

Serverless SSR

操作指南





【版权声明】

©2013-2025 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】

🔗 腾讯云

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

操作指南 快速部署 Nextjs 框架 快速部署 Nuxtjs 框架 高级能力配置 层部署 高级配置 持续构建 静态资源托管改造 自定义路由项目改造 ICP 备案



操作指南 快速部署 Nextjs 框架

最近更新时间: 2024-11-15 15:13:42

操作场景

本文将为您指导如何通过 Web Function,将您的本地 Next.js SSR 项目快速部署到云端。

() 说明:

本文档主要介绍控制台部署方案,您也可以通过命令行完成部署,详情请参见 通过命令行完成框架部署 。

前提条件

在使用腾讯云云函数服务之前,您需要 注册腾讯云账号 并完成 实名认证 。

操作步骤

模板部署:一键部署 Next.js 项目

- 1. 登录 Serverless 控制台,单击左侧导航栏的函数服务。
- 2. 在主界面上方选择期望创建函数的地域,并单击新建,进入函数创建流程。
- 3. 选择使用模板创建来新建函数,在搜索框里输入 webfunc 筛选函数模板,选择 Next.js 框架模板并单击下一步。如下图所示:

模板创建 使用示例模版	从头开始 快速创建一个函数或应用	使用容 器镜像 基于容器确像未创建函数	
模糊搜索	webfunc 多个过滤标签用回车键分码	Q #32个	推荐排序
	Express 框架模版 查看详情 送别 函数	Flask 框架模版 查看详情 供别 函数	Nextjs 框架模版 查看详情 送到 函数
	描述 基于 API 网关和 Web 函数,快速部署 Express 示例项 目 标签 WebFunc Express Nodejs12.16 web	描述 基于 API 网关和 Web 函数,快速部署 Flask 示例项目 标签 WebFunc Flask Python3.6 web 作者 �� 等讯云	描述 基于 API 网关和 Web 函数、快速部署 Nextjs 示例项目 标签 WebFunc Nextjs Nodejs12.16 SSR web
	作者 🛆 腾帆云 部署 19,730次	部署 13,719次	作者 🔗 護讯云 部署 9,973次

- 4. 在新建页面,您可以查看模板项目的具体配置信息并进行修改。
- 5. 单击完成即可创建函数。函数创建完成后,您可在函数管理页面查看 Web 函数的基本信息。
- 您可以通过 API 网关生成的访问路径 URL,访问您部署的 Next.js 项目。单击左侧菜单栏中的触发管理,查看访问路径。如下图所示:



← nextjs_demo-	正常		函数服务帮助文档 记
函数管理	触发管理		
触发管理	创建触发器		
监控信息			
日志查询	API网关触发	沂属别名: 默认流量	删除
并发配额	API服务名	SCF_API_SERVICE 🛛	
部署日志	serviceId	service	
	apild	api	
	请求方法	ANY	
	发布环境	发布	
	鉴权方式	免鉴权	
	启用Base64编码	未启用	
	支持CORS	否	
	后端超时	15s	
	访问路径	https://service	.gz.apigw.tencentcs.com/release/ 💼

7. 单击访问路径 URL,即可访问服务 Next.js 项目。如下图所示:

欢迎访问 Next.js 应用

腾讯云 Serverless为您提供服务

() 说明:

由于 Nextjs 框架每次部署前需要重新构建,请确保本地更新代码并且重新 build 之后再进行部署。

自定义部署:快速迁移本地项目上云

前提条件

本地已安装 Node.js 运行环境。

本地开发

1. 参考 Next.js 官方文档,安装并初始化您的 Next.js 项目:

npx create-next-app

2. 在根目录下,执行以下命令在本地直接启动服务。

cd my-app && npm run dev

3. 打开浏览器访问 http://localhost:3000 ,即可在本地完成 Next.js 示例项目的访问。如下图所示:



部署上云

腾讯云

接下来执行以下步骤,对已初始化的项目进行简单修改,使其可以通过 Web Function 快速部署,此处项目改造通常分为以下两步: • 修改监听地址与端口为 0.0.0.0:9000 。

• 新增 scf_bootstrap 启动文件。

具体步骤如下:

1. 在项目根目录下新建 scf_bootstrap 启动文件,在该文件添加如下内容(用于启动服务并指定启动端口):

```
#!/var/lang/node12/bin/node
const { nextStart } = require('next/dist/cli/next-start');
nextStart([ '--port', '9000', '--hostname', '0.0.0.0' ])
```

▲ 注意:

- 此处仅为示例启动文件,具体请根据您的业务场景进行调整。
- 示例使用的是云函数标准 Node 环境路径,本地调试时,需修改成您的本地路径。

2. 新建完成后,还需执行以下命令修改文件可执行权限,默认需要 777 或 755 权限才可正常启动。示例如下:

chmod 777 scf_bootstrap

- 3. 登录 Serverless 控制台,单击左侧导航栏的函数服务。
- 4. 在主界面上方选择期望创建函数的地域,并单击新建,进入函数创建流程。



5. 选择**从头开始**新建函数,根据页面提示配置相关选项。如下图所示:

模板创建 使用示例模版快速	创建一个函数或应用	<mark>从头开始</mark> 从一个 Hello World 示例开始	使用容器镜像 基于容器镜像未创建函数	
基础配置				
函数类型 *	 ● 事件函数 接收云 API、多种触发器的 JSON ● Web函数 ■接接收 HTTP 请求触发函数执行 	1.格式事件触发函数执行。 查看文档 亿 行, 适用于 Web 服务场景。 查看文档 亿		
函数名称 *	只能包含字母、数字、下划线、道	主字符,以字母开头,以数字或字母结尾,2~60个字符		
地域 * 运行环境 *	③ 广州 Nodejs 12.16	v v		
函数代码 () 上传项目前,请修改您的项目监听	满口为9000		
提交方法 * 函数代码 *	 ○ 在线编辑 ○ 本地上传zipt ○ 本地上传zipt ○ 本地上传zipt 	 本地上传文件夹 通过cos上传zip包 		
	请选择文件夹,最大支持250M			

- 函数类型:选择"Web 函数"。
- 函数名称: 填写您自己的函数名称。
- 地域:填写您的函数部署地域,默认为广州。
- 运行环境:选择 "Nodejs 12.16"。
- 部署方式:选择"代码部署",上传您的本地项目。
- 提交方法:选择"本地上传文件夹"。
- **函数代码:**选择函数代码在本地的具体文件夹。
- 6. 单击完成完成 Next.js 项目的部署。

开发管理

部署完成后,即可在 SCF 控制台快速访问并测试您的 Web 服务,并且体验云函数多项特色功能,例如层绑定、日志管理等,享受 Serverless 架构带来的低成本、弹性扩缩容等优势。







快速部署 Nuxtjs 框架

最近更新时间: 2024-11-15 15:13:42

操作场景

本文将为您指导如何通过 Web Function,将您的本地 Nuxt.js SSR 项目快速部署到云端。

本文档主要介绍控制台部署方案,您也可以通过命令行完成部署,详情请参见 通过命令行完成框架部署 。

前提条件

() 说明:

在使用腾讯云云函数服务之前,您需要 注册腾讯云账号 并完成 实名认证。

操作步骤

模板部署:一键部署 Nuxt.js 项目

- 1. 登录 Serverless 控制台,单击左侧导航栏的函数服务。
- 2. 在主界面上方选择期望创建函数的地域,并单击新建,进入函数创建流程。
- 3. 选择使用模板创建来新建函数,在搜索框里输入 webfunc 筛选函数模板,选择Nuxt.js 框架模板并单击下一步。如下图所示:

模板创建 使用示例模糊	从头开始 快速创建一个函数或应用	使用容器镜象 基于容器镜像未创建函数	
模糊搜索	webfunc 參个过端标签用回车總分碼 Express 重看详情 美別 函数 描述 基于 API 网关和 Web 函数,快速邮署 Express 示例项目 标签 WebFunc Express 标签 WebFunc Express 修會 ● 請訊云 部署 19,730次	Q 共32个 Flask 框架模版 查看详情 與則 函数 描述 基于 API 网关和 Web 函数,快速邮署 Flask 示例项目 标签 WebFunc Flask Python3.6 web 作者 会 驚讯云 部區 13,719次	推荐排序 ▼ Nextjs 框架模版 重看详情 类則 函数 描述 基于 API 网关和 Web 函数,快速邮署 Nextjs 示例项目 际盔 WebFunc Nextjs Nodejs12.16 SSR web 作者 ♂ 購訊云 部署 9,973次
	Express框架模板(Auth) 宣看详情 英則 函数 描述 基于CIAM登录认证能力和 Web 函数,快速部署预集成数字身份管控平台(公众版),即CIAM登录认能力 标签 CIAM WebFunc Express 作者 ② 腾讯云 部署 9,689次	Koa 框架模版 查看详情 幾則 函数 描述 基于 API 网关和 Web 函数, 快速邮署 Koa 示例项目 标签 WebFunc Koa Nodejs12.16 web 作會 ② 驚讯云 部署 9,956次	Node12 Web 模版 查看详情 與則 函数 描述 基于 Node,is 12, 使用 API 网关和 Web 函数,快速部 署 Hello world 示例项目 际签 WebFunc 作者 ② 講讯云 部署 13,188次
	Laravel 框架模版 查看详情 类则 函数 描述 基于 API 网关和 Web 函数,快速部署 Laravel 示例项目 标签 WebFunc Laravel Php7 web 作者 ② 簿讯云 部署 9,693次	Nuxtjs 框架模版 查看详情 类则 函数 描述 基于 API 网关和 Web 函数, 快速部署 Nuxtjs 示例项目 标签 WebFunc Neuxtjs Nodejs12.16 SSR web 作者 ② 廣讯云 部署 9,869次	Egg 框架模版 查看详情 类則 函数 描述 基于 API 网关和 Web 函数, 快速部署 Egg 示例项目 标签 WebFunc Egg Nodejs12.16 web 作者 金 顕讯云 部署 9,579次



- 4. 在新建页面,您可以查看模板项目的具体配置信息并进行修改。
- 5. 单击完成即可创建函数。函数创建完成后,您可在函数管理页面查看 Web 函数的基本信息。
- 6. 您可以通过 API 网关生成的访问路径 URL,访问您部署的 Nuxt.js 项目。单击左侧菜单栏中的**触发管理**,查看访问路径。如下图所示:

← nuxtjs_demo-	正常		函数服务帮助文档 忆
函数管理	触发管理		
触发管理	创建触发器		
监控信息	A B157 34 44444		
日志查询	API购关照友	所属别名:默认流量	
并发配额	API服务名		
部署日志	serviceId	service-	
	apild	api	
	请求方法	ANY	
	发布环境	发布	
	鉴权方式	免鉴权	
	启用Base64编码	未启用	
	支持CORS	否	
	后端超时	15s	
	访问路径	https://service	

7. 单击访问路径 URL,即可访问服务 Nuxt.js 项目。如下图所示:



() 说明:

由于 Nuxtjs 框架每次部署前需要重新构建,请确保本地更新代码并且重新 build 之后再进行部署。

自定义部署:快速迁移本地项目上云

前提条件

本地已安装 Node.js 运行环境。



本地开发

1. 参考 Nuxt.js 官方文档,安装并初始化您的 Nuxt.js 项目:

npx create-nuxt-app nuxt-app

2. 在根目录下,执行以下命令在本地直接启动服务。

d nuxt-app && npm run dev

3. 打开浏览器访问 http://localhost:3000 ,即可在本地完成 Nuxt.js 示例项目的访问。如下图所示:



部署上云

接下来执行以下步骤,对已初始化的项目进行简单修改,使其可以通过 Web Function 快速部署,此处项目改造通常分为以下两步:

- 新增 scf_bootstrap 启动文件。
- 修改监听地址与端口为 0.0.0.0:9000 。

具体步骤如下:

1. 在项目根目录下新建 scf_bootstrap 启动文件,在该文件添加如下内容(用于启动服务并指定启动端口):

```
#!/var/lang/node12/bin/node
require("@nuxt/cli")
   .run(["start", "--port", "9000", "--hostname", "0.0.0.0"])
   .catch(error => {
     require("consola").fatal(error);
     require("exit")(2);
   });
```



▲ 注意:

- 此处仅为示例启动文件,具体请根据您的业务场景进行调整。
- 示例使用的是云函数标准 node 环境路径,本地调试时,需修改成您的本地路径。
- 2. 新建完成后,还需执行以下命令修改文件可执行权限,默认需要 777 或 755 权限才可正常启动。示例如下:

chmod 777 scf_bootstrap

- 3. 登录 Serverless 控制台,单击左侧导航栏的函数服务。
- 4. 在主界面上方选择期望创建函数的地域,并单击新建,进入函数创建流程。
- 5. 选择从头开始新建函数,根据页面提示配置相关选项。如下图所示:

模板创建 使用示例模版快速	的建一个函数或应用	<mark>从头开始</mark> 从一个 Hello World 示例开始	使用容器镜像 基于容器镜像未创建图	段
基础配置				
函数类型 *	 事件函数 接收云 API、多种触发器的 JS Web函数 直接接收 HTTP 请求触发函数 	SON 格式事件触发函数执行, 查看文档 [2 如执行,适用于 Web 服务场景。 查看文档 [2		
函数名称 *	只能包含字母、数字、下划线	载、连字符,以字母开头,以数字或字母结尾,2~	60个字符	
地域★ 运行环境★	⑤ 广州 Nodejs 12.16	v v		
函数代码 (上传项目前,请修改您的项目监	监听满口为9000		
提交方法 *	○ 在线编辑 ○ 本地上传	izip包 🔵 本地上传文件夹 📄 通过cos上传。	zip包.	
函数代码 *	还未选择文件夹	上传		

- 函数类型:选择"Web函数"。
- 函数名称:填写您自己的函数名称。
- 地域:填写您的函数部署地域,默认为广州。
- 运行环境:选择"Nodejs 12.16"。
- 部署方式:选择"代码部署",上传您的本地项目。
- 提交方法:选择"本地上传文件夹"。
- 函数代码: 选择函数代码在本地的具体文件夹。
- 6. 单击完成完成 Nuxt.js 项目的部署。

```
∴ 注意:
访问 URL 时,可能由于前端路由导致访问失败,访问时需去掉 /release 路径。
```

开发管理



部署完成后,即可在 SCF 控制台快速访问并测试您的 Web 服务,并且体验云函数多项特色功能,例如层绑定、日志管理等,享受 Serverless 架构带来的低成本、弹性扩缩容等优势。



高级能力配置

层部署

最近更新时间: 2024-10-09 15:37:31

操作场景

目前 Serverless SSR 只支持上传小于 50MB 的代码,当您的项目过大时,您可以将依赖放在层中而不是部署包中,可确保部署包保 持较小的体积。层的具体使用请参见 层管理相关操作 。

操作步骤

创建层

新建层并上传依赖,您可以通过以下两种方式操作:

- 通过 Serverless SSR 控制台 直接创建
- 使用 Serverless 应用中心的 Layer 组件 (参考 Layer 组件)

使用层

您可以通过控制台配置和本地配置两种方法,在项目配置中使用层部署,具体如下:

控制台配置

- 使用模板创建时, Serverless SSR 会自动为您创建名为 \${appName}-layer 的层,并自动帮您把应用的依赖项 node_modules 上传到该层中。
- 导入已有项目时,您可以选择使用新建层或已有层完成部署,选择新建层时,Serverless SSR 会自动帮您把应用的依赖项 node_modules 上传到该层中。

运行环境	Nodejs 10.15 💌				
使用层部署 🛈	新建层	•	请选择版本	•	φ

本地配置

在本地项目目录下,创建 serverless.yml 文件,完成层的名称与版本配置,再通过 sls deploy 进行上传,模板如下:





Nodejs10.14

查看详细配置,请参见 layer 组件全量配置文档 。

项目部署

完成配置部署后,您 SSR 项目中的依赖文件将自动通过层来管理,再次开发部署时,直接引用该层信息即可,缩短部署时间,提高开发 效率。



高级配置

最近更新时间: 2024-11-15 15:13:42

在 Serverless SSR 控制台中,新建应用时可以选择是否启用高级配置,以实现对应用访问的优化。本文主要对默认优化和建议优化的 配置项进行说明。

默认优化

node_modules 拆分为层

默认会把工程中的 node_modules 拆分为 <mark>层(layer</mark>)部署。简单来说,层就是把项目中不经常变更的部分独立出来,云函数运行时 再把层包含的文件挂载到云函数的执行环境中。

- Next.js 和 Nuxt.js 项目的 node_modules 会比较大,项目稳定后,部署不需要每次都更新 node_modules。
- 把 node_modules 部署为层,每次部署只需要上传业务代码,部署会快很多。
- 更新 node_modules 的层时,建议使用 npm install --production 安装依赖包,减少代码包体积,加快部署和启动速度。

点击查看 层部署指引 >>

启用静态资源托管

- Next.js 应用默认会把 .next/static 目录和 public 目录下的所有文件启用静态资源托管。
- 把静态内容(HTML、JavaScript、CSS、图像、视频等文件)托管到COS,不占用云函数的访问资源,成本更低,访问速度更快。

点击查看 静态资源托管改造 >>

建议优化

CDN 加速

前提条件

- 启用 CDN 加速必须使用您已经在腾讯云备案的自有域名,详情请参见备案指引。
- 设置自定义域名,需要在域名管理设置一条 CNAME 记录,指向配置的对象存储的二级域名。
- 更多 COS 设置 CDN 加速域名的说明请参见 COS 域名管理。

配置指引

• 控制台配置

创建应用时,您可以直接在高级配置里开启 CDN 加速,填入您**已备案且配置好解析**的域名,完成 CDN 域名加速的配置。

CDN加速 (✔ 启用	
	埴入域名	
	选择证书ID	Ŧ
	自动配置CNAME	

选择"自动配置 CNAME"时,Serverless SSR 会尝试自动为您配置域名的 CNAME 解析,一般需要5 – 10分钟生效,您可以在 域名解析控制台 查看解析结果,或手动配置解析。

• 本地配置

腾讯云

在 serverless.yml 文件中,inputs 字段里增加以下配置内容,完成 CDN 加速域名的配置。

org: orgDemo
app: appDemo
stage: dev
component: nextjs
name: nextjsDemo
inputs:
此处省略
静态资源相关配置
staticConf:
cosConf:
这里是创建的 COS 桶名称
bucket: serverless-nextjs
cdnConf: # CDN 加速配置
domain: static.test.com
https:
certId: abcdefg

修改配置文件 next.config.js(nuxt.js 项目为 nuxt.config.js),将 STATIC_URL 参数改为您的自定义域名,以 next.js 为 例:

```
const isProd = process.env.NODE_ENV === 'production';
const STATIC_URL ="static.test.com";
module.exports = {
  env: {
    STATIC_URL: isProd ? STATIC_URL : '',
    },
    assetPrefix: isProd ? STATIC_URL : '',
};
```

完成修改后,重新部署,即可完成 CDN 加速域名的配置。

sls deploy

自定义域名

前提条件

- 系统默认会给应用分配一个二级域名,您可以通过绑定自有域名,用自有域名访问应用。
- 自定义域名必须是已经在**腾讯云备案**的域名,目前,腾讯云已推出 Serverless 备案资源包,帮您快速完成 ICP 备案,详情请参考 ICP 备案。
- 设置自定义域名需要在域名管理配置一条 CNAME 记录,指向系统分配的二级域名。 详情请参考 配置自定义域名。

配置指引

• 控制台配置

🔗 腾讯云

创建应用时,您可以直接在高级配置里开启自定义域名,填入您**已备案旦配置好解析**的域名,完成自定义域名的配置。

自定义域名 🕄	✔ 启用	
	埴入域名	
	选择协议	
	自动配置CNAME (j)	

建议您 手动配置域名解析,您也可选择自动配置,勾选"自动配置 CNAME"时,Serverless SSR 会尝试自动为您配置域名的 CNAME 解析,一般需要5 – 10分钟生效,您可以在 域名解析控制台 查看解析结果,或手动配置解析。

• 本地配置

在 serverless.yml 文件中,inputs 字段里增加以下配置内容,完成自定义域名的配置。



完成修改后,重新部署,即可完成自定义域名的配置。



持续构建

最近更新时间: 2024-09-11 14:55:21

功能介绍

Serverless SSR 支持用户通过代码托管实现应用的持续构建,您可以按需选择您的代码仓库,并配置构建计划运行的触发规则。目 前,Serverless SSR 支持 GitHub、GitLab、Gitee 代码源,您可以授权后直接拉取代码仓库信息,也可以填入公开的自定义仓库 完成部署。

触发规则介绍

Serverless SSR 支持您选择需要的触发规则,包括以下两种触发方式:

- 自动触发构建
- 手动触发构建

自动触发构建

() 说明:

自定义仓库不支持自动触发。自定义仓库即指用户自有的外部代码仓库,如已托管在 GitHub、GitLab、Gitee 上的各类 repo。

选择自动触发构建,会自动监听创建应用时所选择的代码仓库分支,根据其变化来自动触发构建计划。您只需要更新代码仓库指定分支里 的项目代码即可,Serverless SSR 会自动完成应用的持续部署。

手动触发构建

选择手动触发,每次应用变更,您需要在 SSR 控制台手动完成触发构建。

- 1. 在 SSR 控制台 首页,单击目标应用进入应用详情页,在页面顶部单击开发部署。
- 2. 在部署方式中选择代码托管。
- 3. 单击页面底部的更新应用,即可完成应用的重新构建。



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	代码托管	本地开发
	下载项目到本地,完成开发后重新上传	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 🖸	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	 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 🔻	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 ☑		
▲ 高级设置			
使用层	test-github-layer v 1	¥	
静态资源托管 🛈	✔ 启用		
	test-github-dev 🗸		
	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		
更新应用			

您可以在应用详情页的部署日志页面查看每次部署信息。



静态资源托管改造

最近更新时间: 2024-09-03 17:02:41

操作场景

当项目中的静态资源过多时,直接部署会导致每次请求页面时,所有的静态资源也要进行重新请求与加载,使得应用的单位时间并发数会 根据页面静态资源请求数而增加,从而造成冷启动问题。Serverless SSR 支持使用静态资源托管来存储您的静态资源,缩短冷启动时 间。

配置流程

控制台配置

模板部署

通过模板创建时,Serverless SSR 已提前在模板中完成了静态资源的配置,并默认开启,您无需进行任何改造操作,即可创建一个使 用静态资源托管的 SSR 应用。

创建流程:创建过程中,Serverless SSR 会自动为您创建一个新的 COS 存储桶,并将项目中自动将编译生成的 .next(或 .nuxt) 和 public 文件夹静态资源上传到该 COS,使得静态资源均通过访问 COS 获取,无需重复请求云函数获取静态资源。

导入已有项目

导入已有项目时,除了在控制台开启"静态资源托管"功能外,您还需对您已有项目进行如下改造:

- **1. 在项目目录下,创建** next.config.js 配置文件(nuxt项目配置文件名为 nuxt.config.js)。
- 2. 在配置文件中加入如下内容:
 - next.js

```
const isProd = process.env.NODE_ENV === 'production';
module.exports = {
    env: {
      STATIC_URL: isProd ? process.env.STATIC_URL : '',
    },
    assetPrefix: isProd ? process.env.STATIC_URL : '',
};
```

○ nuxt.js

```
export default {
    // ...
env: {
    STATIC_URL: process.env.STATIC_URL || '',
    },
    build: {
    extend(config, { isDev, isClient }) {
        if (!isDev && process.env.STATIC_URL) {
            config.output.publicPath = process.env.STATIC_URL
            }
        },
    },
```



// ... }

改造完成后,将已有项目导入并完成部署,Serverless SSR 会自动帮您注入生成的静态文件托管 URL,完成静态资源托管的配置。

命令行部署配置

如果您使用 Serverless Framework 命令行工具完成项目开发,静态资源托管配置步骤如下:

1. 选择静态资源存储桶,获取存储路径。

您可以通过 COS 控制台 或 Serverless COS 组件 快速创建您的存储桶,也可以选择已有存储桶,创建完成后通过 COS 控制台的存储桶概览页获取存储桶路径。

概览		【用量概览】并非实时数据,约有 2 小时时延。该数据仅作为监控数据以供参考,如需查看准确的计费计量数据,可到【费用中心】中下载用量明细进行查看。						
文件列表		如果子账号无法查看概览数据,请联系主账号管理员赋予【Get	Bucket】权限。					
基础配置	~							
安全管理	~	用量概览 标准存储 ▼						
权限管理	~	対象数量 ▼	存储量		本月总流量 💌	本月总请求数 🔻		
域名与传输管理	~	*	МВ		0 в	0 🛪		
容错容灾管理	~		较昨天 ↑ 0.00%		上月总流量 0 B	上月总请求数 24 次		
日志管理	~		*X11/3/43/41 0.007/0					
数据处理	~							
函数计算	~	基本信息		域名信息				
		存储桶名称		访问域名	https://	.com I		
		所属地域		默认CDN加速域名	未邰署			
		创建时间		自定义CDN加速域名	0条			
		访问权限		自定义源站域名	0条			
				全球加速域名	未开启			
				静态网站域名	未开启			
		告警配置	配置告警策略					

- 2. 创建配置文件 next.config.js (nuxt 项目配置文件名为 nuxt.config.js),填入以下内容,STATIC_URL 改为您存储 桶的 URL 路径:
 - next.js



○ nuxt.js

```
const STATIC_URL =
   "https://xxxxx.cos.ap-guangzhou.myqcloud.com";
export default {
   // ...
env: {
    STATIC_URL: STATIC_URL || '',
```





3. 在 serverless.yml 中,新增静态资源相关配置 staticConf,代码示例如下:



4. 修改好配置后,在根目录下执行 scf deploy ,完成部署。

scf deploy



自定义路由项目改造

最近更新时间: 2024-09-11 14:55:21

操作场景

如果没有用 Express 等 Web 框架替代 Next.js 或 Nuxt.js 默认的 Web Server,上传整个工程即可。如果有用到 Express 等 Web 框架,需要做简单的改造,本文以 Express 为例进行说明。

操作步骤

步骤1: 修改启动文件名

您可以通过命令行开发部署修改启动文件名: 把启动 js 文件重命名为 sls.js ,并把它放在项目的根目录下。

步骤2:修改监听端口

将本地监听端口修改为导出 app 应用,以 next.js 为例:

```
const express = require('express')
const next = require('next')
async function createServer() {
  const app = next({ dev: false })
  const handle = app.getRequestHandler()

  await app.prepare()
  const server = express()
  server.all('*', (req, res) => {
    return handle(req, res)
  })
  // define binary type for response
  // if includes, will return base64 encoded, very useful for images
  server.binaryTypes = ['*/*']
  return server
}
// comment out `listen`
// exports app server
// createServer().litsen(3000);
module.exports = createServer
```

 说明: Nuxt.js 项目修改可参见 Nuxt.js 产品文档。



ICP 备案

最近更新时间: 2024-09-11 14:55:21

为什么要备案?

根据国务院令第292号《互联网信息服务管理办法》和工信部令第33号《非经营性互联网信息服务备案管理办法》规定,国家对经营性互 联网信息服务实行许可制度,对非经营性互联网信息服务实行备案制度。未获取许可或者未履行备案手续的,不得从事互联网信息服务, 否则属于违法行为。

因此,使用中国内地(大陆)的 Serverless 服务开办网站并绑定域名服务时必须先办理网站备案,备案成功并获取通信管理局下发的 ICP 备案号后才能开通域名访问。

• ICP 备案号以工信部网站公共查询为准: 查询入口

• 更多相关法律法规请参见: 法律法规

备案场景

如果您的网站托管在腾讯云中国内地(大陆)的 Serverless 服务中,且网站的主办者和域名从未办理过备案,则在开通 Serverless 服务并且使用云函数 SCF 进行自定义域名的 HTTP 访问服务前,需在腾讯云备案系统进行首次备案的操作。

备案准备

- •为了节约备案时间和顺利通过备案,建议您提前了解备案流程。
- 因各地管局要求不同,需准备的材料也有所不同。建议您提前了解各省、自治区、直辖市管局的备案要求,以及相关备案限制。
- Serverless 备案要求: 备案本身不收取任何费用,但通过 Serverless 方式备案需购买云函数5000万次调用次数包与40万GBs资源用量包。请前往资源包购买页面完成购买。

△ 注意:

Serverless 备案方式已向全部用户开放。

备案流程

请参考 <mark>首次备案</mark>,通过小程序完成备案操作。

⚠ 注意: 在执行填写网站信息步骤时,需先开启云函数备案,即在备案类型中选择Serverless。

常见问题

使用 Serverless 的访问域名必须要备案吗?

需根据实际情况进行判断。若默认采用 API 网关提供的三级域名访问自身是无需进行备案的,只有需要自定义域名且该域名指向中国内地 (大陆)的 Serverless 服务才需要备案。例如,访问博客页面等场景。您可根据以下场景判断是否需要备案:

• 不需要备案:

域名解析指向托管于非中国内地(大陆)的 Serverless 服务时,例如中国香港服务器,则不需要备案。

• 需要备案:

域名指向中国内地(大陆)的 Serverless 服务时,需要完成备案。

Serverless 备案流程与云服务器 CVM 备案流程是否一致?



Serverless 备案与 CVM 备案流程上并无实质性差别,所有体验完全一致。唯一的差异点在于,CVM 备案过程中填写的 IP 将直接暴 露给用户,Serverless 备案过程中 IP 将由系统自动拉取并提供。

为什么备案时 IP 地址不可访问?

备案在腾讯云,解析也需要在腾讯云。用户在使用腾讯云 Serverless 方式备案过程中,域名解析 IP 需指向腾讯云站点的 IP 地址 。

Serverless 备案有限制吗?

由于 Serverless 较为轻量,可能会实时删除或新增。为了迎合用户使用 Serverless 习惯,Serverless 备案将以账号作为维度,每 个账号支持购买1个云函数资源包,支持备案2个网站。

已经在 CVM 备案的域名是否需要重新备案?

若同账号下已在 CVM 进行 ICP 备案的域名,可直接通过 API 网关绑定,无需重复备案。

注册的域名当前不使用,还需要备案吗?

域名本身无需备案的,但需实名认证。仅当该域名开通 Web 服务时,才需要备案。

Serverless 站点未完成搭建,需要办理备案吗?

不需要备案。在您的 Serverless 站点正式提供对外访问前,才需要备案。由于备案需要一定的办理时间,建议您提前在腾讯云办理备 案,以便您的站点做好之后可以马上投入使用。

已备案域名接入腾讯云 Serverless 服务,是否需要重新备案?

不需要重新备案,但要办理接入备案。详情请参见 <mark>接入备案</mark> 。

什么情况下需要新增网站备案?

- 如果您存在多个域名,都需要进行备案。
- 您已经有域名进行过备案,现在需要备案新的域名。

接入备案是否可以接入多个 Serverless 服务?

同一主体备案信息可以同时接入多个网站信息,最多同时可以接入10个云函数或 CVM 备案信息。

Serverless 备案资源包购买后是否支持退款?

资源包购买之后立即生效,暂不支持退款。五天无理由退款及其他特殊退款请通过 提交工单 进行咨询。