



链滴

# Https 就是这么简单之 Caddy2

作者: [yuanhenglizhen](#)

原文链接: <https://ld246.com/article/1578496435040>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

之前博客一直跑的是Caddy1.0版本，很轻松。就博客和小应用来说Caddy做web服务就够了，可以去很多事情，逼格高而且不失优雅。

之前写的一些关于caddy1.0的使用姿势

[基于 caddy 提升 https 安全等级](#)

[caddy 实现 google 镜像站点](#)

## 01. 起因

看同事在用阿里云的赛门铁克ssl证书，实在是看不下去了，就给同事安利了caddy。当时没多想，随丢了一个百度到使用文档，然后同事自由发挥了下，编译安装了caddy2，跑不起来，很纠结。当时为caddy只有1.0版本，2版本一直处于beta阶段，就没有用，结果翻车了。。。

于是被动的去学习了一波。。

## 02. Caddy简介

caddy的文档还是蛮全的，不过有些细节需要你自己的发掘，摆好姿势和造型才能跑的更流畅。

官方文档：<https://caddyserver.com/docs/getting-started>

github文档：<https://github.com/caddyserver/caddy/wiki>

example：<https://github.com/caddyserver/caddy/wiki/v2:-Caddyfile-examples>

caddy2是使用go语言编写的轻量级的应用

作为一个很好的选择：

- 网络服务器
- 反向代理
- 边车代理
- 负载均衡器
- API网关
- 入口控制器
- 系统管理员
- 流程主管
- 任务调度器
- (任何长时间运行的过程)

尽管它具有与其他层一起工作的能力，但主要在OSI模型的 L4（传输层）和L7（应用层）上运行。

Caddy2与Caddy1的主要区别就是，caddy使用api的方式，以上传json文件进行配置站点，同时也容之前的Caddyfile配置文件。

## 03. 快速搭建https站点

1. 下载二进制文件

```
wget https://github.com/caddyserver/caddy/releases/download/v2.0.0-beta12/caddy2_beta1_linux_amd64
```

2. 拷贝到bin下并赋予可执行权限

```
mv caddy /usr/bin/ && chmod 777 caddy
```

3. 运行caddy守护进程，在当前终端继续操作的话可以使用start

```
caddy run
```

```
[root@sun ~]# caddy run
2020/01/08 14:43:21.214 INFO using adjacent Caddyfile
2020/01/08 14:43:21.216 INFO admin admin endpoint started {"address": "localhost:2019", "enforce_origin": false, "origins": ["localhost:2019"]}
2020/01/08 14:43:21.216 INFO http server is only listening on the HTTPS port but has no TLS connection policies; adding one to enable TLS {"server_name": "srv0", "https_port": 443}
2020/01/08 14:43:21.216 INFO http enabling automatic TLS certificate management {"domains": ["opschina.top", "www.opschina.top"]}
2020/01/08 14:43:21.217 INFO http enabling automatic HTTP->HTTPS redirects {"domains": ["opschina.top", "www.opschina.top"]}
2020/01/08 14:43:21.217 INFO tls cleaned up storage units
2020/01/08 14:43:21.218 INFO autosaved config {"file": "/root/.config/caddy/autosave.json"}
2020/01/08 14:43:21.218 INFO serving initial configuration
2020/01/08 22:43:21 [INFO] [cache:0xc0006e4ff0] Started certificate maintenance routine
```

4. 打开另外一个终端，并查看当前配置。默认情况下配置是空的，因为我之前操作过，所有会有一些置内容

```
curl localhost:2019/config/
```

```
[root@sun ~]# curl localhost:2019/config/
{"apps":{"http":{"servers":{"srv0":{"listen":[":443"],"routes":[{"handle":[{"handler":"subroute","routes":[{"handle":[{"handler":"reverse_proxy","health_checks":{"active":{"expect_status":2,"path":"/health"},"transport":{"protocol":"http","read_buffer_size":4096},"upstreams":[{"dial":"127.0.0.1:8080"}]}]}]}]}]}]}]}]}
```

5. 创建一个配置文件

```
vim caddy.json
```

```
{
  "apps":{
    "http":{
      "servers":{
        "srv0":{
          "listen":[
            ":443"
          ],
          "routes":[
            {
              "handle":[
                {
                  "handler":"subroute",
                  "routes":[
                    {
                      "handle":[
                        {
                          "handler":"reverse_proxy", #处理方式为反向代理
                          "health_checks":{
                            "active":{
                              "expect_status":2,
                              "path":"/health" #健康检查的路径
                            }
                          }
                        },
                      "transport":{
                        "protocol":"http",

```

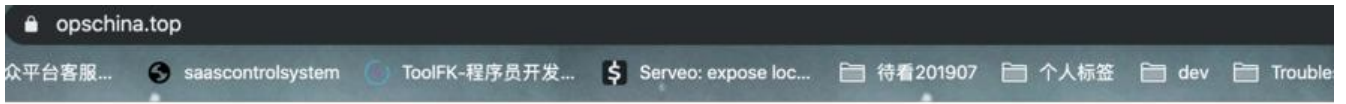
```
        "read_buffer_size":4096
      },
      "upstreams":[
        {
          "dial":"127.0.0.1:8080"
        }
      ]
    },
    ],
    "match":[
      {
        "path":[
          "/"
        ]
      }
    ]
  }
],
"match":[
  {
    "host":[
      "opchina.top",
      "www.opchina.top"
    ]
  }
]
}
]
}
}
}
}
}
```

## 6. 上传配置文件

```
curl localhost:2019/load -X POST -H "Content-Type: application/json" -d @caddy.json
```

## 7. 稍等片刻便可以发现生成好了

访问我们的网站[www.opchina.top](http://www.opchina.top)，这时候发现会自动跳转到https，搞定收工！



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## 04.相关参数

```
[root@sun ~]# caddy help  
Caddy is an extensible server platform.
```

```
usage:  
caddy <command> [<args...>]
```

```
commands:  
adapt          将配置文件转换成json  
environ       打印环境变量  
file-server   启动可付诸生产的文件服务器  
hash-password 用base64加密密码  
help          显示caddy的命令帮助  
list-modules  列出已安装的模块  
reload        热加载  
reverse-proxy 快速且可适用生产的反向代理  
run           启动Caddy进程并无限期阻止  
start         后台运行caddy  
stop         停止caddy  
validate      测试配置文件是否正确  
version       打印版本号
```

Use 'caddy help <command>' for more information about a command.

Full documentation is available at:  
<https://caddyserver.com/docs/command-line>

后续有时间，给大家分享下深入使用下Caddy2.

有个很深的感触，就是很多应用都可以使用api的方式来进行配置启动服务了