

静态网站托管 实践教程



腾讯云

【 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 】



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

实践教程

搭建 Hexo

搭建 VuePress

部署 Hugo

已有应用部署

实践教程

搭建 Hexo

最近更新时间：2024-09-09 10:18:01

操作场景

本文档介绍通过腾讯云静态网站托管服务搭建一个 Hexo 网站，并使用云开发 CLI 工具管理部署 Hexo 网站。云开发提供两种方式进行部署：

- 方式一：一键部署项目至云端。
- 方式二：使用 CLI 工具，手动将本地文件部署至云端。

一键部署 Hexo

[单击链接进入](#)，可一键部署 Hexo 到云开发。

CLI 工具部署

前提条件

在进行后续的内容前，请先确保您的电脑中安装了 Node.js 运行环境。如果未安装，可以前往 [Node.js 官网](#) 下载安装。

操作步骤

步骤1：安装云开发 CLI 工具和 Hexo

执行以下命令，安装云开发 CLI 工具以及 Hexo：

```
npm i -g @cloudbase/cli hexo-cli
```

步骤2：在本地初始化一个 Hexo 项目

1. 执行以下命令，创建一个 Hexo 项目：

```
hexo init
```

命令执行之后，可以查看到如下图输出：

```
→ Downloads hexo init blog
INFO Cloning hexo-starter https://github.com/hexojs/hexo-starter.git
Cloning into '/Users/bestony/Downloads/blog'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 161 (delta 12), reused 12 (delta 4), pack-reused 131
Receiving objects: 100% (161/161), 31.79 KiB | 129.00 KiB/s, done.
Resolving deltas: 100% (74/74), done.
Submodule 'themes/landscape' (https://github.com/hexojs/hexo-theme-landscape.git)
registered for path 'themes/landscape'
Cloning into '/Users/bestony/Downloads/blog/themes/landscape'...
remote: Enumerating objects: 1063, done.
remote: Total 1063 (delta 0), reused 0 (delta 0), pack-reused 1063
Receiving objects: 100% (1063/1063), 3.21 MiB | 250.00 KiB/s, done.
Resolving deltas: 100% (585/585), done.
Submodule path 'themes/landscape': checked out '73a23c51f8487cfd7c6deec96ccc75439
60d350'
INFO Install dependencies
yarn install v1.22.1
info No lockfile found.
[1/4] 🔍 Resolving packages...
info There appears to be trouble with your network connection. Retrying...
warning hexo > hexo-log > hexo-bunyan > mv > mkdirp@0.5.3: Legacy versions of mkdirp
 are no longer supported. Please update to mkdirp 1.x. (Note that the API surface
 has changed to use Promises in 1.x.)
warning hexo-renderer-stylus > stylus > mkdirp@0.5.3: Legacy versions of mkdirp ar
 e no longer supported. Please update to mkdirp 1.x. (Note that the API surface
 has changed to use Promises in 1.x.)
warning hexo-renderer-stylus > nib > stylus > mkdirp@0.5.3: Legacy versions of mkd
irp are no longer supported. Please update to mkdirp 1.x. (Note that the API surfa
ce has changed to use Promises in 1.x.)
info There appears to be trouble with your network connection. Retrying...
[2/4] 📦 Fetching packages...
[3/4] 🔗 Linking dependencies...
[4/4] 🏗 Building fresh packages...
success Saved lockfile.
warning Your current version of Yarn is out of date. The latest version is "1.22.4
", while you're on "1.22.1".
info To upgrade, run the following command:
$ brew upgrade yarn
🌟 Done in 62.82s.
INFO Start blogging with Hexo!
→ Downloads
```

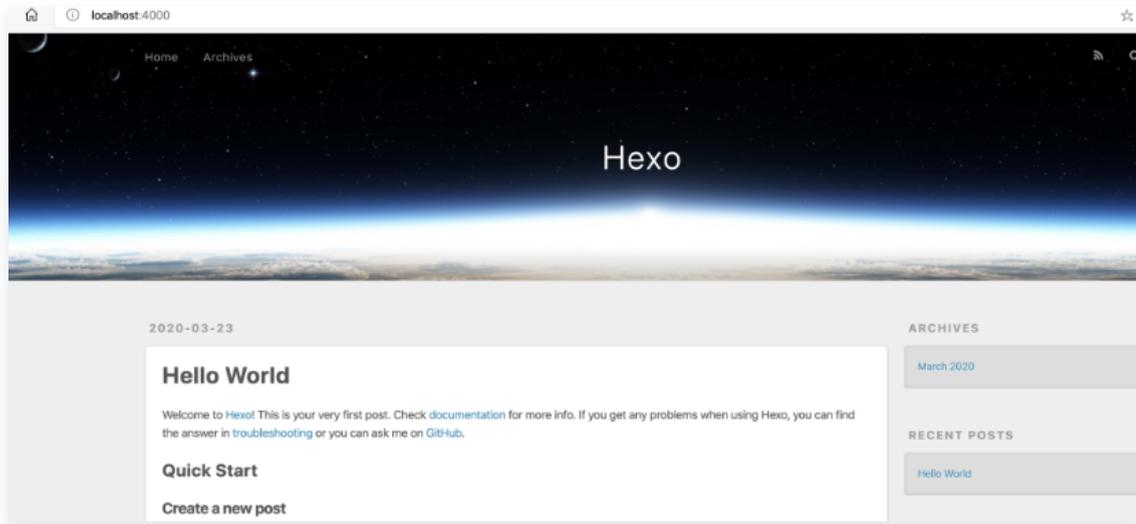
2. 初始化完成后，执行以下命令进入到目录中，并启动预览。

```
cd blog
hexo s
```

命令执行过程如下图所示：

```
hexo s -- 82X5
→ Downloads cd blog
→ blog hexo s
INFO Start processing
INFO Hexo is running at http://localhost:4000 . Press Ctrl+C to stop.
```

3. 在浏览器中输入 `localhost:4000`，看到 Hexo 的界面即说明已成功完成 Hexo 的本地初始化。



步骤3：创建一个云开发环境

完成本地的 Hexo 建设，接下来创建一个云开发环境，用于部署 Hexo。

1. 开通云开发环境

- 1.1 登录腾讯云 [云开发控制台](#)，单击**新建**，新建一个环境进行部署。
- 1.2 应用来源选择**模板仓库**或**代码导入**，并进行相应的配置。
- 1.3 选择环境所属地域，选择**按量计费**计费方式，填写环境名称后单击**下一步**。

创建环境

应用模板 > **2 环境信息** > 应用配置

地域:

计费方式: [模式对比](#)

环境名称:
环境名称支持不超过25个字符的小写字母、数字及'-'字符的组合, 且仅支持字母开头、字母数字结尾, 不支持连续-

标签 (选填) ①:
[+ 添加](#)

费用: 按实际使用的资源量收费。 [查看计费详情](#)
 同意 [《计费规则》](#)

1.4 根据提示进行相应的配置创建一个云开发环境。

❗ 说明

在开通环境以后，请记住您的环境 ID，该 ID 后续部署需要用到。

2. 开通静态网站托管

云开发环境创建完成后，单击左侧菜单栏中的**静态网站托管**，根据页面提示单击**开通静态网站托管**进行开通。



当出现如下图时，则说明静态网站托管已经开通。



您现在可以通过上传文件手动上传一个文件测试，稍后，我们将会使用云开发 CLI 来完成上传。

步骤4：使用 CLI 部署 Hexo

1. 初始化云开发 CLI

完成云开发环境配置后，需要初始化云开发 CLI，从而实现借助 CLI 工具来上传页面（您也可以通过网页端直接上传，但如果您博客的文章比较多，建议使用 CLI 工具进行上传）。

1.1 在命令行中执行以下命令登录云开发：

```
tcb login
```

1.2 执行之后将提醒您需要在网页中进行授权：



```
tcb login
→ dist tcb login
✓ 已打开云开发 CLI 授权页面，请在云开发 CLI 授权页面同意授权
∴ 获取授权中 ...
```

在弹出的页面中单击**确认授权**。



确认授权后，您会看到控制台输出相应的命令部署。至此，您的云开发 CLI 已初始化完毕。

2. 构建 Hexo 并上传

2.1 在 Hexo 目录，执行 `Hexo g` 命令来生成文件，Hexo 会默认将文件生成在 Public 目录下。如下图所示：

```
→ blog hexo g
INFO Start processing
INFO Files loaded in 80 ms
INFO Generated: index.html
INFO Generated: archives/index.html
INFO Generated: fancybox/blank.gif
INFO Generated: fancybox/jquery.fancybox.css
INFO Generated: fancybox/jquery.fancybox.pack.js
INFO Generated: fancybox/jquery.fancybox.js
INFO Generated: fancybox/fancybox_overlay.png
INFO Generated: fancybox/fancybox_loading@2x.gif
INFO Generated: fancybox/fancybox_loading.gif
INFO Generated: fancybox/fancybox_sprite.png
INFO Generated: fancybox/fancybox_sprite@2x.png
INFO Generated: archives/2020/03/index.html
INFO Generated: archives/2020/index.html
INFO Generated: css/fonts/fontawesome-webfont.woff
INFO Generated: js/script.js
INFO Generated: fancybox/helpers/jquery.fancybox-media.js
INFO Generated: fancybox/helpers/jquery.fancybox-buttons.css
INFO Generated: fancybox/helpers/jquery.fancybox-buttons.js
INFO Generated: fancybox/helpers/jquery.fancybox-thumbs.css
INFO Generated: fancybox/helpers/jquery.fancybox-thumbs.js
INFO Generated: css/style.css
INFO Generated: fancybox/helpers/fancybox_buttons.png
INFO Generated: css/fonts/FontAwesome.otf
INFO Generated: css/fonts/fontawesome-webfont.eot
INFO Generated: css/fonts/fontawesome-webfont.svg
INFO Generated: css/fonts/fontawesome-webfont.ttf
INFO Generated: css/images/banner.jpg
INFO Generated: 2020/03/23/hello-world/index.html
INFO 28 files generated in 317 ms
```

2.2 文件生成完成后，可以执行以下命令来进行部署（需要将 EnvID 替换为前面您的环境 ID）：

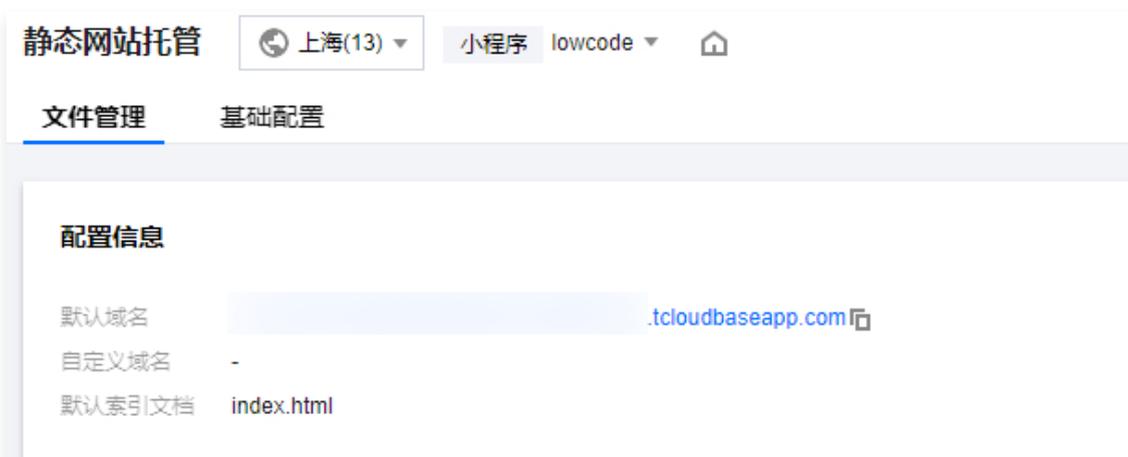
```
cd public
tcb hosting deploy ./ -e EnvID
```

稍等片刻等待部署完成，即可预览页面。



步骤5: 浏览 Hexo

1. 登录腾讯云 [云开发控制台](#)，单击左侧菜单栏中的**静态网站托管** > **文件管理**，进入文件管理页面。
2. 在文件管理页面可以找到默认的域名，单击域名，即可访问您部署的 Hexo。



相关命令

卸载 Hexo

```
npm uninstall -g @cloudbase/cli hexo-cli
```

搭建 VuePress

最近更新时间：2024-10-09 15:24:41

操作场景

本文档介绍通过腾讯云静态网站托管服务搭建一个 VuePress 网站，并使用云开发 CLI 工具管理部署 VuePress 网站。

- 方式一：一键部署项目至云端。
- 方式二：使用 CLI 工具，手动将本地文件部署至云端。

一键部署 VuePress

[单击链接进入](#)，可一键部署 VuePress 到云开发。

CLI 工具部署

前提条件

在进行后续的内容前，请先确保您的电脑中已安装 Node.js 运行环境。如果未安装，可以访问 [Node.js 官网](#) 下载安装。

操作步骤

步骤1：安装云开发 CLI 工具和 VuePress

执行以下命令，安装云开发 CLI 工具以及 VuePress。

```
npm i -g @cloudbase/cli vuepress
```

步骤2：在本地初始化一个 VuePress 项目

1. 执行以下命令，在本地创建一个目录，本文以 tcb 为例：

```
mkdir tcb && cd tcb
```

2. 进入到 tcb 目录后，执行以下命令，创建一个默认的 hello world 文件。

```
echo "# Hello TCB & Vuepress" > README.md
```

```
→ tcb echo "# Hello TCB & Vuepress" > README.md  
→ tcb █
```

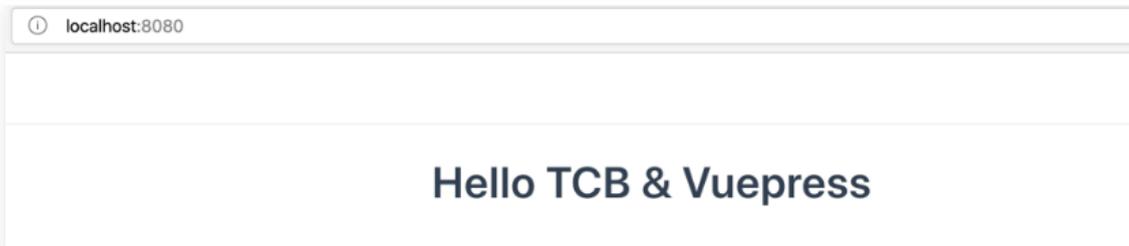
3. 执行以下命令，启动 VuePress。

```
vuepress dev
```

等待其编译完成，完成后如下图所示：

```
success [15:26:13] Build 394910 finished in 113 ms! ( http://local  
host:8080/ )  
█
```

4. 打开浏览器访问 `localhost:8080`，可以看到上述步骤创建的文档产生的页面，说明已成功完成 VuePress 的本地初始化。



步骤3：创建一个云开发环境

完成本地的 Vuepress 建设，接下来创建一个云开发环境，用于部署 VuePress。

1. 开通云开发环境

1.1 登录腾讯云 [云开发控制台](#)，单击**新建**，新建一个环境进行部署。

1.2 应用来源选择**模板仓库**或**代码导入**，并进行相应的配置。

1.3 选择环境所属地域，选择按量计费计费方式，填写环境名称后单击下一步。

创建环境

应用模板 > **2 环境信息** > 应用配置

地域

计费方式 按量计费 包年包月 [模式对比](#)

环境名称

免费资源 开启免费资源
免费资源按照最早开通环境时的周期每个月赠送用量，超过用量后进行计费。[了解免费额度说明](#)

费用 按实际使用的资源量收费。[查看计费详情](#)

1.4 根据提示进行相应的配置创建一个云开发环境。

说明

在开通环境以后，请记住您的环境 ID，该 ID 后续部署需要用到。

2. 开通静态网站托管

云开发环境创建完成后，单击左侧菜单栏中的**静态网站托管**，根据页面提示单击**开通静态网站托管**进行开通。



当出现如下图时，则说明静态网站托管已经开通。



您现在可以通过上传文件手动上传一个文件测试，稍后，我们将会使用云开发 CLI 来完成上传。

步骤4: 使用 CLI 部署 VuePress

1. 初始化云开发 CLI

完成云开发环境配置后，需要初始化云开发 CLI，从而实现借助 CLI 工具来上传页面（您也可以通过网页端直接上传，但如果您博客的文章比较多，建议使用 CLI 工具进行上传）。

1.1 在命令行中执行以下命令登录云开发：

```
tcb login
```

1.2 执行之后将提醒您需要在网页中进行授权：



在弹出的页面中单击**确认授权**。



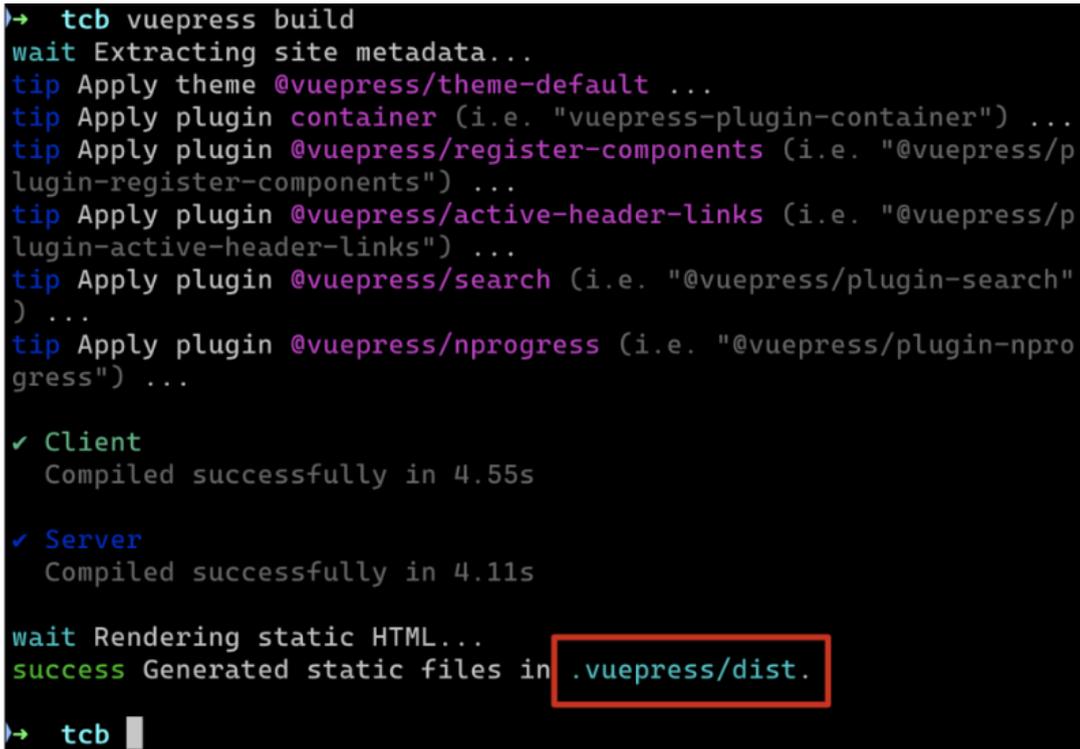
确认授权后，您会看到控制台输出相应的命令部署。至此，您的云开发 CLI 已初始化完毕。

2. 部署 VuePress

2.1 在 VuePress 的目录，执行以下命令构建静态页面：

```
vuepress build
```

构建完成后，系统将提醒您生成的静态文件在 `.vuepress/dist` 。如下图所示：



```
→ tcb vuepress build
wait Extracting site metadata...
tip Apply theme @vuepress/theme-default ...
tip Apply plugin container (i.e. "vuepress-plugin-container") ...
tip Apply plugin @vuepress/register-components (i.e. "@vuepress/p
lugin-register-components") ...
tip Apply plugin @vuepress/active-header-links (i.e. "@vuepress/p
lugin-active-header-links") ...
tip Apply plugin @vuepress/search (i.e. "@vuepress/plugin-search"
) ...
tip Apply plugin @vuepress/nprogress (i.e. "@vuepress/plugin-npro
gress") ...

✓ Client
  Compiled successfully in 4.55s

✓ Server
  Compiled successfully in 4.11s

wait Rendering static HTML...
success Generated static files in .vuepress/dist.
→ tcb
```

2.2 需要将 `.vuepress/dist` 中的文件夹中的内容上传到云开发静态网站托管。

执行以下命令进入到 `dist` 目录：

```
cd .vuepress/dist
```

在 `dist` 目录中执行以下命令上传文件，此处需要将 `EnvID` 替换为您自己环境的环境 ID：

```
tcb hosting:deploy ./ -e EnvID
```

稍等片刻，文件即可上传完毕，如下图所示：

```

→ dist tcb hosting:deploy ./ -e docs-85a181
文件传输中 [ ] 0%
文件传输中 [= ] 1%
文件传输中 [= ] 1%
文件传输中 [=====] 19%
文件传输中 [=====] 30%
文件传输中 [=====] 30%
文件传输中 [=====] 100%
0.0s
✓ 文件共计 9 个
✓ 文件上传成功 9 个

```

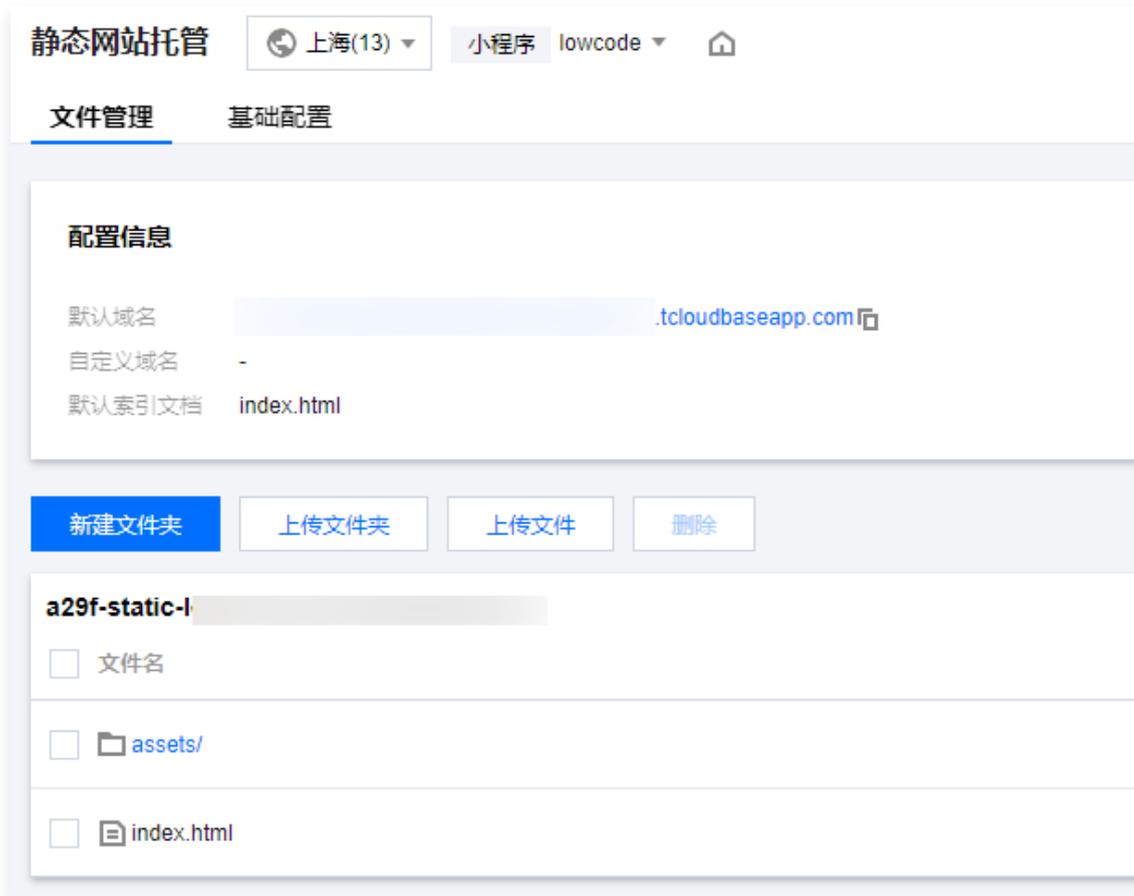
状态	文件
✓	assets/img/search.83621669.svg
✓	index.html
✓	assets/js/3.9eae86a0.js
✓	assets/js/2.8f362bbe.js
✓	assets/css/0.styles.68aa17eb.css
✓	assets/js/4.cf28fdb8.js
✓	assets/js/6.4f6ea00d.js
✓	assets/js/5.6bee3498.js
✓	assets/js/app.e85c920e.js

```

✗ 文件上传失败 0 个
→ dist

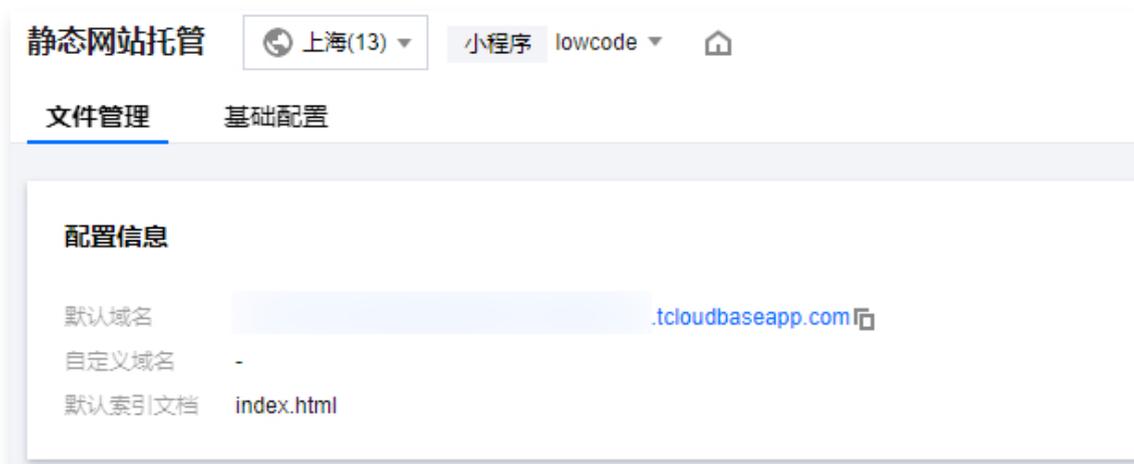
```

此时，您在云开发控制台 [静态网站托管](#) 页面同样可以查看到已上传的文件，说明成功上传。



步骤5: 浏览 VuePress

1. 登录腾讯云 [云开发控制台](#)，单击左侧菜单栏中的 [静态网站托管](#) > [文件管理](#)，进入文件管理页面。
2. 在文件管理页面可以找到默认的域名，单击域名，即可访问您部署的 VuePress。



部署 Hugo

最近更新时间：2024-06-27 14:33:11

操作场景

云开发 CloudBase 是一款云端一体化的产品方案，采用 serverless 架构，免环境搭建等运维事务，支持一云多端，助力快速构建小程序、Web 应用、移动应用。

云开发静态网站托管支持通过云开发 SDK 调用服务端资源，例如云函数、云存储、云数据库等，从而将静态网站扩展为全栈网站。

无论是腾讯云云开发用户，还是小程序云开发用户，只需开通按量付费，即可享有云开发静态网站托管服务。Hugo 是一款用 Go 编写的静态站点生成器，具有丰富的主题资源。本文将为您介绍如何在云开发静态网站托管部署 Hugo。

操作步骤

步骤1：安装 Hugo

1. 执行以下命令安装 Hugo：

```
brew install hugo
```

ⓘ 说明

Windows 用户可以前往 Hugo 的 Github 仓库下载 Hugo 的可执行程序进行安装，具体安装流程请参见 [Hugo官方操作文档](#)。

2. 执行以下命令，使用 Hugo 创建一个 blog 项目：

```
hugo new site hugo-demo && cd hugo-demo
```

3. 使用 Hugo 添加一个主题。

3.1 执行以下命令，将主题添加进项目中：

```
git init
git submodule add https://github.com/budparr/gohugo-theme-ananke.git
themes/ananke
```

ⓘ 说明

对于无法使用 Git 的用户，可以使用如下方法来添加主题。

- a. 下载最新版的 [主题安装包](#)。
- b. 解压主题安装包，找到 “gohugo-theme-ananke-master” 文件夹。
- c. 重命名文件夹为 “ananke”，并将其移动到 `hugo-demo` 项目中的 “themes/” 文件夹下。

3.2 执行以下命令，添加主题至配置文件中：

```
echo 'theme = "ananke"' >> config.toml
```

4. 执行以下命令，创建一篇用于测试的文章：

```
hugo new posts/my-first-post.md
```

5. 执行以下命令，在目录中运行：

```
hugo server
```

6. 自定义主题（可选）

截止上一步骤，我们建立的博客已经可以访问，如果您需要按照自己的需求继续美化博客，可以按照下列步骤进行：

6.1 打开配置文件 `config.toml`，文件示例如下：

```
baseUrl = "http://example.org/"
languageCode = "en-us"
title = "My New Hugo Site"
theme = "ananke"
```

6.2 修改 “title” 的值为网站名称。

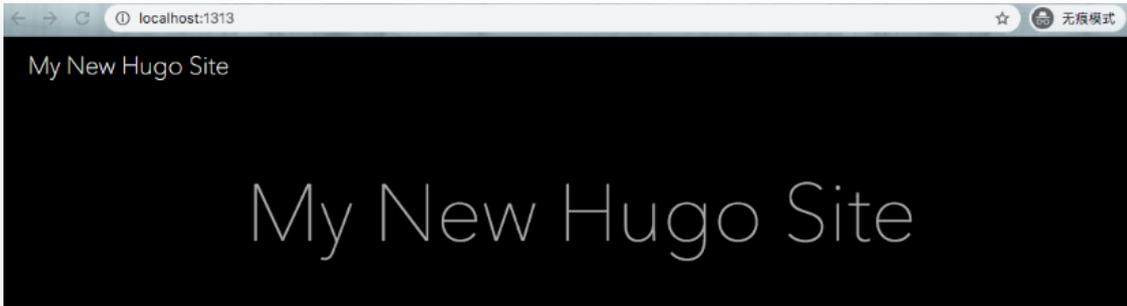
6.3 设置域名 “baseUrl” 为默认或者自定义域名。

ⓘ 说明

此处默认/自定义域名可以使用云开发提供的域名，请完成后续的 [步骤2：静态托管部署](#)。

6.4 如需了解主题 “ananke”，请参见 [gohugo-theme-ananke](#)。如需配置更多主题，请参见 [自定义主题](#)。

7. 在浏览器输入 `http://localhost:1313` 即可查看效果：



8. 使用如下代码部署编译完成的静态页面文件：

```
hugo -D
```

生成好的静态页面文件将放在项目的 `public` 目录中，目录结构如下：

```
|— 404.html
|— categories
|   |— index.html
|   └─ index.xml
|— dist
|   |— css
|   |   └─ app.1cb140d8ba31d5b2f1114537dd04802a.css
|   └─ js
|       └─ app.3fc0f988d21662902933.js
|— images
|   └─ gohugo-default-sample-hero-image.jpg
|— index.html
|— index.xml
|— posts
|   |— index.html
|   |— index.xml
|   |— my-first-post
|   |   └─ index.html
|   └─ page
|       └─ 1
|           └─ index.html
|— sitemap.xml
└─ tags
    |— index.html
    └─ index.xml
```

步骤2：静态托管部署

方式一：结合 Github 自动部署

自动部署将会在每次更新仓库内文件时，自动提交更新到静态网站托管中，实现博客的自动更新。

1. 提交本地文件至 GitHub 仓库中，具体操作步骤请参见 [添加现有项目](#) 到 GitHub。
2. 本地文件同步到 GitHub 仓库中后，需要配置自动化部署流程，具体步骤请参见 [GitHub Action 文档](#)，通过使用 Action 自动化，调用命令行工具完成部署。

方式二：手动部署

1. 登录 [云开发控制台](#)，单击**新建**，选择空模板，单击**下一步**，填写环境名称并开通“按量计费”环境。



开通环境以后，进入 [环境概览](#) 页面请记住您的 `环境 Id`，这个 ID 后续部署需要用到。

2. 在本地安装 Node.js。如未安装请前往 [Node.js 官网](#) 下载安装，并确保 Node.js 安装成功。
3. 打开命令提示符，执行以下命令安装 cloudbase cli:

```
npm install -g @cloudbase/cli
```

4. 执行以下登录命令登录云开发:

```
tcb login
```

登录成功如下图所示：

```
→ ~ cloudbase login  
✓ 已打开云开发 CLI 授权页面，请在云开发 CLI 授权页面同意授权！  
: 获取授权中... █
```

5. 在弹出的页面中单击**确认授权**进行授权：



6. 执行以下命令，在 hugo-site 中部署 public 目录中的文件：

```
cloudbase hosting deploy ./public -e EnvID
```

此处的 EnvID 替换为上述 [步骤](#) 创建好的环境 ID。

```
→ hugo-demo git:(master) x cloudbase hosting:deploy ./public -e blog-19f8b  
a  
文件部署中...  
文件传输中 [-----] 100% 0.0s  
✓ 文件部署成功!  
→ hugo-demo git:(master) x █
```

7. 登录 [云开发控制台](#)，进入 [静态网站托管](#) 页面，可以找到默认的域名，单击域名，即可看到您刚部署的 Hugo。


```
CloudBase Framework info      Version v1.2.10
CloudBase Framework info      Github:
https://github.com/Tencent/cloudbase-framework

CloudBase Framework info      EnvId webpage
检测到当前项目包含 Vue.js 项目

构建脚本 `npm run build`
本地静态文件目录 `dist`

是否需要修改默认配置 No
请输入应用唯一标识 (支持大小写字母数字及连字符, 同一账号下不能相同) test-vue
是否需要保存当前项目配置, 保存配置之后下次不会再次询问 Yes
CloudBase Framework info      install plugins
```

❗ 说明

如果您的应用未被自动检测识别出语言和框架配置, 您可以查看 [配置说明](#) 和 [插件体系](#) 来进行自定义配置, 再执行 `cloudbase` 命令进行部署。

项目访问

前往 [我的应用](#) 中访问和管理应用。

仓库项目导入

准备工作

1. [注册腾讯云账号](#)。
2. 创建按量计费 [云开发环境](#), 获取 [环境 ID](#)。
3. 安装 [Node.js](#)。
4. 安装 [Cloudbase CLI](#)。
5. 已有 Github/Gitee/Gitlab 项目链接。

项目部署

1. 代码导入。

进入 [云开发控制台](#) 首页，单击**新建**，切换至**代码导入**标签页。如下图所示：

创建环境

1 应用模板 > 2 环境信息 > 3 应用配置

应用来源 模板仓库 代码导入

来源地址
请输入代码来源，支持 Github/Gitlab/Gitee 公开仓库地址，代码规则[详见](#) [🔗](#)

应用目录
请输入应用所在的子目录

部署分支
请输入部署分支

① 说明

如果您需要部署项目至已有**按量计费**环境中，在该环境标签内，单击**应用部署**，切换至**代码导入**标签页。



2. 输入应用来源信息。

输入应用来源地址、应用目录及应用分支相关信息，同步项目到云开发环境中。

说明

您可前往 [云开发示例仓库](#) 中，查找更多应用；您也可以将示例中的项目拷贝至个人仓库中，进行二次开发后同步至云开发环境中进行部署。

3. 输入应用配置信息。

如果您导入的项目中已经包含项目配置信息，单击**完成**开始部署应用；若当前仓库没有配置文件，请手动输入 [应用配置信息](#)，进行应用构建和部署。

项目访问

前往 [我的应用](#) 中访问和管理应用。