

Serverless SSR

快速入门

产品文档



腾讯云

【 版权声明 】

©2013–2022 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

快速入门

控制台部署应用

账号权限管理

快速入门

控制台部署应用

最近更新时间：2022-04-14 15:28:12

操作场景

通过 [Serverless SSR 控制台](#)，您可以快速创建 Web 应用模板，或导入本地已有项目，实现传统 Web 框架的快速迁移、持续部署、二次开发、监控运维，在控制台完成应用项目的闭环开发体验。

权限配置

主账号授权

1. 主账号登录 [Serverless SSR 控制台](#)，单击[前往授权](#)进入访问管理控制台。
2. 在访问管理控制台的[角色](#)列表页，查看 `SLS_QcsRole` 和 `CODING_QCSRole` 服务角色是否创建成功。

⚠ 注意：

如果您已经创建过 `CODING_QCSRole`，请检查角色拥有权限是否完整，该角色需要基本策略列表如下：`QcloudSLSFullAccess`、`QcloudSSLFullAccess`、`QcloudAccessForCODINGRole`，如有缺少，请手动添加。

角色

🔔 为什么我的账户出现了新角色？
在服务中完成特定操作（如授权创建服务角色）时，服务可能为您创建服务相关角色。或者，如果您在某项服务开始支持服务相关角色之前已在使用该服务，则该服务可能自动在您的账户中创建角色。

新建角色

角色名称	角色载体	角色描述	操作
<code>CODING_QCSRole</code>	产品服务 - coding.cloud.tencent.com	当前角色为 CODING DevOps 服务角色，该角色将在已关联策略的权限范围内访问您的其他...	删除
<code>SLS_QcsRole</code>	产品服务 - sls.cloud.tencent.com	-	删除
<code>Faceid_QCSRole</code>	产品服务 - faceid.qcloud.com	当前角色为 人脸核身 服务角色，该角色将在已关联策略的权限范围内访问您的其他云服务资...	删除

3. 确定角色与权限都符合要求后，即可开始使用服务。

子账号授权

子账号如果未开通 [Serverless Framework](#) 和 [Coding DevOps](#) 的服务，请先与主账号联系，完成服务开通与角色创建（步骤 [同上](#)）。

开通服务后，需要主账号（或拥有授权操作的子账号）给当前子账号进行授权操作（参考[子账号授权指引](#)）。完成授权后，即可开始使用服务。

创建应用

模板创建

1. 登录 [Serverless SSR 控制台](#)，单击**新建应用**，进入应用创建页。
2. 根据页面提示，填写应用名，并选择环境、地域、创建方式和模板等基本信息。



- 创建方式：选择**应用模板**。
 - 模板：选择您想要使用的**框架模板**。
3. （可选）打开**高级设置**卡片，根据您的实际需要进行相关配置，您也可以在完成部署后，在应用详情控制台完成这部分配置。

高级配置

静态资源托管  启用

新建存储桶 

使用静态资源托管时，需要您对项目配置文件做简单改造，[查看改造指引](#)

CDN加速  启用

自定义域名  启用

函数配置

内存  128MB 

超时时间  3 秒

时间范围：1-900秒

环境变量 

key	value
<input type="text" value="请输入key"/>	<input type="text" value="请输入value"/>

固定出口IP  启用

标签 启用

说明：

目前，静态资源托管与 CDN 加速的高级能力仅支持 **Next.js**、**Nuxt.js** 框架。

4. 单击**创建**，Serverless 控制台会自动开始为您部署应用。

部署完成后，进入应用详情页，可以查看创建的云上资源、监控日志、部署记录等信息，也支持在**开发部署**页面

修改配置，重新部署。

导入已有项目

1. 登录 [Serverless SSR 控制台](#)，单击**新建应用**，进入应用创建页。
2. 根据页面提示，填写应用名，并选择环境、地域、创建方式和类型选择。

创建方式

应用模板 使用模板创建 Serverless SSR 应用	导入已有项目 快速导入您的本地项目
---	-----------------------------

类型选择

Next.js 应用 ▾

如果使用 Express 等 web 框架替代 Next.js 默认的 web server，必须进行部分改造，点击查看[自定义路由项目改造指引](#)

上传方式

代码托管 ▾

选择代码托管方式，每次更新仓库中的代码，Serverless SSR 都会自动为您进行部署，点击查看[代码托管部署指引](#)

代码源

 Github	 Gitlab	 Gitee	 自定义仓库
--	--	---	---

选择当前代码源内项目及仓库，如当前账号尚未授权该代码源，可[授权代码源](#)

代码仓库

请选择仓库地址 ▾ 请选择代码分支 ▾

触发规则

自动触发构建 手动触发构建

选择应用自动更新的触发规则，查看[详细触发规则](#)

- 创建方式：选择**导入已有项目**。
- 类型选择：选择使用的框架模板。

说明：

部分框架项目需要进行简单项目改造，才可以进行部署，详情请参考 [项目改造](#)

3. 选择代码上传方式，Serverless SSR 控制台支持您直接上传本地项目部署，也可以选择导入代码仓库。

- 代码托管

目前支持 **GitHub**、**GitLab**、**Gitee** 的代码仓库地址，也支持公开的自定义代码库，您可以通过选择应用的触发方式，完成应用的自动更新，详情请参考 [项目触发方式管理](#)。

- 文件夹上传

您可以通过上传文件夹的方式直接导入本地项目，对于 Node.js 框架，Serverless Framework 将自动为您创建层，并将依赖包 `node_modules` 传入层中完成部署。

4. (可选) 打开高级配置卡片, 根据您的实际需要进行相关配置, 您也可以在完成部署后, 在应用详情控制台完成这部分配置。

▲ 高级设置

静态资源托管  启用

新建存储桶



使用静态资源托管时, 需要您对项目配置文件做简单改造, [查看改造指引](#)

CDN加速  启用

自定义域名  启用

函数配置

内存  128MB

超时时间  3 秒

时间范围: 1-900秒

环境变量 

key	value
<input type="text" value="请输入key"/>	<input type="text" value="请输入value"/>

固定出口IP  启用

创建

 说明:

目前, 静态资源托管与 CDN 加速的高级能力仅支持 **Next.js**、**Nuxt.js** 框架。

5. 单击**创建**, Serverless SSR 控制台会自动开始为您部署应用。

部署完成后, 进入应用详情页, 可以查看创建的云上资源、监控日志、部署记录等信息, 也支持在**开发部署**页面修改配置, 重新部署。

管理应用

应用创建完成后, 您可以在应用详情页, 完成查看项目具体信息, 主要支持以下几部分信息查看。

资源管理

在资源列表页面，支持查看当前应用为您创建的云资源，并查看基本配置信息。

← test·

dev ▾

资源列表

开发部署

应用监控

部署日志

基础信息

应用名称 test-github
实例名称 nextjs-T9bueMqlr
地域 ap-guangzhou

API网关

服务ID [service-1ko24yh8](#) 
域名 service-
环境 release
URL

云函数

函数名称 [nextjs_component_s4sr8r5](#) 
命名空间 default
运行环境 Nodejs10.15
使用层 test-github-layer (版本1)

开发部署

在开发部署页面，您可以轻松地实现应用的配置修改与二次部署上传，支持本地上传、代码托管、CLI 开发三种方式。

同时，您也可以在该页面修改应用配置信息，单击**保存**完成重新部署。

资源列表 **开发部署** 应用监控 部署日志

更新代码

部署方式

文件夹上传

下载项目到本地，完成开发后重新上传

代码托管

每次更新仓库中的代码，自动为您进行部署，点击查看[代码托管部署指引](#)

本地开发

通过命令行开发工具 Serverless Framework，实现本地快速开发部署，查看[产品文档](#)

代码源

Github

Gitlab

Gitee

自定义仓库

选择当前代码源内项目及仓库，如当前账号尚未授权该代码源，可[授权代码源](#)

代码仓库

请选择仓库地址

请选择代码分支

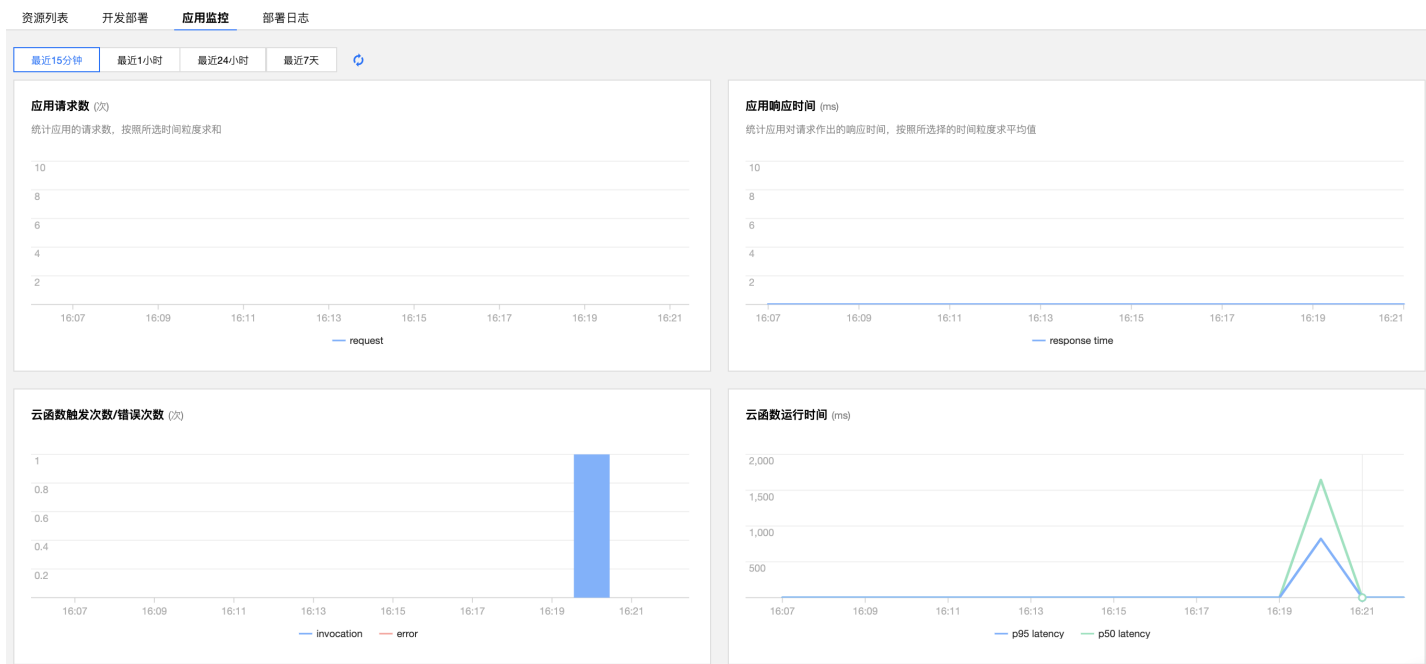
触发规则

自动触发构建 手动触发构建

选择应用自动更新的触发规则，查看[详细触发规则](#)

应用监控

在**应用监控**页面，您可以查看项目部署后输出的基本信息、项目请求次数、项目报错统计等多项监控指标，方便您轻松实现项目的管理运维。



部署日志

在**部署日志**页面，可以看到**通过控制台部署**或**自动触发**的部署日志，以及部署结果。

说明：

目前通过 CLI 命令行部署的应用日志暂时无法在控制台查看。

[资源列表](#) [开发部署](#) [应用监控](#) [部署日志](#)

部署时间	来源	状态
2020-11-23 11:27:16	控制台部署	操作成功

```
Downloading code
Initializing node.js environment
Installing plugins
Installing dependencies
Processing serverless config files
Deploying serverless project
Warming up serverless project
serverless deploy --debug
Initializing...
Action: "deploy" - Stage: "dev" - App: "test-github1" - Instance: "nextjs-NZxUinkEd"
the code will be uploaded to https://sp-ins-prod-1300963013.cos.ap-guangzhou.myqcloud.com/3d3fa644-df00-447b-84e3-6ea9fceb16e.zip?q-sign-algorithm=sha1&q-ak=AKID1qcBelcVaOwSmtn67y7pJJIAVj4jftBE&q-sign-time=1606102128;1606102728&q-key-time=1606102128;1606102728&q-header-list=&q-url-param-list=&q-signature=8636f1062672c47bce3d27b1bde60072e98c912b
Uploading...
Deploying...
Deploying Next.js App...
Packaging Next.js application...
Code zip path /tmp/31r9da/source.zip
Creating bucket: sls-cloudfunction-ap-guangzhou-code-***** in ap-guangzhou
Bucket sls-cloudfunction-ap-guangzhou-code-***** already exist.
Removing cors for ap-guangzhou's bucket: sls-cloudfunction-ap-guangzhou-code-*****
Removing tags for ap-guangzhou's bucket: sls-cloudfunction-ap-guangzhou-code-*****
```

账号权限管理

最近更新时间：2022-04-14 15:28:20

操作场景

Serverless SSR 基于 [Serverless Framework](#) 与 [Coding DevOps](#) 构建计划完成部署，用户在使用前，必须保证已经开通这两项服务，并拥有相关角色的调用权限。

本文主要介绍首次登录 SSR 控制台时，Serverless Framework 与 Coding DevOps 的授权流程。

操作步骤

主账号授权

1. 登录 [Serverless SSR 控制台](#)，单击[前往授权](#)进入访问管理控制台。
2. 在访问管理控制台的[角色列表页](#)，查看 `SLS_QcsRole` 和 `CODING_QCSRole` 服务角色是否创建成功。

⚠ 注意：

如果您已经创建过 `CODING_QCSRole`，请检查角色拥有权限是否完整，该角色需要基本策略列表如下：`QcloudSLSFullAccess`、`QcloudSSLFullAccess`、`QcloudAccessForCODINGRole`，如有缺少，请手动添加。

3. 确定角色与权限都符合要求后，即可开始使用服务。

子账号授权

如果未开通 [Serverless Framework](#) 和 [Coding DevOps](#) 的服务，请先与主账号联系，完成服务开通与角色创建（步骤 [同上](#)）。

开通服务后，需要主账号（或拥有授权操作的子账号）给当前子账号进行如下授权操作。完成授权后，即可开始使用服务。

步骤1：授权操作产品的权限

1. 在 [CAM 用户列表页](#)，选取对应子账号，单击用户名称，进入用户详情页。
2. 单击[关联策略](#)，在添加策略页面单击[从策略列表中选择策略关联](#)。
3. 搜索并关联 `QcloudSLSFullAccess` 与 `QcloudCODINGFullAccess`，单击[下一步](#)。
4. 单击[确定](#)，即可授予子账号 [Serverless Framework](#) 所有资源的操作权限。

步骤2：授权调用 `SLS_QcsRole` 角色的权限

1. 在 [CAM 用户列表](#) 页，选取对应子账号，单击用户名称，进入用户详情页。

2. 单击**关联策略**，在添加策略页面单击从**策略列表**中选取策略关联。
3. 选择**新建自定义策略** > **按策略语法创建** > **空白模板**，填入如下内容。

注意：

角色参数替换为您的 uin（账号 ID）。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "cam:PassRole"
      ],
      "resource": [
        "qcs::cam::uin/${填入账号的uin}:roleName/SLS_QcsRole"
      ],
      "effect": "allow"
    },
    {
      "resource": [
        "*"
      ],
      "action": [
        "name/sts:AssumeRole"
      ],
      "effect": "allow"
    }
  ]
}
```

4. 单击**确定**，即可授予子账号 SLS_QcsRole 的操作权限。