微服务平台 TSF

快速入门

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

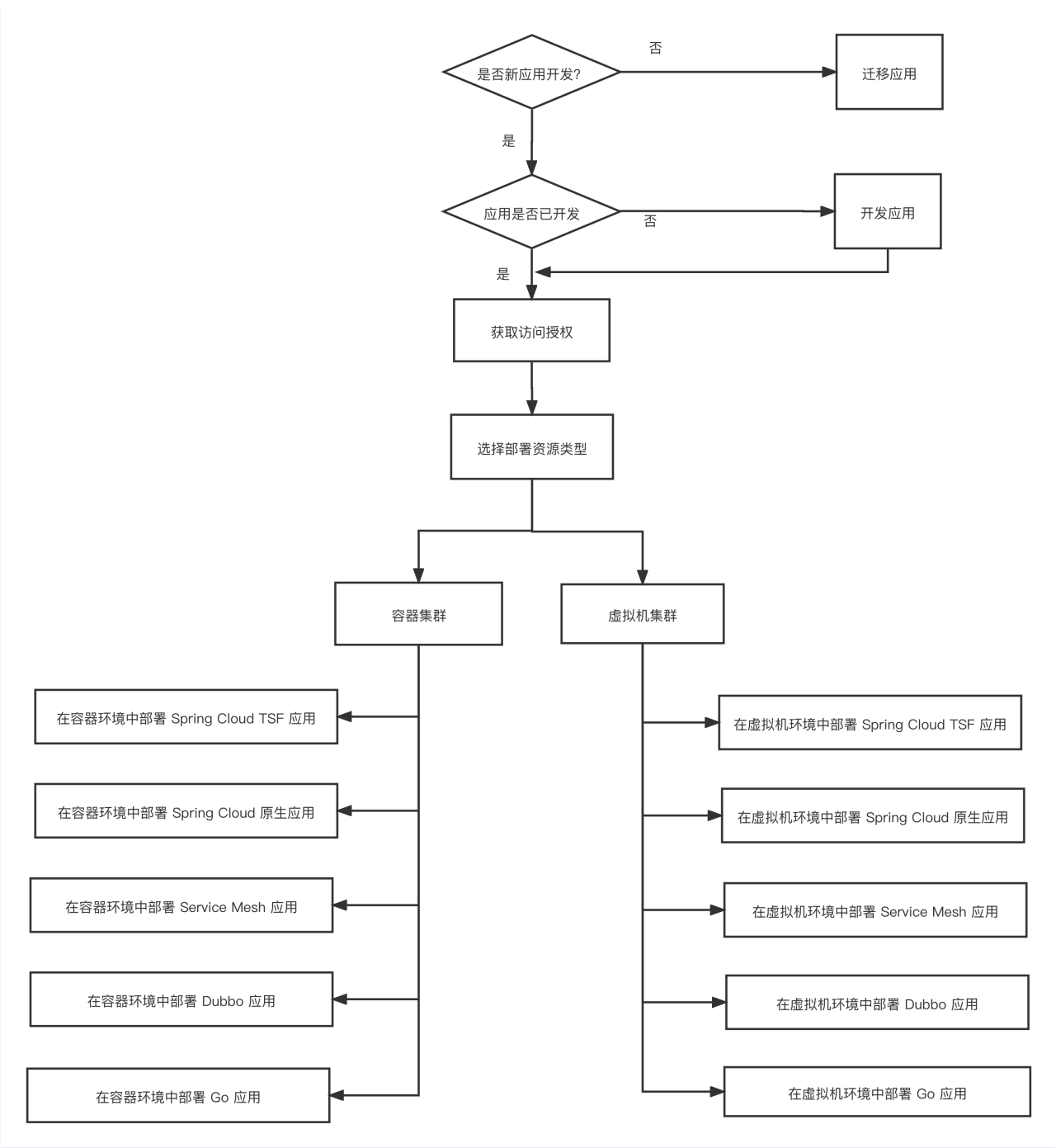
快速入门
- 入门流程指引
- 获取访问授权
  - 主账号获取访问授权
  - 子账号获取访问授权
- 在容器环境中部署微服务
  - 快速创建一个容器集群
  - 部署 Spring Cloud TSF 应用
  - 部署 Spring Cloud 原生应用
  - 部署 Service Mesh 应用
  - 部署 Dubbo 应用
  - 部署 Spring Cloud TSF 应用（Java Agent 版）
- 在虚拟机环境中部署微服务
  - 快速创建一个虚拟机集群
  - 部署 Spring Cloud TSF 应用
  - 部署 Spring Cloud 原生应用
  - 部署 Service Mesh 应用
  - 部署 Dubbo 应用
  - 部署 Spring Cloud TSF 应用（Java Agent 版）

版权所有：腾讯云计算（北京）有限责任公司

快速入门
入门流程指引

最近更新时间：2023-08-11 11:19:51

快速入门以一组简单的 Demo 为例介绍将不同类型的应用（如 Spring Cloud 应用、Mesh 应用、Dubbo 应用等）部署到 TSF 的不同环境中的操作流程，帮助您快速了解 TSF 的使用流程。



- 迁移应用请参见 应用迁移 。
- 开发应用请参见 应用开发 。
- 部署应用到 TSF 请参见 在容器环境部署微服务 和 在虚拟机环境部署微服务 。

版权所有：腾讯云计算（北京）有限责任公司

获取访问授权
主账号获取访问授权

最近更新时间：2022-06-09 15:54:58

操作背景

由于 TSF 需要访问其他云产品的 API（例如 TKE），所以需要授权 TSF 创建服务角色。

前提条件

注册腾讯云账号

> **说明**
> 当您注册腾讯云账号后，系统默认为您创建了一个主账号，用于快捷访问腾讯云资源。

操作步骤

1. 使用 **主账号**登录 TSF 控制台，进入概览页。由于没有授权腾讯微服务平台 TSF 服务角色权限无法访问其他云产品资源。

2. 单击**前往授权**，进入 CAM 控制台 授权，单击**同意授权**，则为腾讯微服务平台 TSF 授权服务角色访问您其他云产品资源。

子账号获取访问授权

最近更新时间：2024-01-19 11:48:11

CAM 基本概念

主账号通过给子账号绑定策略实现授权，策略设置可精确到 **[API，资源，用户/用户组，允许/拒绝，条件]** 维度。

账户

- **主账号**：拥有腾讯云所有资源，可以任意访问其任何资源。
- 子账号：包括子用户和协作者。
  - **子用户**： 由主账号创建，完全归属于创建该子用户的主账号。
  - **协作者**：本身拥有主账号身份，被添加作为当前主账号的协作者，则为当前主账号的子账号之一，可切换回主账号身份。
- **身份凭证**：包括登录凭证和访问证书两种，**登录凭证** 指用户登录名和密码，**访问证书** 指云 API 密钥（SecretId 和 SecretKey）。

资源与权限

- **资源**：资源是云服务中被操作的对象，如一个云服务器实例、COS 存储桶、VPC 实例等。
- **权限**：权限是指允许或拒绝某些用户执行某些操作。默认情况下，**主账号拥有其名下所有资源的访问权限**，而**子账号没有主账号下任何资源的访问权限**。
- **策略**：策略是定义和描述一条或多条权限的语法规范。**主账号**通过将**策略关联**到用户/用户组完成授权。

子账号使用 TSF

> **说明**
> 在进行 tsf_passrole 授权之前请确定主账户已完成获取访问授权，并已拥有 TSF_QCSRole 的角色，详情参考 主账号获取访问授权 。

协作者与子账号使用TSF平台时，需要对三方面进行授权：

1. 要将角色（及其许可策略）传递至 TSF 服务，用户必须具有**传递角色**至服务的许可，即创建 tsf_PassRole策略，详细操作参考 授予 tsf_PassRole 策略 。
2. 配置使用 TSF 平台的权限，平台支持为不同子账号灵活配置管理权限，包含对TSF中不同集群、命名空间、应用授予读权限或写权限，详细操作可以参考 管理中心 。具体支持以下三种使用场景：
   - 为子账号或协作者配置全部资源的全读写策略。您可以通过授予子账号 QcloudTSFFullAccess 策略给予子账号使用TSF平台的全量权限。
   - 为子账号或协作者配置全部资源的部分操作权限，如可以为部分用户配置应用、微服务、配置的全读写策略，以及集群、命名空间的只读策略。
   - 为子账号和协作者配置某些资源（一个或多个）的读或写权限，如可以为不同的子账号配置不同命名空间、不同应用的不可见、只读、全读写权限。

   > **说明**
   > 该场景下不支持使用六段式资源自定义配置使用权限和灵活配置管理权限。

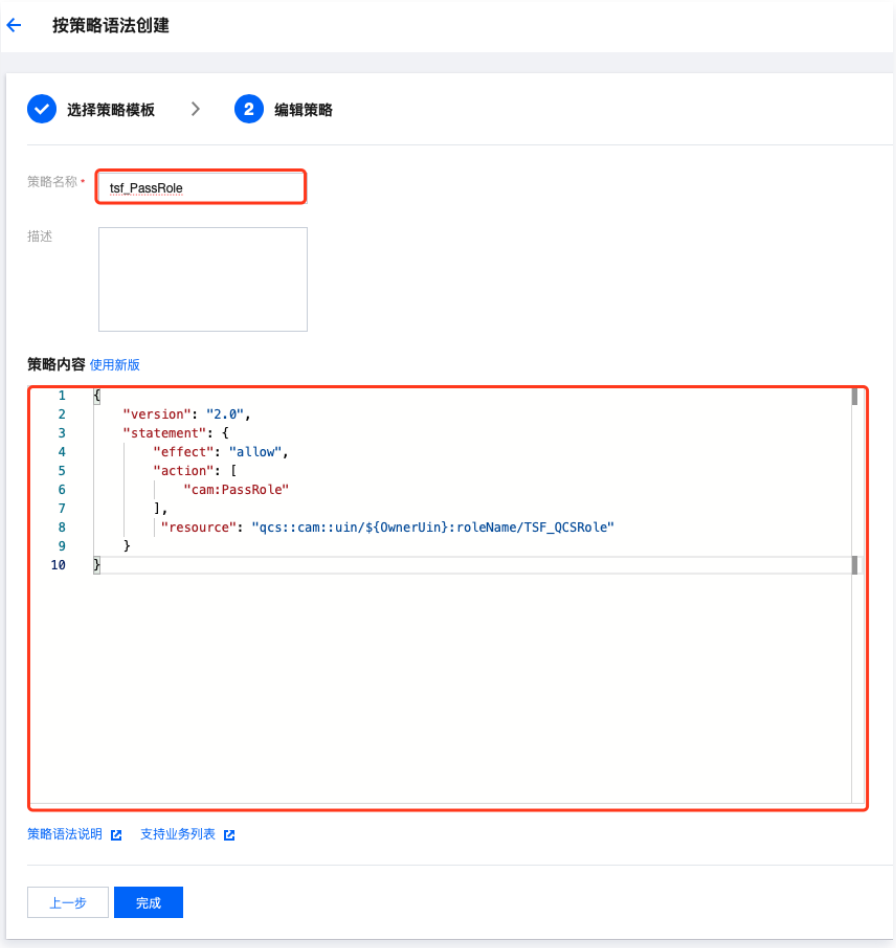
3. 使用 TSF 平台过程中，涉及到对其他产品的调用，如云服务器（CVM）、私有网络（VPC）、标签（TAG）、容器服务（TKE）、镜像仓库（TCR）等，需要主账号对子账号进行授权。详细说明参考 授予访问其他云产品权限 。

授予 tsf_PassRole 策略

步骤1：新建 tsf_PassRole 策略

1. 登录 访问管理控制台 。

2. 在左侧导航栏，单击**策略**，进入策略管理列表页。
3. 单击**新建自定义策略**。
4. 在选择创建策略方式的弹出框中，单击**按策略语法创建**，进入按策略语法创建页。
5. 在 按策略语法创建页 中，选择**空白模板**，并单击**下一步**。
6. 填写策略名和内容，并单击**创建策略**。

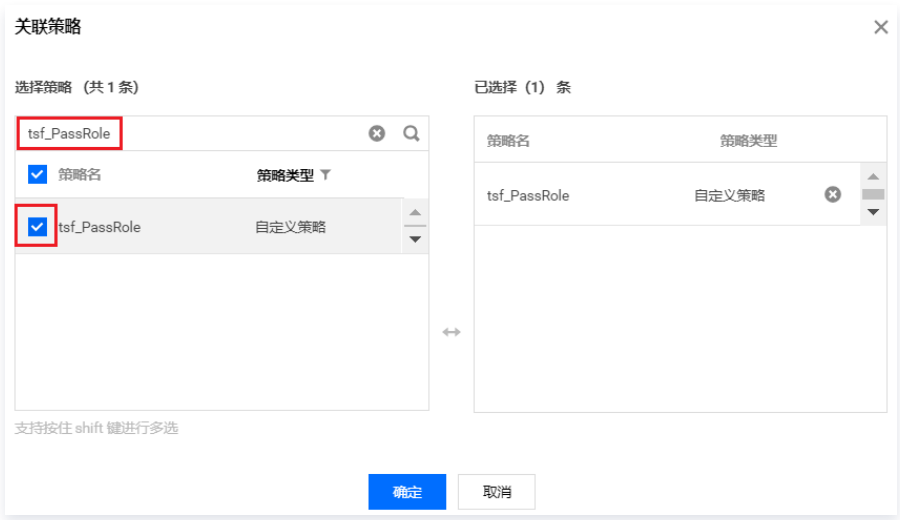


使用主账号或具有管理权限的子账号创建如下自定义策略，其中 `${OwnerUin}` 为主账号 ID，从控制台账号信息页面获取。具体策略语法如下：

```
{
  "version": "2.0",
  "statement": {
    "effect": "allow",
    "action": [
      "cam:PassRole"
    ],
     "resource": "qcs::cam::uin/${OwnerUin}:roleName/TSF_QCSRole"
  }
}
```

步骤2：将 tsf_PassRole 策略绑定到用户

1. 在左侧导航栏，单击**用户** > 用户列表，进入用户管理页面。
2. 选择要授予 TSF 使用权限的用户，单击操作列的**授权**。
3. 从策略列表中筛选出**步骤一**中的创建的策略（ 如 `tsf_PassRole` ）。



4. 单击**确定**，绑定策略。该策略会显示在用户的策略列表中。

授予访问其他云产品权限

TSF 平台使用中涉及到以下云产品的调用。主账号需要对子账号进行单独授权才能保证对应 TSF 产品功能的使用。TSF 中涉及到的对云产品的调用如下：

云产品	接口名	接口作用	影响到TSF平台的操作
云服务器（CVM）	DescribeKeyPairs	查询密钥对信息	创建集群后通过重装云主机方式导入云主机，选择密钥
云服务器（CVM）	DescribeInstances	查询实例列表	向集群中导入云主机时，查看可导入的云主机列表
云服务器（CVM）	DescribeZones	查询可用区	创建集群时查看可用区列表
云服务器（CVM）	ResetInstance	重装云主机实例	创建集群后通过重装云主机方式导入云主机
云服务器（CVM）	ModifyInstancesAttribute	修改云主机名称与安全组	通过重装云主机方式导入云主机到集群中的过程中修改云主机安全组
云服务器（CVM）	DescribeSecurityGroups	查看安全组	导入云主机到集群中查询安全组信息
私有网络（VPC）	DescribeVpcs	查询 VPC 列表	创建集群过程中选择集群所属VPC
私有网络（VPC）	DescribeVpcsEx	查询 VPC 列表	创建集群过程中选择集群所属VPC
私有网络（VPC）	DescribeSubnets	查询子网列表	选择子网
私有网络（VPC）	DescribeSecurityGroups	查看安全组	通过重装云主机方式导入云主机到集群中的过程中修改云主机安全组
容器服务（TKE）	DescribeClusters	查看容器集群列表	查询容器集群列表
标签（TAG）	DescribeResourceTagsByResourceIds	按顺序查看资源关联的标签	查看按照标签查看对应资源

标签（TAG）	ModifyResourceTags	修改资源的标签信息	修改 TSF 中资源的标签信息
标签（TAG）	DescribeTagKeys	查看标签键	查看 TSF 中资源的标签键
标签（TAG）	DescribeResourcesByTagsUnion	通过标签查询资源列表并集	在 TSF 中通过标签查询资源列表并集
标签（TAG）	DescribeTagValues	查看标签值	查看 TSF 中资源的标签值
腾讯云可观测平台（Monitor）	DescribeProductEventList	获取产品事件列表	在概览页中查看产品事件列表
腾讯云可观测平台（Monitor）	GetMonitorData	拉取指标监控数据	查看 TSF 中监控数据
腾讯云可观测平台（Monitor）	DescribeAlarmHistories	查询告警信息历史记录	查看 TSF 相关的告警记录
容器镜像服务（TCR）	全部接口	容器镜像云端托管服务	使用容器部署过程中镜像相关能力

云服务器（CVM）、标签（TAG）、容器服务（TKE）、云监控（Monitor）授权说明：

您可以通过腾讯云上访问管理能力对以上接口进行授权，详细可参考 创建自定义策略 。
在此给出授权示例：
如您希望子账号使用TSF中针对所有资源的创建集群、导入云主机、配置与查询标签、查看事件与监控能力，且该能力不对任何资源进行区分，可配置接口级别授权策略如下：

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cvm:DescribeKeyPairs",
        "cvm:DescribeInstances",
        "cvm:ResetInstance",
        "cvm:ModifyInstancesAttribute",
        "cvm:DescribeSecurityGroups",
        "tke:DescribeClusters",
        "tag:DescribeResourceTagsByResourceIds",
        "tag:ModifyResourceTags",
        "tag:DescribeTagKeys",
        "tag:DescribeTagValues",
        "monitor:DescribeProductEventList",
        "monitor:GetMonitorData"
      ],
      "resource": [
        "*"
      ]
    }
  ]
}
```

注意

- 这里进行授权后，子账号也直接获得了相关产品的直接使用权限。建议您按照需求严格管控，可以选择对相关产品接口进行资源级别授权。

- 配置 "cvm:ResetInstance"和"cvm:ModifyInstancesAttribute"接口后子账号将获得重装实例和修改实例的属性的权限，请谨慎配置，避免出现权限过大的场景。

镜像服务（TCR）授权

如果主账号未开通过镜像仓库，会提示如下图所示信息，此时需要主账号登录 TSF 控制台，开通镜像仓库。主账号开通镜像仓库后协作者/子账号才能继续使用镜像仓库。

如子账号需要通过容器部署微服务，需要使用镜像仓库能力，建议为子账号授予 QcloudTCRFullAccess 策略。

私有网络（VPC）授权说明：

建议为子账号授予私有网络读权限 QcloudVPCReadOnlyAccess。注意子账号也将同时可在VPC及相关产品中对私有网络资源拥有读权限。

总结

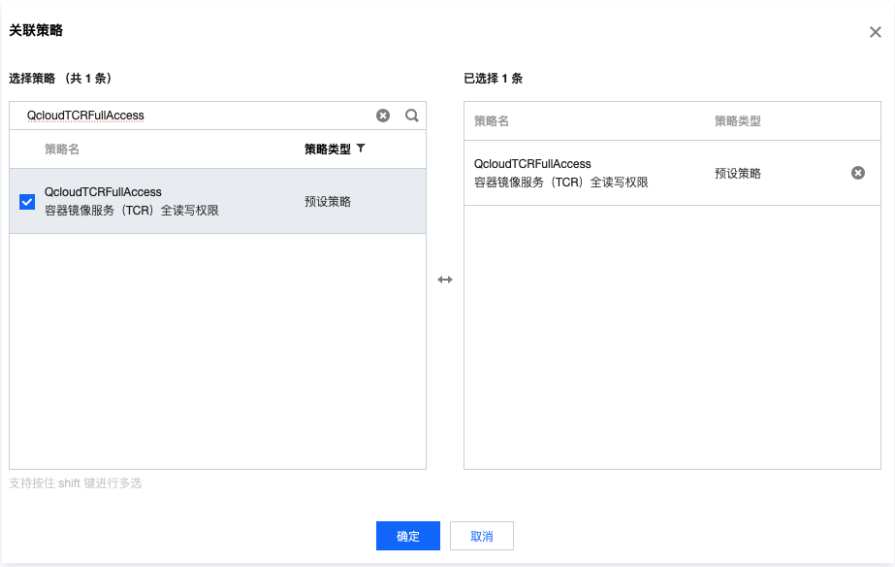
最终子账号可以授予如下策略：

策略	是否必选	说明
tsf_PassRole	必选	手动创建。
QcloudCamSubaccountsAuthorizeRoleFullAccess	必选	访问管理（CAM）子账号授权服务角色相关权限，包含子账号在授权服务角色过程中涉及的全部权限。
QcloudTSFFullAccess	可选	腾讯微服务平台（TSF）全读写访问权限，也可以参考 管理中心 进行细粒度授权。
QcloudTCRFullAccess	可选	与镜像仓库相关，如果需要使用 TSF 中容器相关功能需要授权。
QcloudVPCReadOnlyAccess	可选	如果需要读取集群 VPC 等信息需要授权。
QcloudMonitorReadOnlyAccess	可选	如果需要读取监控数据需要授权，也可以按照上文说明进行接口级别授权。
QcloudTAGFullAccess	可选	如果需要读写TSF中资源的标签需要授权，也可以按照上文说明进行接口级别授权。
QcloudCVMFullAccess	可选	如果需要向TSF集群中导入云主机需要授权，也可以按照上文说明进行接口（DescribeKeyPairs，DescribeInstances，ResetInstance，ModifyInstancesAttribute，DescribeSecurityGroups）、资源级别授权。
QcloudTKEReadOnlyAccess	可选	如果需要创建容器集群需要授权，也可以按照上文说明进行接口（DescribeClusters）、资源级别授权。

补充说明

如何通过访问管理对子账号进行策略授权。

1. 登录 访问管理控制台 。
2. 在左侧导航栏，单击**用户** > **用户列表** ，进入用户管理页面。
3. 选择要授予 TSF 使用权限的用户，单击操作列的**授权**。
4. 从策略列表中选择 `QcloudTCRFullAccess` 策略。



5. 单击**确定**，绑定策略。该策略会显示在用户的策略列表中。

在容器环境中部署微服务
快速创建一个容器集群

最近更新时间：2023-11-01 17:50:41

操作场景

部署应用之前需要新建一个集群，集群是指云资源管理的集合，包含了运行应用的云主机等资源。TSF 中的集群分为虚拟机集群和容器集群。使用虚拟机或者容器计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。
本文介绍在 TSF 控制台快速创建一个容器集群并导入云主机的操作方法。

前提条件

- 已 获取访问授权
- 已 购买云服务器

创建容器集群

1. 登录 TSF 控制台 。
2. 在左侧导航栏中单击**集群**，选择好地域后，单击**新建**。
3. 设置集群的基本信息。

- 集群类型：选择**容器集群**。
- 新建类型：选择**直接创建**。
- Kubernetes 版本：选择容器集群的 Kubernetes 版本，建议选择最新版本或者次新版本。
- 所在可用区：选择容器集群所在的可用区，建议选择**默认可用区**。
- 集群网络：为集群内主机分配在主机网络地址范围内的 IP 地址，选择与已购买云服务器相同的 VPC 网络，用来保证后续导入集群的云服务器属于同一 VPC。
- 容器网络：为集群内容器分配在容器网络地址范围内的 IP 地址，这里我们选择可用的容器网络。
- 标签：可选，可留空。用于分类管理资源，详情参见 标签 。
- 备注：可选，可留空。集群的描述，不超过200个字符。
- 数据集：可选，可留空。用于细粒度管理子账号数据权限，详情参见 数据集管理 。

4. 单击**提交**，在弹窗中选择**确认**，进入导入云主机页面。如果选择**取消**，您后续可以在集群列表中，单击集群操作列的**导入云主机**重新导入云主

机。

导入云主机

1. 从集群所在 VPC 的云主机列表中，选择需要添加到集群的云主机。
2. 单击**下一步**，进行云主机配置。
   - 导入方式：容器集群只支持**重装系统**部署方式，该方式会使 root 用户安装 agent。
   - 操作系统：支持 CentOS，TencentOS 和 Ubuntu 三种类型。
   - 登录方式：选择**设置密码**，并根据提示设置对应的密码。
3. 单击**提交**，等待云主机导入完成，云主机的可用状态将变为“可用”。
4. 返回集群列表页面，集群状态变为**运行中**，可用云主机数量变为1。

集群关联的系统命名空间

命名空间（Namespace）是对一组资源和对象的抽象集合，用于对服务相互访问的隔离，在网络连通性的前提下，同一命名空间内的服务可以相互发现和相互调用。
创建每个集群时会自动创建一个系统命名空间，命名规则是 `<cluster-name>_default` 。您可以单击创建好的集群的“ID”，选择**命名空间**页签，可以查看到该命名空间。

您可以直接在该系统命名空间部署应用，也可以重新新建，新建命名空间的具体操作请参见 命名空间管理 。

后续操作：在容器集群中部署应用

通过本文，您已经了解如何在腾讯微服务平台中创建集群。在已创建的容器集群和命名空间中，您可以创建并部署应用，常见的应用类型有：

- 部署 Spring Cloud TSF 应用：Spring Cloud 普通应用使用 TSF SDK 接入，支持 TSF 全栈服务治理、应用性能监控、应用配置管理能力。
- 部署 Spring Cloud 原生应用：原生 Spring Cloud 应用0代码改造即可接入，支持服务发现、服务治理、应用性能监控能力。
- 部署 Service Mesh 应用：Mesh 应用支持 Go/Python/C++ 等不同编程语言，通过 Service Mesh 技术接入 TSF，无需修改代码，支持全套 TSF 服务治理能力。
- 部署 Dubbo 应用：Dubbo 应用使用 TSF Atom-SDK 框架接入，支持 TSF 全栈服务治理、应用配置管理能力。

版权所有：腾讯云计算（北京）有限责任公司

部署 Spring Cloud TSF 应用

最近更新时间：2024-01-19 10:39:01

操作场景

Spring Cloud 普通应用使用 TSF SDK 接入，支持 TSF 全栈服务治理、应用性能监控、应用配置管理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Spring Cloud TSF 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在容器环境中部署 Spring Cloud 普通应用，并实现简单的服务调用。

前提条件

- 已参见 快速创建一个容器集群 创建好一个容器集群并导入云主机。
- 已下载并解压 应用 Demo 。

操作步骤

步骤1：新建应用

1. 在左侧导航栏，单击**应用管理**，进入应用列表页。
2. 在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

- **应用名**：填写 provider-demo。
- **部署方式**：选择 **容器部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。

- **开发框架**：选择 SpringCloud。
- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情请参见 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击**确认**，前往上传镜像并部署应用。

步骤2：上传镜像

1. 在镜像标签页面，单击**上传程序包/镜像**。
2. 在个人仓库上传程序包页面，上传程序包。
   - **文件上传方式**：选择**JAR包部署**。
   - **上传程序包**：单击**选择文件**，选择提前准备好的 Demo 中的 `provider-demo-1.29.0-Finchley-RELEASE` jar 程序包。
   - **程序包版本**：填写版本号，或单击**用时间戳作为版本号**。
   - **备注**：填写备注。
3. 单击**上传程序包并制作镜像**，我们将自动为您制作镜像并上传到镜像仓库，右上角将出现任务进行的状态。
4. 任务完成后，在镜像标签页的镜像列表中将看到上传好的镜像。

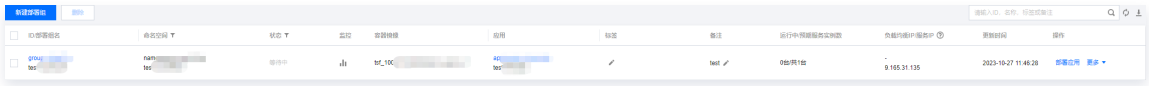
步骤3：部署应用

1. 在应用列表中，单击在 步骤1：新建应用 中创建的应用的 “ID”。
2. 在部署组页面，单击**新建部署组**，设置部署组相关信息。
   - **组名**：填写 provider。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：用于采集应用的业务日志数据，此处可选择**无**。关于日志配置项的使用说明可参见 日志配置项 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
3. 单击**保存&下一步**，进入部署应用页面。
4. 设置部署相关信息。

版权所有：腾讯云计算（北京）有限责任公司

- **选择镜像**：选择 步骤2：上传镜像 中推送到镜像仓库的镜像版本。
- **启动参数**（选填）：设置 Java 应用的启动参数。
- **资源配置**：应用容器的 CPU 和内存限制使用默认值即可，实例数设置为1。
- **访问配置**：
  - 网络访问方式决定了部署组内应用的网络属性，不同访问方式的应用可以提供不同网络能力。此处设置**集群内访问**。
  - 端口映射中选择 TCP 协议，容器端口和服务端口设置为18081。

5. 单击**提交**，完成应用部署。应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。



6. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

7. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查看上报的 API 定义。

步骤4：验证服务调用

使用与之前相同的流程部署一组 consumer-demo 和 provider-demo。

请求 consumer 来调用 provider

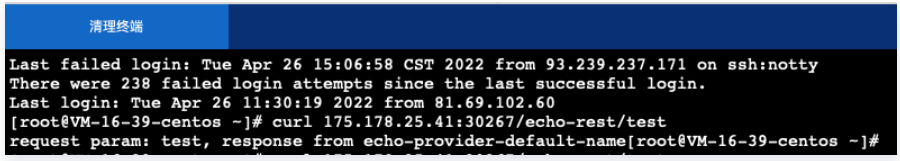
1. 在 部署组 列表页面，单击 consumer 部署组的“ID”，进入服务实例列表页面。
2. 在页面上方选择“基本信息”页签，在**服务访问**模块获取**主机端口（NodePort）**。

3. 在集群页面找到目标集群的ID，点进去，进入云主机列表页面，获取集群中任一云主机的 IP。

4. 单击云主机操作栏的**登录**，输入登录密码，登录云服务器。
5. 执行 curl 命令调用 provider 服务，其中 `<云主机 IP>` 和 `<NodePort>` 为上述步骤获取的主机端口（NodePort）和云主机公网 IP。

```
curl <云主机 IP>:<NodePort>/echo-rest/test
```

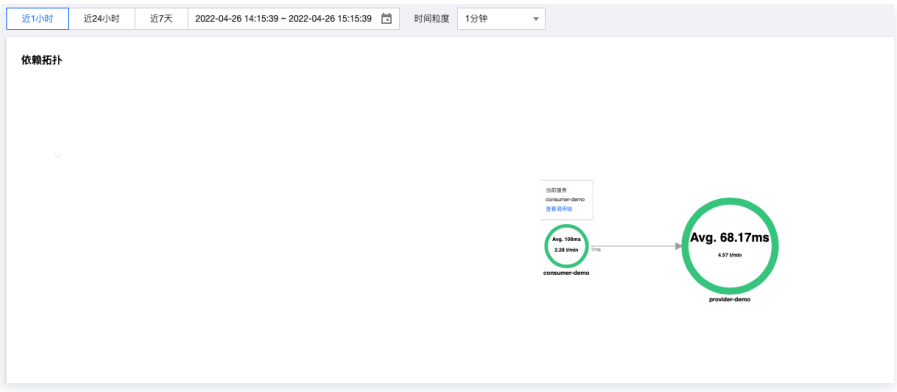
调用结果如下：



在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 provider 和 consumer 服务的运行状况。服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 单击 consumer-demo 服务的“ID”，进入服务详情页面，可以看到两个服务的依赖关系。



在选中时间范围内， consumer-demo 调用了 provider-demo 服务，调用成功比例为 100%（绿色部分）。其中平均每次调用耗时106ms，请求频率为每分钟2.28次。

部署 Spring Cloud 原生应用

最近更新时间：2024-01-19 10:39:01

操作场景

原生 Spring Cloud 应用0代码改造即可接入，支持服务发现、服务治理、应用性能监控能力。如果您想了解如何实现微服务应用的相关功能，请参见 Spring Cloud 原生应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在容器环境中部署 Spring Cloud 原生应用，并实现简单的服务调用。

前提条件

- 已参见 快速创建一个容器集群 创建好一个容器集群并导入可用的云主机。
- 已下载并解压 应用 Demo 。

操作步骤

步骤1：新建应用

1. 在左侧导航栏，单击**应用管理**，进入应用列表页。
2. 在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

版权所有：腾讯云计算（北京）有限责任公司

- **应用名**：填写 consul-provider。
- **部署方式**：选择 **容器部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **SpringCloud**。
- **应用类型**：选择 **原生应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击**确认**，前往上传镜像并部署应用。

步骤2：上传镜像

1. 在镜像标签页面，单击**上传程序包/镜像**。
2. 在个人仓库上传程序包页面，上传程序包。
   - 文件上传方式：选择**JAR包部署**。
   - 上传程序包：单击**选择文件**，选择提前准备好的 Demo 中的 `consul-provider-0.1.1-SNAPSHOT` 的 jar 程序包。
   - 程序包版本：填写版本号，或单击**用时间戳作为版本号**。
   - 备注：填写备注。
3. 单击**上传程序包并制作镜像**，我们将自动为您制作镜像并上传到镜像仓库，右上角将出现任务进行的状态。
4. 任务完成后，在镜像标签页的镜像列表中将看到上传好的镜像。

步骤3：部署应用

1. 在应用列表中，单击在 步骤1. 新建应用 中创建的应用的 “ID” 。
2. 在部署组页面，单击**新建部署组**，设置部署组相关信息。
   - **组名**：填写 consumer。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：用于采集应用的业务日志数据，此处可选择**无**。关于日志配置项的使用说明可参见 日志配置项 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
3. 单击**保存&下一步**，进入部署应用页面。

版权所有：腾讯云计算（北京）有限责任公司

4. 设置部署相关信息。

部署组名称
provider
镜像仓库类型
个人仓库
企业仓库
选择镜像
输入关键字搜索
镜像名称
镜像版本
tsf_100011913960/consul-provider
20220426111913
若现有的镜像版本不合适，您可以推送镜像
数据卷
添加数据卷
为容器提供存储，目前支持临时路径、主机路径、云硬盘数据卷、文件存储NFS、配置文件、PVC，还需挂载到容器的指定路径中。
启动参数
-Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m
环境变量
自定义
JAVA_TOOL_OPTIONS
-Xloggc:/data/tsf_apm/monitor/j
新增变量
变量名只能包含字母、数字及分隔符("-"、"_"、".")，且必须以字母或"_"开头
展开高级设置
资源配置
应用容器
CPU限制
内存限制
request
0.25
limit
0.5
核
request
640
limit
1280
MiB
request 用于预分配资源，当集群中的节点没有 request 所要求的资源数量时，会导致无法创建容器。
limit 用于设置容器使用资源的最大上限，避免异常情况下节点资源消耗过多。
agent容器
部署 agent 容器
CPU限制：request 0.1 核 limit 0.2 核； 内存限制：request 125 MiB limit 400 MiB
实例数量
1
个

- **选择镜像**：选择 步骤2：上传镜像 中推送到镜像仓库的镜像版本。
- **启动参数**（选填）：设置 Java 应用的启动参数。
- **资源配置**：应用容器的 CPU 和内存限制使用默认值即可，实例数设置为1。
- **访问配置**：
  - 网络访问方式决定了部署组内应用的网络属性，不同访问方式的应用可以提供不同网络能力。此处设置**主机端口访问**。
  - 端口映射中选择 TCP 协议，容器端口和服务端口设置为8002。

5. 单击**提交**，完成应用部署。应用部署成功后，部署组中**运行中/预期服务实例数**的数值发生变化。

6. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

7. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查看上报的 API 定义。

步骤4：验证服务调用

使用与之前相同的流程部署一组 consumer 和 provider（如 consul-consumer 和 consul-provider）。

请求 consumer 来调用 provider

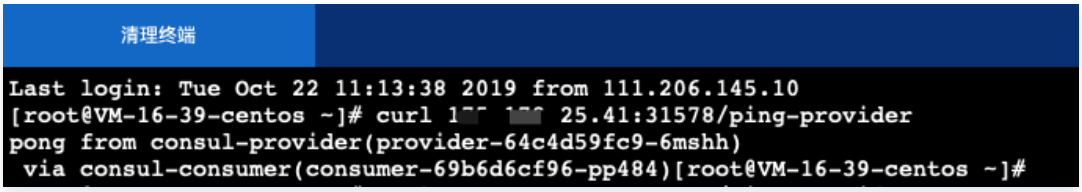
1. 在 部署组 列表页面，单击 consumer 部署组的“ID”，进入服务实例列表页面。
2. 在页面上方选择“基本信息”页签，在**服务访问**模块获取**主机端口**（NodePort）。

3. 在集群页面找到目标集群的ID，点进去，进入云主机列表页面，获取集群中任一云主机的 IP。

4. 单击云主机操作栏的**登录**，输入登录密码，登录云服务器。
5. 执行 curl 命令调用 provider 服务，其中 `<云主机 IP>` 和 `<NodePort>` 为上述步骤获取的主机端口（NodePort）和云主机公网 IP。

```
curl <云主机 IP>:<NodePort>/ping-provider
```

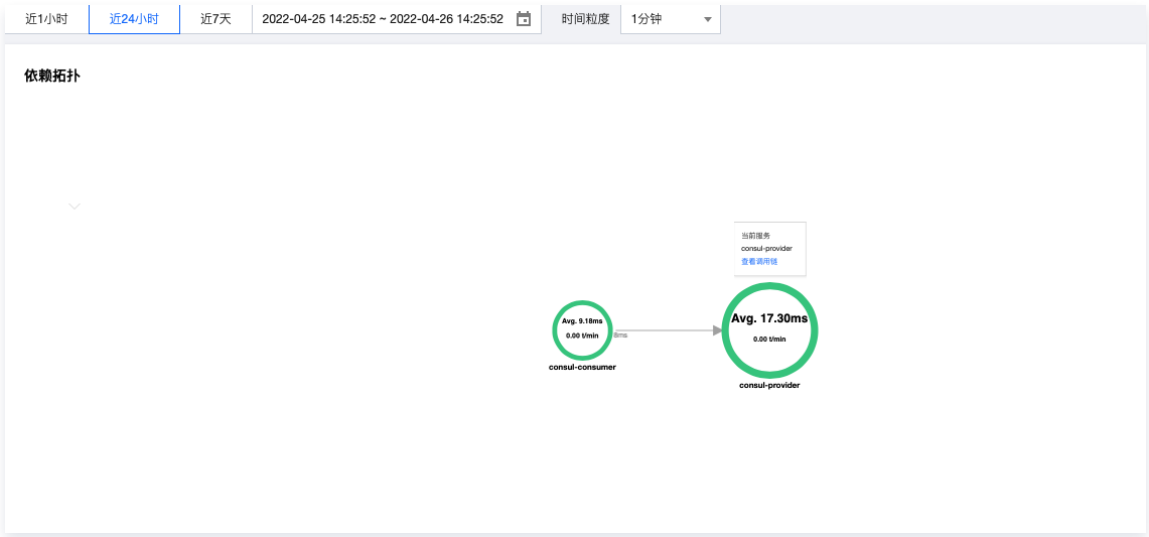
调用结果如下：



在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 consul-provider 和 consul-consumer 服务的运行状况。服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 在服务治理页面，单击 consul-consumer 服务的“ID”，进入服务详情页面，可以看到两个服务的依赖关系。



在选中时间范围内，consul-consumer 调用了 consul-provider 服务，调用成功比例为 100%（绿色部分）。

部署 Service Mesh 应用

最近更新时间：2024-01-19 10:39:01

操作场景

Mesh 应用支持 Go/Python/C++ 等不同编程语言，通过 Service Mesh 技术接入 TSF，无需修改代码，支持全套 TSF 服务治理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Mesh 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一组 Mesh 应用 Demo，包含 user、shop 和 promotion 三个应用。本文以一个示例介绍如何在容器环境中部署 Service Mesh 应用，并实现简单的服务调用。

前提条件

已参见 快速创建一个容器集群 创建好一个容器集群并导入云主机。

操作步骤

步骤1：新建应用

1. 在左侧导航栏，单击**应用管理**，进入应用列表页。
2. 在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

- **应用名**：填写 user。
- **部署方式**：选择 **容器部署**。

- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **其他语言**。
- **开发框架**：选择 **其他框架**。
- **应用类型**：选择 **Mesh应用**。
- **服务配置**：选择使用 **使用本地Spec.yaml**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击**确认**，前往上传镜像并部署应用。

步骤2：上传镜像

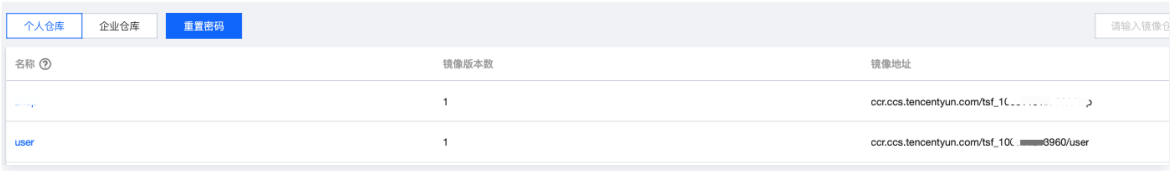
前提条件

- 安装 docker 。
- 使用 `sudo` 允许系统管理员让普通用户执行 `docker` 命令。
- 下载并解压 Demo（包含 user、shop 和 promotion 三个应用）。

1. 初始化镜像仓库

首次使用 镜像仓库 时，需要进行初始化操作，设置登录镜像仓库的登录密码。

TSF 会针对每个容器应用创建一个名为 `tsf_<账号ID>/<应用名>` 的镜像仓库。



2. 制作镜像

1. 解压下载的 Demo 程序包，进入 `demo-mesh-user/` 目录，在 dockerfile 文件所在目录下，执行如下命令制作镜像。

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 <主账号 ID> 对应用户腾讯云的**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在腾讯云控制台账号信息页面获取。），<应用名> 表示控制台上刚刚创建的应用名。[tag] 为镜像的 tag，用户可自定义。
参见示例如下：

```
docker build . -t ccr.ccs.tencentyun.com/tsf_ 3960/user:test
```

2. 命令执行完成后，执行 docker image ls 命令查看创建的镜像。

```
[ @LILYJLLIU-MB0 demo-mesh-user % docker image ls
REPOSITORY                                TAG       IMAGE ID       CREATED          SIZE
ccr.ccs.tencentyun.com/tsf_ 960/user      test      c10851baf09d   8 seconds ago    192MB
```

可查看到该镜像 tag 和 ImageId，这两个参数将用于推送镜像到镜像仓库。
更多关于制作镜像相关的操作请参见 制作容器镜像 。

3. 推送镜像到镜像仓库

1. 在 应用列表 中，单击在 步骤1. 新建应用 中创建的应用“ID/应用名”，进入部署组页面。
2. 选择**镜像**标签页，单击**上传程序包/镜像**，可以获得登录镜像仓库、拉取镜像和推送镜像到仓库的命令。

3. 复制**使用指引**中登录腾讯云 docker registry 的命令并执行。

```
sudo docker login --username=<账号 ID>  ccr.ccs.tencentyun.com
```

> **说明**

> 用户需要输入两次密码，首次为 sudo 密码，第二次为镜像仓库登录密码。
> 命令行工具显示 `Login Succeeded` 即表示登录成功。

4. 登录成功后，复制**使用指引**中给镜像打tag的命令并执行。

```
sudo docker tag [ImageId] ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

其中 [ImageId] 和 [tag] 是在制作镜像时获取。

> **说明**
> 若此时需要输入密码，请输入sudo 密码。

5. 复制**使用指引**中推送镜像到仓库的命令并执行，其中 [tag] 和步骤4相同。

```
sudo docker push ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

运行结果如下：

```
demo-mesh-user % sudo docker push ccr.ccs.tencentyun.com/tsf_      60/user:test
The push refers to repository [ccr.ccs.tencentyun.com/tsf_      3960/user]
d60536810874: Pushed
b244919fdf41: Pushed
7a609c85ccb9: Mounted from qcloud/centos
b6b3bb28b476: Mounted from qcloud/centos
36018b5e9787: Mounted from qcloud/centos
test: digest: sha256:8f561810b64d75283004898eb457b05947e152a3332d2314ac      size: 1359
```

6. 推送镜像成功后，在控制台刷新镜像列表页面，可以看到上传镜像仓库中的镜像。

步骤3：部署应用

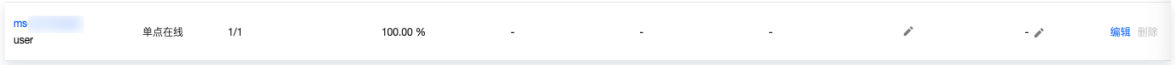
1. 在应用列表中，单击在 步骤1. 新建应用 中创建的应用的 “ID” 。
2. 在部署组页面，单击**新建部署组**，设置部署组相关信息。
   - **组名**：填写 user。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：用于采集应用的业务日志数据，此处可选择**无**。关于日志配置项的使用说明可参见 日志配置项 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
3. 单击**保存&下一步**，进入部署应用页面。
4. 设置部署相关信息。

版权所有：腾讯云计算（北京）有限责任公司

- **选择镜像**：选择 步骤2：上传镜像 中推送到镜像仓库的镜像版本。
- **启动参数**（选填）：设置 Java 应用的启动参数。
- **环境变量**（选填）：此处可不填写。
- **资源配置**：应用容器的 CPU 和内存限制使用默认值即可，实例数设置为1。
- **访问配置**：
  - 网络访问方式决定了部署组内应用的网络属性，不同访问方式的应用可以提供不同网络能力。此处设置为**主机端口访问**。
  - 端口映射中选择 TCP 协议，容器端口和服务端口设置为8089。

5. 单击**提交**，完成应用部署。应用部署成功后，部署组中**运行中/预期服务实例数**的数值发生变化。

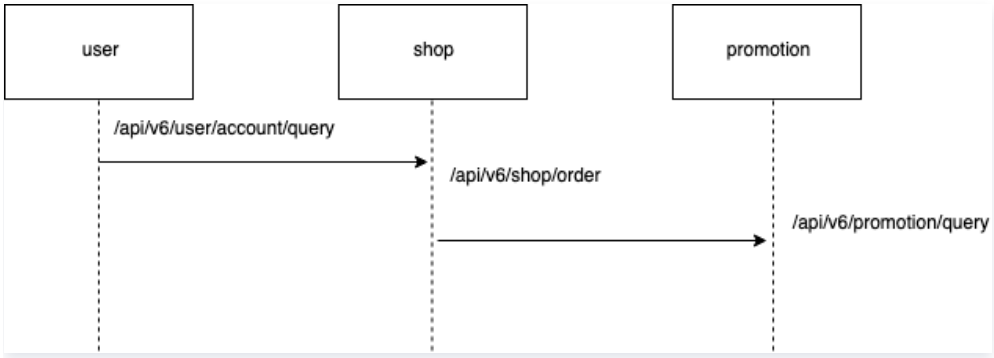
6. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。



7. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查查看上报的 API 定义。

步骤4：验证服务调用

使用同样的步骤部署应用 Demo 中的 shop 和 promotion 应用。user、shop、promotion 三个服务的接口间调用关系如下：



对应的服务名和应用监听端口为：user（8089），shop（8090），promotion（8091）。

1. 触发 user 服务调用 shop 和 promotion 服务

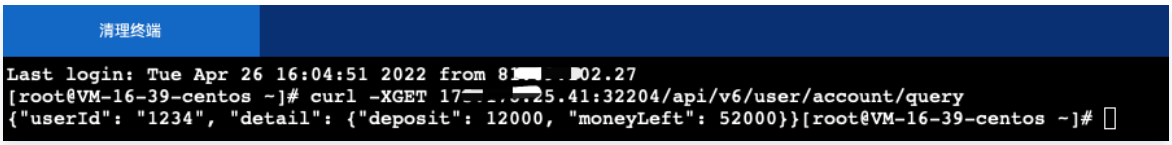
1. 在 部署组 列表页面，单击 user 部署组的“ID”，进入服务实例列表页面。
2. 在页面上方选择“基本信息”页签，在**服务访问**模块获取**主机端口（NodePort）**。

3. 在 集群 页面找到目标集群的ID，点进去，进入云主机列表页面，获取集群中任一云主机的 IP。

4. 单击云主机操作栏的**登录**按钮，输入登录密码，登录云服务器。
5. 执行 curl 命令调用 user 服务，其中 <云主机 IP> 和 <NodePort> 为上述步骤获取的主机端口（NodePort）和云主机 IP。

```
curl <云主机 IP>:<NodePort>/api/v6/user/account/query
```

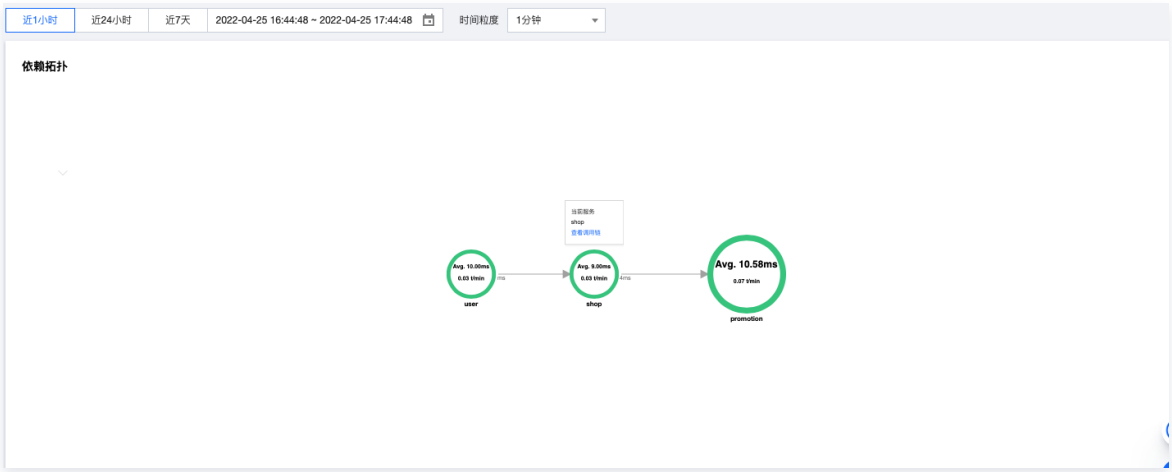
调用结果如下：



2. 在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 user、shop 和 promotion 服务的运行状况。服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 在服务治理页面，单击 shop 服务的“ID”，进入服务详情页面，可以看到三个服务的依赖关系。



部署 Dubbo 应用

最近更新时间：2023-10-09 17:13:31

操作场景

Dubbo 应用使用 TSF Atom-SDK 框架接入，支持 TSF 全栈服务治理、应用配置管理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Dubbo 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在虚拟机环境中部署 Dubbo 普通应用。

前提条件

- 已参见 快速创建一个容器集群 创建好一个虚拟机集群并导入云主机。
- 已下载并解压 Demo 。

操作步骤

步骤1：新建应用

1. 在左侧导航栏，单击**应用管理**，进入应用列表页。
2. 在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

- **应用名**：填写 provider。
- **部署方式**：选择 **容器部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **Dubbo**。

- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击**确认**，前往上传镜像并部署应用。

步骤2：上传镜像

1. 在镜像标签页面，单击**上传程序包/镜像**。
2. 在个人仓库上传程序包页面，上传程序包。
   - **文件上传方式**：选择**JAR包部署**。
   - **上传程序包**：单击**选择文件**，选择提前准备好的 Demo 中的 provider 的 jar 程序包。
   - **程序包版本**：填写版本号，或单击**用时间戳作为版本号**。
   - **备注**：填写备注。
3. 单击**上传程序包并制作镜像**，我们将自动为您制作镜像并上传到镜像仓库，右上角将出现任务进行的状态。
4. 任务完成后，在镜像标签页的镜像列表中将看到上传好的镜像。

步骤3：部署应用

1. 在应用列表中，单击在 步骤1. 新建应用 中创建的应用的 “ID” 。
2. 在部署组页面，单击**新建部署组**，设置部署组相关信息。
   - **组名**：部署组的名称。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：用于采集应用的业务日志数据，此处可选择**无**。关于日志配置项的使用说明可参见 日志配置项 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
3. 单击**保存&下一步**，进入部署应用页面。
4. 设置部署相关信息。

- **选择镜像**：选择 步骤2：上传镜像 中推送到镜像仓库的镜像版本。
- **启动参数**（选填）：设置 Java 应用的启动参数。
- **资源配置**：应用容器的 CPU 和内存限制使用默认值即可，实例数设置为1。

- **访问配置：**
  - 网络访问方式决定了部署组内应用的网络属性，不同访问方式的应用可以提供不同网络能力。此处设置**集群内访问**。
  - 端口映射中选择 TCP 协议，容器端口和服务端口设置为8080。

5. 单击**提交**，完成应用部署。应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

6. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

部署 Spring Cloud TSF 应用（Java Agent 版）

最近更新时间：2023-10-09 17:13:31

操作场景

Spring Cloud 普通应用使用 Java Agent 接入 TSF 平台，支持 TSF 服务注册发现、服务治理、应用配置管理和应用性能监控能力。
Java Agent 是 Java 1.5 版本之后引　的特性，可以被理解为 JVM 虚拟机级别的 AOP，使用该技术可以做到无需对原有应用做任何修改，就可以对原有应用的实现类进行动态修改和增强。利用 Java Agent 技术来做无侵入埋点，可以使得业务无需做任何改造就可以接入 TSF 平台并使用 TSF 一系列配套的能力。
本文主要介绍开源 Spring Cloud 应用如何 0 改造就可以将应用部署接入到 TSF 平台。

Agent 插件功能说明

- 以下表格是目前支持的 Agent 插件以及其对应功能的详细说明

Agent 插件名称	插件说明
服务 Agent	支持 TSF 服务注册发现、服务治理以及应用配置管理能力
可观测 Agent	支持 TSF 应用性能监控能力，包括调用链、数据指标监控等

- 以下表格是 Spring Cloud 版本以及对应支持 Agent 插件的详细说明

开源 Spring Cloud 版本	服务 Agent	可观测 Agent
Spring Cloud 2020	支持	支持

> **说明**
> 目前仅支持 **Spring Cloud 2020** 使用 Java Agent 接入 TSF 平台。

前提条件

1. 已创建好集群并导入云主机，容器场景请参见 容器集群管理 。
2. 下载 官方开源 Spring Cloud 2020 Demo ，在pom.xml文件所在目录下执行 `mvn clean package` 将应用程序打包，在 `femas-agent-example-springcloud-provider/target/` 目录下可以看到打包好的 jar 程序包。

操作步骤

1. 登录 TSF 控制台 。
2. 这里跳过 **新建应用** 步骤，直接到 **部署应用配置 Agent 插件**，如果需要了解 **新建应用** 和其他 **部署应用** 相关操作可以参见 容器应用管理 。
3. **部署应用配置 Agent 插件**：在部署容器应用的时候单击**展开高级设置**，勾选相应的 Agent 插件进行部署（建议服务 Agent 和可观测 Agent 都选上）。

版权所有：腾讯云计算（北京）有限责任公司

4. 其他部署配置都选择好以后，单击 **提交** 进行部署。
5. 完成部署以后的服务调用和拓扑图展示等验证操作继续参考 操作步骤第2点 中的容器应用管理文档。

在虚拟机环境中部署微服务
快速创建一个虚拟机集群

最近更新时间：2023-10-09 17:13:31

操作场景

部署应用之前需要新建一个集群，集群是指云资源管理的集合，包含了运行应用的云主机等资源。TSF 中的集群分为虚拟机集群和容器集群。使用虚拟机或者容器计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。
本文介绍在 TSF 控制台快速创建一个虚拟机集群并导入云主机的操作方法。

前提条件

- 已 获取访问授权
- 已 购买云服务器

新建集群

1. 登录 TSF 控制台 。
2. 在左侧导航栏中单击**集群**，选择好地域后，单击**新建**。
3. 设置集群的基本信息。

   - 集群类型：选择**虚拟机集群**。

- 集群名：填写集群名称，最长60个字符。
- 所在可用区：选择集群所在的可用区，建议选择**默认可用区**。
- 集群网络：选择与已购买云服务器相同的 VPC 网络，用来保证后续导入集群的云服务器属于同一 VPC。
- 标签：可选，可留空。用于分类管理资源，详情参见 标签 。此处不填写。
- 备注：可选，可留空。集群的描述，不超过200个字符。
- 数据集：可选，可留空。用于细粒度管理子账号数据权限，详情参见 数据集管理 。

4. 单击**提交**，在弹窗中选择**确认**，进入导入云主机页面。如果选择**取消**，您后续可以在集群列表中，单击操作列的**导入云主机**重新导入云主机。

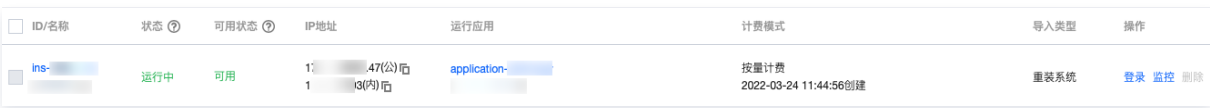
导入云主机

使用虚拟机计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。

1. 从集群所在 VPC 的云主机列表中，选择需要添加到集群的云主机。
2. 单击**下一步**，进行云主机配置。

- 导入方式：选择**重装系统**。
- 操作系统：支持两种不同操作系统选择，Ubuntu 18.04版本与 CentOS 7.5版本。
- 登录方式：选择**设置密码**，并根据提示设置对应密码。

3. 单击**提交**，导入的云主机将出现在云主机名列表中。等待几分钟，刷新列表，正常情况下云主机的状态将变为运行中，可用状态变为"可用"。



4. 返回集群列表页面，集群状态变为**运行中**，可用云主机数量变为1。

集群关联的系统命名空间

命名空间（Namespace）是对一组资源和对象的抽象集合，用于对服务相互访问的隔离，在网络连通性的前提下，同一命名空间内的服务可以相互发现和相互调用。
创建每个集群时会自动创建一个系统命名空间，命名规则是 `<cluster-name>_default` 。您可以单击创建好的集群的“ID”，选择**命名空间**页签，可以查看到该命名空间。

您可以直接在该系统命名空间部署应用，也可以重新新建，新建命名空间的具体操作请参见 命名空间管理 。

后续操作：在虚拟机集群中部署应用

通过本文，您已经了解如何在腾讯微服务平台中创建虚拟机集群，在已创建的虚拟机集群和命名空间中，您可以创建并部署应用，常见的应用类型有：

- 部署 Spring Cloud TSF 应用：Spring Cloud 普通应用使用 TSF SDK 接入，支持 TSF 全栈服务治理、应用性能监控、应用配置管理能力。
- 部署 Spring Cloud 原生应用：原生 Spring Cloud 应用0代码改造即可接入，支持服务发现、服务治理、应用性能监控能力。
- 部署 Service Mesh 应用：Mesh 应用支持 Go/Python/C++ 等不同编程语言，通过 Service Mesh 技术接入 TSF，无需修改代码，支持全套 TSF 服务治理能力。
- 部署 Dubbo 应用：Dubbo 应用使用 TSF Atom-SDK 框架接入，支持 TSF 全栈服务治理、应用配置管理能力。

版权所有：腾讯云计算（北京）有限责任公司

部署 Spring Cloud TSF 应用

最近更新时间：2024-01-19 10:39:01

操作场景

Spring Cloud 普通应用使用 TSF SDK 接入，支持 TSF 全栈服务治理、应用性能监控、应用配置管理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Spring Cloud TSF 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在虚拟机环境中部署 Spring Cloud 普通应用，并实现简单的服务调用。

前提条件

已参见 快速创建一个虚拟机集群 创建好一个虚拟机集群并导入可用的云主机。

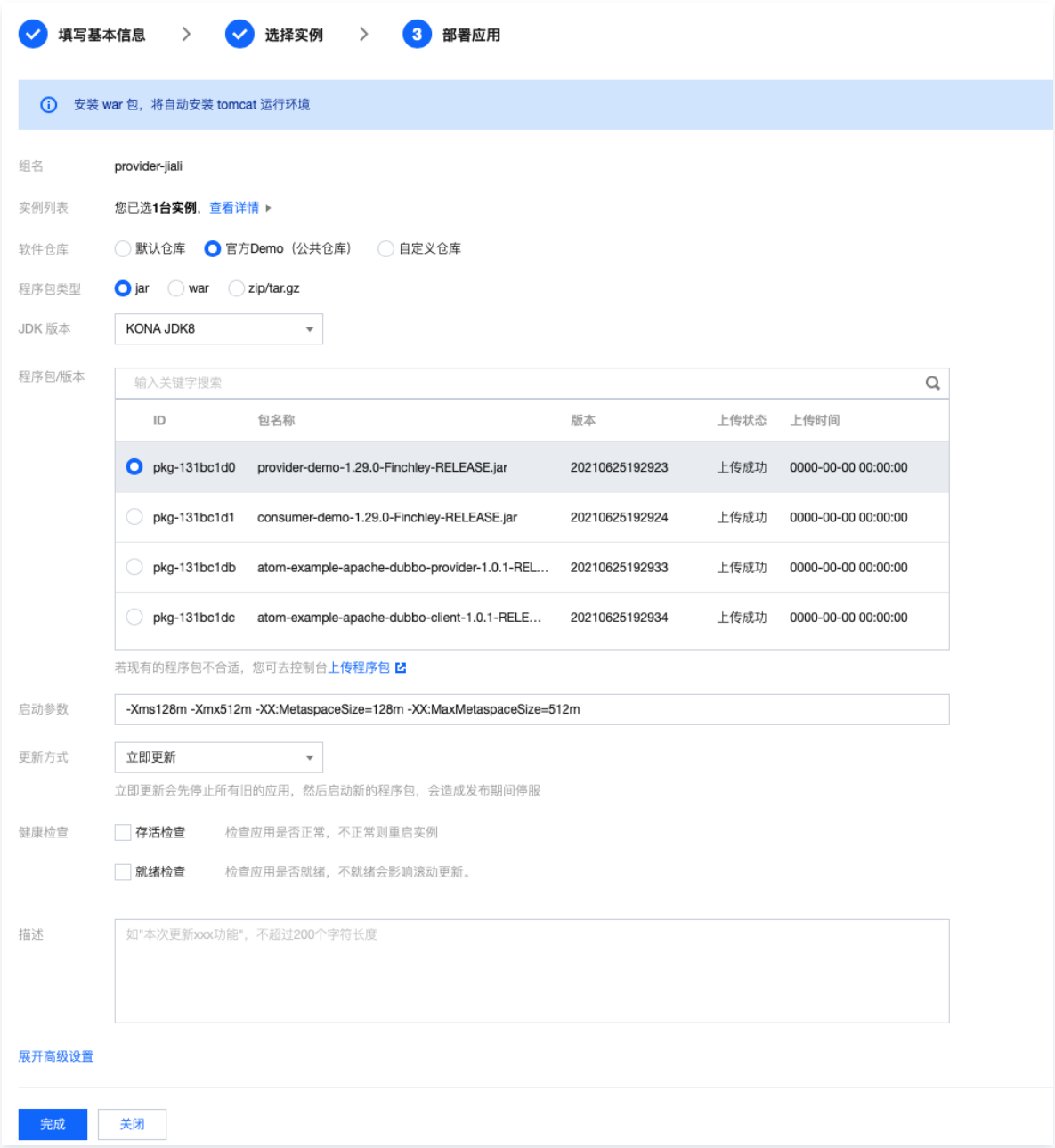
部署应用

1. 登录 TSF 控制台 。
2. 在左侧导航栏选择**应用管理**，选择好地域后，在应用列表上方单击**新建应用**。
3. 设置应用信息后，单击**提交**。

- **应用名**：填写 provider-demo。
- **部署方式**：选择 **虚拟机部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 JAVA。
- **开发框架**：选择 SpringCloud。
- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情请参见 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的提醒弹窗中选择**使用官方 Demo 新建部署组**，进入新建部署组页面。

5. 在新建部署组页面，填写部署组信息。
   - **组名**：部署组的名称，不超过60个字符。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：应用的日志配置项用于指定 TSF 采集应用的日志路径。此处可选择**无**。参考 日志服务 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参考 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参考 标签 。
   - **备注**：选填，可留空。
6. 单击**保存&下一步**，从关联集群的可用云主机列表勾选用于部署的云主机。
7. 单击**部署应用**，设置部署信息。



- 软件仓库：选择**官网Demo（公共仓库）**。
- 程序包类型：选择 **jar**。
- JDK 版本：选择 **KONA JDK8**。
- 程序包类型：选择名称为 `provider-demo-1.29.0-Finchley-RELEASE.jar` 的程序包。
- 启动参数：选填。
- 更新方式：选择立即更新。
- 健康检查：可选。详情参见 健康检查 。
- 描述：可选。

8. 单击**完成**，应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

9. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

10. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查看上报的 API 定义。

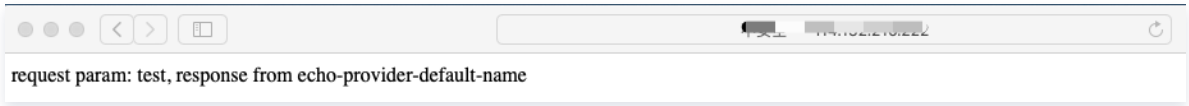
验证服务调用

使用与 部署应用 Demo 相同的流程在同一个集群和命名空间下部署一个 consumer-demo 服务，可以通过以下两种方式访问触发 consumer 服务调用 provider 服务。

方式1：使用公网访问验证服务间调用

1. 在集群云主机列表页面，获取 consumer 服务所在云服务器的公网 IP。

2. 使用浏览器访问 consumer 应用的 URL 并调用 provider 服务，格式为 `http://<云服务器实例公网 IP>:18083/echo-rest/test` ，返回结果如下：



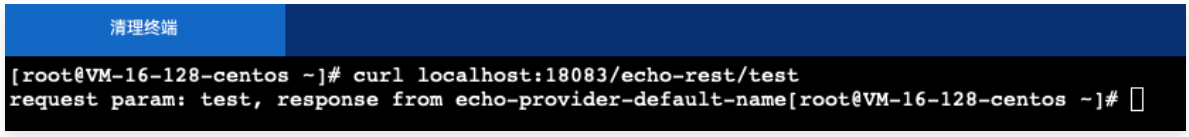
方式2：登录云服务器验证服务间调用

1. 在集群云主机列表页面，单击 consumer 服务所在云服务器操作栏的**登录**，输入登录密码，登录云服务器。

2. 执行 curl 命令调用 provider 服务。

```
curl localhost:18083/echo-rest/test
```

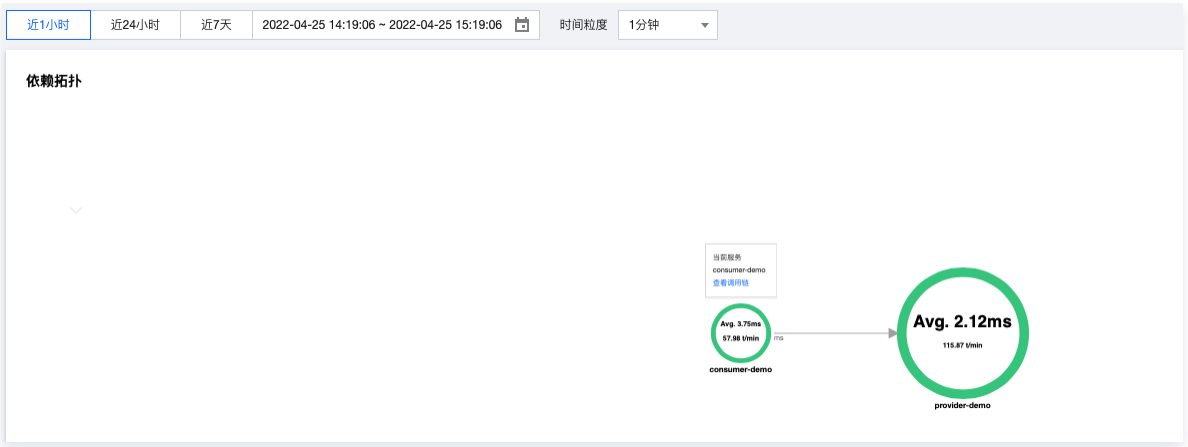
调用结果如下：



在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 provider 和 consumer 服务的运行状况。服务状态为**单点在线**，表示服务被代理注册成功。服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 单击 consumer 服务的“ID”，进入服务详情页面，可以看到两个服务的依赖关系。



在选中时间范围内，consumer-demo 调用了 provider-demo 服务，调用成功比例为 100%（绿色部分）。其中平均每次调用耗时 3.75ms，请求频率为每分钟57.98次。

版权所有：腾讯云计算（北京）有限责任公司

部署 Spring Cloud 原生应用

最近更新时间：2024-01-19 10:39:01

操作场景

原生 Spring Cloud 应用0代码改造即可接入 TSF，支持服务发现、服务治理、应用性能监控能力。如果您想了解如何实现微服务应用的相关功能，请参见 Spring Cloud 原生应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在虚拟机环境中部署 Spring Cloud 原生应用，并实现简单的服务调用。

前提条件

已参见 快速创建一个虚拟机集群 创建好一个虚拟机集群并导入可用的云主机。

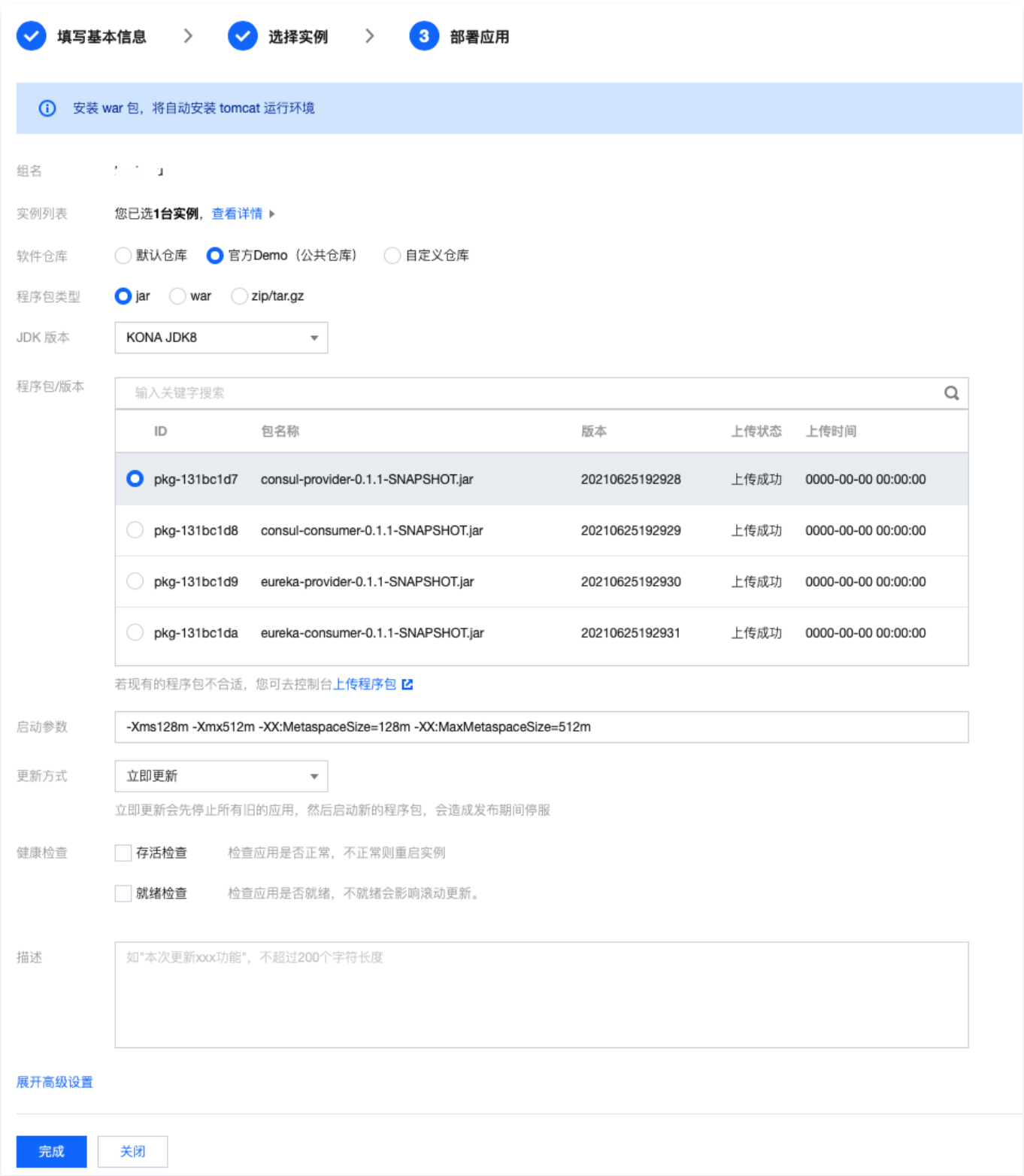
部署应用

1. 登录 TSF 控制台 。
2. 在左侧导航栏，单击**应用管理**，选择好地域后，在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

- **应用名**：填写 **consul-provider**。
- **部署方式**：选择 **虚拟机部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **SpringCloud**。
- **应用类型**：选择 **原生应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的提醒弹窗中选择**使用官方Demo新建部署组**，进入新建部署组页面。

5. 在新建部署组页面，填写部署组信息。
   - **组名**：部署组的名称，不超过60个字符。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：应用的日志配置项用于指定 TSF 采集应用的日志路径。此处可选择**无**。参见 日志服务 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
6. 单击**保存&下一步**，从关联集群的可用云主机列表勾选用于部署的云主机。
7. 单击**部署应用**，设置部署信息。



- 软件仓库：选择**官网Demo（公共仓库）**。
- 程序包类型：选择 **jar**。
- JDK 版本：选择 **KONA JDK8**。
- 程序包类型：选择名称为 `consul-provider-0.1.1-SNAPSHOT.jar` 的程序包。
- 启动参数：选填。
- 更新方式：选择立即更新。
- 健康检查：可选。详情参见 健康检查 。

- 描述：可选。

8. 单击**完成**，应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

9. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

10. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查看上报的 API 定义。

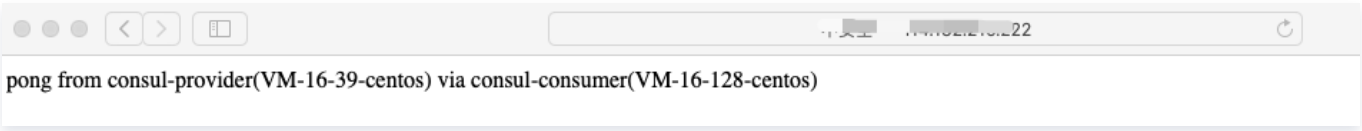
验证服务调用

使用与 部署应用 Demo 相同的流程在同一个集群和命名空间下部署一个 consul-consumer 服务，可以通过以下两种方式访问触发 consul-consumer 服务调用 consul-provider 服务。

方式1：使用公网访问验证服务间调用

1. 在集群云主机列表页面，获取 consumer 服务所在云服务器的公网 IP。

2. 使用浏览器访问 consumer 应用的 URL 并调用 provider 服务，格式为 `http://<云服务器实例公网IP>:8001/ping-provider`，返回结果如下：



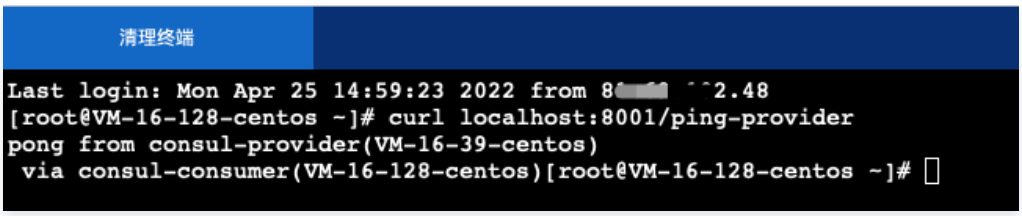
方式2：登录云服务器验证服务间调用

1. 在集群云主机列表页面，单击 consul-consumer 服务所在云服务器操作栏的**登录**，输入登录密码，登录云服务器。

2. 执行 curl 命令调用 provider 服务。

```
curl localhost:8001/ping-provider
```

调用结果如下：



在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 provider 和 consumer 服务的运行状况。服务状态为**单点在线**，表示服务被代理注册成功。服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 在服务治理页面，单击 consumer 服务的“ID”，进入服务详情页面，可以看到两个服务的依赖关系。

在选中时间范围内，consul-consumer 调用了 consul-provider 服务，调用成功比例为 100%（绿色部分）。其中平均每次调用耗时 7.77ms，请求频率为每分钟0.03次。

部署 Service Mesh 应用

最近更新时间：2024-01-19 10:39:01

操作场景

Mesh 应用支持 Go/Python/C++ 等不同编程语言，通过 Service Mesh 技术接入 TSF，无需修改代码，支持全套 TSF 服务治理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Mesh 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一组 Mesh 应用 Demo，包含 user、shop 和 promotion 三个应用。本文以一个示例介绍如何在虚拟机环境中部署 Service Mesh 应用，并实现简单的服务调用。

前提条件

已参见 快速创建一个虚拟机集群 创建好一个虚拟机集群并导入可用的云主机。

部署应用

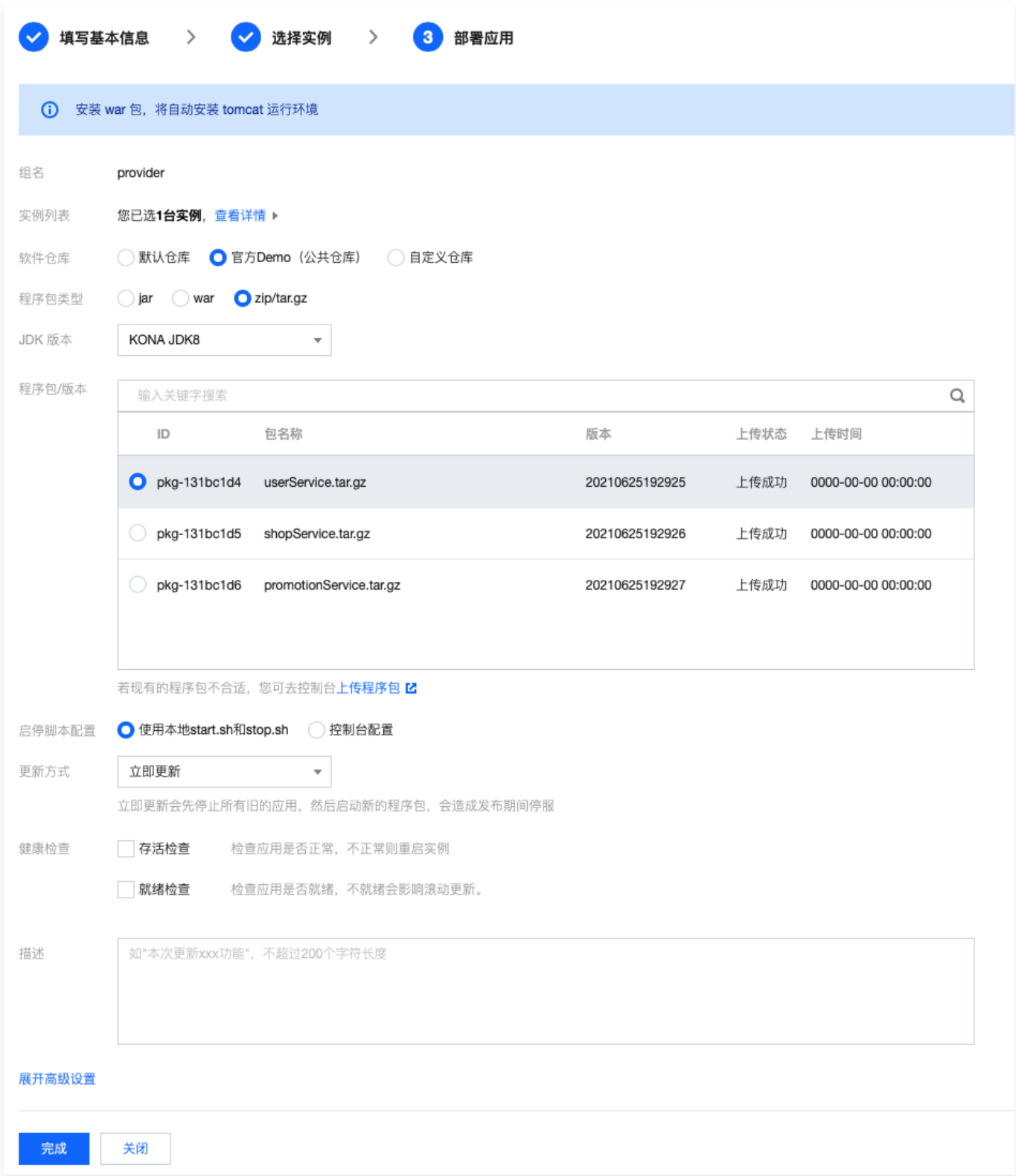
1. 登录 TSF 控制台 。
2. 在左侧导航栏，单击**应用管理**，选择好地域后，在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

版权所有：腾讯云计算（北京）有限责任公司

- **应用名**：填写 **user**。
- **部署方式**：选择 **虚拟机部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **其他语言**。
- **开发框架**：选择 **其他框架**。
- **应用类型**：选择 **Mesh 应用**。
- **服务配置**：选择**使用本地Spec.yaml**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的提醒弹窗中选择**使用官方Demo新建部署组**，进入新建部署组页面。

5. 在新建部署组页面，填写部署组信息。
   - **组名**：部署组的名称，不超过60个字符。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：应用的日志配置项用于指定 TSF 采集应用的日志路径。此处可选择**无**。参见 日志服务 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
6. 单击**保存&下一步**，从关联集群的可用云主机列表勾选用于部署的云主机。
7. 单击**部署应用**，设置部署信息。

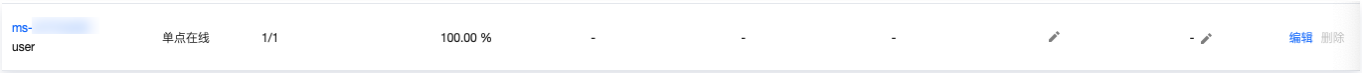


- 软件仓库：选择**官网Demo（公共仓库）**。
- 程序包类型：选择 **zip/tar.gz**。
- JDK 版本：选择**KONA JDK8**。
- 程序包类型：选择程序包名称为 `userService.tar.gz` 的程序包。
- 启动参数：选填。
- 更新方式：选择立即更新。
- 健康检查：可选。详情参见 健康检查 。

- 描述：可选。

8. 单击**完成**，应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

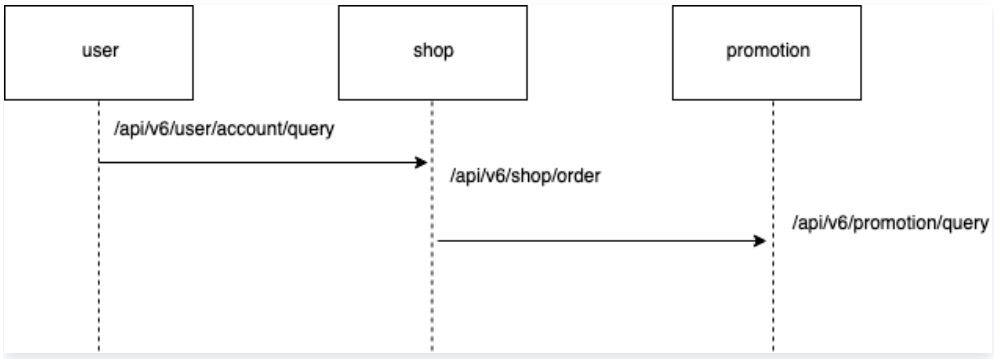
9. 在 服务治理 页面，选择地域和应用关联的命名空间后，可以看到服务实例显示**单点在线**状态，表示服务注册成功。



10. 在服务列表页单击服务的“ID”，进入服务详情页，单击**接口列表**标签页，可以查看上报的 API 定义。

验证服务调用

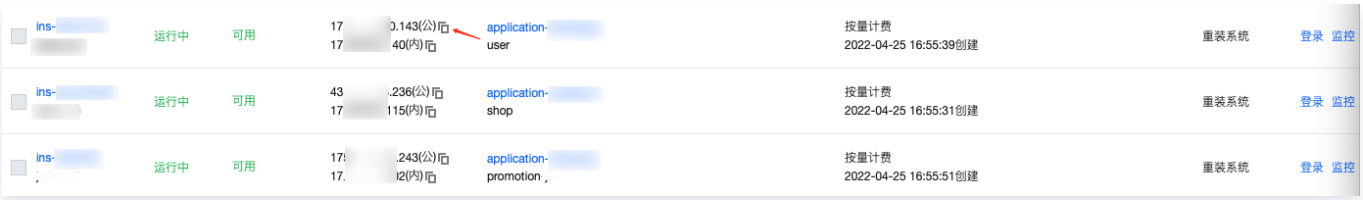
使用同样的步骤在同一个集群和命名空间中部署 user、shop 和 promotion 三个应用。user、shop、promotion 三个服务的接口间调用关系如下：



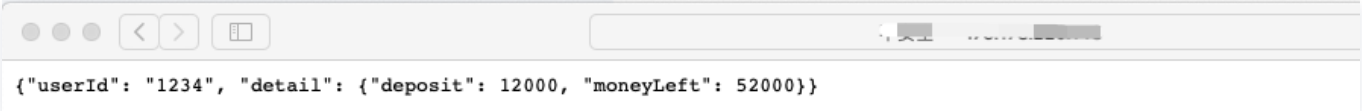
对应的服务名和应用监听端口为：user（8089），shop（8090）和 promotion（8091）。
用户可以登录虚拟机集群 VPC 下的任一机器，然后通过 `curl` 命令验证 user 服务是否健康，以及触发 user 服务调用 shop 和 promotion 服务。

方式1：使用公网访问验证服务调用

1. 在集群云主机列表页面，获取 user 服务所在云服务器的公网IP。

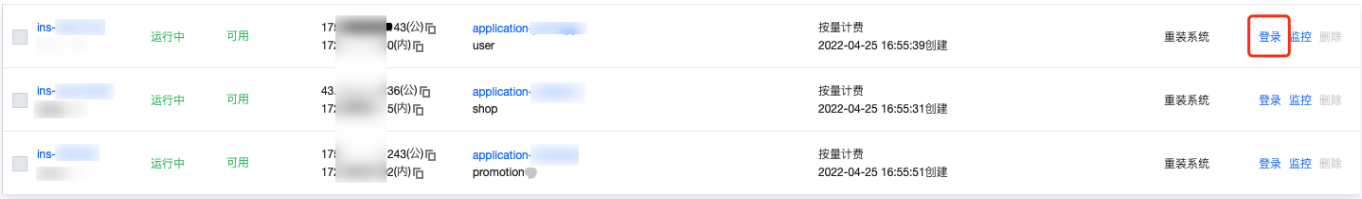


2. 使用浏览器访问 consumer 应用的 URL 并触发 user 服务调用 shop 和 promotion 服务。格式为 `http://<云服务器实例公网IP>:8089/api/v6/user/account/query`，返回结果如下：



方式2：登录云服务器验证服务间调用

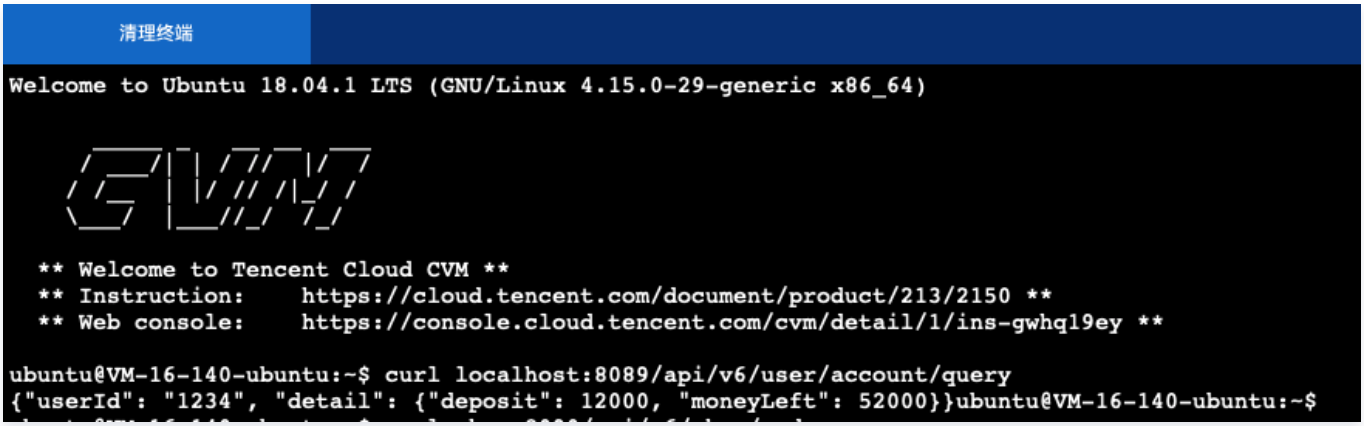
1. 在集群云主机列表页面，单击 user 服务所在云服务器操作栏的**登录**，输入登录密码，登录云服务器。



2. 执行 curl 命令调用 user 服务接口。

```
curl localhost:<user端口>/api/v6/user/account/query
```

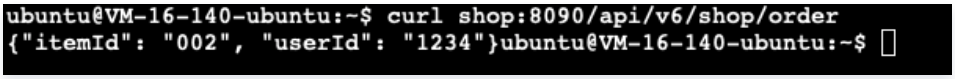
调用结果如下：



或者执行 curl 命令调用 shop 服务接口（注意使用服务名来调用）。

```
curl shop:<shop端口>/api/v6/shop/order
```

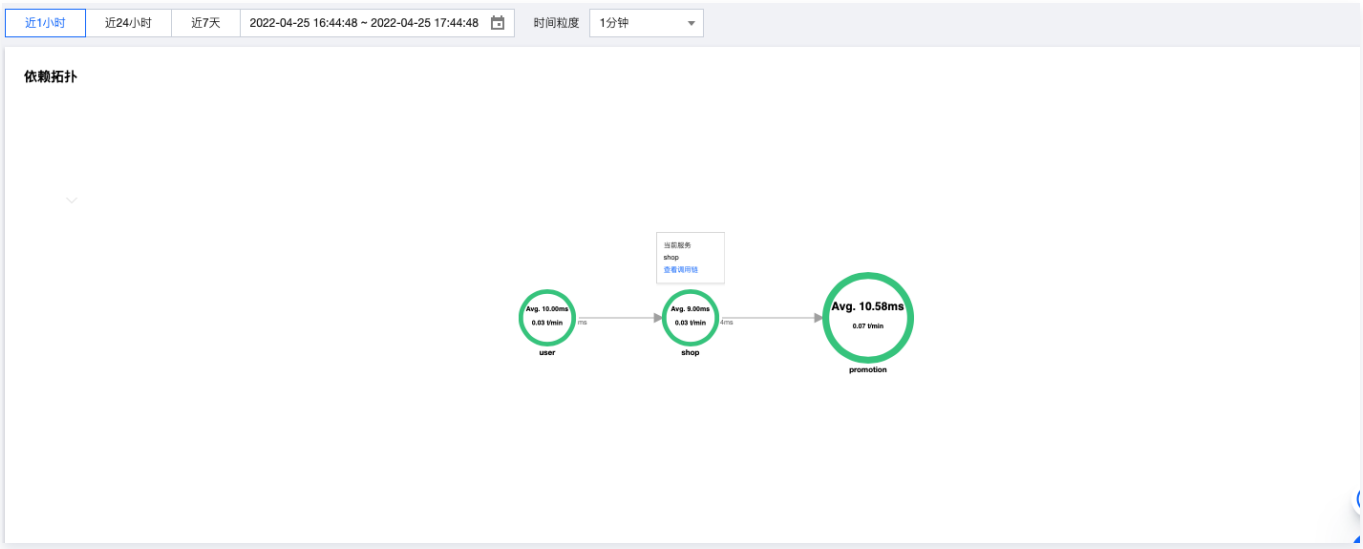
调用结果如下



在 TSF 控制台查看服务依赖拓扑图

1. 在 服务治理 页面，选择创建集群和命名空间后，可以看到 user、shop 和 promotion 服务的运行状况。服务状态为**单点在线**，表示服务被代理注册成功。服务提供者的请求量大于0，请求成功率为100%，表示服务提供者被服务消费者请求成功。

2. 在服务治理页面，单击 shop 服务的“ID”，进入服务详情页面，可以看到三个服务的依赖关系。


近1小时
近24小时
近7天
2022-04-25 16:44:48 ~ 2022-04-25 17:44:48
时间粒度
1分钟
依赖拓扑
当前服务
shop
查看调用链
Avg. 10.00ms
0.03 t/min
user
Avg. 9.00ms
0.03 t/min
shop
Avg. 10.58ms
0.07 t/min
promotion

部署 Dubbo 应用

最近更新时间：2024-01-19 10:39:01

操作场景

Dubbo 应用使用 TSF Atom-SDK 框架接入，支持 TSF 全栈服务治理、应用配置管理能力。如果您想了解如何实现微服务应用的相关功能，请参见 Dubbo 应用开发 。
为了帮助您快速体验如何在 TSF 中部署微服务应用，TSF 提供了一对应用 Demo，包含一个 provider 应用和一个 consumer 应用。本文以一个示例介绍如何在虚拟机环境中部署 Dubbo 应用。

前提条件

已参见 快速创建一个虚拟机集群 创建好一个虚拟机集群并导入两个可用的云主机。

部署应用

1. 登录 TSF 控制台 。
2. 在左侧导航栏，单击**应用管理**，选择好地域后，在应用列表上方单击**新建应用**。
3. 设置应用信息，单击**提交**。

- **应用名**：填写 provider。
- **部署方式**：选择 **虚拟机部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **Dubbo**。
- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。详情参见 标签 。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限，详情参阅 数据集管理 。
- **备注**：选填，可留空。

4. 在弹出的提醒弹窗中选择**使用官方Demo新建部署组**，进入新建部署组页面。

5. 在新建部署组页面，填写部署组信息。
   - **组名**：部署组的名称，不超过60个字符。
   - **集群**：选择提前创建好的集群。
   - **命名空间**：选择集群关联的默认命名空间。
   - **日志配置项**：应用的日志配置项用于指定 TSF 采集应用的日志路径。此处可选择**无**。参见 日志服务 。
   - **日志投递**：用于日志转储，此处可选择**无**。关于日志投递的详情说明可参见 日志投递 。
   - **标签**：用于分类管理资源，可不选。详情参见 标签 。
   - **备注**：选填，可留空。
6. 单击**保存&下一步**，从关联集群的可用云主机列表勾选用于部署的云主机。
7. 单击**部署应用**，设置部署信息。

- 软件仓库：选择**官网Demo（公共仓库）**。
- 程序包类型：选择 **jar**。
- JDK 版本：选择**KONA JDK8**。
- 程序包类型：选择程序包名称为 `atom-example-apache-dubbo-provider-1.0.1-RELEASE.jar` 的程序包。
- 启动参数：选填。
- 更新方式：选择立即更新。

- 健康检查：可选。详情参见 健康检查 。
- 描述：可选。

8. 单击**完成**，应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

9. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

部署 Spring Cloud TSF 应用（Java Agent 版）

最近更新时间：2022-08-19 09:55:12

操作场景

Spring Cloud 普通应用使用 Java Agent 接入 TSF 平台，支持 TSF 服务注册发现、服务治理、应用配置管理和应用性能监控能力。
Java Agent 是 Java 1.5 版本之后引　的特性，可以被理解为 JVM 虚拟机级别的 AOP，使用该技术可以做到无需对原有应用做任何修改，就可以对原有应用的实现类进行动态修改和增强。利用 Java Agent 技术来做无侵入埋点，可以使得业务无需做任何改造就可以接入 TSF 平台并使用 TSF 一系列配套的能力。
本文主要介绍开源 Spring Cloud 应用如何 0 改造就可以将应用部署接入到 TSF 平台。

Agent 插件功能说明

- 以下表格是目前支持的 Agent 插件以及其对应功能的详细说明

Agent 插件名称	插件说明
服务 Agent	支持 TSF 服务注册发现、服务治理以及应用配置管理能力
可观测 Agent	支持 TSF 应用性能监控能力，包括调用链、数据指标监控等

- 以下表格是 Spring Cloud 版本以及对应支持 Agent 插件的详细说明

开源 Spring Cloud 版本	服务 Agent	可观测 Agent
Spring Cloud 2020	支持	支持

> **说明**
> 目前仅支持 **Spring Cloud 2020** 使用 Java Agent 接入 TSF 平台。

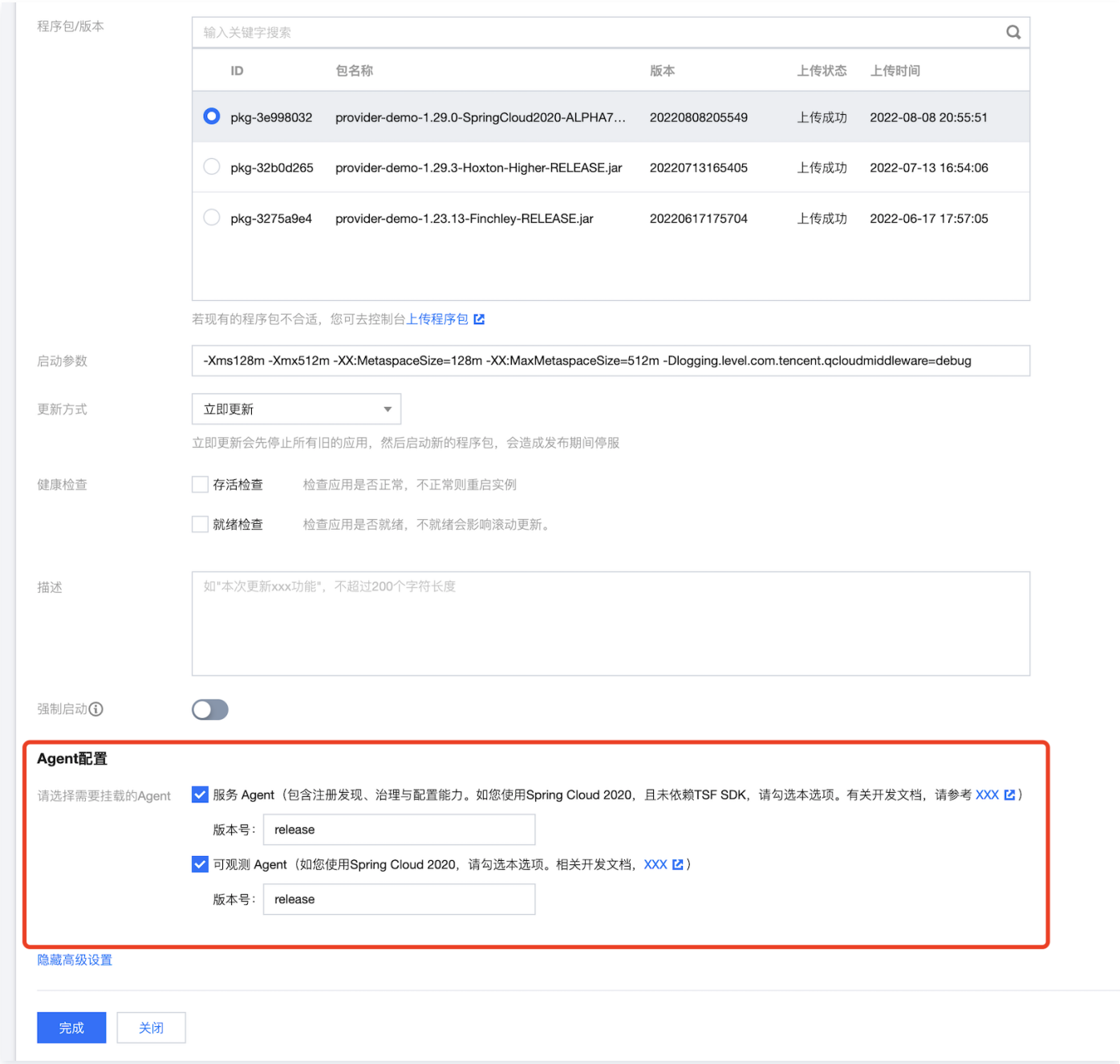
前提条件

1. 已创建好集群并导入云主机，虚拟机场景请参见 虚拟机集群管理 。
2. 下载 官方开源 Spring Cloud 2020 Demo ，在pom.xml文件所在目录下执行 `mvn clean package` 将应用程序打包，在 `femas-agent-example-springcloud-provider/target/` 目录下可以看到打包好的 jar 程序包。

操作步骤

1. 登录 TSF 控制台 。
2. 这里跳过 **新建应用** 步骤，直接到 **部署应用配置 Agent 插件**，如果需要了解 **新建应用** 和其他 **部署应用** 相关操作可以参见 虚拟机应用管理 。
3. **部署应用配置 Agent 插件**：在部署虚拟机应用的时候单击**展开高级设置**，勾选相应的 Agent 插件进行部署（建议服务 Agent 和可观测 Agent 都选上）。

版权所有：腾讯云计算（北京）有限责任公司



4. 其他部署配置都选择好以后，单击 完成 进行部署。

5. 完成部署以后的服务调用和拓扑图展示等验证操作继续参考文档上述虚拟机应用管理文档。