

## 使用OwnCloud搭建自己的网盘

最近在自己的服务器上建了一个OwnCloud的网盘。

服务器版本为Ubuntu Server

18.10，在创建OwnCloud服务时，因不想污染服务器的环境，用到了Docker，在Docker中安装OwnCloud。

整套系统需要3个Docker，分别用于安装OwnCloudServer、OnlyOffice Server和Nginx。

Nginx用作反向代理，当访问OwnCloud Server和OnlyOffice Server的域名时，分别指向各自对应的Docker。OwnCloud Server是网盘服务器，而OnlyOffice Server用来配合OwnCloud Server，在线显示文档，支持的格式包括word、excel、ppt、epub等。



### 1. 安装Docker

UbuntuServer自身不带Docker，需要安装。运行命令

```
sudo apt install docker.io
```

安装后运行一下命令检查是否安装成功。

```
docker -version
```

### 2. 安装ownCloudServer

1) 先在当前目录创建目录ownCloud。

```
mkdir ownCloud
```

2) 接着到docker的官网查找ownCloud Server的Docker。

<https://hub.docker.com/r/owncloud/server/>

运行如下命令下载docker到本地，我使用的版本是10.0。

```
color:#24292E;background:#F6F8FA">sudo docker pull owncloud/server:10.0
```

3) 然后运行OwnClouder Server的docker。

```
color:#24292E;background:#F6F8FA">sudo docker run -d --name ownCloud  
-p 8081:8080 --rm owncloud/server:10.0
```

ownCloud默认的端口是8080，把它映射成宿主机（即Ubuntu服务器）的端口8081，使用参数 -p

8081:8080，首次运行时，我的目的只是把它的目录 /mnt/data/中的文件复制到宿主机，所以使用了--rm参数，意味着一旦停止运行ownCloud，它会自动销毁。

浏览器输入http://<ip>:8081，可以打开网页，证明ownCloud正常运行。

4) 之后把 /mnt/data/中的文件复制到宿主机的 ~/ownCloud 目录，停止容器ownCloud。

```
cd ownCloud
```

```
sudo docker cp ownCloud:/mnt/data/ ./
```

```
sudo docker stop ownCloud
```

此时可看到目录 ~/ownCloud/data（在我的环境中，绝对路径是 /home/ubuntu/ownCloud/data）中的有若干文件夹，即ownCloud的个人数据。

5) 重新启动 ownCloud。

```
sudo docker run -d --name ownCloud -p 8081:8080
```

```
-v/home/ubuntu/ownCloud/data:/mnt/data owncloud/server:10.0
```

这次运行时，参数稍有不同，去掉了 --rm，这样以后把 ownCloud停止了，可以再次start，数据不会丢失。加上了参数-v

/home/ubuntu/ownCloud/data:/mnt/data，把宿主机的目录/home/ubuntu/ownCloud/data映射到宿主机的目录 /mnt/data。

6) 浏览器中登录ownCloud，修改相关设置。

浏览器中打开 http://<ip>:8081，输入默认用户名/密码：admin/admin

登录，修改密码、语言。

进入Market，安装Apps。安装Text Editor、PDF

Viewer、Gallery、Onlyoffice，此时可以查看txt文件、图片、PDF，但word、ppt、excel无法查看，即OnlyOffice没法使用。

Google之后，原来还需要安装OnlyOffice服务器。

### 3. 安装OnlyOffice Server

1)

```
font-stretch: normal"> 到 /apps目录中查看是否已经安装onlyOffice 插件。
```

在我的环境中已经安装，如果没有安装，可参见文档<https://github.com/ONLYOFFICE/onlyoffice-owncloud>。

打开页面`http://<ip>/settings/admin?sectionid=additional`，这是OnlyOffice的设置，我们先不设置。

□

2) 下载OnlyOffice Server的docker镜像，运行它。

```
sudo docker run -i -t -d --name onlyofficeServer -p 8082:80onlyoffice/documentserver
```

打开浏览器 `http://<ip>:8082`，验证服务器已经在运行。

设置 `http://<documentServer>`，为 `http://<ip>:8082`，保存。

□

浏览器中打开 `http://<ip>:8081`登录，点击odt、epub文件，可以在线查看。新建word文档，可以编辑。

说明OnlyOffice已经安装成功，此时ownCloud已经可以运行了。不过有点美中不足，只能IP访问。稍后的步骤把域名绑定到相应IP，通过nginx反向代理访问OwnCloud。

#### 4. 配置域名解析和Nginx

首先要有个域名，在域名解析里面，配置记录类型A，把域名指向<ip>。

域名解析成功之后，等一段时间，就可以通过`http://<domain>:8081`，可以访问OwnCloud了。为了去掉端口（即输入`http://<domain>`）也能访问，我们需要做些配置。

1) 配置Nginx。

```
sudo docker run -d -p 80:80 --rm --name mynginx nginx
```

创建目录 `nginx_files/conf`，

然后把nginx中的目录复制到里面（别忘了末尾的 `.`，它代表当前目录）。

```
sudo docker container cp mynginx:/etc/nginx .
```

然后停止它。

```
sudo docker container stop mynginx
```

再次运行。

```
sudo docker run -d -p 80:80 -p 443:443
```

```
-v/home/ubuntu/nginx_files/conf:/etc/nginx --name mynginx nginx
```

2)

`font-stretch: normal">` 修改nginx.conf文件、复制证书，重新加载nginx。

修改的步骤可以参见文档Nginx 容器教程(<http://www.ruanyifeng.com/blog/2018/02/nginx-docker.html>)，不再赘述。

```
service nginx reload
```

OK，此时http、https均工作。

□

3) http强制跳转到https。

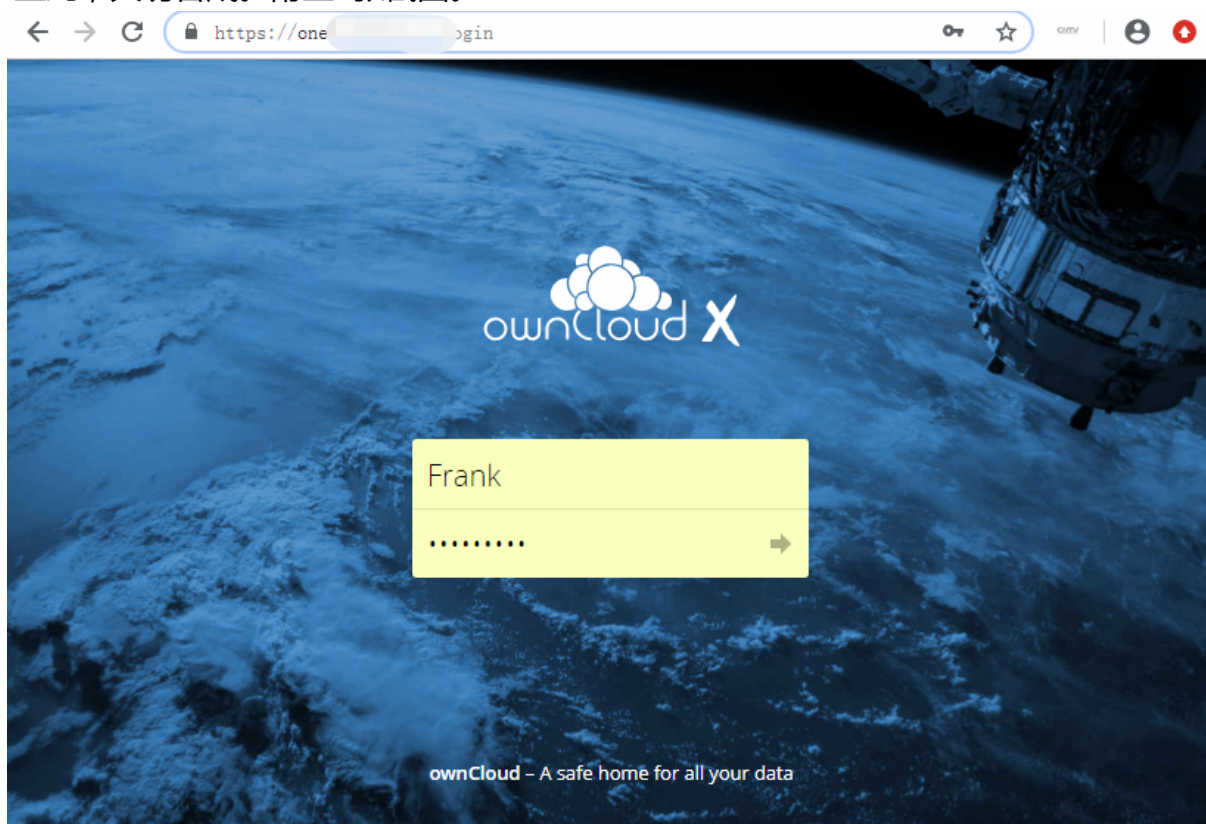
可以参见这篇文档 nginx强制使用https访问 ( [https://blog.csdn.net/wzy\\_1988/article/details/8549290](https://blog.csdn.net/wzy_1988/article/details/8549290) ) 。

□

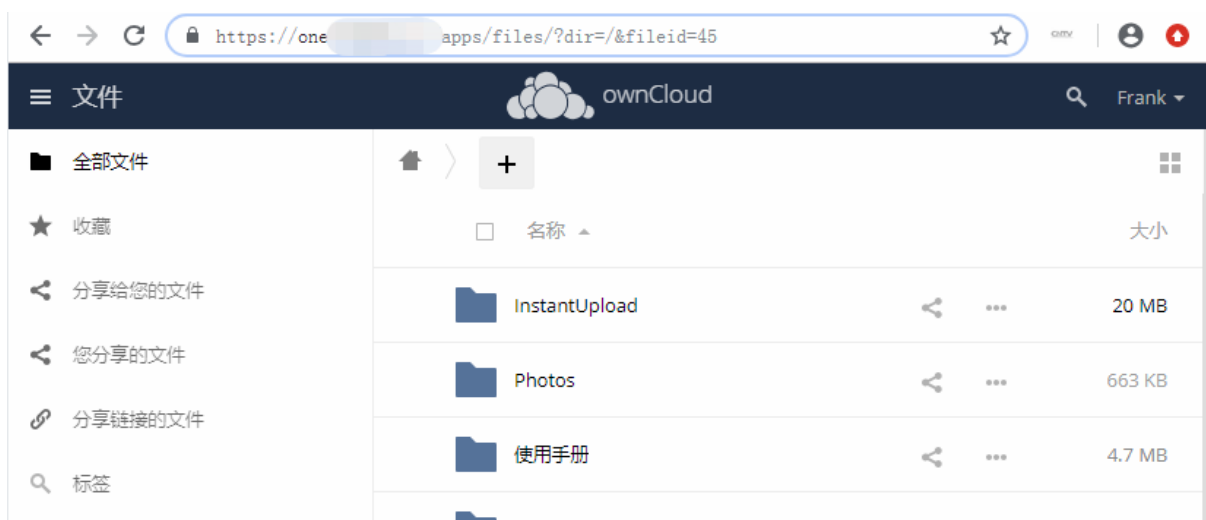
稍加说明：在ownCloud强制使用https之后，在步骤3.2中的OnlyOffice Server也应强制改成https访问，否则仍旧无法使用。

□

至此，大功告成。附上2张截图。



## 登录界面



ownCloud主界面

## 参考文档

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC"><https://hub.docker.com/r/owncloud/server/>

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC"><https://github.com/owncloud-docker/server>

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC"><https://github.com/ONLYOFFICE/onlyoffice-owncloud>

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC"><https://github.com/ONLYOFFICE/Docker-DocumentServer>

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC"><http://www.ruanyifeng.com/blog/2018/02/nginx-docker.html>

10.0pt;font-family:Symbol;color:black">·  
font-family:"Arial","sans-serif";color:#1155CC">[https://blog.csdn.net/wzy\\_1988/article/details/8549290](https://blog.csdn.net/wzy_1988/article/details/8549290)

## 相关文章

10.0pt;font-family:Symbol;color:black">·  
font-family:"Helvetica","sans-serif";color:black">  
使用Nginx实现服务器中多容器共存

10.0pt;font-family:Symbol;color:black">·  
font-family:"Helvetica","sans-serif";color:#576B95">怎样让小程序服务器免备案

10.0pt;font-family:Symbol;color:black">·  
利用CSS3的animation实现闪亮的效果