

Docker 部署 fdfs 的镜像

1、环境说明

要使用 docker 容器技术，必须在 centos 系统上安装 docker。Docker 目前的版本是 1.17.3 版本。但由于 docker 从 1.8 版本开始，只能在 centos7 以上的系统上使用，并且内核要求大于 3.10.0。在 centos6.5 上只能安装到 docker 的 1.7.1 版本，根据我们实际情况也只能安装 1.7.1 版本。具体可以参考官网：
<https://docs.docker.com/v1.7/docker/installation/centos/>，
下面是官网的一个截图：

1	21	12	12
12			
2			
12			
12			
12			

Testtesttesttest

12	12	12	12
12	12	12	12
12	12	12	
12	12	12	12

Install

You use the same installation procedure for all versions of CentOS, only the package you install differs. There are two packages to choose from:

Version	Package name
6.5 and higher	https://get.docker.com/rpm/1.7.0/centos-6/RPMS/x86_64/docker-engine-1.7.0-1.el6.x86_64.rpm https://get.docker.com/rpm/1.7.0/centos-6/SRPMS/docker-engine-1.7.0-1.el6.src.rpm
7.X	https://get.docker.com/rpm/1.7.0/centos-7/RPMS/x86_64/docker-engine-1.7.0-1.el7.centos.x86_64.rpm https://get.docker.com/rpm/1.7.0/centos-7/SRPMS/docker-engine-1.7.0-1.el7.centos.src.rpm

This procedure depicts an installation on version 6.5. If you are installing on 7.X, substitute that package for your installation.

所以鉴于上述原因，我们选择 **docker** 的 **1.7.1** 版本安装。

(1) Docker 的 yum 方式在线安装

1) 第一步安装:

直接执行: `yum install -y docker-io`

如果提示说找不到 `docker-io` 的安装包, 则再执行:

```
yum -y install
```

```
http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

2) 第二步启动:

启动执行: `service docker start`

3) 第三步测试安装成功

直接执行: `docker -v`

```
[root@localhost fdfs3]# docker -v
Docker version 1.7.1, build 786b29d/1.7.1
[root@localhost fdfs3]#
```

看到输出版本说明 `docker` 安装成功!

4) 第四步 `docker` 加入开启启动

直接执行: `chkconfig docker on`

2、配置 **fdfs**

提示：以下的操作需要熟练 **docker** 的基本命令的操作

在本文件夹下看到的 **Fdfs.tar** 已经是一个能够安装和配置好的 **docker** 容器的归档文件，我们把它导入到我们的 **centos** 系统，然后更改一下主机 **ip** 的配置即可使用。

第一步：开放端口

开放 **fdfs** 这个 **docker** 这个容器需要用到的端口，在这里是 **22122, 23000, 8088**。执行 **shell** 命令开放端口：

```
/sbin/iptables -I INPUT -p tcp --dport 8088 -j  
ACCEPT
```

```
/sbin/iptables -I INPUT -p tcp --dport 22122 -j  
ACCEPT
```

```
/sbin/iptables -I INPUT -p tcp --dport 23000 -j  
ACCEPT
```

```
/etc/rc.d/init.d/iptables save #保存配置
```

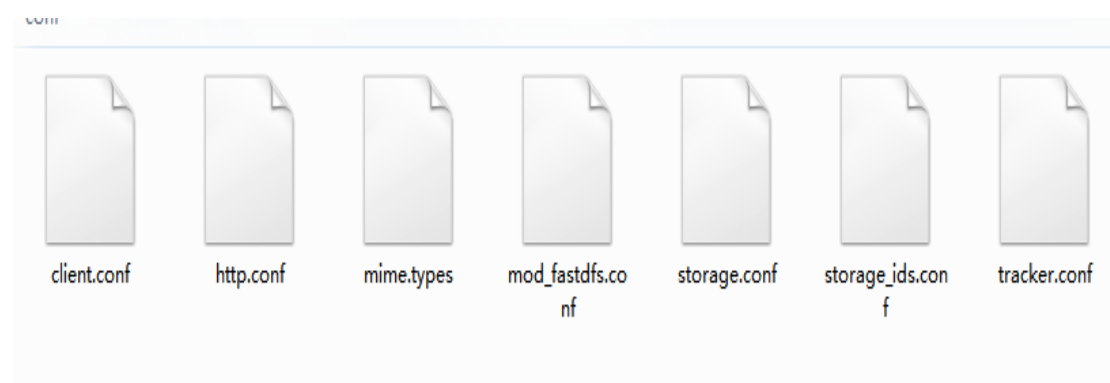
```
/etc/rc.d/init.d/iptables restart #重启防火墙服务
```

第二步：配置

建立必要的文件夹。在本文件夹下有一个 **fdfs** 文件，在 **fdfs** 中有 **conf**，**basedata**，**tracker**，**storedata** 四个文件夹



我们只需要配置 **conf** 下的文件即可，**conf** 文件夹里面文件如下：



我们只需要配置 **storage.conf**，**mod_fastdfs.conf**，**client.conf** 这三个文件即可

(1) 配置 **storage.conf**

只需要配置一下三项:

tracker_server=

```
tracker_server=192.168.1.67:22122
```

主机ip+端口 22122

主机 ip+端口 22122

以上三项改好即可

(2) 配置 **mod_fasdfs.conf**

只需要配置一下几项:

tracker_server=

```
40  
41 tracker_server=192.168.1.67:22122  
42
```

更改对应的 ip 即可.

(3) 配置 **client.conf**

只需配置

tracker_server=

```
14  
15 tracker_server=192.168.1.67:22122  
16
```

更改对应 ip 即可。

第三步：移动 fdfs 整个文件夹至 Resource 下

比如：

```
mv fdfs /home/www/vpr/Resource/
```

在这里注意是要把 fdfs 这个文件夹移动过去，最终是

/home/www/vpr/Resource/fdfs

然后配置一下权限，进入到 resource 目录。

```
chmod -R 777 fdfs
```

至此，整个 fdfs 的容器已经配置完毕。

下面将开始 docker 的操作。

3、Docker 操作

这一步需要学习一下 docker 的基本操作命令

第一步：导入 fdfs.tar

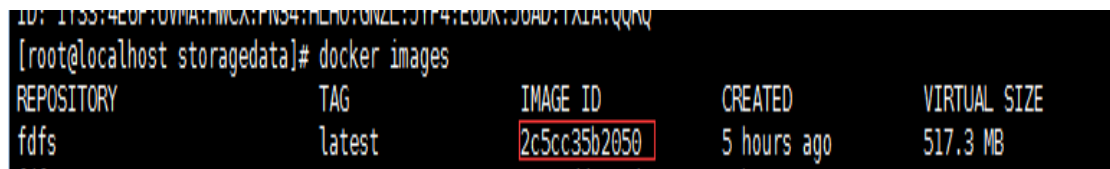
在本文件夹下面有一个 fdfs.tar 的 docker 容器对象，我们需要将它导入

(1) 进入到本文件，在终端执行：

```
docker load -i fdfs.tar
```

(2) 导入完成后，查看镜像

```
docker images
```



REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
fdfs	latest	2c5cc35b2050	5 hours ago	517.3 MB

可以看到这里列出一些信息

(3) 创建一个容器（最重要）

在本文件夹下面有一个 **start.sh**，这个文件创建 **fdfs** 容器的创建脚本

里面有四个参数需要我们配置：



```
1 #!/bin/sh
2 #定义变量,下面的变量只是例子,最终根据项目设置,每一个路径后必须有"
3 tracker="/home/vpr/www/Resource/fdfs/tracker/"
4 conf="/home/vpr/www/Resource/fdfs/conf/"
5 storagedata="/home/vpr/www/Resource/fdfs/storagedata/"
6 basedata="/home/vpr/www/Resource/fdfs/basedata/"
7 #docker run |
8 docker run --name=testfdfs -v ${tracker}:/opt/fdfs/tracker/ -v ${co
9
```

我们只需要每个配置红色部分改为我们项目实际对应的目录即可实现，路径的最后就是我们移动过来的 **fdfs** 文件夹

(4) 查看 **docker** 容器是否启动成功

`docker ps`

启动的容器都会在这里显示，例如

```
[root@localhost storagedata]# docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED
c70f8cc5d956	fdfs	"/start.sh"	5 hours ago

到这里说明容器已经启动了，整个 **fdfs** 的容器已经完毕。

测试一下：可以在浏览器：<http://ip:8088/>

将会出现如下界面说明成功启动了



welcome to fdfs

you successful start the fdfs container!

(5) 停止这个 **fdfs** 容器

直接执行

`docker stop c70f8cc5d956`

c70f8cc5d956 是这个容器的 id

那么如何再次启动它呢？

`docker start c70f8cc5d956`

这样子 **fdfs** 容器就可以启动了

再次提示：请学习一下 **docker** 的基本操作，不然出问题不知道如何解决的！

(4) 安装 **php** 的 **fastdfs** 的驱

第一步：安装驱动

在本文件夹下可以看到一个 **php_client** 的文件夹，这个是 **php** 的 **fastdfs** 的驱动安装包，首先需要编译安装

(1) 进入到 **php_client**

```
cd php_client
```

(2) 执行一下命令

```
phpize
```

```
./configure
```

```
make
```

```
make install
```

(3) 配置 **php.ini**

找到 **php_client** 文件夹下的 **fastdfs_client.ini** 这个配置文件，只需要修改其中一项，关联到客户端的操作配置文件：

Testtesttesttest

```
40  
41 fastdfs_client.tracker_group0 = /home/www/vpr/Resource/fdfs/client.conf  
42
```

将红色部分改为项目实际的资源的路径

改完之后，将 **fastdfs_client.ini** 这个文件的全部内容复制（追加到） **php.ini** 这个文件后面，当然也可以先复制这个文件内容到 **php.ini** 后面，再修改上面的配置值。

然后重新启动 **httpd**，通过命令 **php -m** 查看是否安装成功这个驱动

```
[hao@vpr ~]$ php -m  
[PHP Modules]  
bz2  
calendar  
Core  
ctype  
curl  
date  
dom  
ereg  
exif  
fastdfs_client  
fileinfo  
filter  
ftp  
gd  
gettext  
gmp  
hash  
iconv  
intl  
json  
libxml  
mbstring  
mcrypt  
mhash  
mongo  
mysql  
mysqli  
openssl
```

看到这个红色部分代表驱动安装成功，然后就可以在 **php** 使用了！

Testtesttesttest