

使用docker部署nginx并配置https

我只有一台服务器，但我想在这台服务器上运行多个项目，怎么办？

总不能靠加端口区分吧？

百度和Google是个好东西，于是我找到了答案，使用nginx。

通过nginx，我可以给我的一台服务器配置两个域名，然后通过不同的域名访问我的两个项目。

公告

昵称： TandK

园龄： 2年9个月

粉丝： 0

关注： 1

[+加关注](#)

实现方式

大概流程是这么玩的，你在服务器配置好nginx，让nginx监听80（http端口）和443（https端口），然后在nginx配置文件写明转发到哪里。

比如我有一个博客系统，还有一个邮件系统。

为了区分开，我给了这两个项目分别两个域名：blog.tandk.com和mail.tandk.com。

是的，这两个域名都是同属一个主域名tandk.com下的，以我上面的例子来说，你只要在域名服务商那里购买一个叫tandk.com的域名，就可以通过配置，无需购买获得上面两个域名。

具体的配置过程其实也挺简单，我只在腾讯云配过，其他的平台应该大同小异，如果你不大了解，可以看看我的这篇文章：[《腾讯云配置多个域名前缀》](#)

回到nginx，两个域名有了，我要配置nginx的配置文件，要告诉nginx，当我访问blog.tandk.com时，转发到内网的8080端口，当我访问mail.tandk.com时，转发到内网的8181端口。

再配好ssl证书，就可以正常访问了。

ssl证书可以让你用https访问你的域名，如果你觉得http可以满足的话，那么不需要ssl也行。

具体实现

思路有了，接下来便是一步一步完成。

看到标题会点进来的，我就当你对于docker有一定了解了，这里就不提怎么安装docker了。

我们直接用docker安装nginx：

```
# 安装
docker pull nginx
```

<

2021年10月

>

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

找找看

谷歌搜索

我的标签

索引(2)

MySQL(2)

Spring(1)

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

PDFCROWD

```
# 安装后查看docker镜像
```

```
docker images
```



安装就只需要一个命令，连安装包都不用自己找，这就是我用docker的原因。

建目录用于存放nginx配置文件、证书文件：

```
mkdir /opt/docker/nginx/conf.d -p
```

```
mkdir /opt/docker/nginx/cert -p
```

nginx的ssl证书长这个样子，两个都要，中间打码的是你的域名：

 1_ [redacted].bundle.crt	2021/1/9 13:59	安全证书	4 KB
 2_ [redacted].key	2021/1/9 13:59	KEY 文件	2 KB

把他们俩放到服务器的/opt/docker/nginx/cert目录下。

查看内网的ip，记录下来，等下写配置文件要用：

```
# linux查看内网ip
```

```
ifconfig
```

接下来说说配置文件的一点注意事项。

了解docker的都知道，docker容器启动后，容器内部有自己的文件系统，配置文件也在里面。

因此我们需要把配置文件挂载到外部，就是我们的linux服务器上。

这样我们每次需要修改nginx的配置文件的时候，只要修改挂载到外部的文件，容器内的文件也会相应改变，能省很多功夫。

强烈建议挂载文件！！！不然的话会有很多坑，自身的血泪说出这句话。

要挂载文件，首先需要启动nginx，进去容器内部查看配置文件在哪里，并将其复制出来：

SpringBoot(1)

java(1)

随笔档案

2021年10月(1)

2021年6月(1)

2021年5月(1)

2021年1月(4)

2020年12月(2)

阅读排行榜

1. SpringBoot同时接收单个对象和List<object>参数(1769)

2. RESTful API实践总结(486)

```
# 启动nginx
docker run --name nginx -p 80:80 -d nginx

# 进入docker的nginx容器
docker exec -it nginx bash
```

确定进入容器内部后，用下面两个命令查找nginx的配置文件在哪：

```
# 查找nginx配置文件default.conf
# 这个在/etc/nginx/conf.d/default.conf
find / -name "default.conf"

# 查找nginx配置文件nginx.conf
# 这个在/etc/nginx/nginx.conf
find / -name "nginx.conf"
# 注：两者不在同一目录下。
```

接下来退出容器（进入容器内部只是确认一下配置文件的位置而已）：

```
exit
```

使用docker的cp命令把这两个配置文件复制到刚刚建好的目录下：

```
# 把docker内的default.conf复制到外部
docker cp nginx:/etc/nginx/conf.d/default.conf
/opt/docker/nginx/conf.d/default.conf

# 把docker内的nginx.conf复制到外部
docker cp nginx:/etc/nginx/nginx.conf /opt/docker/nginx/conf.d/nginx.conf
```

保险起见，这个时候先不修改容器外部的配置文件，先挂载文件启动docker，启动成功了，确定没问题了，再去修改配置文件。

3. MySQL索引性能分析(369)

4. 浅谈sql索引(273)

5. 使用docker部署nginx并配置https(236)

评论排行榜

1. 使用docker部署nginx并配置https(4)

2. RESTful API实践总结(3)

推荐排行榜

1. 使用docker部署nginx并配置https(3)

2. RESTful API实践总结(2)

3. 浅谈sql索引(1)

最新评论

首先停掉之前的容器并删除：

停止容器

```
docker stop nginx
```

删除容器

```
docker rm -f nginx
```

带着挂载文件的命令启动nginx：

挂载

```
docker run --name nginx -p 80:80 -p 443:443 -v
/opt/docker/nginx/conf.d/nginx.conf:/etc/nginx/nginx.conf -v
/opt/docker/nginx/conf.d/default.conf:/etc/nginx/conf.d/default.conf -v
/opt/docker/nginx/cert:/etc/nginx -d nginx
```

-p命令表示容器内部和外部监听的端口，这部分不细说。

-v表示挂载的文件，-v [该文件在容器外部的位置]:[该文件在容器内部的位置]。

使用下面的命令查看是否启动成功：

查看正在运行的容器

```
docker ps
```

查看全部容器，包括没有启动的容器

如果上面的命令出现的列表没有nginx，使用这个命令

```
docker ps -a
```

如果确定启动失败了，查看日志定位失败的原因

```
docker logs nginx
```

当nginx启动成功后，我们只需要修改配置文件，就可以实现多个域名和https访问的需求啦。

1. Re:使用docker部署nginx并配置https

@tensor_zhang 哈哈，其实是写给自己看的，前阵子配起来踩了不少坑，这东西又不会经常用到，生怕下次需要配的时候忘记了，又要从头开始踩坑。...

--TandK

2. Re:使用docker部署nginx并配置https

有免费的泛域名证书, 有效期三个月. 试试

--MyCoolDog

3. Re:使用docker部署nginx并配置https

真-手模手教学

--love喵

4. Re:使用docker部署nginx并配置https

真-手把手教学

--tensor_zhang

nginx的配置文件有两个，nginx.conf和default.conf。

点开看nginx.conf的话，会发现里面引入了default.conf的内容，也就是说这是一个文件分成两个文件来写。

我们只需要修改default.conf的内容即可达到我们的目的。

这里粘一下linux编辑文本的命令：

```
# 打开文件
vim 文件名

# 进入默认不可编辑，需要按下i后才可以编辑

# 编辑完成后，需要按esc退出编辑状态

# 退出并保存
:wq

# 退出不保存
:q!
```

修改default.conf的内容为如下，需根据你的情况做出改变：

```
server {
    listen 80; #侦听80端口
    listen 443 ssl; #侦听443端口，用于SSL
    server_name tandk.com www.tandk.com; # 自己的域名
    # 注意证书文件名字和位置，是从/etc/nginx/下开始算起的
    ssl_certificate 1_tandk.com_bundle.crt;
    ssl_certificate_key 2_tandk.com.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

5. Re:RESTful API实践总结

@xuanbg 我的思路是这样的，假设你在自己的电脑上有文章这么一个文件夹，那么你想指向审核通过和不通过的资源，通常情况下是新建两个对应的文件夹分类好。按照这么一个思路，我个人的答案是这样的：#

...

--TandK

```

ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
ssl_prefer_server_ciphers on;

client_max_body_size 1024m;

location / {
    proxy_set_header HOST $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    # 这里写的是我的腾讯云内网地址,不知道为啥,不能用127.0.0.1...
    proxy_pass http://xx.xx.xx.xx:8090;
}
}

```

这上面便配好了一个域名的https和http访问,假如你想多配一个域名,拿我上面的blog.tandk.com和mail.tandk.com来举例,每个域名要加多一个server:

```

server {
    listen 80; #侦听80端口
    listen 443 ssl; #侦听443端口,用于SSL
    server_name blog.tandk.com; # 自己的域名
    # 注意证书文件位置,是从/etc/nginx/下开始算起的
    ssl_certificate 1_blog.tandk.com_bundle.crt;
    ssl_certificate_key 2_blog.tandk.com.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    client_max_body_size 1024m;
}

```

```

location / {
    proxy_set_header HOST $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    # 这里写的是我的腾讯云内网地址,不知道为啥,不能用127.0.0.1...
    proxy_pass http://xx.xx.xx.xx:8080;
}

server {
    listen 80; #侦听80端口
    listen 443 ssl; #侦听443端口,用于SSL
    server_name mail.tandk.com; # 自己的域名
    # 注意证书文件位置,是从/etc/nginx/下开始算起的
    ssl_certificate 1_mail.tandk.com_bundle.crt;
    ssl_certificate_key 2_mail.tandk.com.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    client_max_body_size 1024m;

    location / {
        proxy_set_header HOST $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # 这里写的是我的腾讯云内网地址,不知道为啥,不能用127.0.0.1...
        proxy_pass http://xx.xx.xx.xx:8181;
    }
}

```



```
}  
}
```

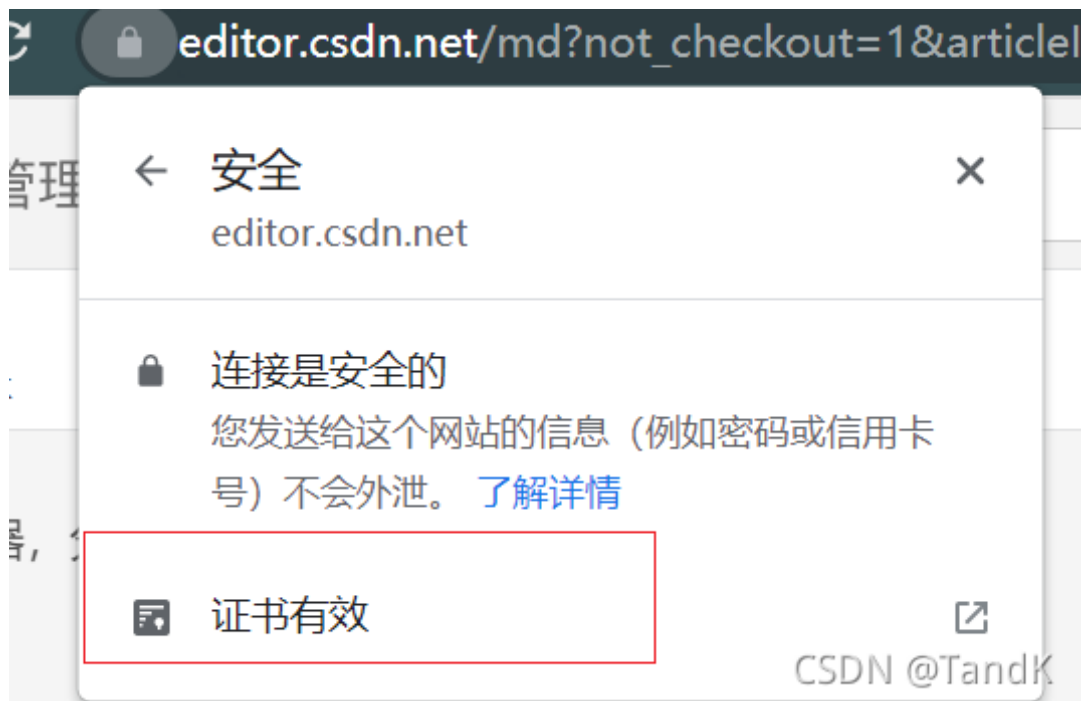
配置完毕后，保存、停止容器再重启：

```
# 停止nginx  
docker stop nginx  
  
# 启动容器  
docker start nginx  
  
# 查看是否启动成功  
docker ps
```

打开浏览器，分别用http和https访问你配置的域名，来确认是否成功。

查看https是否生效，可以按下面的截图来查看：





关于ssl证书的坑

如果你细心点，看我上面的配置文件会发现，两个server，随着域名的不同，ssl证书的名字也不一样了。

对的，对于每一个不同的子域名，我都需要去申请一个ssl证书。

对于mail.tandk.com，我需要去申请一个ssl证书，对于blog.tandk.com，我还需要去申请一个新的ssl证书。

那么能不能申请一个通配的证书去适配一个主域名下的所有子域名呢。

有的，花钱。

别的平台我不太清楚，腾讯云有那种证书，比如你可以申请一个*.tandk.com的证书，那样你所有子域名就都能用这个证书了。

但是我看了一下这种证书一年要花1600，像我这种穷逼还是算了，反正申请一个免费证书也花不了多少时间。

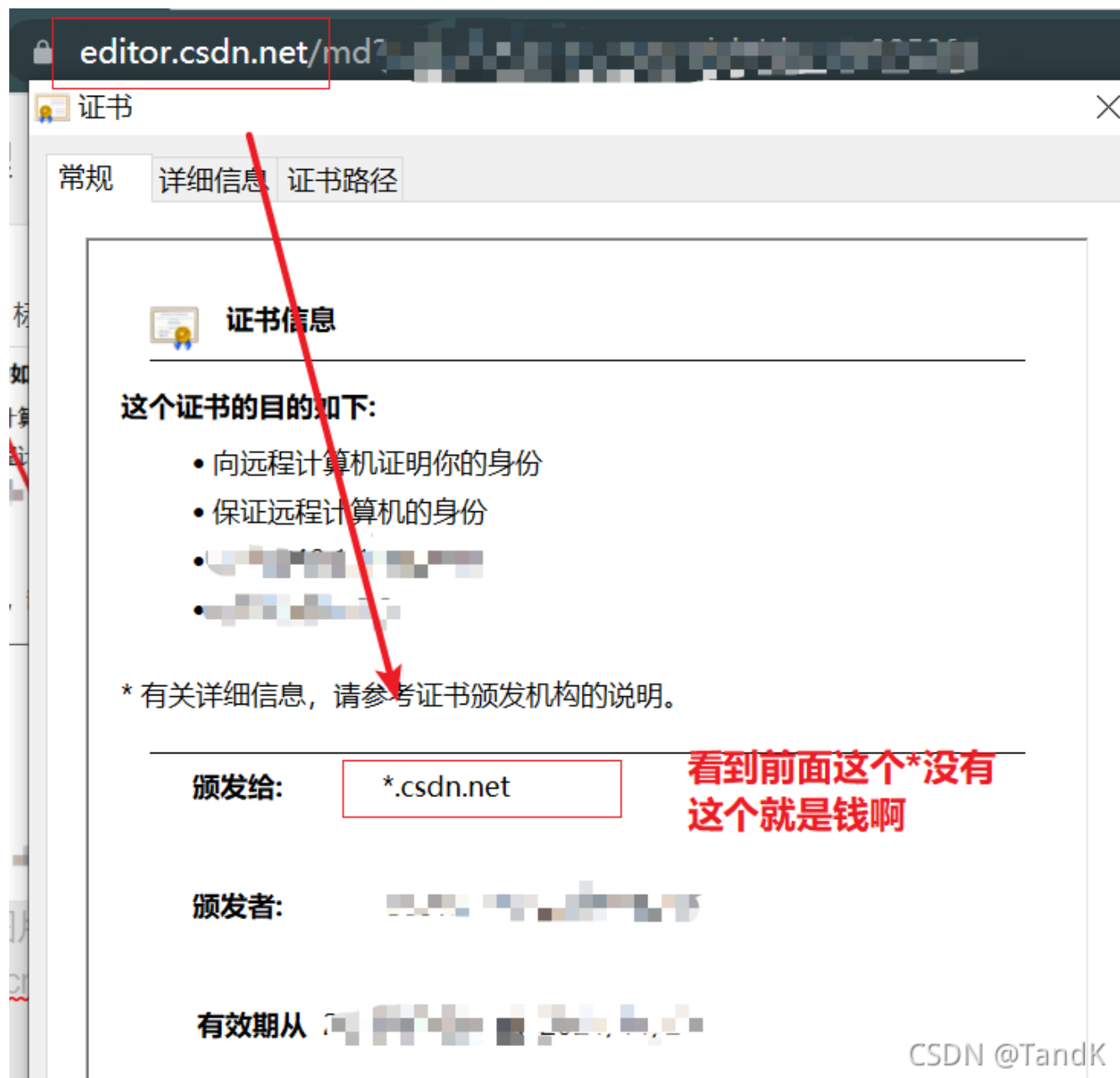
给你看看有钱人家和穷逼的区别：

我（穷逼），一个子域名一个证书：



CSDN @TandK

有钱人家，一个通配证书配所有子域名：



腾讯云对于免费的证书限制是这样的：

- 免费证书不能通配。就是上面说的，虽然同属于tandk.com这个主域名，但是我每多一个子域名，比如blog.tandk.com，就需要再去申请一个专属于blog.tandk.com的ssl证书。
- 一个主域名只能申请20张ssl证书。
- 证书吊销后，还会在15个月内，占用这20个证书的名额。

好家伙，这是逼着我们去买通配证书啊，不过想了想，免费证书对我这种自己玩一玩的其实够用了。

20个项目，我是用不了那么多的，而且要真要那么多，我也多买几个服务器和域名了，腾讯还是良心的。
(腾讯快给广告费！)

以上，参考自下面的内容：

[《docker安装nginx并配置https》](#)

[《docker上启动nginx,并配置修改nginx的配置文件》](#)

[《一台服务器通过nginx配置多个域名（80端口）》](#)

感谢各位大佬的无私教程！



 TandK
关注 - 1
粉丝 - 0

+加关注

3
推荐

0
反对

« 上一篇： RESTful API实践总结

posted @ 2021-10-23 22:54 TandK 阅读(246) 评论(4) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

【推荐】HeapDump性能社区2021性能领域问卷调研, 参与即可抽奖!

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载!

【推荐】并行超算云面向博客园粉丝推出“免费算力限时申领”特别活动

【推荐】华为全面助力青少年编程教育普及, 花瓣少儿编程正式上线



穿山甲

App开发者高效成长

📈 增长变现闭环
💰 收入提升 **28%**

[立即注册](#)

The advertisement features a dark red background with a subtle pattern of red circles and lines. A large, 3D white cube with a red arrow pointing upwards is positioned on the right side. The text is in white and red, with the percentage '28%' in a bold red font.

编辑推荐:

- 聊聊我在微软外服的工作经历及一些个人见解
- 死磕 NIO — Reactor 模式就一定意味着高性能吗?

- 消息队列那么多，为什么建议深入了解下RabbitMQ？
- 技术管理进阶——管人还是管事？
- 以终为始：如何让你的开发符合预期

最新新闻：

- 苹果iOS 15的新功能「同播共享」怎么玩？先用这款App来场「跳绳 PK」吧（2021-10-25 15:10）
 - 搞副业、玩滑板、做KOL，不加班的大厂员工“活”过来了（2021-10-25 15:00）
 - 何小鹏：如果只做没太大变化的车，那我创业干什么？（2021-10-25 14:46）
 - 全球供应链大混乱：船只短缺、码头拥堵，“卡”住汽车的可不止芯片（2021-10-25 14:35）
 - 想拿下中国市场？大众需要华为（2021-10-25 14:23）
- » 更多新闻...

Copyright © 2021 TandK

Powered by .NET 6 on Kubernetes