



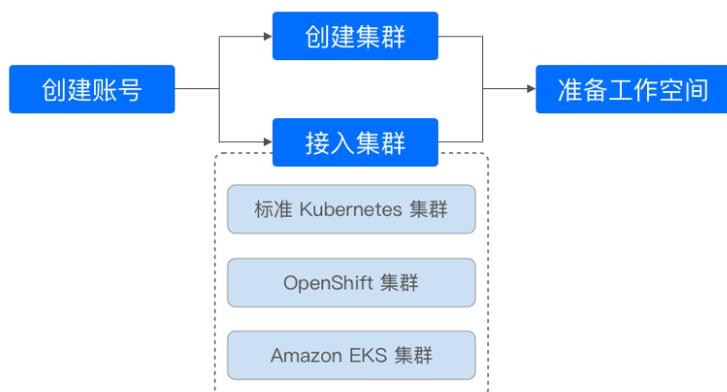
# 全栈云原生开放平台 快速入门 v3.4

# 快速入门

当您首次使用平台中心时，您可参考本文档快速体验平台中心。

# 入门概述

当您首次使用平台中心时，您可参照下图所示流程，快速创建平台账号、在平台上搭建集群并创建一个工作空间。



## 常见问题

以下为平台中心相关的常见问题，帮助您更顺畅地体验平台中心。

- Q：我的账号需要拥有哪些权限才可以完整地体验平台中心功能？

您的账号需要已绑定系统自定义角色 **平台管理员**，或已绑定拥有以下权限的自定义角色：

- 平台中心相关视图（平台管理、项目管理、运维中心）的查看权限
- 平台管理、项目管理、运维中心所有功能的查看、创建、更新、删除权限

- Q：如何查看我的账号拥有哪些权限？

平台的用户权限基于 RBAC（Role-Based Access Control，角色的权限访问控制）实现，通过查看您的账号拥有的角色的权限，即可得知您的账号所拥有的权限。

- 使用账号登录平台后，单击界面右上角的 **账号名称** > **个人信息** 进入个人信息页面。
- 在角色区域，可查看当前账号拥有的角色列表，请记录角色的名称。
- 单击界面左上方的 ，在展开的下拉框中选择 **平台管理**，进入平台管理视图。
- 在左侧导航栏中单击 **用户角色管理** > **角色管理**，进入角色列表页面。
- 找到账号拥有的角色。单击 **角色名称**，进入角色详情页并下拉至 **权限** 区域，即可查看角色拥有的权限。

- Q：如何给账号授权？

可通过以下几种方式，为平台上已有的账号授权：

- 为用户添加角色：通过为已有账号添加平台上已有的角色，将角色具有的操作权限授予用户账号；
  - 为角色导入成员：通过为平台上已有的角色导入成员，将角色具有的操作权限授予成员的用户账号；
  - 为项目/命名空间添加/导入成员：通过为平台上已有的项目/命名空间导入平台已有用户的方式，或为项目/命名空间添加用户（仅当平台通过 IDP 配置对接了 OIDC 时可用）的方式，可为用户授予项目/命名空间相关角色的操作权限。例如：导入项目成员、添加项目成员。
- Q：系统预设的角色有哪些？

角色名称	说明	角色级别
平台管理员	具备平台所有业务及资源的全部权限。	平台级别

角色名称	说明	角色级别
平台审计人员	负责整个平台的审计工作，可查看平台所有资源与操作记录，没有查看外的其他操作权限。	平台级别
项目管理员	负责绑定、解绑命名空间管理员，并管理命名空间的配额。	项目级别
命名空间管理员	负责给命名空间添加成员并设置角色。	命名空间级别
开发人员	负责在该命名空间下开发、部署、维护自定义应用。	命名空间级别

■ Q：平台支持纳管已有的外部集群吗？

平台支持接入已有的 标准 Kubernetes 集群、OpenShift 集群、Amazon EKS 集群，并纳管集群下已有的 Kubernetes 资源。详情请参考 纳管 OpenShift 集群。

■ Q：平台支持对接企业已有的用户体系吗？

除了支持创建本地用户外，平台支持通过配置 IDP (Identity Provider)，对接企业已有的 LDAP (Lightweight Directory Access Protocol, 轻量级目录访问协议)，导入企业已有的用户体系；或通过 OIDC (OpenId Connect) 协议，使用平台认可的第三方账号登录平台。详情请参考 IDP 配置。

同时，可通过解决方案对接 Azure Active Directory (Azure AD)。详情请参考 如何对接客户的 Azure Active Directory？

■ Q：平台的运维中心能提供哪些智能运维体验？

平台的运维中心，基于可视化界面为您实时地展示平台上各类资源的监控数据、日志和事件信息。同时，可结合监控数据、日志和事件为资源配置告警规则。告警被触发时能够将告警信息通过消息发送平台及时地推送给您，可以有效地帮助您避免平台故障，降低业务风险，提升平台的运维效率。

# 创建账号

平台的用户管理、角色管理功能，基于 RBAC（Role-Based Access Control，角色的权限访问控制）实现。支持平台级别、项目级别和命名空间级别的权限控制，帮助您便捷地设置租户下的用户的操作权限。

平台基于企业常用的权限使用场景预设了 5 个系统角色，同时也支持平台管理员根据实际使用场景自定义角色。将权限和角色进行关联后，通过为不同的用户分配角色，即可将权限赋予用户；同理，通过删除用户已拥有的角色，即可实现权限的回收。

提示：为方便您在工作空间创建完成后，为用户设置工作空间的访问权限，本文仅向您说明如何快速创建一个或多个仅能登录平台但暂无功能操作权限的账号。待工作空间创建完成后，可通过导入成员的方式为创建的账号绑定工作空间的角色，授予相应的操作权限。

## 操作步骤

1. 使用平台管理员账号登录平台后，切换至 **平台中心** 的 **平台管理** 视图。
2. 在左侧导航栏中单击 **用户角色管理 > 用户管理**，进入用户列表页面。
3. 单击 **创建用户** 按钮，进入创建用户页面。请参照以下说明，配置用户相关参数。

参数	是否必填	说明
用户名	是	用户名，登录平台的账号。 支持输入 <b>a-z</b> 、 <b>0-9</b> 、 <b>-</b> ，以 <b>a-z</b> 开头，长度不超过 253 个字符。
显示名称	是	用户的显示名称。长度不超过 64 个字符。
密码类型	是	单击选择一种类型。 <b>随机</b> ：会在密码输入框中随机生成一个安全系数较高的新密码。 <b>自定义</b> ：需要用户在密码输入框中，输入密码。
密码	是	根据密码类型，随机生成密码或输入密码。 输入密码时，密码必须同时满足以下规则： 长度 8~32 位； 包含英文字母和数字； 包含特殊符号，特殊符号支持 <b>~!@#\$%^&amp;*() -_+=?</b> 。  单击密码输入框右侧的  图标，可显示密码，再次单击可隐藏密码； 单击密码输入框右侧的  图标，可快速复制密码。
邮箱	否	用户的电子邮箱地址，和用户名关联，也可以作为登录平台的账号使用。
角色	否	暂不配置该参数。

4. 单击 **创建** 按钮，创建本地用户。

# 准备业务集群

集群是容器运行所需要的资源合集，包含节点、网络、存储等资源，是您在平台上搭建业务环境（Namespace）运行您的自定义应用的先决条件。

您可以通过平台创建一个新的 Kubernetes 集群，或将已有的标准 Kubernetes 集群、OpenShift 集群、Amazon EKS 集群接入平台进行托管。

本文将说明如何在平台上创建一个集群，如需接入集群，请参考 [接入集群](#)。

## 前提条件

- 已准备好待添加至集群的节点。如需为集群配置 GPU 参数，且控制节点会作为 GPU 节点使用，需要先行为 GPU 节点安装 GPU 驱动，请参考 [GPU 节点安装 GPU 驱动](#)。
- 集群的控制节点和网络需根据实际业务和网络情况提前规划好，否则集群部署完成后将无法调整。
- 请确保集群节点防火墙、selinux、swap 分区已关闭，否则可能会导致集群部署失败。

## 操作步骤

- 使用平台管理员账号登录平台后，切换至 [平台中心](#) 的 [平台管理](#) 视图。
- 在左侧导航栏中单击 [集群管理](#) > [集群](#)，进入集群列表页面。
- 单击 [创建集群](#) 按钮，进入创建集群页面。
- 参照以下说明，配置集群基本信息参数。

参数	是否必填	说明
名称	是	集群的名称。 支持输入 <code>a-z</code> 、 <code>0-9</code> 、 <code>-</code> ，以 <code>a-z</code> 、 <code>0-9</code> 开头或结尾，且总长度不超过 32 个字符。
显示名称	是	集群的显示名称。
GPU	否	是否为集群部署 GPU 资源的滑块开关，默认为关闭，单击后开启。当开关开启时，可通过单击下拉选择框，可选择： <b>vGPU</b> ：虚拟化 GPU 核心，100 个单位的虚拟核心等于 1 个物理核心；1 个单位的显存为 256Mi。 <b>pGPU</b> ：物理 GPU，只能整颗分配。
集群地址	是	集群对外暴露 API Server 的访问地址，使用 https 协议。 <b>IP 地址/域名</b> ： 当集群只有一个控制节点（master）时，输入 master 节点的 <b>私有 IP</b> 或域名。 当集群有 2 个或 2 个以上控制节点（master）时，需要输入 <b>VIP</b> 或域名。 <b>端口</b> ：端口号。 <b>说明</b> ：有负载均衡时，请输入负载均衡的端口，没有负载均衡时，必须输入 <code>6443</code> 。
节点名称设置	是	支持使用节点 IP 或主机名称（Hostname）作为平台上节点的名称，可通过单击复选框进行选择，默认使用节点 IP 作为节点名称。 <b>注意</b> ：当选择使用主机名称作为平台上节点名称时，需确保添加至集群的节点的主机名称唯一。
节点信息	是	参见 <a href="#">配置节点信息</a> 。
监控组件部署节点	否	选择为业务集群部署监控组件的节点，最多支持选择 3 个。集群创建成功后，会为集群部署存储类型为 <b>本地卷</b> 的监控组件。 <b>提示</b> ：如果创建集群时选择不配置该参数（暂不部署监控组件），当集群创建成功且状态为正常时，可前往集群的服务组件页面为集群部署监控组件，此时，存储类型可选 <b>本地卷</b> 、 <b>存储类</b> 或 <b>持久卷</b> ，具体操作参见 <a href="#">为业务集群部署监控组件</a> 。 <b>说明</b> ：为避免影响集群性能，建议优先选择计算节点；且当控制节点不支持部署自定义应用时，也不支持部署监控组件。
网卡	是	网卡设备的名称，默认为 eth0。

参数	是否必填	说明
网络模式	否	<p>集群的网络模式，默认为 <b>Kube-OVN</b>。支持选择：</p> <p><b>Kube-OVN</b>：该模式下不同 Namespace 对应着不同的子网，同一台机器的不同 Pod 可以使用不同的子网，多个子网之间网络可以联通。在容器访问外部网络的策略中，Kube-OVN 设计了两种方案：一种是分布式网关，每台主机都可以作为当前主机上 Pod 的出网节点，做到出网的分布式；一种是和 Namespace 绑定的集中式网关，可以做到一个 Namespace 下的 Pod 流量使用一个集中式的主机出网。</p> <p><b>Calico</b>：Calico 是一种可为容器提供安全网络连接的三层网络方案，具有简单、易扩展、安全性高、性能好等优势。</p> <p><b>Flannel</b>：为集群内所有容器提供一个扁平化的网络环境。所有容器在 Flannel 提供的网络平面上可以看作是在同一网段，可自由通信。其模型让集群上全部的容器使用一个网络，从网络中为每个节点划分一个子网。节点上的容器创建网络时，从子网中划分一个 IP 给容器，提高了容器之间通信效率，不用考虑 IP 转换问题。</p> <p><b>Galaxy-FloatingIP</b>：Galaxy 是由腾讯提供的 CNI 框架，实现了 Overlay、Floating-IP、NAT、Host 四种网络解决方案，可以动态地为容器配置 Underlay/Overlay 网络以及端口映射。</p> <p><b>无网络</b>：如需安装其他网络插件，请选择 <b>无网络</b> 模式，集群创建成功后可手动安装。</p>
FloatingIP 参数	否	<p>当集群的 <b>网络模式</b> 为 <b>Galaxy-FloatingIP</b> 时，该参数有效，支持配置 FloatingIP 高级参数。该参数用于配置能够被 Pod 使用的 IP，以及 Pod IP 与节点的拓扑关系（即 Pod IP 能在哪些节点上使用）。</p> <div data-bbox="284 678 1177 1205" data-label="Form"> </div> <p>节点子网：是节点的子网网段，仅支持一个节点对应一个节点子网网段。支持输入 IP 或子网掩码，例如：<b>10.2.0.0/24</b>。支持添加多个节点子网网段。</p> <p>IP 地址范围：Pod IP 的地址范围，支持输入 IP 地址或 IP 段，IP 段的输入格式为：<b>10.0.0.2~10.0.0.182</b>。</p> <p>子网：Pod IP 的网段，支持输入 IP 或子网掩码，例如：<b>10.0.0.0/24</b>。</p> <p>网关：Pod IP 的网关 IP 地址，例如：<b>10.0.0.1</b>。</p> <p>（可选）VLAN 号：VLAN 号，当子网与节点子网 VLAN 不同时，需要配置。支持输入正整数，例如：<b>2</b>。</p> <p>详细的配置示例请参考 FloatingIP 参数配置示例。</p>
容器网络	是	<p>集群的容器网络（CNI）的配置信息。</p> <p><b>Cluster CIDR</b>：容器网络的子网地址（CIDR 格式），例如：<b>10.1.0.0/16</b>，该地址应与节点子网不同。集群创建时的 Pod 可使用该网段内的 IP。</p> <p><b>Service CIDR</b>：Kubernetes Service 的网络地址范围（CIDR 格式），例如：<b>10.96.0.0/12</b>，不可与 <b>Cluster CIDR</b> 的网段重叠。</p> <p>（可选）节点 IP 个数：仅当集群网络模式为 <b>Flannel</b>、<b>Galaxy-FloatingIP</b> 时，支持设置该参数。单击下拉选择框，选择当前容器网络下允许在每个节点上分配给容器组的最大 IP 个数。可选：128、256，默认为 128。</p> <p>说明：会根据以上配置，自动计算集群最多可容纳的节点个数，并在输入框下方的提示内容中展示。</p>

## 配置节点信息

1. 单击 **添加节点** 按钮，在弹出的 **添加节点** 对话框中，配置节点信息。

说明：

- 添加节点 对话框关闭后，支持单击  图标修改已添加的节点信息；单击  图标删除已添加节点信息。再次单击 **添加节点** 按钮，可在新的对话框中配置节点信息；
- 在同一个 **添加节点** 对话框中的添加的节点的 SSH 端口以及认证信息需要保证统一。

* 主机信息:		
SSH 端口: 22	认证方式: 密码	用户名: admin@cpaas.io
控制节点: 10.168.10.3 (master1测试)	GPU 主机: 否	可部署应用: 是
计算节点: 10.168.10.4 (node1)	GPU 主机: 是	-
控制节点: 10.168.10.2 (master2)	GPU 主机: 否	可部署应用: 否
SSH 端口: 20		
认证方式: 密码	用户名: admin@cpaas.io	
控制节点: 10.168.10.6	GPU 主机: 是	可部署应用: 否
控制节点: 10.168.10.8	GPU 主机: 否	可部署应用: 是
控制节点: 10.168.10.0	GPU 主机: 否	可部署应用: 否

+ 添加主机

2. 单击 添加 按钮，可添加一条节点配置记录，支持添加多条。

提示：单击  图标，可移除一条节点配置记录。

### 添加主机 ✕

**1.** 控制节点至少添加1个，不支持设置为2个，大于等于3个时为高可用集群（高可用集群，建议设置为奇数个，推荐3或5）；

**2.** 添加控制节点时，若开启“可部署应用”，该节点可部署业务应用；

**3.** 添加控制节点或计算节点时，若开启“GPU 主机”，需要手动安装 GPU 驱动和 runtime；

* 主机:	* 主机类型	* 主机 IP	显示名称	GPU 主机	可部署应用	
	控制节点	10.168.10.3	master1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
	计算节点	10.168.10.4	node1	<input checked="" type="checkbox"/>	-	
	控制节点	10.168.10.2	master2	<input type="checkbox"/>	<input type="checkbox"/>	

+ 添加

\* SSH 端口:

认证方式:  密码  密钥

\* 用户名:

\* 密码:

说明：

- 控制节点至少添加 1 个，不支持设置为 2 个，大于等于 3 个时为高可用集群（高可用集群，建议设置为奇数个，推荐 3 或 5）；
- 添加控制节点时，若开启 可部署自定义应用 开关，该节点可部署业务自定义应用；
- 添加控制节点或计算节点时，若开启 GPU 节点，需要手动安装 GPU 驱动 和 runtime。

参数	是否必填	说明
节点类型	是	节点的类型，包括 控制节点 (master) 和 计算节点 (node) 。 单击 节点类型 下拉选择框，选择节点类型。 控制节点：控制节点负责运行集群中的 kube-apiserver、kube-scheduler、kube-controller-manager、etcd 和容器网络以及平台的部分管理组件； 计算节点：计算节点是 Kubernetes 集群中承担业务运行计算组件的节点，可以是虚拟机也可以是物理机。计算节点承担实际的 Pod 调度以及与管理节点的通信等。 注意：如果集群中没有计算节点，集群仅能运行平台组件和少量业务，无法正常运行生产业务。集群创建完成后，可通过 添加节点 为集群添加计算节点。
节点 IP	是	在 IP 地址 输入框中，输入节点的 IP 地址。处于内网模式创建集群，请输入节点的 私有 IP。  注意：一台节点只能属于一个集群，待添加节点不能被其他集群占用。否则，无法通过 Kubernetes 校验，进而导致集群创建失败。

参数	是否必填	说明
显示名称	否	节点的显示名称。
GPU 节点	否	控制是否在该节点上部署 GPU 相关组件的滑块开关，通过单击可控制开启或关闭。开关显示为  时为关闭状态，单击后开启；开关显示为  时为开启状态，单击可关闭。
可部署自定义应用	否	当节点类型为控制节点时该参数有效。控制是否在该控制节点上部署业务自定义应用的滑块开关，通过单击可控制开启或关闭。开关显示为  时为关闭状态，单击后开启；开关显示为  时为开启状态，单击可关闭。
SSH 端口	是	节点的 SSH 协议端口号，例如： <code>22</code> 。
认证方式	是	登录已添加节点的认证方式及对应的认证信息。可选择： 密码：需要输入具有 root 权限的用户名及对应的密码。 密钥：需要输入具有 root 权限的用户名、私钥以及私钥密码。

## 后续操作

### 查看执行进度

在集群列表页面，可查看已创建的集群的列表信息，其中，处在 **创建中** 状态的集群，可查看执行进度。

### 操作步骤

- 单击处在 **创建中** 状态的集群右侧的  图标 **查看执行进度**。
- 在弹出的执行进度对话框中，可查看集群的执行进度（status.conditions）。

提示：当某个类型正在执行中或状态为失败且有原因时，光标悬停在对应的原因（显示蓝色字体）上时，可查看原因（status.conditions.reason）的详细信息。

## 参考信息

### FloatingIP 参数配置示例

#### ■ 示例 1

IP 地址范围 10.0.0.2-10.0.0.182、子网 10.0.0.0/24 只能配置在节点子网为 10.2.0.0/24 的机器上。；且 IP 地址范围 10.1.0.2-10.1.0.254、子网 10.1.0.0/24 只能配置在节点子网为 10.2.1.0/24 的机器上。配置信息如下：

```
[{
  "nodeSubnets": ["10.2.0.0/24"],    ## 节点子网
  "ips": ["10.0.0.2~10.0.0.182"],    ## IP 地址范围
  "subnet": "10.0.0.0/24",    ## 子网
  "gateway": "10.0.0.1"    ## 网关
}, {
  "nodeSubnets": ["10.2.1.0/24"],
  "ips": ["10.1.0.2~10.1.0.254"],
  "subnet": "10.1.0.0/24",
  "gateway": "10.1.0.1"
}]
```

#### ■ 示例 2

两个 IP 地址范围在集群下所有的机器都能用（假设集群的所有机器只有 10.2.0.0/24、10.2.1.0/24 两个网段），配置信息如下：

```
[{
  "nodeSubnets": ["10.2.0.0/24","10.2.1.0/24"], ## 节点子网
  "ips": ["10.0.0.2~10.0.0.182"], ## IP 地址范围
  "subnet": "10.0.0.0/24", ## 子网
  "gateway": "10.0.0.1" ## 网关
}, {
  "nodeSubnets": ["10.2.0.0/24","10.2.1.0/24"],
  "ips": ["10.1.0.2~10.1.0.254"],
  "subnet": "10.1.0.0/24",
  "gateway": "10.1.0.1"
}]
```

# 准备工作空间

本文主要向您介绍通过创建项目、命名空间，并为用户分配项目、命名空间的操作权限，从而根据实际的租户场景创建出资源和人员均隔离的工作空间。

项目是平台为了满足企业多租户的使用场景，基于平台上一个或多个集群的资源配额，划分出的可以实现资源和人员相互隔离的项目环境，能够代表企业中不同的子公司、部门或项目组。通过项目管理，能够轻松实现项目组之间的资源隔离、租户内的配额管理。

平台的命名空间是项目下相互隔离的更小的资源空间，也是用户实现作业生产的工作区间。一个项目下可以创建多个命名空间，可占用的资源配额总和不能超过项目配额。命名空间更细粒度的划分了资源配额的同时，还限制了命名空间下容器的大小（CPU、内存），有效的提升了资源利用率。

请参考如下流程，准备工作空间。

1. 创建项目并导入成员
2. 创建命名空间并导入成员

提示：当您还订阅了 Container Platform 的其他子产品（Container Platform、DevOps、Service Mesh）时，项目和命名空间创建成功后，即可通过产品切换入口，前往子产品界面体验自定义应用、流水线、服务网格等功能。

## 创建项目并导入成员

提示：

- 新的项目的资源配额，可以来源于平台上的多个集群包括联邦集群。
- 可结合项目的使用规模和集群的资源使用情况，为项目合理分配资源配额。

1. 登录平台后，切换至 **平台中心** 的 **项目管理** 视图。
2. 单击 **创建项目** 按钮，进入创建项目页面。
3. 在 **基本信息** 页面，参照以下说明配置相关参数。

参数	是否必填	说明
名称	是	项目的名称，不可与已有项目同名，且不可与项目名称黑名单列表中的名称同名，否则不允许创建项目。 支持输入 <b>a-z</b> 、 <b>0-9</b> 、 <b>-</b> 、 <b>.</b> ，并且以 <b>a-z</b> 、 <b>0-9</b> 开头或结尾，长度不超过 32 字符。 说明：项目名称黑名单中的名称为平台集群下特殊命名空间的名称，默认包括：cpaas-system、cert-manager、default、global-credentials、kube-ovn、kube-public、kube-system、nsx-system、alauda-system、kube-federation-system、ALL-ALL、true。
显示名称	否	项目的显示名称。
描述	否	项目的描述信息。
所属集群	是	项目关联的集群，管理员可以为项目分配所选集群的资源配额。单击下拉选择框，支持选择一个或多个集群。 在下拉选项中，非联邦集群名称下方将显示集群资源的可分配配额和总额信息，格式如： <b>可分配 CPU 配额 (CPU 总额)</b> ；联邦集群下方将显示联邦集群包含的成员名称。  说明： - 当普通集群异常时，普通集群不可选； - 可以选择联邦集群的某个或多个成员作为所属集群，也可选择整个联邦集群作为所属集群。且存在以下几种限制： * 当选择成员时，成员所在的联邦集群不可选； * 当选择联邦集群时，联邦下的成员不可选； * 当联邦集群的控制集群异常时，联邦集群不可选；非控制集群异常时，联邦集群可选。  - 当联邦集群解除联邦后，项目和原联邦成员的关联关系（配额信息）不变，不影响项目下已有资源的使用。

- 单击 **下一步** 按钮，在设置项目配额环节，依次为已选择的集群，设置在当前创建项目中的资源配额。支持设置 CPU（核）、内存（Gi）、存储（Gi）、PVC 数（个）、Pods（个）、虚拟 GPU/物理 GPU、显存的资源配额。

说明：

- 仅当集群中部署了 GPU（虚拟 GPU/物理 GPU）资源时，支持配置 GPU 资源配额。  
当 GPU 资源为 **虚拟 GPU** 资源时，同时支持配置 **显存** 配额；当 GPU 资源为 **物理 GPU** 资源时，仅需配置 **物理 GPU** 配额。  
**GPU 单位**：100 个单位的虚拟核心等于 1 个物理核心；1 个单位的显存为 256 Mi；物理 GPU 的单位为颗，只能整颗分配。
- 如果未输入某项资源的配额，默认为该资源无限制配额。
- 设置的项目配额数值应在页面提示的配额范围之内。在每项资源配额的输入框下方，将会展示该项资源的已分配配额和总额信息以供参考。
- 当集群为 **联邦集群** 时，可选择 **配额设置方式** **为联邦集群下所有集群设置相同配额**。当勾选该单选框时，即可统一设置联邦成员分配给当前项目的配额；不勾选该单选框时，则可分别设置联邦成员分配给当前项目的配额。
- 当集群为 **联邦集群** 时，当联邦成员的配额参数不一致（例如：个别成员可配置 GPU 相关参数）时，**配额设置方式** **为联邦集群下所有集群设置相同配额** 不可勾选。

- 设置完成后单击 **创建项目**，项目创建成功后进入项目概览页面。

- 在左侧导航栏中单击 **项目成员**，进入项目成员列表页面。

- 单击 **导入成员** 按钮，弹出 **导入成员** 对话框。

- 单击用户名前的选择框，选择一个或多个用户。

提示：可通过对话框右上方的用户组下拉选择框选择用户组，并在用户名搜索框中输入用户名对用户进行模糊搜索。

- 单击对话框底部 **设置角色** 项右侧的下拉选择框，选择名称为项目管理员的角色。

- 单击 **导入** 按钮，将成员用户导入项目并授予项目管理相关权限。

## 创建命名空间并导入成员

- 进入项目概览页面后，再左侧导航栏中单击 **命名空间 > 命名空间** 选项，进入命名空间列表页面。

- 单击 **创建命名空间** 按钮，进入 **创建命名空间** 页面。

- 参见以下参数说明，配置命名空间的 **基本信息**。

- **所属集群**：单击下拉选择框，在项目已关联的集群中，选择一个要创建命名空间的集群。
- **命名空间**：输入命名空间的名称，名称的固定前缀为项目名称，支持输入 **a-z**、**0-9**、**-**，并且以 **a-z**、**0-9** 开头或结尾，前缀及输入字符的总体长度不超过 63 字符。
- **(非必填) 显示名称**：输入命名空间的显示名称。

- 在 **资源配额** 区域，配置命名空间的 **资源配额**（ResourceQuota）。

支持配置 CPU（核）、内存（Gi）、存储（Gi）、Pod（个）、PVC 数（个）、虚拟 GPU/物理 GPU、显存的配额。创建容器实例时，将依据请求值，请求占用相关的计算或存储资源配额。

说明：

- 命名空间的资源配额，是集群项目配额的一部分。对于某一种资源，可设置的最高配额为在项目中的未使用的资源配额，如果任意一种资源的可用量为 0，将无法继续创建命名空间，请联系平台管理员获得支持。
- 仅当集群中部署了 GPU（虚拟 GPU/物理 GPU）资源时，支持配置 GPU 资源配额。  
当 GPU 资源为 **虚拟 GPU** 资源时，同时支持配置 **显存** 配额；当 GPU 资源为 **物理 GPU** 资源时，仅需配置 **物理 GPU** 配额。

**GPU 单位**：100 个单位的虚拟核心等于 1 个物理核心；1 个单位的显存为 256 Mi；物理 GPU 的单位为颗，只能整颗分配。

5. 在 **容器限额** 区域，参照以下参数说明，配置 **容器限额** (LimitRange)。

通过设置容器限额，可以控制在当前命名空间下创建的容器的 **CPU/内存** 大小。在命名空间下创建计算组件时，单一容器的 CPU、内存大小不可超过最大值。

- **最大值**：命名空间下单一容器可使用的资源 (CPU、内存) 的最大限制值 (limits)。在该命名空间下创建计算组件时，容器大小的限制值 (limits，界面显示为 **资源限制**)，界面输入时不可超过所在命名空间的 **最大值**。  
注意：如果通过 YAML 编排文件设置了大于该 **最大值** 的值，计算组件将创建失败。
- **默认值**：命名空间下单一容器可使用的资源 (CPU、内存) 的默认限制值 (limits)。在该命名空间下创建计算组件时，容器大小的限制值 (limits，界面显示为 **资源限制**) 默认为命名空间的 **默认值**，可修改，修改后的值不可超过 **最大值**。
- **requests** (请求值)：命名空间设定的容器对资源 (CPU、内存) 的请求值，(requests) 默认为 **limits/集群超售比**。在该命名空间下创建计算组件时，容器大小的请求值 (request，界面显示为 **资源限制**) 默认为命名空间的 **默认值**，可通过编辑计算组件的 YAML 编排文件修改，修改后的值不可超过 **最大值**，否则会导致计算组件创建失败。  
requests 值供调度器调度容器时参考和判断。调度器会检测每个节点可用于分配的资源 (资源总额 - 已调度到节点上 Pod 内的容器的 requests 之和)，当新建的 Pod 内的容器的 requests 值的总和，超过了节点的剩余可用资源，该 Pod 将不会被调度到此节点上。

6. 单击 **创建** 按钮，命名空间创建成功后进入命名空间的详情页面。

7. 单击 **命名空间成员** 页签，进入命名空间成员列表页面。

8. 单击 **导入成员** 按钮，弹出 **导入成员** 对话框。

9. 单击用户名前的选择框，选择一个或多个用户。

提示：可通过对话框右上方的用户组下拉选择框选择用户组，并在用户名搜索框中输入用户名对用户进行模糊搜索。

0. 单击对话框底部 **设置角色** 项右侧的下拉选择框，选择要分配的角色名称 (例如：命名空间管理员、开发人员)。

1. 单击 **导入** 按钮，将成员用户导入命名空间并授予命名空间相关权限。

# 快速入门

本文旨在介绍如何快速使用容器平台，帮助您快速了解从快速创建应用到接入服务的基本使用流程。

## 1. 入门必读

在体验容器平台功能之前，完成必要的准备工作，参考 [入门必读](#)。

### 1. 快速创建自定义应用（通过镜像）

使用已上传到镜像仓库的镜像创建应用，参考 [快速创建自定义应用（通过镜像）](#)。

### 2. 快速创建模板应用（通过应用目录）

使用模板仓库中的应用模板创建模板应用，参考 [快速创建应用（通过应用市场）](#)。

### 3. 快速配置外部访问应用

创建内部路由并一键开启外网访问，参考 [快速配置外部访问应用](#)。

# 入门必读

在体验容器平台功能之前，管理员需要完成以下操作，方可在容器平台上创建应用。

通过镜像创建自定义应用，需满足以下前提条件：

1. 平台管理员已通过 DevOps 工具链集成制品仓库，并分配给相应项目使用，可参考 DevOps 文档。
2. 管理员已在镜像仓库中上传了可用的镜像。
3. 平台管理员已通过 平台中心 > 项目管理 创建项目及项目下的命名空间。
4. 创建自定义应用的人员拥有业务视图操作权限，例如：项目管理员角色、命名空间管理员角色、开发人员角色。

通过应用目录创建模板应用，需满足以下前提条件：

1. 平台可以正常访问远端存放模板的仓库。
2. 仓库中的模板可以正常使用。
3. 平台管理员已通过 平台中心 > 项目管理 创建项目及项目下的命名空间。
4. 平台管理员负责添加模板仓库。
5. 创建模板应用的人员拥有业务视图操作权限，例如：项目管理员角色、命名空间管理员角色、开发人员角色，负责使用仓库中的模板。

# 快速创建自定义应用（通过镜像）

在业务视图，创建自定义应用的人员进入要创建应用的命名空间，使用镜像仓库中的镜像，快速创建自定义应用。

## 操作步骤 - 选择镜像

1. 单击左导航栏中的 **应用管理** > **自定义应用**，再单击 **创建自定义应用**，通过镜像创建应用。
2. 在 **选择镜像** 窗口，选择镜像地址，并在右侧框中选择版本，单击 **创建** 按钮，进入创建应用页面。

## 操作步骤 - 创建应用

1. 在创建自定义应用页面，根据提示，依次填写 4 张卡片的内容，分别是应用、计算组件、内部路由和访问规则。单击 **更多**，可填写高级配置项。

- **应用**：名称 为必填项。
- **计算组件**：
  - **基本信息**：名称、部署模式、实例数量都有推荐的默认配置，可以手动修改；
  - **容器组**：均为非必填项，支持引用或关联存储卷声明、配置字典、保密字典等资源，但需要先创建对应的资源，可在创建应用后再配置。
  - **容器**：名称、镜像地址、资源限制都有推荐的默认配置，可以手动修改。
- **内部路由**：支持添加内部路由，添加的内部路由在“网络-内部路由”中可查看。都是非必填项，可创建自定义应用后再添加。
- **访问规则**：支持添加访问规则，添加的访问规则在“网络-访问规则”中可查看。都是非必填项，可创建自定义应用后再添加。

2. 根据提示填写完所有必填项后，单击 **创建**，并自动跳转至自定义应用的详情页，详情页包含以下页签。

- **详细信息**：可以查看自定义应用的运行状态，计算组件的运行状态。可以增减 Pod 实例数量，对应用执行更新、导出、删除等操作。
- **YAML**：查看 YAML 文件，不支持编辑。
- **容器组**：查看应用所用容器组的基本信息，也可在 **计算组件** > **容器组** 中查看。
- **拓扑**：展示应用组件的关系。
- **版本快照**：支持快速创建应用的快照，支持回滚和导出。
- **日志、事件、监控、告警**：可以查看应用的监控记录信息。

# 快速创建模板应用（通过应用目录）

在管理视图，平台管理员添加模板仓库。

在业务视图，创建模板应用的人员进入要创建应用的命名空间，使用应用目录中的应用模板，快速创建模板应用。

## 操作步骤 - 添加模板仓库

1. 平台管理员进入管理视图，单击左侧导航栏中的 **模板仓库**，进入模板仓库列表页，单击 **添加模板仓库** 按钮。
2. 在 **添加模板仓库** 窗口，配置模板仓库参数。选择一种模板仓库类型，并填写必填项后，单击 **确定** 按钮，即可添加成功。

添加模板仓库

\* 名称:

描述:

类型:  Chart  Git  SVN  Local ?

\* 仓库地址:

分配项目:

用户名:

密码:

**说明：**支持 Chart、Git、SVN 和 Local 这四种模板仓库类型。除了 Local 类型外，均需要填写远端仓库地址。

3. 支持更新、删除已添加的模板仓库，单击列表页的仓库名称，进入对应仓库的详情页。当远端仓库内的模板发生变化时，单击 **同步模板** 进行同步。
4. 平台管理员在管理视图添加模板仓库成功后，可将仓库分配给项目，创建模板应用的人员在业务视图进入对应项目，在应用目录即可查看并使用模板仓库中的模板。

## 操作步骤 - 通过应用目录创建模板应用

1. 创建模板应用的人员在业务视图，单击左导航栏中的 **应用目录**，进入应用目录页面。
2. 选择创建模板应用需要使用的模板，直接单击 **部署** 按钮。
3. 在部署页面，配置以下参数。

应用市场 / 模板应用 / 部署

基本信息

\* 名称:

显示名称:

应用模板: haproxy.public-charts

模板版本: 0.1.5

同步创建应用:  ?

参数配置

replicaCount:

键	值
pullPolicy	IfNotPresent
registry	harbor.alauda.cn

- **基本信息**：名称为必填项。支持选择模板的版本，以及是否同步创建自定义应用。如选择不同步创建应用，则仅在 **应用管理 > 模板应用** 列表页中，可以查看此应用；如选择同步创建自定义应用，则在 **应用管理 > 自定义应用** 列表页中，也可以同步查看到此应用。

- **参数配置**：根据实际情况修改。

配置完成后，单击 **部署**。

4. 单击左导航栏中的 **应用管理 > 模板应用**，在模板应用列表页面，单击模板应用名称，进入模板应用详情页，详情页包含以下页签。

- **详细信息**：查看应用的基本信息，支持更新、删除应用。

- **参数配置**：查看应用的 YAML 配置，不可编辑。

# 快速配置外部访问应用

通过创建内部路由，并一键开启外网访问，快速配置外部访问自定义应用。

## 前提条件

- 要访问的自定义应用状态正常。
- 拥有业务视图操作权限的人员可执行该操作，例如：项目管理员角色、命名空间管理员角色、开发人员角色。

## 操作步骤

1. 在业务视图，单击左导航栏中的 **网络 > 内部路由**，单击 **创建内部路由**。
2. 在创建内部路由页面，配置以下参数。
  - **名称**：必填项。
  - **外网访问**：一键开启。
  - **目标组件**：选择 **计算组件**。
  - **计算组件类型**：选择关联的计算组件类型，并在 **计算组件名称** 框中，选择要关联的计算组件。
  - **端口**：选择访问应用使用的协议，例如 HTTP，使用有效的服务端口，服务端口与容器端口自动映射，无需多余配置。
3. 创建成功后跳转至内部路由详情页，查看分配的主机端口。
4. 以“主机 IP+ 主机端口”的形式访问应用，主机 IP 可以使用集群节点的任意一台状态正常的主机 IP。  
说明：在平台中心，进入 **集群管理 > 集群**，在节点列表中查看主机 IP。

# 快速入门

DevOps 快速入门将平台的核心功能拆分出来形成一个快速使用指南，帮助您快速使用平台核心能力。

# 入门必读

本章节主要介绍 DevOps 快速入门的前期准备工作，并概述整体使用流程，在您开始之前，请仔细阅读以下内容：

- **适用对象**：初次接触平台的使用者。
- **资源准备**：以下为您介绍使用平台快速入门需要准备的资源。
  - **项目资源准备**：您已经在 **平台中心** 的 **项目管理** 下创建了资源充足的 **项目**。
  - **业务集群资源准备**：您已经在业务集群中部署了 **Jenkins**、**GitLab**、**Harbor** 工具。

**提示**：平台已经提供页面部署的方式部署 DevOps 工具链，请您登录容器平台，切换至管理视图，在左侧导航栏中单击 **操作器 > 操作器中心**，找到 **DevOps Tools Operator**，根据容器平台的帮助文档部署 DevOps 工具链。

- **总体流程**：以下为您展示了快速入门的实践流程，为您快速使用平台能力提供参考。
  1. **快速集成绑定工具**：集成绑定已部署的 DevOps 工具链。
  2. **快速创建流水线**：使用平台流水线模版创建流水线。
  3. **快速执行流水线**：对创建好的流水线进行执行操作。

# 快速集成绑定工具链

集成绑定 DevOps 工具，您可以在项目中重复使用工具，无需反复输入授权。本节会帮助您集成绑定持续集成工具 Jenkins、代码仓库工具 GitLab、制品仓库工具 Harbor。

## 基本流程

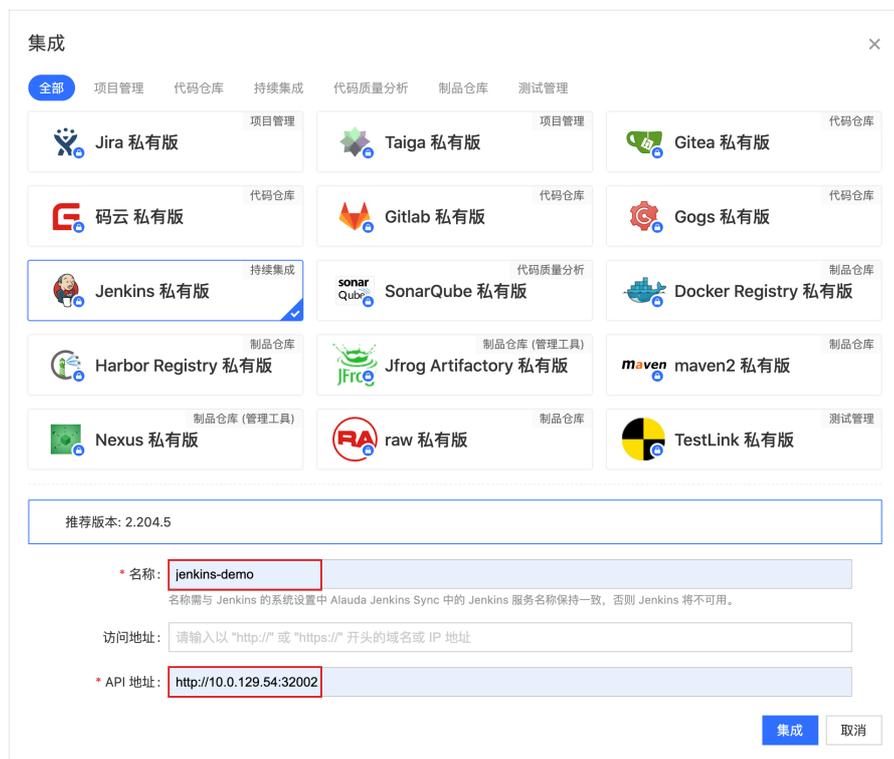
1. 快速集成绑定 Jenkins
2. 快速集成绑定 GitLab
3. 快速集成绑定 Harbor

## 操作步骤 1 快速集成绑定 Jenkins

在此步骤中您需要集成绑定 Jenkins，并且给需要使用此工具的项目分配资源。

### 集成 Jenkins

1. 登录 DevOps 平台，进入管理视图后，单击 **DevOps 工具链 > 集成**。
2. 在 **集成** 窗口，单击 **Jenkins 私有版**。



- 在 **名称** 框中，输入集成的 Jenkins 工具名称，例如：`jenkins-demo`。

注意：该名称需与 Jenkins 系统设置中 **Alauda Jenkins Sync** 的 Jenkins 服务名称保持一致，否则 Jenkins 将不可用。

您需使用管理员账户登录 Jenkins，在 **系统管理 -> 系统设置** 中启用 **Alauda Jenkins Sync** 选项卡并且在 **Jenkins 服务名称** 框中填写集成 Jenkins 服务的名称，例如：`jenkins-demo`。

- 在 **API 地址** 框中，输入集成 Jenkins 工具的 API 地址，例如：`http://10.0.129.54:32002`。

3. 单击 **集成**，进入集成详情页面。

### 绑定 Jenkins

1. 在集成详情页面中，单击 **绑定**。
2. 在弹出的 **选择项目** 页面中，选择您已经创建好的项目，例如：`project`，选择项目后在绑定页面配置绑定参数。

- 在 **基本信息** 的 **名称** 框中，输入绑定 Jenkins 服务的名称，此名称仅用于在该项目下区分不同工具链资源，例如：**jenkins-binding**。
- 在 **认证** 区域，选择认证方式和凭据。
  - 认证方式：目前仅支持并默认选择 **用户名/Token** 认证。
- 在 **凭据** 框右侧，单击 **添加凭据**，在弹框中填写以下参数：

- 在 **基本信息** 区域的 **凭据名称** 框中，填写凭据名称，例如：**jenkins-secret**。
- 在 **数据** 区域的 **用户名** 框中，填写登录 Jenkins 时使用的用户名，例如：**admin**。在 **Token** 框中，填写在 Jenkins 中生成的 **Token**。

提示：生成 **Token** 方法如下：单击凭据框下方的 **前往 Jenkins**，在跳转页面中的 **API Token** 区域，单击 **添加新 Token**，生成 Token 后复制并粘贴在添加凭据页面的 **Token** 框中。

单击 **创建**，创建 Jenkins 凭据。

3. 单击 **绑定账号**，单击绑定账号后，平台会立即跳转至绑定详情页面。至此，您已经在该项目下集成并绑定了 Jenkins 服务。

## 操作步骤 2 快速集成绑定 GitLab

集成绑定 GitLab 代码仓库工具，实现在项目中重复使用代码仓库，而无需反复输入授权，方便操作，易于管理。

## 集成 GitLab

1. 在集成窗口，单击 **GitLab 私有版**。

集成

全部 项目管理 代码仓库 持续集成 代码质量分析 制品仓库 测试管理

Jira 私有版 Taiga 私有版 Gitea 私有版

码云 私有版 Gitlab 私有版 Gogs 私有版

Jenkins 私有版 SonarQube 私有版 Docker Registry 私有版

Harbor Registry 私有版 Jfrog Artifactory 私有版 maven2 私有版

Nexus 私有版 raw 私有版 TestLink 私有版

推荐版本: 11.4.0

\* 名称: gitlab-demo

访问地址: 请输入以 "http://" 或 "https://" 开头的域名或 IP 地址

\* API 地址: http://10.0.100.34:32001

集成 取消

- 在名称框中，输入集成的 GitLab 工具名称，例如：`gitlab-demo`。
- 在 API 地址框中输入集成工具的服务地址，例如：`http://10.0.129.54:32001`。

2. 单击 **集成**，进入集成详情页面。

## 绑定 GitLab

1. 在集成详情页面中，单击 **绑定**。

2. 在绑定窗口中绑定账号。

1. 在 **绑定账号** 步骤下您需进行如下操作：

1 绑定账号 ———— 2 分配仓库

\* 名称: gitlab-binding

描述:

认证方式: 用户名/Token OAuth2  
前往 gitlab-demo 生成 Token, 并将 Token 信息添加到凭据

\* 凭据: gitlab-secret 添加凭据

绑定账号 取消

- 在名称框中，输入绑定 GitLab 代码仓库服务的名称，例如：`gitlab-binding`。
- 在认证方式区域，您可选择 Token 或 OAuth2，本流程以 Token 为例。

- 在 **凭据** 区域，单击右侧框中 **添加凭据**，在弹出的 **创建凭据** 页面中填写凭据信息。

创建凭据

基本信息

\* 凭据名称: gitlab-secret

显示名称:

工具类型: gitlab-enterprise

工具集成名称: gitlab-demo

数据

类型:  用户名/Token  OAuth2

\* 用户名: root  
请输入登录GitLab时使用的用户名

\* Token: .....  
[前往GitLab 设置页面生成Token](#)

创建 取消

- 在 **基本信息** 区域的 **凭据名称** 框中，填写凭据名称，例如：**gitlab-secret**。
- 在 **数据** 区域的凭据 **类型** 处，选择 **用户名/Token**。
- 在 **用户名** 框中，填写登录 GitLab 时使用的用户名，例如：**root**。
- 在 **Token** 框中，填写在 GitLab 中生成的 **Token**。

提示：生成 **Token** 方法如下：单击凭据框下方的 **前往 GitLab**，在跳转页面中的 **Personal Access Tokens** 区域，填写 **Name** 并在 **Scopes** 区域勾选所有权限后，单击 **Create personal access token**，生成 Token 后复制并粘贴在平台添加凭据页面的 **Token** 框中，单击 **创建**。

单击 **绑定账号**。绑定账号后，会判断选择的凭据是否有权限去访问代码仓库，如果没有权限，会提示错误。

2. 在 **分配仓库** 步骤下您需进行如下操作：

- 选择绑定的个人或组织账号的一个或多个代码仓库，或选择 **自动同步全部仓库**，同步当前账号下所有的仓库，并保持与当前账号下仓库的同步。

提示：如果选择了同步某个账号下的全部代码仓库，当这个账号下的仓库在远端的代码仓库发生了变化，若对代码仓库进行增、删、改等操作，平台也会对相关变化进行同步操作。

3. 单击 **分配仓库**。跳转至绑定详情页，等待一段时间，分配的代码仓库会显示，至此，您已经在该项目下集成并绑定了 GitLab 服务。

## 操作步骤 3 快速集成绑定 Harbor

### 集成 Harbor

通过集成 Harbor，在 Docker Registry 的基础上，使用增加了安全、标识的镜像。

1. 在 **集成** 窗口，单击 **Harbor Registry 私有版**。

集成

全部 项目管理 代码仓库 持续集成 代码质量分析 制品仓库 测试管理

Jira 私有版 项目管理

Taiga 私有版 项目管理

Gitea 私有版 代码仓库

码云 私有版 代码仓库

Gitlab 私有版 代码仓库

Gogs 私有版 代码仓库

Jenkins 私有版 持续集成

SonarQube 私有版 代码质量分析

Docker Registry 私有版 制品仓库

Harbor Registry 私有版 制品仓库

Jfrog Artifactory 私有版 制品仓库 (管理工具)

maven maven2 私有版 制品仓库

Nexus 私有版 制品仓库 (管理工具)

raw 私有版 制品仓库

TestLink 私有版 测试管理

推荐版本: 2.0

\* 名称: harbor-demo

访问地址: 请输入以 "http://" 或 "https://" 开头的域名或 IP 地址  
若镜像仓库中心为 http 协议, 需要在集群内每个节点上将镜像仓库地址添加到 docker daemon 配置中, 方可使用。

\* API 地址: http://10.0.129.54:32000  
若镜像仓库中心为 http 协议, 需要在集群内每个节点上将镜像仓库地址添加到 docker daemon 配置中, 方可使用。

集成 取消

- 在名称框中, 输入集成的 Harbor Registry 工具名称, 例如: harbor-demo。
- 在 API 地址框中, 输入集成 Harbor Registry 工具的服务地址, 例如: https://10.0.129.54:32000。

2. 单击集成。进入集成详情页面。

## 绑定 Harbor

绑定 Harbor 制品仓库工具, 实现在项目中重复使用 Docker 镜像, 而无需反复输入授权, 方便操作, 易于管理。

- 在集成 Harbor 详情页面, 单击绑定。
- 在绑定页面中绑定账号:

1 绑定账号 ———— 2 分配仓库

\* 名称: harbor-binding

描述:

认证方式: 用户名/密码 无认证

\* 凭据: harbor-secret 添加凭据

绑定镜像仓库时, 会在绑定的项目内, 自动生成镜像仓库类型的保密字典 (DevOps默认不展示), 以便在执行部署应用或执行流水线等操作时, 可正常进行。修改绑定凭据时, 该自动生成的保密字典将自动更新, 若修改或删除此保密字典, 系统会自动再次生成。

绑定账号 取消

- 在名称框中, 输入绑定 Harbor 制品仓库工具的名称, 例如: harbor-binding。
- 在认证方式区域, 选择镜像服务或无认证。本流程推荐选择用户名/密码认证。
- 在凭据区域, 单击凭据框右侧的添加凭据, 填写凭据信息。

**创建凭据**

**基本信息**

\* 凭据名称: harbor-secret

显示名称:

工具类型: harbor-registry

工具集成名称: harbor-demo

**数据**

类型: 用户名/密码

\* 用户名: admin  
请输入登录Harbor时使用的用户名

\* 密码: .....  
请输入登录Harbor时使用的密码

创建 取消

- 在 **基本信息** 区域的 **凭据名称** 框中，填写凭据名称，例如：**harbor-secret**。
- 在 **用户名** 框中，填写登录 Harbor 时使用的用户名，例如：**admin**。
- 在 **密码** 框中，填写在 Harbor 的登录密码，例如：**password**。

单击 **创建**，完成创建 Harbor 凭据。

3. 单击 **绑定账号**。

单击绑定账号后，会判断选择的凭据是否有权限去访问制品仓库，如果没有权限，会提示错误。

4. 绑定账号成功后，分配制品仓库。单击 **1** 处空白选择框，选择要使用的镜像仓库或仓库中的某个镜像。

绑定账号 ———— 分配仓库

✓ 账号绑定成功，请为项目分配镜像仓库

添加地址

操作

1

- ▾ 库 /
- ▾ e2e-automation
- ▾ library
- ▾ new11
- ▾ new2
- ▾ new33
- ▾ test
- ▾ tsf\_1

库 取消

5. 选择镜像后，单击 **2** 处 **添加地址**，将选择的镜像仓库或镜像添加到仓库地址列表。单击 **3** 处 **分配仓库**，将选择的制品仓库绑定到项目中。

① 绑定账号 ————— ② 分配仓库

✔ 账号绑定成功，请为项目分配镜像仓库

2 添加地址

仓库地址	操作
没有选中地址	

3 分配仓库 取消

6. 添加成功后，将跳转至制品仓库的详情页面，至此，您已经在该项目下集成并绑定了 Harbor 服务。

# 快速创建流水线

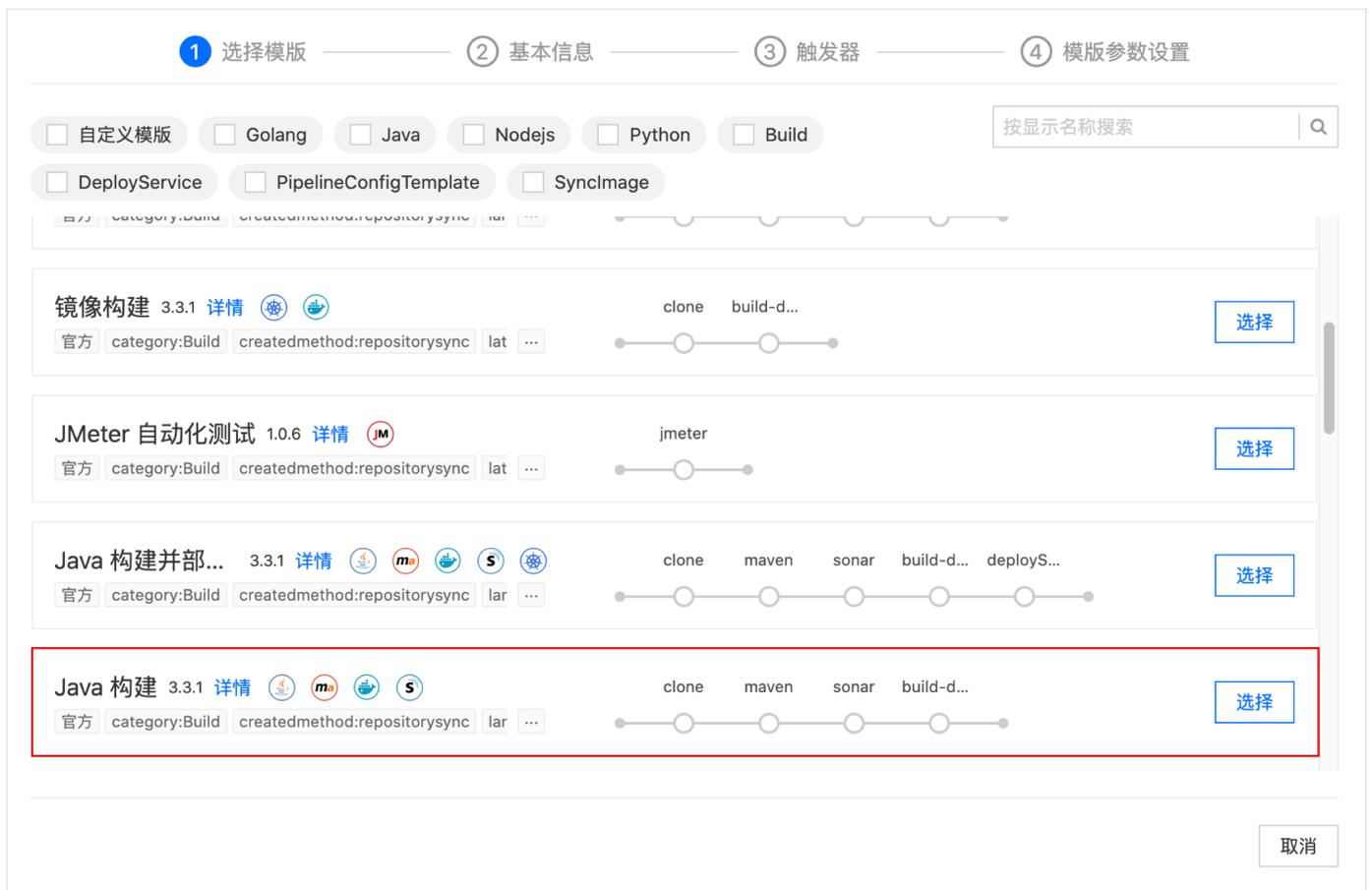
本节将使用 **Java 构建** 官方模版创建流水线，通过详细的步骤操作描述，帮助您快速创建一条流水线。

## 基本流程

1. 选择模版
2. 配置基本信息
3. 配置触发器
4. 配置模版参数

## 操作步骤 1 选择模版

1. 登录 DevOps 平台，进入业务视图后，在项目列表页面，单击您已创建的 **项目名称**。
2. 进入项目后，单击 **持续交付 > 流水线 > 创建流水线 > 模版创建**。
3. 在 **选择模版** 步骤，选择 **Java 构建** 官方模版创建流水线，在流水线模版右侧单击 **选择** 按钮。



## 操作步骤 2 配置基本信息

在 **基本信息** 步骤，配置流水线的基本信息：

- 在 **名称** 框中，输入 Jenkins 流水线的名称，例如：**pipeline**。
- 在 **Jenkins 实例** 框中，选择您刚在该项目中绑定的 Jenkins 实例，例如：**jenkins-demo**。

1 选择模版 — 2 基本信息 — 3 触发器 — 4 模版参数设置

\* 名称:

显示名称:

\* Jenkins:

执行方式:  依次执行  并发执行 ?

禁用流水线:  ?

上一步 下一步 取消

配置完成后，单击 下一步。

## 操作步骤 3 配置触发器

在 **触发器** 步骤，选择启用定时扫描触发器，配置相应的触发规则，可以实现流水线的自动触发。

- 定时扫描触发器**：在指定的时间检查代码仓库是否有变更，如有变更，将触发流水线。支持选择预设的触发器规则。您可以选择 **每 2 分钟检查新提交**，则每隔 2 分钟系统会自动检查代码仓库是否有新提交的变更，若有新提交的变更则会执行流水线。

1 选择模版 — 2 基本信息 — 3 触发器 — 4 模版参数设置

**定时扫描触发器**  
 设置扫描间隔，系统会自动检查流水线中配置的代码仓库分支是否有新提交，若有新提交则会自动触发

\* 扫描规则：

例如：选择“每 2 分钟检查新提交”，则每隔 2 分钟系统会自动检查代码仓库是否有新提交，若有新提交则会执行流水线

**代码仓库触发器**  
 可设置代码仓库分支发现规则，符合分支规则的分支代码有新的提交时，自动触发流水线执行（执行变动的分支代码）

**定时触发器**  
 可设置一个或几个时间，到时间后会自动触发流水线执行

上一步 下一步 取消

配置完成后，单击 下一步。

## 操作步骤 4 配置模版参数

在 **模版参数设置** 步骤，配置模版参数。不同模版类型的配置信息不同，您可以单击右上角的模版版本号，查看模版详情了解每个参数的定义。

1. 在 **代码检出** 区域，配置代码仓库。

- 在 **代码仓库** 框中，单击 **选择** 后，在 **代码仓库** 窗口，选择使用已绑定的 GitLab 代码仓库中的项目。
- 在 **分支** 框中，从下拉框中选择分支，例如：**master**。

2. 在 **Docker 构建** 区域，配置使用 Docker 镜像构建的参数。

- 在 **镜像仓库** 框中，单击 **选择** 后，在 **镜像** 窗口，选择 Harbor 中的一个镜像仓库，并在后面的框中，输入镜像的版本，默认为：latest。单击 **确定**。

选择模版 — 基本信息 — 触发器 — **4 模版参数设置**

代码检出 \* 流水线全局配置

\* 代码仓库:  🔗 ?

分支:  × ?

[展开高级选项](#)

---

Maven 构建

构建命令:  ?

展示 JUnit 测试报告:  ?

---

代码扫描

开启代码扫描:  ?

[展开高级选项](#)

---

Docker 构建

\* 镜像仓库:  🔗 ?

📁

[展开高级选项](#)

---

通知

开启通知:  ?

[预览 Jenkinsfile](#)

完成以上 4 个步骤后，请单击 **创建** 完成创建。

# 快速执行流水线

对您已创建好的流水线进行执行操作。

## 操作步骤

1. 登录 DevOps 平台，默认进入业务视图，在顶部的 **项目名称** 处，切换项目，例如：**project**，再单击 **持续交付 > 流水线**。
2. 在流水线页面，可以通过以下两种方式手动执行流水线：
  - 找到要手动执行的 **流水线名称**，例如：**pipeline**，单击 ，再单击 **执行**。
  - 单击要手动执行的 **流水线名称**，例如：**pipeline**，在流水线详情页面，单击 **操作 > 执行**。

# 快速入门

本文旨在介绍如何快速使用 Service Mesh，帮助您快速了解从部署网格到接入服务的基本使用流程。

## 1. 入门必读

在体验微服务治理功能之前，完成必要的准备工作，参考 [入门必读](#)。

## 2. 流量策略设置

为服务设置流量策略，进行流量治理，参考 [流量策略设置](#)。

## 3. 查看服务调用关系

通过服务拓扑，查看服务间的调用关系，参考 [查看服务调用关系](#)。

## 4. 追踪服务调用链

追踪服务间的完整调用链，参考 [追踪服务调用链](#)。

# 入门必读

在您开始体验 Service Mesh 的各项微服务治理功能之前，管理员需要在业务集群部署服务网格，用户还需添加服务至 Service Mesh 平台，方可在 Service Mesh 上对服务进行治理。

## 操作步骤-在业务集群部署服务网格

部署包括 Istio 服务网格管理平台在内的所有 Service Mesh 组件，并接入必要的外部组件，使集群具备容器化微服务治理的能力。服务网格管理功能将部署 Service Mesh 组件流程可视化，管理员只需配置主要参数，可以方便快捷地执行部署服务网格、修改服务网格参数等操作。

1. 管理员在管理视图的左导航栏中，单击 **服务网格**，再在列表页中，单击 **创建服务网格**。
2. 在 **基本信息** 区域，配置服务网格的名称、部署集群等基本信息。
  - 集群中已部署 Prometheus 工具。
  - 已部署集群可访问的外部 Elasticsearch 工具，也支持使用系统默认 Elasticsearch 工具。
3. 在 **调用链配置** 区域，配置相关参数，可使用默认值。
4. 在 **网关部署** 区域，配置集群的入口网关和出口网关参数。
5. 在 **Sidecar 配置** 区域，配置相关参数，可使用默认值。
6. 在 **接入组件** 区域，配置接入服务网格外部的组件。
  - **Elasticsearch**：调用链数据的存储。支持系统默认及外接 ES。
  - **Prometheus**：Istio 监控功能数据采集，系统默认。

## 操作步骤-添加微服务至 Service Mesh 平台

您可以在微服务治理平台上，将您的业务作为微服务进行治理，以在服务列表中添加内部路由及其关联的计算组件，作为服务接入方式。

注意：在容器平台，已创建内部路由及关联的 Deployment（部署）类型计算组件。内部路由必须与计算组件是一对一关联关系，方可添加到服务列表中。

1. 切换至业务视图，进入需要添加微服务的命名空间。
2. 单击 **微服务列表**，在列表页中单击 **添加微服务**，选择要添加到服务列表中的内部路由即可。

注意：添加微服务后，微服务中的计算组件将重启。

### 添加内部路由

1. 内部路由与工作负载为一对一关系时，方可加入 Service Mesh。若下述列表为空，可转至 **部署 / 内部路由** 创建 Deployment 或 Service。  
2. 添加后，会更新 Service 与 Deployment 的 app 标签，使其与 Service 名称一致，并重启 Deployment。

<input type="checkbox"/>	内部路由(service)	工作负载
<input checked="" type="checkbox"/>	test-fy	test-fy

**添加** **取消**

# 设置流量策略

完成入门必读的准备工作后，可为服务设置流量策略，如安全、负载均衡、熔断等，进行流量治理。

在业务视图，进入服务详情页。单击下方的 **策略** 页签，可设置相关流量策略，例如设置服务的熔断策略，针对与服务关联的计算组件实例进行异常值检测。

### 创建熔断策略

**阈值** ?

连续错误响应个数:  ?

**规则** ?

检查周期:  s ?

实例最短隔离时间:  s ?

实例最大隔离比例:  % ?

更多配置可参考用户手册相关功能介绍。

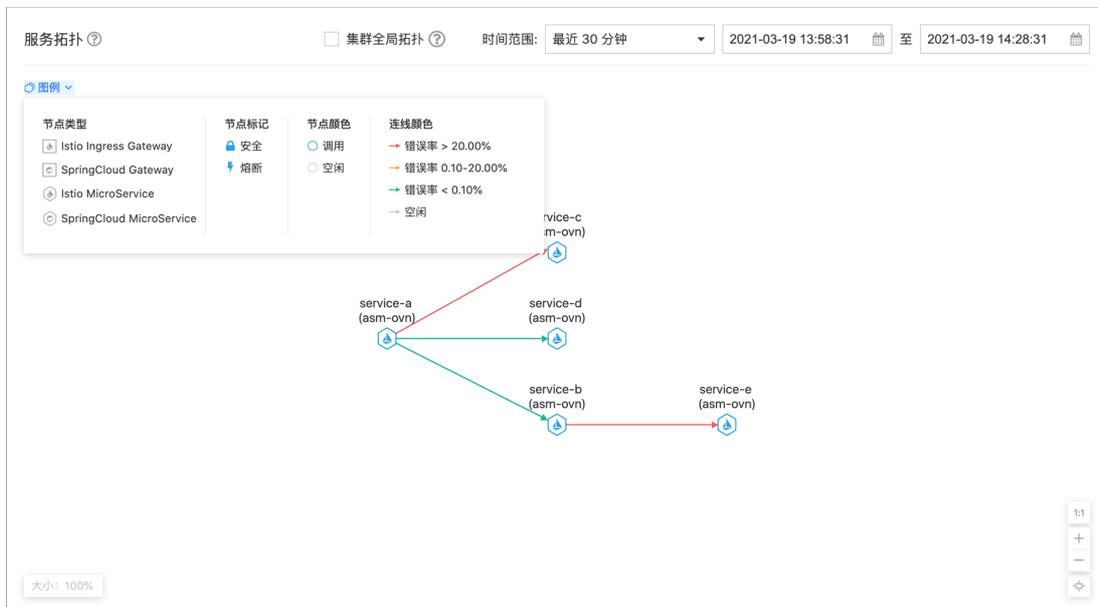
- 管理安全策略，为服务创建安全规则，对流量进行加密。
- 管理负载均衡策略，调用服务时，根据负载均衡规则，自动在服务关联的 Pod 实例之间分配流量。
- 管理熔断策略，针对与服务关联的计算组件实例进行异常值检测。
- 管理连接池设置策略，帮助您抵御流量负载过大的情况。

# 查看服务调用关系

通过服务拓扑图，可视化命名空间中的服务间调用关系，方便查看服务之间的调用信息。选中某个服务节点或连线时，可以显示该服务或连线的详细信息，例如请求数量、性能、错误率、正确和错误的 RPS 等。

## 操作步骤

1. 将具有调用关系的服务添加至 Service Mesh 平台后，在左导航栏单击 **服务拓扑**，即可查看拓扑图。



2. 单击节点或者连线，即可查看相应的流量信息。



# 追踪服务调用链

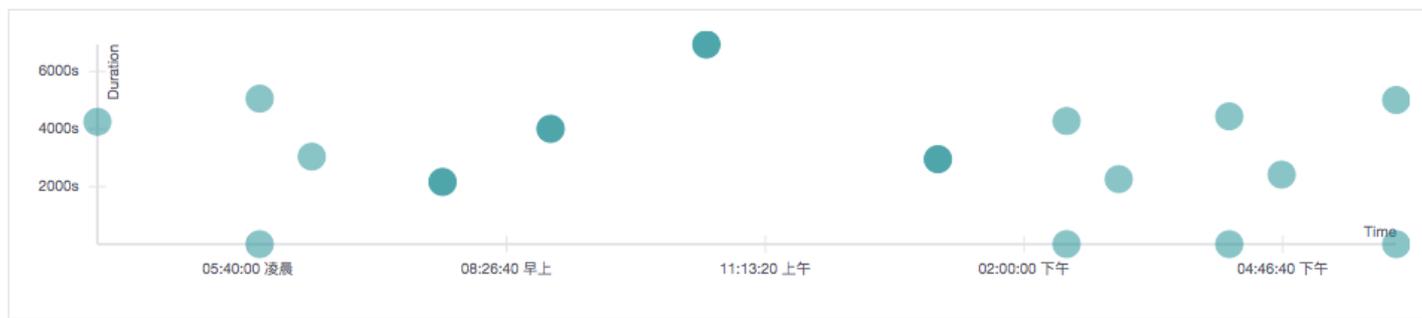
本平台通过集成 Jaeger，用来追踪项目中微服务之间的完整调用链。通过调用链跟踪系统，可以快速了解各节点的响应状况。

在调用链页面，输入相应的查询条件，即可查询相应的调用链。

服务(10):  操作(1):  标签:  ?

时间范围:  最小调用时间:  最大调用时间:

最多显示数量:



20 Traces

Sort:

## Compare traces by selecting result items

<input type="checkbox"/>	service-a: service-b.asm-demo.svc.cluster.local:80/* 69da005	5.01s
<input type="text" value="1 Span"/>	<input type="text" value="service-a (1)"/>	Today   5:45:14 下午 几秒前
<input type="checkbox"/>	service-a: service-b.asm-demo.svc.cluster.local:80/* 827e890	2.22ms
<input type="text" value="1 Span"/>	<input type="text" value="service-a (1)"/>	Today   5:45:14 下午 几秒前
<input type="checkbox"/>	service-a: service-b.asm-demo.svc.cluster.local:80/* d751c63	2.42s

说明：

- 服务：当前命名空间下已注入 sidecar 的 Kubernetes 服务。
- 操作：要查询的服务操作。