b> <ul style="list-style-type: none"> <li>• 仅成功安装的历史版本支持回滚操作。</li> <li>• 支持一键将应用回滚到某个历史版本，默认保留最近 10 个历史版本记录。</li> </ul> </div>

# 13.基础元数据

## 13.1. 逻辑单元

在创建单元化工作空间或者工作空间组时，系统会自动进行逻辑单元和部署单元的规划。您可以根据实际需求，对逻辑单元进行修改或删除。本文介绍如何通过控制台为逻辑单元配置灾备及相关操作。

### 背景信息

逻辑单元是单元化架构的基础，一个单元被称为一个 Zone。根据业务特点不同，您可以将系统部署在不同类型的逻辑单元中。

- RZone（Region Zone）：部署按用户维度拆分的关键业务系统。核心业务和数据单元化拆分，拆分后分片均衡，单元内尽量自包含（调用封闭），拥有自己的数据，能完成所有业务。一个可用区可以有多个 RZone。
- GZone（Global Zone）：部署未按用户维度拆分的系统，被 RZone 依赖，提供不可拆分的数据和服务，如配置型的业务。数据库可以和 RZone 共享，多租户隔离，全局只有一组，可以配置流量权重。
- CZone（City Zone）：部署未按用户维度拆分的系统，被 RZone 高频访问，解决跨域通信延时问题。为了解决异地延迟问题而特别设计，适合读多写少且不可拆分的业务。一般每个城市一套应用和数据，是 GZone 的快照。

### 配置灾备

灾备即容灾与备份结合的保障企业数据、业务高可用的方案之一。具体配置操作如下：

1. 登录控制台，在左侧导航栏单击 **基础元数据 > 逻辑单元**。
2. 在逻辑单元列表页，选择需要配置灾备信息的逻辑单元，单击 **灾备信息** 列中的 **配置**。
3. 在对话框中完成以下配置后，单击 **提交**。

配置项	说明
同城灾备组	可选。同城容灾场景下的灾备分组。 <div>② 说明 不支持和 异地灾备组 选择相同的逻辑单元。</div>
异地灾备组	可选。异地容灾场景下的灾备分组。 <div>② 说明 不支持和 同城灾备组 选择相同的逻辑单元。</div>
正式容灾状态	正式容灾或容灾演练场景下的状态。根据实际容灾场景进行选择。支持 正常、同城、异地。

## 查看灾备信息

您可以在逻辑单元列表页查看灾备信息。

1. 登录控制台，在左侧导航栏单击 **基础元数据 > 逻辑单元**。
2. 在逻辑单元列表页，选择需要查看灾备信息的逻辑单元，将鼠标光标悬浮于 **灾备信息** 列中的 **查看** 上方，即可查看当前逻辑单元配置的灾备信息。

## 删除逻辑单元

如果不再需要某个逻辑单元，您可以通过 **基础元数据 > 逻辑单元**，也可以通过 **全局设置** 进行删除（更多信息，请参见 [创建逻辑单元](#)）。

### 注意

删除逻辑单元前要先删除部署单元，若直接删除会报错。

1. 登录控制台，在左侧导航栏单击 **基础元数据 > 逻辑单元**。
2. 在逻辑单元列表页，选择需要删除的逻辑单元，单击 **操作** 列中的 **删除**。
3. 在弹出框中，单击 **确定**。

### 说明

删除逻辑单元会导致应用服务不可用，请谨慎操作。

## 相关文档

您可以参考以下文档，了解逻辑单元更多功能及操作：

- [创建逻辑单元](#)
- [删除部署单元](#)

# 13.2. 部署单元

在创建单元化工作空间或者工作空间组时，系统会自动进行逻辑单元和部署单元的规划。您可以根据实际需求，对部署单元进行修改或删除。

## 背景信息

部署单元（Cell）定义了一个服务通信的边界，用以规整通信的链路，为发布部署时切流做无损发布、高可用应用架构，以及容灾切换和恢复打下基础。部署单元（Cell）也是实现单元化的基础。

- 双机房双活场景下，一个应用的应用服务可以跨部署单元（Cell）部署。
- 单元化场景下，一个应用可以创建多个应用服务在不同部署单元（Cell）部署。

## 修改部署单元

1. 登录控制台，在左侧导航栏单击 **基础元数据 > 部署单元**。
2. 在部署单元列表页，选择需要修改的部署单元，单击 **操作** 列中的 **编辑**。
3. 在 **编辑部署单元** 页，根据需要修改相应信息。仅支持修改默认权重，即部署单元所承载的业务流量权重。

4. 完成后单击 **提交** 保存修改。

## 删除部署单元

如果不再需要某个部署单元，可以在部署单元列表页将其删除。

1. 登录控制台，在左侧导航栏单击 **基础元数据 > 部署单元**。
2. 在部署单元列表页，选择需要删除的部署单元，单击 **操作** 列中的 **删除**。
3. 在弹出框中，单击 **确定**。

### ② 说明

删除部署单元会导致应用服务不可用，请谨慎操作。

## 相关文档

您可以参考以下文档，了解部署单元更多功能及操作：

- [创建部署单元](#)
- [逻辑单元](#)

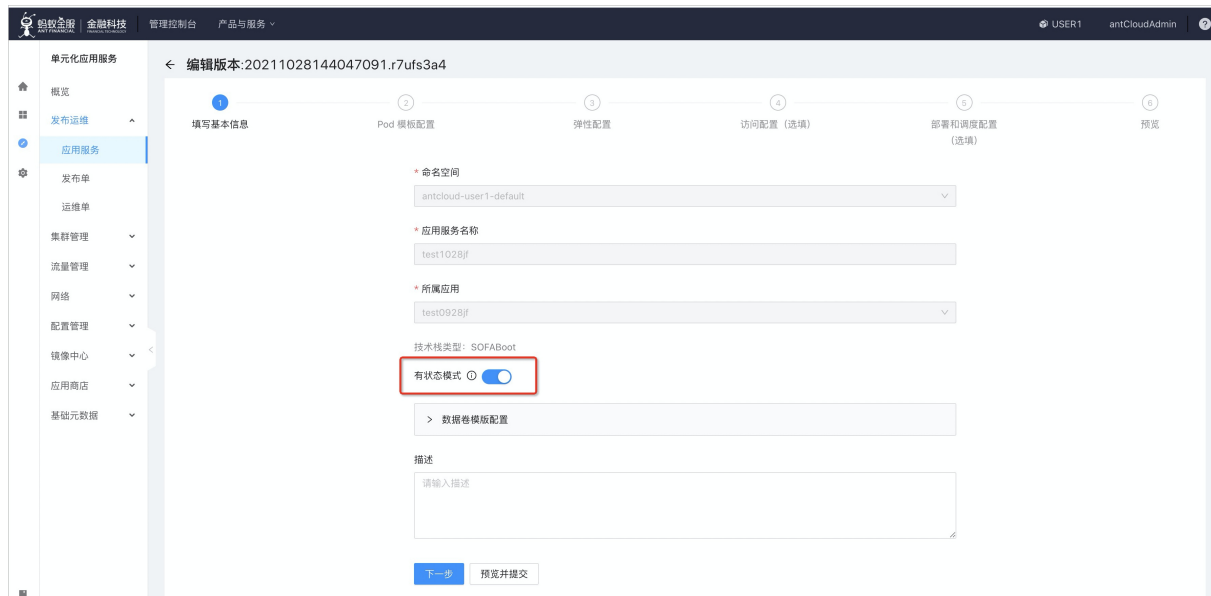
# 14.最佳实践

## 14.1. LHC Pod 域名配置

本文介绍如何通过域名访问 Pod。

### 步骤一：开启有状态模式

在创建应用服务或者编辑应用服务的 [填写基本信息](#) 阶段，开启 [有状态模式](#)。



开启后，Pod 名称会加上 `-0`、`-1` ... 数字后缀。待应用服务发布成功后，可前往应用服务详情页查看 Pod 名称是否已加上 `-0`、`-1` ... 数字后缀。



### 步骤二：配置 Headless Service

在创建应用服务或者编辑应用服务的 [访问配置](#) 阶段，添加 Headless Service 名称。本例中输入 `svc`。Headless Service 名称即为 `test0928jf-svc`。

### 步骤三：基于域名规则计算 Pod 域名

域名规则如下：

- Service 命名规则：<Headless Service 名称>-cell-<部署单元名>  
示例：test0928jf-svc-cell-rz00a
- Hostname 命名规则：<应用服务名>-<部署单元名>-<Pod number>

#### ? 说明

- Pod Number：Cell 维度从 0 开始递增的数字。
- 应用服务发布完成后，可前往应用服务详情页查看 Pod 名称，Pod 名称最后一位即为 Pod Number。

示例：test0928jf-rz00a-0

- Pod 域名规则：<hostname>.<service name>  
示例：test0928jf-rz00a-0.test0928jf-svc-cell-rz00a

Pod 域名规则		
< hostname > . < service name >		
示例：test0928jf-rz00a-0.test0928jf-svc-cell-rz00a		
Hostname 命名规则		Service 命名规则
< 应用服务名 >-< 部署单元名 >-< Pod number >		< Headless Service 名称 >-cell-< 部署单元名 >
示例：test0928jf-rz00a-0		示例：test0928jf-svc-cell-rz00a

假设部署单元 rz00a 有 2 个副本，基于上述规则，计算结果如下：

计算项	计算结果
Service	test0928jf-svc-cell-rz00a
Hostname	test0928jf-rz00a-0 test0928jf-rz00a-1
Pod 域名	test0928jf-rz00a-0.test0928jf-svc-cell-rz00a test0928jf-rz00a-1.test0928jf-svc-cell-rz00a

通过 Ping 命令验证能否访问 Service: test0928jf-svc-cell-rz00a。

```
sh-4.2# ping test0928jf-cell-rz00a
PING test0928jf-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183) 56(84) bytes of data.
64 bytes from test0928jf-cell-rz00a-66tq5-1.test0928jf-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183): icmp_seq=1 ttl=62 time=0.219 ms
64 bytes from test0928jf-cell-rz00a-66tq5-1.test0928jf-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183): icmp_seq=2 ttl=62 time=0.201 ms
^C
```

通过 Ping 命令验证能否访问 Pod 域名: test0928jf-rz00a-0.test0928jf-svc-cell-rz00a、test0928jf-rz00a-1.test0928jf-svc-cell-rz00a。

```
sh-4.2# ping test0928jf-rz00a-0.test0928jf-svc-cell-rz00a
PING test0928jf-rz00a-0.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.184) 56(84) bytes of data.
64 bytes from test0928jf-rz00a-0.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.184): icmp_seq=1 ttl=62 time=0.219 ms
64 bytes from test0928jf-rz00a-0.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.184): icmp_seq=2 ttl=62 time=0.201 ms
^C
--- test0928jf-rz00a-0.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.201/0.210/0.219/0.009 ms
sh-4.2# ping test0928jf-rz00a-1.test0928jf-svc-cell-rz00a
PING test0928jf-rz00a-1.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183) 56(84) bytes of data.
64 bytes from test0928jf-rz00a-1.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183): icmp_seq=1 ttl=62 time=0.253 ms
64 bytes from test0928jf-rz00a-1.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183): icmp_seq=2 ttl=62 time=0.216 ms
64 bytes from test0928jf-rz00a-1.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local (10.110.3.183): icmp_seq=3 ttl=62 time=0.214 ms
^C
--- test0928jf-rz00a-1.test0928jf-svc-cell-rz00a.antcloud-user1-default.svc.cluster.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.214/0.227/0.253/0.025 ms
```

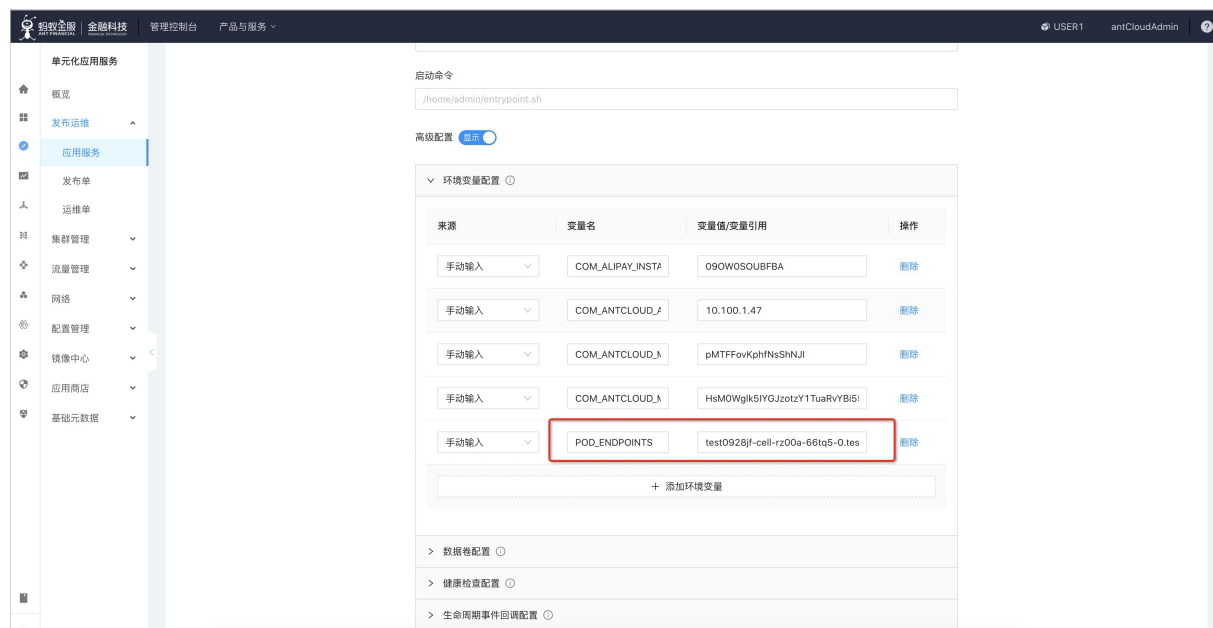
#### 注意

Pod 域名只能在集群内部访问，如果应用服务发布的部署单元属于不同的集群，则不同的部署单元之间通过 Pod 域名无法访问。

## 步骤四：添加环境变量 POD\_ENDPOINTS = <Pod 域名>

在编辑应用服务的 **Pod 模板配置** 阶段，开启 **覆盖配置** 后，新建环境变量，将上面的域名列表写入环境变量，变量名支持自定义：

```
POD_ENDPOINTS= test0928jf-rz00a-0.test0928jf-svc-cell-rz00a,test0928jf-rz00a-1.test0928jf-svc-cell-rz00a
```



待应用服务发布后，即可从环境变量中获取所有 Pod 的域名列表。环境变量名可以按需定义。