

# 腾讯微服务平台 TSF

## Serverless 微服务平台

### 产品文档



腾讯云

**【 版权声明 】**

©2013–2021 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。

## 文档目录

### Serverless 微服务平台

TSF Serverless 简介

TSF Serverless 购买指南

TSF Serverless 运行环境

TSF Serverless 使用须知

TSF Serverless 操作指南

概述

自定义域名及证书部署示例

使用 TSF Serverless 部署微服务

Express Hello World 示例

Koa Hello World 示例

# Serverless 微服务平台

## TSF Serverless 简介

最近更新时间：2020-12-17 16:36:02

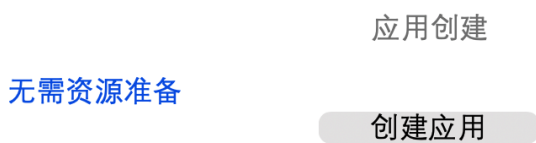
Serverless 微服务平台（Tencent Service Framework Serverless，TSF Serverless）是可承载核心生产应用的一站式 Serverless 开发平台，为用户提供按需使用、按量计费、免运维的使用体验。

### 产品优势

与 TSF 当前的虚拟机部署和容器部署模式相比，TSF Serverless 无需提前准备资源，也无需操心资源扩缩容等传统运维流程，具有简化流程、免运维 IaaS、节约资源等优势。

#### 简化流程，免运维 IaaS

TSF Serverless 创建应用流程



#### 节约资源闲置成本

TSF Serverless 创建应用后的用户成本 = 实际消耗资源费用，Serverless 自动按业务负载弹性扩缩

精益成本，不为闲置资源付费

TSF 创建应用流程

资源准备 应用创建



TSF 创建应用后的用户成本

= TSF 平台授权费 + 集群CVM费用

购买的CVM存在闲置可能，用户需为此买单

### 产品功能

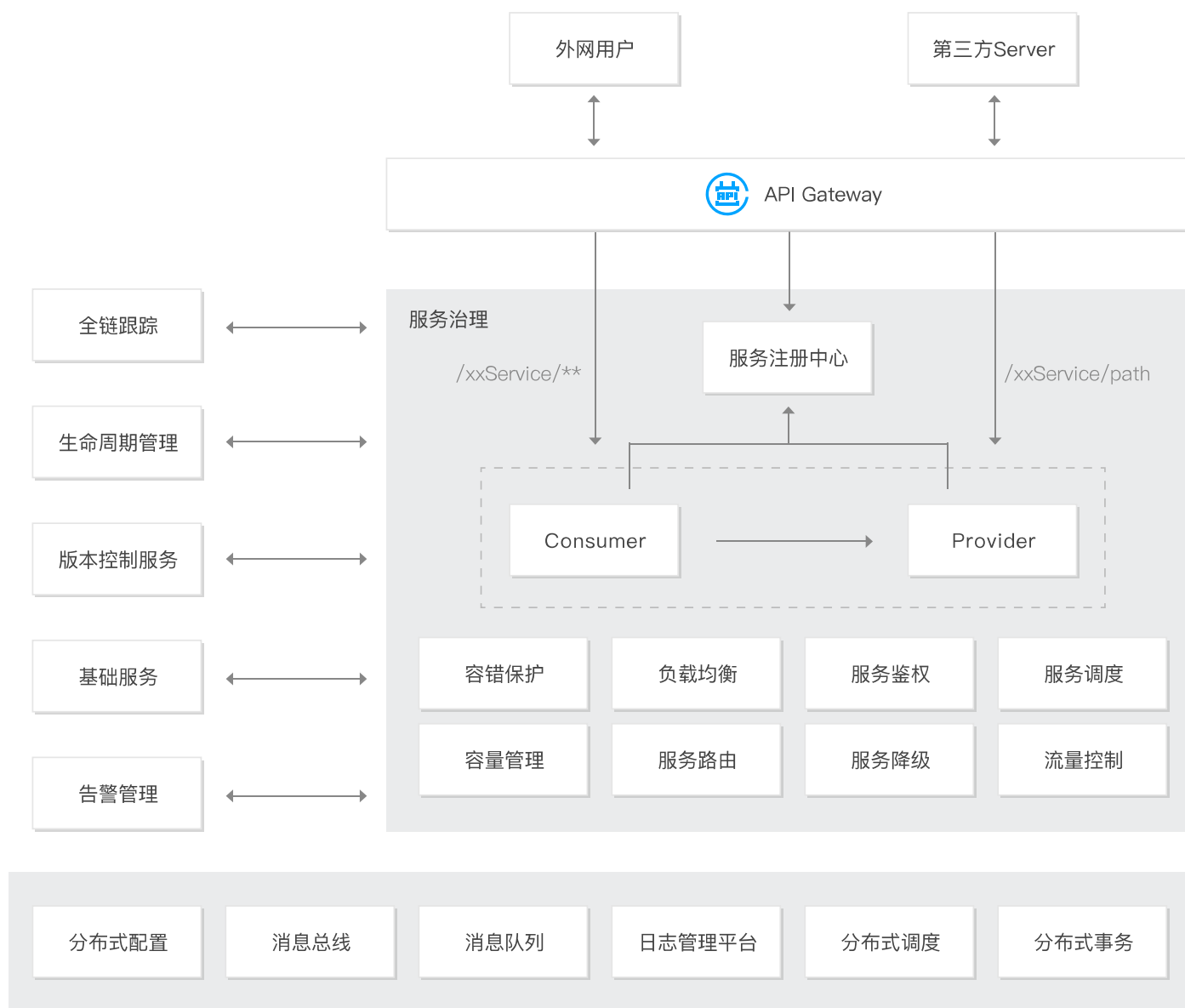
- 微服务的应用托管平台**  
 主要支持东西向微服务框架（如 Spring Cloud 和 Service Mesh）。
- 精益成本，不为闲置资源付费**  
 无需提前为业务峰值准备资源，按需使用、按量计费，无需为闲置资源付费。
- 完善的微服务中台能力**  
 提供应用全生命周期管理（支持创建、部署、回滚、扩容、下线、启动和停止应用）、细粒度微服务治理（支持

服务路由、服务限流、服务鉴权规则，支持分布式配置管理）、分布式事务等能力。

## 应用场景

### • 分布式系统

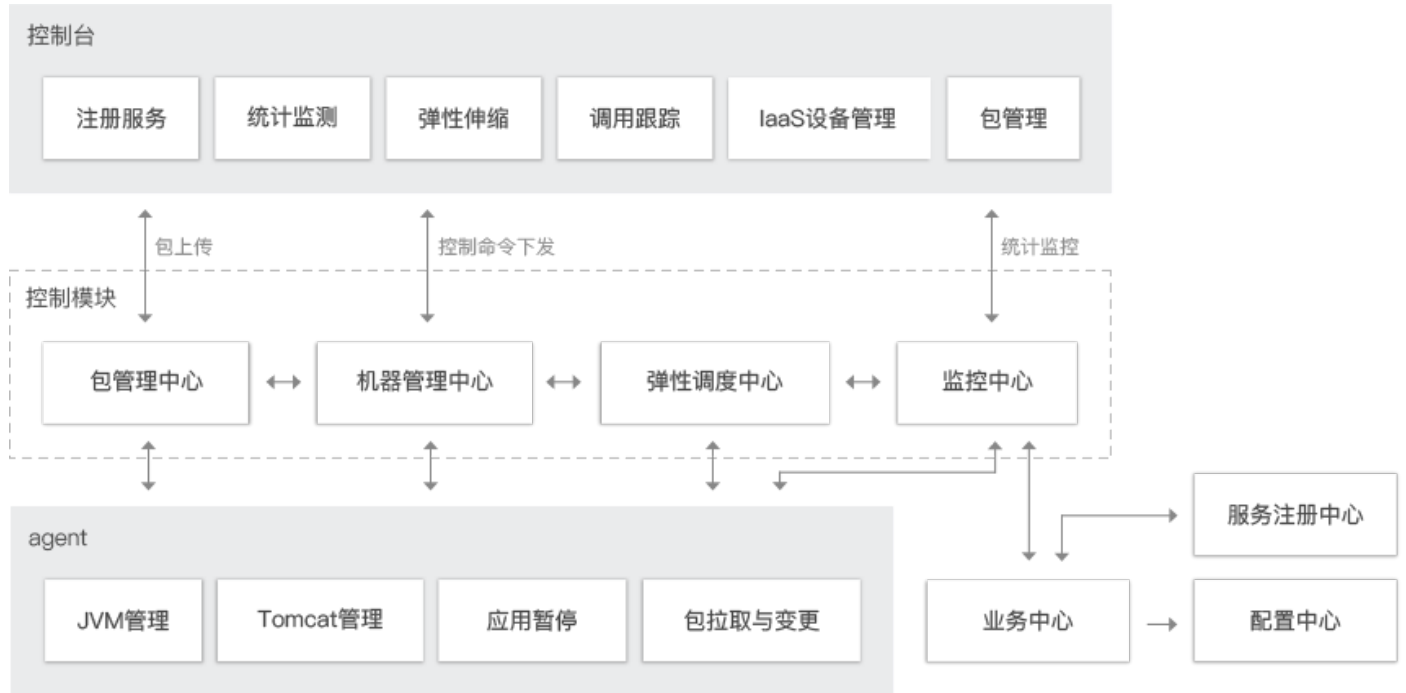
TSF Serverless 帮助用户屏蔽底层资源购买和运维细节，低门槛部署微服务应用，让企业聚焦核心业务本身。



### • 应用发布和管理

传统的应用发布需要运维人员登录到每一台服务器进行发布和部署，TSF Serverless 针对分布式系统的应用发布和管理，提供了简单易用的可视化控制台。用户可以通过控制台发布应用（包括创建、部署、启动应用），

并可以查看应用的部署状态，无需关心底层机器。



# TSF Serverless 购买指南

最近更新时间：2020-06-11 15:25:49

TSF Serverless 内测名额已满，暂时不接受新的申请。

内测期间，您可以免费使用本产品。内测结束之后，本产品将进行公测。届时我们将在 [产品发布台](#) 发布公告，如有定价方面的变动也将提前通过站内信、邮件等方式通知您。

当您使用本产品遇到问题时，请 [联系我们](#) 寻求相应的帮助。

# TSF Serverless 运行环境

最近更新时间：2020-06-29 15:40:36

TSF Serverless 目前支持的运行环境如下，您可在创建应用时选取所需的运行环境：

- Java 8
- Golang 1
- Python 2.7
- Python 3.6
- Nodejs 8.9
- Nodejs 10.15

## 示例下载

TSF Serverless 提供各语言的 Demo 示例包。您可以在控制台的上传程序包页面下载，也可以直接单击下方链接下载。

运行环境	下载地址
Java 8	<a href="#">单击下载</a>
Golang 1	<a href="#">单击下载</a>
Python 2.7	<a href="#">单击下载</a>
Python 3.6	<a href="#">单击下载</a>
Nodejs 8.9	<a href="#">单击下载</a>
Nodejs 10.15	<a href="#">单击下载</a>

## 相关文档

您可参考以下文档，使用相关功能：

- [TSF Serverless 使用须知](#)
- [Express Hello World 示例](#)
- [使用 CLI 快速创建并部署 Serverless 应用](#)



# TSF Serverless 使用须知

最近更新时间：2020-08-13 10:56:03

## 监听端口

- 如果您的业务是微服务（Spring、Mesh），则无需考虑端口监听问题。
- 如果您的业务是 Web 服务，您的服务需要主动监听8080端口。

## 上传程序包要求

目前 Serverless 应用支持的程序包格式包括 jar、war、tar.gz 和 zip。

- jar：FatJar 格式的程序包，用户可以参考 [如何打 FatJar 包](#)。
- war：war 格式的程序包，在部署 war 包时，TSF 会自动安装 Tomcat 环境。示例 [demo 下载](#)，示例 demo 部署成功后，使用 `http://<IP>:8080/sample` 访问，其中 IP 可以是实例的外网 IP。
- tar.gz、zip：压缩包中必须包含三个文件，确保文件名正确：
  - start.sh：启动脚本
  - stop.sh：停止脚本
  - cmdline：用于检查应用进程是否存在，没有.sh后缀

文件类型	启动方式
war	云服务器上的 agent 会使用 <code>java -jar</code> 命令启动程序。
jar	云服务器上的 agent 会使用 <code>java -jar</code> 命令启动程序。
tar.gz	云服务器上的 agent 会解压压缩包，使用解压目录下的 <code>start.sh</code> 脚本启动应用程序。
zip	云服务器上的 agent 会解压压缩包，使用解压目录下的 <code>start.sh</code> 脚本启动应用程序。

更多说明请参考 [上传程序包要求](#)。

## 本地临时存储

- 如果您需要在本地写临时文件，推荐写到 `/tmp` 路径下，其他任意路径均可。
- `/tmp` 路径下空间有限，您需要定时清理。

## 旧版部署组/应用迁移说明

为提供更好的服务并全面支持微服务平台相关能力，我们在 1.19.0 版本中对 Serverless 应用及部署组进行了升级。关于历史部署组，您需要关注以下几点：

- 历史部署组上的业务可照常运行。但为了保证您能享受新版本功能，我们强烈建议您重新创建部署组，并对旧版部署组做迁移。具体操作请参考 [部署组迁移指引](#)。
- 1.19.0 版本后默认历史集群不可新建部署组，请重新创建并选择新的集群进行操作。
- 1.19.0 版本后的访问管理入口将从【应用】迁移到【部署组】。
- 如果需要查看历史部署组的域名，您可以通过登录 [API 网关控制台](#)，基于 TSF 应用名查找对应 API 网关。例如：旧版应用名为 demo，则对应 API 网关的默认域名为 demo-<appid>.<region>.apigw.tencentcs.com。
- 1.19.0 版本后 Serverless 应用的程序包需同时包含 start.sh、stop.sh、cmdline 三个文件，详见 [上传程序包要求](#)。

## 部署组迁移指引

1. 登录 [TSF 控制台](#)，在左侧导航栏单击【集群】，新建一个集群。详细操作可参考 [集群](#) 文档。

### 新建集群



集群类型

虚拟机集群

容器集群

Serverless 集群

集群名

Serverless

最长60个字符

所在可用区

广州一区

集群网络

tsf-test2

如现有的网络不合适，您可以去控制台 [新建私有网络](#)

VPC子网

subnet-q65fjho6

备注

最长200个字符

提交

关闭

## 2. 单击【部署组】，创建 Serverless 应用部署组。

具体操作可参考 [Serverless 应用部署组](#) 文档。**历史部署组不可直接创建 Serverless 集群。**

**1** 填写基本信息 > **2** 选择实例 > **3** 部署应用

组名  ❗ 创建后名称不可修改

最长60个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

集群

命名空间

关联应用

若现有的应用不合适，您去控制台[新建应用](#)

保存 & 下一步

关闭

3. 部署 Serverless 应用。具体操作可参考 [Serverless 应用部署组](#) 文档。

### 新建部署组



1 填写基本信息 > 2 部署应用

① 安装 war 包, 将自动安装 tomcat 运行环境

组名 serverless213213213

程序包/版本

输入关键字搜索



ID	包名称	版本	上传时间
<input checked="" type="radio"/> pkg-baf570d6	consumer-demo-1....	20200114193840	2020-01-14 19:41:12

若现有的程序包不合适, 您去控制台[上传程序包](#)

是否开启外网访问  当前命名空间暂未配置code, 不能开启外网访问[前往配置](#)

高级设置 ▾

完成

关闭

4. 上传程序包。

本次程序包上传必须同时包含 `start.sh`、`stop.sh`、`cmdline`, 详情请参考 [上传程序包要求](#)。

5. 开启外网访问。


单击部署组 ID, 在部署组详情页开启外网访问, 即可在外网访问已迁移至部署组的服务。

← group-oydmeey2

日志 外网访问 基本信息

外网访问路径

外网访问生效状态 

访问路径 <https://service-fia-251006131.gz.apigw.tencentcs.com> 

API网关访问服务名 [service-fia3544u](#) 

API网关监控 [点此前往](#) 

自定义域名 

# TSF Serverless 操作指南

## 概述

最近更新时间：2020-12-17 16:27:21

### 操作场景

本文旨在通过 Express 和 Koa 的 Hello World 示例介绍 TSF Serverless 的使用方法，帮助您了解 TSF Serverless 的基本使用流程和控制台的基本操作。例如创建应用、上传程序包、创建部署组、配置外网访问和查看运行日志和监控。

您可以结合示例步骤进行项目开发，服务创建及访问验证。本文示例步骤如下：

1. 使用 Node.js Web 框架（Express、Koa）在本地完成项目开发和本地测试。
2. 在 TSF 控制台创建 Serverless 应用，并上传打包后的项目代码。
3. 创建部署组并开启外网访问，验证服务的运行情况。

### 前提条件

- 本地已安装 [Node.js](#)
- 已 [开通 TSF Serverless 白名单](#)

### 操作步骤

- [Express Hello World 示例](#)
- [Koa Hello World 示例](#)

# 自定义域名及证书部署示例

最近更新时间：2020-08-04 16:08:43

## 操作场景

用户可通过域名绑定功能，将 TSF Serverless 中的外网访问路径设置为用户所拥有的自定义域名绑定到服务，使得服务能以用户自身自定义域名的方式提供。

## 前提条件

确保您的自定义域名已完成腾讯云备案。

## 操作步骤

### 打开外网访问路径

1. 登录 [TSF 控制台](#)，在左侧导航栏单击【部署组】。
2. 在部署组管理页，选择已创建的 Serverless 部署，打开外网访问生效状态。
3. 单击【API 网关访问服务名】进入 API 网关服务详情页。



### 关联自定义域名

1. 在 API 网关的服务详情页，选择【自定义域名】，单击页面左上角的【新建】，填写配置信息后，单击【提交】。

## 新增自定义域名



请确认需绑定的域名已经完成解析配置，指向此服务的二级域名。确保您的域名已完成备案，参考[域名备案要求](#)

公网二级域名:  0226.gz.apigw.tencentcs.com

服务 SCF\_API\_SERVICE

域名

访问方式  公网

协议

路径映射  默认路径映射  自定义路径映射

路径

环境



[添加路径映射\(1/3\)](#)

提交

关闭

**说明:**

如果需要在自定义域名的 https 协议，需要提交域名的 SSL 证书。证书可以通过文件上传的方式或通过填写证书名称、内容和私钥的方式提交，具体可查看 [域名型 \(DV\) 免费 SSL 证书](#)。

2. 配置 CNAME 解析完成后，在服务内配置绑定自定义域名（请确保先配置 CNAME 解析，再配置绑定）。

3. 若需要解绑，需先删除服务内绑定的自定义域名，再删除自定义域名的 CNAME。

### 配置域名路径映射

在配置自定义域名时可使用默认路径映射，此时路径的 URL 为 自定义域名/环境名，例如

www.yingxiong.com/release 指向本服务中的发布环境，www.yingxiong.com/prepub 指向本服务中的预发布



环境，`www.yingxiong.com/test` 指向本服务中的测试环境。

← service-lj4r2pb6

服务信息
API管理
使用计划
自定义域名
环境管理
版本管理
监控

新建

请输入ID/名称 Q

↓

域名	解析状态	SSL证书	操作
www.yingxiong.com	正常解析	BB73JGnP	<a href="#">编辑路径映射</a> <a href="#">删除</a>

路径	目标环境	操作
使用默认路径映射①		

您也可以配置自定义路径，使用自定义路径时的 URL 为 自定义域名/自定义路径，这个 URL 则指向您所映射的环境。例如配置路径为 `/mypath`，环境为发布环境，则发布环境的 URL 为 `www.yingxiong.com/mypath`。当您想使用根路径时，可直接配置路径为 `/`。

#### 修改路径映射 ×

所属服务 test

域名 www.yingxiong.com

路径映射  默认路径映射①

自定义路径映射

路径	目标环境	
/mypath	测试	<a href="#">删除</a>
/	发布	<a href="#">删除</a>
/pre	预发布	<a href="#">删除</a>

提交

关闭

使用自定义路径映射时，原本默认路径映射不生效，即 自定义域名/环境名 均不生效。配置后自定义路径映射与默认路径映射还可以进行编辑。

← service-lj4r2pb6

服务信息 API管理 使用计划 **自定义域名** 环境管理 版本管理 监控

新建				请输入ID/名称	
域名	解析状态	SSL证书	操作		
▼ www.yingxiong.com	正常解析	BB73JGnP	<a href="#">编辑路径映射</a>	<a href="#">删除</a>	
路径	目标环境	操作			
/pre	预发布	<a href="#">删除</a>			
/	发布	<a href="#">删除</a>			
/mypath	测试	<a href="#">删除</a>			

# 使用 TSF Serverless 部署微服务

最近更新时间：2020-08-26 15:03:32

## 操作场景

本文指导您通过 TSF 控制台，完成 TSF Serverless 部署微服务的整体操作流程。

## 操作步骤

### 步骤1：创建 Serverless 集群

首先您需要创建 Serverless 集群。集群是实例、Serverless 等云资源的集合。

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏中，单击【[集群](#)】，进入集群列表页。
3. 在集群列表页，单击【新建集群】。
4. 设置集群的基本信息。
  - **集群类型**：选择 **Serverless 集群**。
  - **集群名称**：集群名称，不超过60个字符。
  - **集群网络**：为集群内主机分配在云服务器网络地址范围内的 IP 地址。参阅 [私有网络和子网](#)。
  - **集群描述**：集群的描述，不超过200个字符。

### 步骤2：创建 Serverless 应用

1. 在左侧导航栏，单击【[应用管理](#)】，进入应用列表。
2. 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】。
  - **部署方式**：选择 **Serverless 部署**
  - **运行环境**：选择 **Java8**（目前仅 Java8 支持微服务应用）

### 步骤3：上传程序包

1. 在 [应用管理列表](#) 页，单击目标应用的 ID/应用名，进入应用详情页。

2. 在应用详情页的上方，单击**程序包管理**标签页，单击【上传程序包】。

← application-6yomj9nv (21312321)

部署组    **程序包管理**    基本信息

<input type="checkbox"/>	程序名	版本号	包类型	备注	MD5	上传时间
<input type="checkbox"/>	pkg-93eb2ad2 consumer-demo-1.18.0-Finchle...	20200211175503	fatjar		6791261700d0b350eaece26ff797bb28f	2020-02-11 17:55:22

共 1 条 20 ▾ 条/页

3. 在**上传程序包**对话框中填写相关参数。

- 上传程序包：单击【选择文件】，选择编译为 jar 格式的程序包
- 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

4. 单击【提交】，程序包上传成功后出现在程序包列表中。

#### 步骤4：创建部署组

1. 在 [应用管理列表](#) 页，单击目标应用的 ID/应用名，进入应用详情页。

2. 单击【新建部署组】，设置部署组相关信息：

- 部署组名称：部署组的名称，不超过60个字符。
- 集群：选择步骤1中创建的集群。
- 命名空间：选择集群关联的系统命名空间。
- 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。参考 [日志服务](#)。

3. 单击【提交】。

#### 步骤5：部署应用

1. 在上步操作中单击【下一步】即可完成部署，如部署失败，可在部署组列表页的右侧，单击【部署应用】重试。

新建部署组

×

✓ 填写基本信息 > 2 部署应用

**i** 安装 war 包，将自动安装 tomcat 运行环境

组名 test

程序包/版本

输入关键字搜索

Q

ID	包名称	版本	上传时间
<input checked="" type="radio"/> pkg-93eb2ad2	consumer-demo-1....	20200211175503	2020-02-11 17:55:22

若现有的程序包不合适，您去控制台[上传程序包](#)

是否开启外网访问  当前命名空间暂未配置code, 不能开启外网访问[前往配置](#)

高级设置 ▲

内存规格 8GB

最小实例数 3

启动参数 -Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m

完成

关闭

2. 选择步骤3中已上传成功的程序包后，单击【提交】。

3. 应用部署成功后，部署组中**运行实例数**的数值发生变化。

← application-6yomj9nv (21312321)

部署组    程序包管理    基本信息

新建部署组					
ID/部署组名	集群	命名空间	状态	已部署程序版本/包名	运行实例数
group-6a77576a sssss	cluster-maejlm6v sdasdasdas	namespace-by8ml3la sdasdasdas_default	已停止	-	0
group-2vz6ggwa ceshi1111	cluster-maejlm6v sdasdasdas	namespace-by8ml3la sdasdasdas_default	运行中	20200211175503 consumer-demo-1.18.0-Finchley- RELEASE.jar	1

共 2 条

## 相关文档

您可参考以下文档，使用 TSF Serverless 相关功能：

- [TSF Serverless 使用须知](#)
- [Spring Cloud 概述](#)
- [Spring Cloud 快速入门](#)

# Express Hello World 示例

最近更新时间：2020-12-17 16:26:54

## 说明：

- TSF Serverless 是微服务的应用托管平台，主要支持东西向微服务框架（如 Spring Cloud 和 Service Mesh）。
- Web 服务（Express、Koa）推荐使用 [Serverless Framework](#)。

## 创建 Nodejs Express 项目

1. 根据实际需求，选择目录路径，并在该路径下创建新的目录，用作于项目目录。  
例如，创建一个名称为 helloexpress 的项目目录。
2. 进入 helloexpress 目录，并在该目录下执行以下命令，初始化 node 项目。

```
npm init
```

## 说明：

命令执行后所有选项均保持默认即可。

3. 执行以下命令，安装 Express 包。

```
npm install express --save
```

4. 在项目目录下创建 index.js 文件，并输入以下内容：

```
const express = require('express')
const app = express()
app.get('/', (req, res) => res.send('Hello World!'))
const port = 8080
console.log('listening port',port)
app.listen(port, () => console.log('Example app listening on',port))
```

## 说明：

服务需要监听8080端口。

5. 在项目目录下创建 start.sh 启动脚本文件，并输入以下内容：

```
#!/bin/bash
node index.js
```

6. 执行以下命令，进行项目的本地验证。

```
/bin/bash start.sh
```

显示结果如下，则说明服务已启动。

```
listening port 8080
Example app listening on 8080
```

在浏览器地址栏输入 localhost:8080 后访问，窗口显示 Hello World!，本地创建项目成功。

7. 打包本地项目，准备上传。

您需要在项目的根路径（start.sh 所在路径）下执行打包。您可以在 [TSF Serverless 使用须知](#) 查看程序包的要求。

```
zip code.zip * -r
```

## 创建 Hello World 服务

### 步骤1：新建 Serverless 集群

首先您需要创建 Serverless 集群。集群是实例、Serverless 等云资源的集合。

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏中，单击【[集群](#)】，进入集群列表页。
3. 在集群列表页，单击【[新建集群](#)】。
4. 设置集群的基本信息。
  - **集群类型**：选择 **Serverless 集群**。
  - **集群名称**：集群名称，不超过60个字符。
  - **集群网络**：为集群内主机分配在云服务器网络地址范围内的 IP 地址。参阅 [私有网络和子网](#)。
  - **集群描述**：集群的描述，不超过200个字符。



## 步骤2：创建 Serverless 应用

1. 在左侧导航栏，单击【[应用管理](#)】，进入应用列表。
2. 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】。
  - 部署方式：选择 **Serverless部署**
  - 运行环境：选择 **Nodejs**

## 步骤3：上传程序包

1. 在 [应用管理列表](#) 页，单击目标应用的ID/应用名，进入应用详情页。
2. 在应用详情页的上方，单击[程序包管理](#)标签页，单击【上传程序包】。

← application-6yomj9nv (21312321)

部署组    **程序包管理**    基本信息

<input type="checkbox"/>	程序名	版本号	包类型	备注	MD5	上传时间
<input type="checkbox"/>	pkg-93eb2ad2 consumer-demo-1.18.0-Finchle...	20200211175503	fatjar		6791261700d0b350eaece26ff797bb28	2020-02-11 17:55:22

共 1 条 20 ▾ 条/页

3. 在[上传程序包](#)对话框中填写相关参数。
  - 上传程序包：单击【选择文件】，选择编译为 jar 格式的程序包
  - 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
4. 单击【提交】，程序包上传成功后出现在程序包列表中。

## 步骤4：创建部署组并添加实例

1. 在 [应用管理列表](#) 页，单击目标应用的ID/应用名，进入应用详情页。
2. 单击【新建部署组】，设置部署组相关信息：
  - 部署组名称：部署组的名称，不超过60个字符。
  - 集群：选择[步骤1](#)中创建的集群。
  - 命名空间：选择集群关联的系统命名空间。
  - 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。参考 [日志服务](#)。
3. 单击【提交】。

## 步骤5：部署应用

1. 在上步操作中单击【下一步】即可完成部署，如部署失败，可在部署组列表页的右侧，单击【部署应用】重试。

新建部署组

×

✓ 填写基本信息 > 2 部署应用

**i** 安装 war 包，将自动安装 tomcat 运行环境

组名 test

程序包/版本

输入关键字搜索			
ID	包名称	版本	上传时间
<input checked="" type="radio"/> pkg-93eb2ad2	consumer-demo-1....	20200211175503	2020-02-11 17:55:22

若现有的程序包不合适，您去控制台[上传程序包](#)

是否开启外网访问  当前命名空间暂未配置code, 不能开启外网访问[前往配置](#)

高级设置 ▲

内存规格

最小实例数

启动参数

完成

关闭

2. 选择步骤3中已上传成功的程序包后，单击【提交】。

3. 应用部署成功后，部署组中**运行实例数**的数值发生变化。

[←](#) application-6yomj9nv (21312321)

**部署组**    程序包管理    基本信息

新建部署组

ID/部署组名	集群	命名空间	状态	已部署程序版本/包名	运行实例数
group-6a77576a sssss	cluster-maejlm6v sdasdasdas	namespace-by8ml3la sdasdasdas_default	已停止	-	0
group-2vz6ggwa ceshi1111	cluster-maejlm6v sdasdasdas	namespace-by8ml3la sdasdasdas_default	运行中	20200211175503 consumer-demo-1.18.0-Finchley- RELEASE.jar	1

共 2 条

# Koa Hello World 示例

最近更新时间：2021-01-12 16:10:49

## 说明：

- TSF Serverless 是微服务的应用托管平台，主要支持东西向微服务框架（如 Spring Cloud 和 Service Mesh）。
- Web 服务（Express、Koa）推荐使用 [Serverless Framework](#)。

## 操作步骤

1. 根据实际需求，选择目录路径，并在该路径下创建新的目录，用作于项目目录。

例如，创建一个名称为 `hellokoa` 的项目目录。

2. 进入 `hellokoa` 目录，并在该目录下执行以下命令，初始化 `node` 项目。

```
npm init
```

## 说明：

命令执行后所有选项均保持默认即可。

3. 执行以下命令，安装 Koa 包。

```
npm install koa --save
```

4. 在项目目录下创建文件 `index.js`，并输入以下内容：

```
const koa = require('koa')
const app = new koa()
app.use(async ctx => {
  ctx.body = 'Hello World';
});
const port = 8080
console.log('listening port',port)
app.listen(port, () => console.log('Example app listening on',port))
```

7. 打包本地项目，准备上传。您需要在项目的根路径（start.sh所在路径）下执行打包。

```
zip code.zip * -r
```

8. 创建服务步骤同 [Express示例项目的创建 Hello World 服务](#)。