

文件状态：正式发布

保密级别：内部公开

MCP 产品详细部署文档

广东以大供应链管理有限公司

编写人：罗荣熙

2019.09.09

变更记录

序号	版本号	内容	修改人	审批人	发布日期
1	V1.0	创建	罗荣熙		

目录

变更记录	2
1. 引言	6
1.1. 编写目的	6
1.2. 历史背景	6
1.3. 参考资料	6
2. 产品部署	7
2.1. 部署环境	7
2.1.1. 服务器	7
2.1.2. Iptables 安装	7
2.1.2.1. 停止 firewall	8
2.1.2.2 禁止 firewall 开机自启	8
2.1.2.2. 安装 iptables 防火墙	8
2.1.3. JDK 安装	9
2.1.3.1. 查看系统是否已安装了 JDK	9
2.1.3.2. 卸载已安装的 JDK	9
2.1.3.3. 安装	9
2.1.3.4. 配置环境变量，切换到 root 用户，执行命令	9
2.1.3.5. 配置完后使配置立即生效命令	10
2.1.3.6. 检查是否安装成功，查看 java 的版本信息命令	10
2.1.4. MariaDB 安装	10
2.1.4.1. 查看是否安装了 MariaDB	10

2.1.4.2. 删除已安装的 MariaDB	10
2.1.4.3. 查看是否已经安装了 mysql 数据库	10
2.1.4.4. 删除已安装的 Mysql	11
2.1.4.5. 使用 yum 命令安装 MariaDB	11
2.1.4.6. 启动 MariaDB 数据库	11
2.1.4.7. 设置 MariaDB 开机启动	11
2.1.4.8. 设置 MariaDB 的基本配置	11
2.1.4.9. 登录数据库	12
2.1.4.10. 配置 MariaDB 的字符集	12
2.1.4.11. MySQL 授权远程访问（先用 root 登录 mysql）	14
2.1.4.12. 为了方便远程管理，防火墙中打开 3306 端口	14
2.1.5. Nginx 安装	15
2.1.5.1. 安装 nginx 依赖包，执行命令	15
2.1.5.2. 下载 nginx 依赖包，执行命令	15
2.1.5.3. 安装 nginx	15
2.1.5.4. 查看 nginx 版本号	16
2.1.5.5. 开启 80 端口	16
2.1.5.6. 配置 nginx，执行命令	16
2.1.5.7. 启动 nginx	17
2.1.5.8. 重启 nginx	17
2.1.6. FastDFS 安装	17
2.1.6.1. 所有跟踪服务器和存储服务器均执行如下操作	17

2.1.6.2.	配置 FastDFS 跟踪器(192.168.4.121)	22
2.1.6.3.	配置 FastDFS 存储(192.168.4.125)	24
2.1.6.4.	文件上传测试	26
2.1.6.5.	在每个存储节点上安装 nginx	27
2.2.	数据 SQL 脚本	32
2.2.1.	数据库 PDM 设计	32
2.2.2.	系统管理 SQL 脚本	33
2.2.3.	机构管理 SQL 脚本	33
2.2.4.	统一编码 SQL 脚本	33
2.2.5.	商品管理 SQL 脚本	33
2.2.6.	采购供货 SQL 脚本	33
2.2.7.	证照管理 SQL 脚本	34
2.2.8.	退货管理 SQL 脚本	34
2.2.9.	骨科跟台 SQL 脚本	34
2.2.10.	视图信息 SQL 脚本	34
2.3.	项目打包命令	34
2.3.1.	前端打包命令	34
2.3.2.	后端打包命令	35
2.4.	项目运行命令	36
2.4.1.	前端开发环境部署（使用普通用户进行操作）	36
2.4.2.	后端开发环境部署（使用普通用户进行操作）	36
2.4.3.	Nginx 配置（使用 root 用户进行操作）	36

1.引言

1.1. 编写目的

本文档的目的是为 MCP 产品部署提供完整的部署文档,包括部署环境、数据库 SQL 脚本、项目打包命令、项目运行命令等。本文档的预期读者为软件开发人员,项目实施人员和系统运维人员。

1.2. 历史背景

目前部署的资料有 SQL 脚本、打包命令、部署资料等,但是比较分散,缺乏一个完整的部署文档。为了方便软件开发人员,项目实施人员和系统运维人员快速实现部署,在此提供一份完成的 MCP 产品部署文档,在部署过程中起到指导性意义。

1.3. 参考资料

资料名称	作者	文件编号、版本	资料存放地址
库表设计	罗荣熙	V2.0	http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/model
初始化脚本	罗荣熙	V2.0	http://192.168.1.12:9005/svn/ebig

			gsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/ 库表设计/sql
平台部署	罗荣熙	V2.0	http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/ 平台部署

2. 产品部署

2.1. 部署环境

2.1.1. 服务器

部署服务器选择 CentOS7 以上版本，CPU：Inter2.4GHz 2 核以上，内存：2G 以上，硬盘：100G 以上。

2.1.2. Iptables 安装

CentOS7 默认的防火墙不是 iptables，而是 firewalle，使用 root 用户操作，如果已安装这个步骤可忽略。

2.1.2.1. 停止 firewall

```
[root@localhost ~]# systemctl stop firewalld.service
```

2.1.2.2 禁止 firewall 开机自启

```
[root@localhost ~]# systemctl disable firewalld.service
```

2.1.2.2. 安装 iptables 防火墙

(1) 先检查是否安装了 iptables :

```
[root@localhost ~]# service iptables status
```

(2) 安装 iptables :

```
[root@localhost ~]# yum install -y iptables
```

(3) 升级 iptables :

```
[root@localhost ~]# yum update iptables
```

(4) 安装 iptables-services :

```
[root@localhost ~]# yum install iptables-services
```

(5) 编辑防火墙文件 :

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

(6) 保存退出 :

```
[root@localhost ~]# x
```

(7) 重启防火墙使配置文件生效:

```
[root@localhost ~]# systemctl restart iptables.service
```

(8) 设置 iptables 防火墙为开机自启:


```
[root@localhost ~]# systemctl enable iptables.service
```

2.1.3. JDK 安装

安装 1.8 以上版本的 JDK

2.1.3.1. 查看系统是否已安装了 JDK

```
[root@localhost ~]# rpm -qa | grep jdk
```

2.1.3.2. 卸载已安装的 JDK

```
[root@localhost ~]# yum -y remove java java-1.7.0-openjdk-1.7.0.0-1.49.1.11.4.el6_3.x86_64
```

2.1.3.3. 安装

下载 jdk-8u171-linux-x64.tar.gz

解压命令：

```
[root@localhost ~]# tar -zxvf jdk-8u171-linux-x64.tar.gz -C /opt/java
```

2.1.3.4. 配置环境变量，撤换到 root 用户，执行命令

```
[root@localhost ~]# vi /etc/profile
```

添加以下配置：

```
# java env
```

```
export JAVA_HOME=/opt/java/jdk1.8.1_71
```

```
export JRE_HOME=$JAVA_HOME/jre
```

```
export
```

```
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib/rt.
```

```
jar
```

```
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

2.1.3.5. 配置完后使配置立即生效命令

```
[root@localhost ~]# source /etc/profile
```

2.1.3.6. 检查是否安装成功，查看 java 的版本信息命令

```
[root@localhost ~]# java -version
```

2.1.4. MariaDB 安装

安装 5.7 以上版本的 MySQL，MySQL 已经不再包含在 CentOS7 的源中，而是改用了

MariaDB，使用 root 用户执行以下命令：

2.1.4.1. 查看是否安装了 MariaDB

```
[root@localhost ~]# rpm -qa | grep MariaDB
```

2.1.4.2. 删除已安装的 MariaDB

```
[root@localhost ~]# rpm -e --nodeps MariaDB-*
```

2.1.4.3. 查看是否已经安装了 mysql 数据库

```
[root@localhost ~]# rpm -qa | grep mysql
```

2.1.4.4. 删除已安装的 Mysql

```
[root@localhost ~]#yum remove mysql mysql-server mysql-libs compat-  
mysql51
```

2.1.4.5. 使用 yum 命令安装 MariaDB

```
[root@localhost ~]# yum -y install mariadb mariadb-server
```

2.1.4.6. 启动 MariaDB 数据库

```
[root@localhost ~]# systemctl start mariadb
```

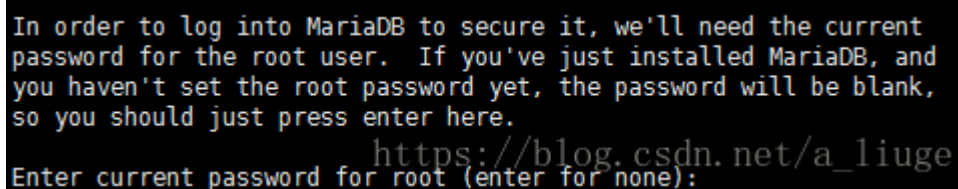
2.1.4.7. 设置 MariaDB 开机启动

```
[root@localhost ~]# systemctl enable mariadb
```

2.1.4.8. 设置 MariaDB 的基本配置

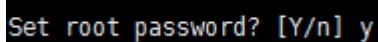
```
[root@localhost ~]# mysql_secure_installation
```

(1) 如图直接回车即可因为还没有设置初始密码 :



```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.  
https://blog.csdn.net/a_liuge  
Enter current password for root (enter for none):
```

(2) 设置 root 的密码 :



```
Set root password? [Y/n] y
```

(3) 删除匿名用户登录-根据你自己的需求选择 , 在此我选择删除匿名用户 :

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] y https://blog.csdn.net/a\_liuge
```

(5) 是否禁止远程登录-根据你自己的需求选择 :

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] n https://blog.csdn.net/a\_liuge
```

(6) 删除测试数据 :

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] y https://blog.csdn.net/a\_liuge
```

(7) 现在重新加载特权表 :

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

```
Reload privilege tables now? [Y/n] y https://blog.csdn.net/a\_liuge
```

2.1.4.9. 登录数据库

```
[root@localhost ~]# mysql -u root -p
```

2.1.4.10. 配置 MariaDB 的字符集

(1) 设置 my.cnf 字符集

```
[root@localhost ~]# vi /etc/my.cnf
```

在 my.cnf 文件中[mysqld_safe]下添加 :

```
init_connect='SET collation_connection = utf8_unicode_ci'
```

```
init_connect='SET NAMES utf8'
```

character-set-server=utf8

collation-server=utf8_unicode_ci

skip-character-set-client-handshake

如图：

```
[mysqld_safe]
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
skip-character-set-client-handshake
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
```

保存退出：

[root@localhost ~]# x

(2) 设置 client.cnf 字符集

[root@localhost ~]# vi /etc/my.cnf.d/client.cnf

在 client.cnf 文件中[client]下添加：

default-character-set=utf8

如图：

```
[client]
default-character-set=utf8
# This group is not read by mysql client library.
# If you use the same .cnf file for MySQL and MariaDB,
# use it for MariaDB-only client options
[client-mariadb]
```

保存退出：

[root@localhost ~]# x

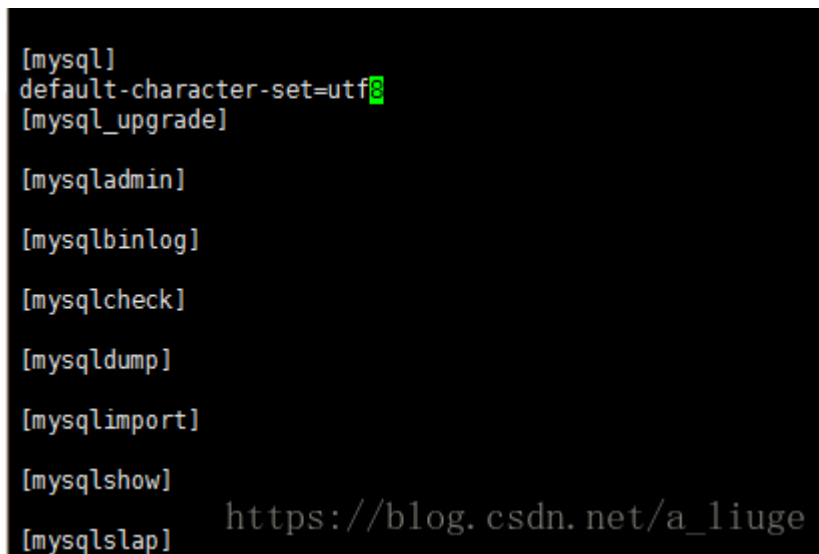
(3) 设置 mysql-clients.cnf 字符集

[root@localhost ~]# vi /etc/my.cnf.d/mysql-clients.cnf

在 mysql-clients.cnf 文件中[mysql]下添加：

```
default-character-set=utf8
```

如图：

A terminal window with a black background and white text. It lists various MySQL command-line options: [mysql], default-character-set=utf8, [mysql_upgrade], [mysqladmin], [mysqlbinlog], [mysqlcheck], [mysqldump], [mysqlimport], [mysqlshow], and [mysqlslap]. A URL https://blog.csdn.net/a_liuge is visible in the background.

```
[mysql]
default-character-set=utf8
[mysql_upgrade]

[mysqladmin]

[mysqlbinlog]

[mysqlcheck]

[mysqldump]

[mysqlimport]

[mysqlshow]

[mysqlslap] https://blog.csdn.net/a_liuge
```

(4) 重启 mariadb 服务：

```
[root@localhost ~]# systemctl restart mariadb
```

2.1.4.11. MySQL 授权远程访问 (先用 root 登录 mysql)

```
[root@localhost ~]# mysql -u root -p
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '654321'
```

```
WITH GRANT OPTION;
```

```
mysql> FLUSH PRIVILEGES
```

2.1.4.12. 为了方便远程管理，防火墙中打开 3306 端口

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

添加以下内容：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
```

重启防火墙：

```
[root@localhost ~]# systemctl restart iptables.service
```

2.1.5. Nginx 安装

安装 1.13 以上版本的 Nginx，使用 root 用户执行以下命令：

2.1.5.1. 安装 nginx 依赖包，执行命令

```
[root@localhost ~]# yum install gcc-c++ pcre pcre-devel zlib zlib-devel openssl  
openssl--devel
```

2.1.5.2. 下载 nginx 依赖包，执行命令

```
[root@localhost ~]# cd /opt/  
  
[root@localhost ~]# mkdir tools  
  
[root@localhost ~]# wget http://nginx.org/download/nginx-1.13.12.tar.gz
```

2.1.5.3. 安装 nginx

执行解压命令：

```
[root@localhost ~]# tar -zxvf nginx-1.13.12.tar.gz  
  
[root@localhost ~]# mv nginx-1.13.12.tar.gz ./tools/  
  
[root@localhost ~]# cd nginx-1.13.12/  
  
指定安装在/opt/nginx 目录下：  
  
[root@localhost ~]# ./configure --prefix=/opt/nginx  
  
[root@localhost ~]# make && make install
```

(注意 , 请不要删除安装包 , 安装 FastDFS 时 , 需要与 Nginx 集成。)

2.1.5.4. 查看 nginx 版本号

```
[root@localhost ~]# cd /opt/nginx/sbin/
```

```
[root@localhost ~]# ./nginx -h
```

2.1.5.5. 开启 80 端口

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

添加以下配置：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

重启防火墙，使端口配置生效，执行命令：

```
[root@localhost ~]# systemctl restart iptables.service
```

2.1.5.6. 配置 nginx , 执行命令

```
[root@localhost ~]# vi /opt/nginx/conf/nginx.conf
```

添加以下配置：

```
#禁止 IP 访问
```

```
server {
```

```
    listen 80 default;
```

```
    server_name _;
```

```
    server_name www.mcp.com mcp.cn
```

```
    return 500;
```



```
}
```

2.1.5.7. 启动 nginx

```
[root@localhost ~]# ./opt/nginx/sbin/nginx
```

2.1.5.8. 重启 nginx

```
[root@localhost ~]# ./opt/nginx/sbin/nginx -s reload
```

2.1.6. FastDFS 安装

FastDFS 分布式文件系统，**tracker 跟踪服务(192.168.4.12 21 1)**，**storage 存储服务(192.168.4.12 255)**，安装使用 root 用户操作，数量目录：/fastdfs，安装包：

FastDFS_v5.05.tar.gz

libfastcommon-master.zip

fastdfs-nginx-module_v1.16.tar.gz

fastdfs_client_java_v1.25.tar.gz

fastdfs_client_v1.24.jar

nginx-1.13.12.tar.gz

2.1.6.1. 所有跟踪服务器和存储服务器均执行如下操作

2.1.6.1.1. 编译和安装所需的依赖包

```
[root@localhost ~]# yum install make cmake gcc gcc-c++
```

2.1.6.1.2. 安装 libfastcommon

上传或下载 libfastcommon-master.zip 到/usr/local/src 目录

解压：

```
[root@localhost ~]# cd /usr/local/src/
```

```
[root@localhost ~]# unzip libfastcommon-master.zip
```

```
[root@localhost ~]# cd libfastcommon-master
```

```
[root@edu-dfs-tracker-01 libfastcommon-master]# ll
total 28
-rw-r--r--. 1 root root 2913 Feb 27 17:27 HISTORY
-rw-r--r--. 1 root root 582 Feb 27 17:27 INSTALL
-rw-r--r--. 1 root root 1342 Feb 27 17:27 libfastcommon.spec
-rwxr-xr-x. 1 root root 2151 Feb 27 17:27 make.sh
drwxr-xr-x. 2 root root 4096 Feb 27 17:27 php-fastcommon
-rw-r--r--. 1 root root 617 Feb 27 17:27 README
drwxr-xr-x. 2 root root 4096 Feb 27 17:27 src
```

2.1.6.1.3. 编译安装

```
[root@localhost ~]# ./make.sh
```

```
[root@localhost ~]# ./make.sh install
```

libfastcommon 默认安装到了

/usr/lib64/libfastcommon.so

/usr/lib64/libdfsclient.so

2.1.6.1.4. 因为 FastDFS 主程序设置的 lib 目录是 /usr/local/lib，所以需要创建软链接

```
[root@localhost ~]# ln -s /usr/lib64/libfastcommon.so
```

```
/usr/local/lib/libfastcommon.so
```

```
[root@localhost ~]# ln -s /usr/lib64/libfastcommon.so
```

```
/usr/lib/libfastcommon.so
```

```
[root@localhost ~]# ln -s /usr/lib64/libbdfsclient.so
```

```
/usr/local/lib/libbdfsclient.so
```

```
[root@localhost ~]# ln -s /usr/lib64/libbdfsclient.so /usr/lib/libbdfsclient.so
```

2.1.6.1.5. 安装 FastDFS

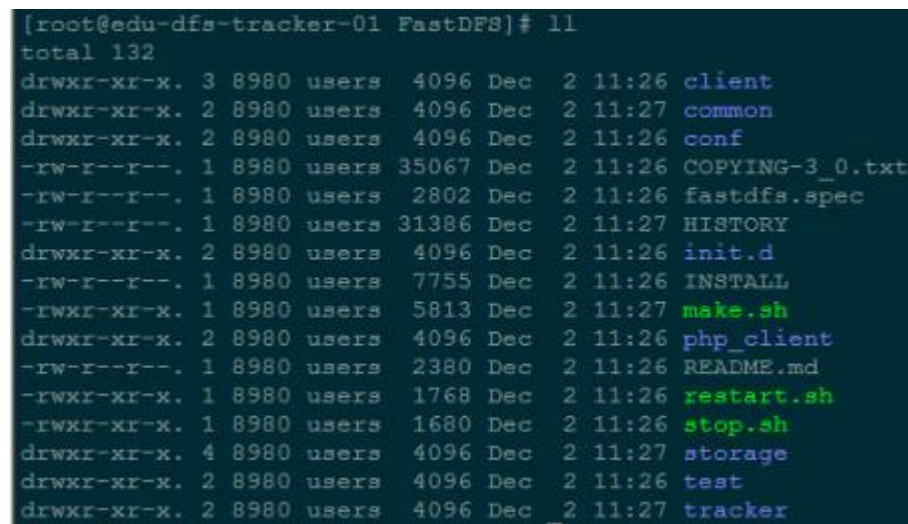
上传或下载 FastDFS 源码包 (FastDFS_v5.05.tar.gz) 到 /usr/local/src 目录

解压：

```
[root@localhost ~]# cd /usr/local/src/
```

```
[root@localhost ~]# tar -zxvf FastDFS_v5.05.tar.gz
```

```
[root@localhost ~]# cd FastDFS
```



```
[root@edu-dfs-tracker-01 FastDFS]# ll
total 132
drwxr-xr-x. 3 8980 users 4096 Dec  2 11:26 client
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:27 common
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:26 conf
-rw-r--r--. 1 8980 users 35067 Dec  2 11:26 COPYING-3_0.txt
-rw-r--r--. 1 8980 users 2802 Dec  2 11:26 fastdfs.spec
-rw-r--r--. 1 8980 users 31386 Dec  2 11:27 HISTORY
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:26 init.d
-rw-r--r--. 1 8980 users 7755 Dec  2 11:26 INSTALL
-rwxr-xr-x. 1 8980 users 5813 Dec  2 11:27 make.sh
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:26 php_client
-rw-r--r--. 1 8980 users 2380 Dec  2 11:26 README.md
-rwxr-xr-x. 1 8980 users 1768 Dec  2 11:26 restart.sh
-rwxr-xr-x. 1 8980 users 1680 Dec  2 11:26 stop.sh
drwxr-xr-x. 4 8980 users 4096 Dec  2 11:27 storage
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:26 test
drwxr-xr-x. 2 8980 users 4096 Dec  2 11:27 tracker
```

2.1.6.1.6. 编译安装 (编译前要确保已经成功安装了 libfastcommon)

```
[root@localhost ~]# ./make.sh
```

```
[root@localhost ~]# ./make.sh install
```

采用默认安装的方式安装,安装后的相应文件与目录：

A、服务脚本在：

```
/etc/init.d/fdfs_storaged
```

```
/etc/init.d/fdfs_tracker
```

B、配置文件在（ 样例配置文件 ）：

```
/etc/fdfs/client.conf.sample
```

```
/etc/fdfs/storage.conf.sample
```

```
/etc/fdfs/tracker.conf.sample
```

C、命令工具在/usr/bin/目录下的：

```
fdfs_appender_test
```

```
fdfs_appender_test1
```

```
fdfs_append_file
```

```
fdfs_crc32
```

```
fdfs_delete_file
```

```
fdfs_download_file
```

```
fdfs_file_info
```

```
fdfs_monitor
```

```
fdfs_storaged
```

```
fdfs_test
```

```
fdfs_test1
```

```
fdfs_trackerd
```

fdfs_upload_appender

fdfs_upload_file

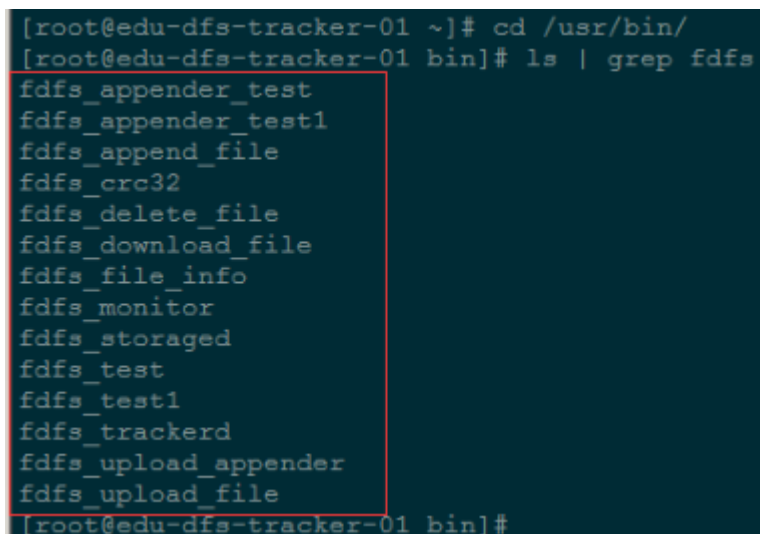
stop.sh

restart.sh

2.1.6.1.7. 因为 FastDFS 服务脚本设置的 bin 目录是 /usr/local/bin , 但实际命令安装在/usr/bin , 可以进入 /usr/bin 目录使用以下命令查看 fdfs 的相关命令

```
[root@localhost ~]# cd /usr/bin/
```

```
[root@localhost ~]# ls | grep fdfs
```



```
[root@edu-dfs-tracker-01 ~]# cd /usr/bin/
[root@edu-dfs-tracker-01 bin]# ls | grep fdfs
fdfs_appender_test
fdfs_appender_test1
fdfs_append_file
fdfs_crc32
fdfs_delete_file
fdfs_download_file
fdfs_file_info
fdfs_monitor
fdfs_storaged
fdfs_test
fdfs_test1
fdfs_trackerd
fdfs_upload_appender
fdfs_upload_file
[root@edu-dfs-tracker-01 bin]#
```

因此需要修改 FastDFS 服务脚本中相应的命令路径 , 也就是把 /etc/init.d/fdfs_storaged 和/etc/init.d/fdfs_tracker 两个脚本中的/usr/local/bin 修改成/usr/bin :

```
[root@localhost ~]# vi fdfs_trackerd
```

使用查找替换命令进统一修改 : %s+/usr/local/bin+/usr/bin

```
[root@localhost ~]# vi fdfs_storaged
```

使用查找替换命令进统一修改：%s+/usr/local/bin+/usr/bin

2.1.6.2. 配置 FastDFS 跟踪器(192.168.4.121)

2.1.6.2.1. 复制 FastDFS 跟踪器样例配置文件,并重命名

```
[root@localhost ~]# cd /etc/fdfs/
```

```
[root@edu-dfs-tracker-01 fdfs]# ll
total 20
-rw-r--r--. 1 root root 1461 Mar 25 23:15 client.conf.sample
-rw-r--r--. 1 root root 7829 Mar 25 23:15 storage.conf.sample
-rw-r--r--. 1 root root 7102 Mar 25 23:15 tracker.conf.sample
```

```
[root@localhost ~]# cp tracker.conf.sample tracker.conf
```

2.1.6.2.2. 编辑跟踪器配置文件

```
[root@localhost ~]# vi /etc/fdfs/tracker.conf
```

修改的内容如下：

disabled=false

port=22122

base_path=/fastdfs/tracker

(其它参数保留默认配置，具体配置解释请参考官方文档说明：

<http://bbs.chinaunix.net/thread-1941456-1-1.html>)

2.1.6.2.3. 创建基础数据目录 (参考基础目录 base_path 配置)

```
[root@localhost ~]# mkdir -p /fastdfs/tracker
```

2.1.6.2.4. 防火墙中打开跟踪器端口（默认为 22122）

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

添加如下端口行：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT
```

重启防火墙：

```
[root@localhost ~]# systemctl restart iptables.service
```

2.1.6.2.5. 启动 Tracker

```
[root@localhost ~]# /etc/init.d/fdfs_trackerd start
```

（初次成功启动，会在/fastdfs/tracker 目录下创建 data、logs 两个目录）

查看 FastDFS Tracker 是否已成功启动：

```
[root@localhost ~]# ps -ef | grep fdfs
```

```
[root@edu-dfs-tracker-01 init.d]# ps -ef | grep fdfs
root      2415      1  0 01:47 ?        00:00:00 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf
root      2423    2160  0 01:47 pts/1    00:00:00 grep fdfs
[root@edu-dfs-tracker-01 init.d]#
```

2.1.6.2.6. 关闭 Tracker

```
[root@localhost ~]# /etc/init.d/fdfs_trackerd stop
```

2.1.6.2.7. 设置 FastDFS 跟踪器开机启动

```
[root@localhost ~]# vi /etc/rc.d/rc.local
```

添加以下内容：

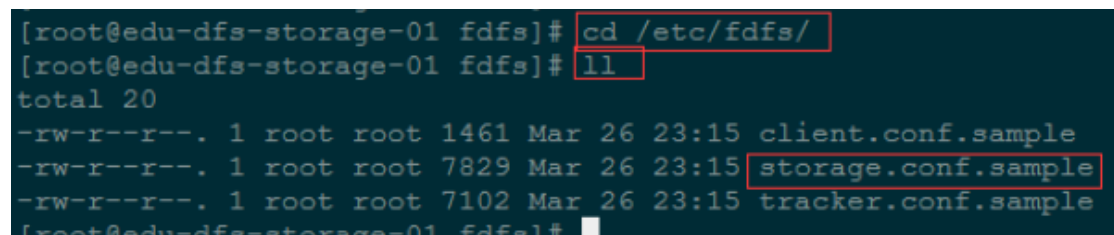
```
## FastDFS Tracker
```

```
/etc/init.d/fdfs_trackerd start
```

2.1.6.3. 配置 FastDFS 存储(192.168.4.125)

2.1.6.3.1. 复制 FastDFS 存储器样例配置文件,并重命名

```
[root@localhost ~]# cd /etc/fdfs/
```



```
[root@edu-dfs-storage-01 fdfs]# cd /etc/fdfs/
[root@edu-dfs-storage-01 fdfs]# ll
total 20
-rw-r--r--. 1 root root 1461 Mar 26 23:15 client.conf.sample
-rw-r--r--. 1 root root 7829 Mar 26 23:15 storage.conf.sample
-rw-r--r--. 1 root root 7102 Mar 26 23:15 tracker.conf.sample
[root@edu-dfs-storage-01 fdfs]#
```

```
[root@localhost ~]# cp storage.conf.sample storage.conf
```

2.1.6.3.2. 编辑存储器样例配置文件

```
[root@localhost ~]# vi /etc/fdfs/storage.conf
```

修改的内容如下:

```
disabled=false
```

```
port=23000
```

```
base_path=/fastdfs/storage
```

```
store_path0=/fastdfs/storage
```

```
tracker_server=192.168.4.121:22122
```

```
http.server_port=8888
```

(其它参数保留默认配置, 具体配置解释请参考官方文档说明:

<http://bbs.chinaunix.net/thread-1941456-1-1.html>)

2.1.6.3.3. 创建基础数据目录（参考基础目录 base_path 配置）

```
[root@localhost ~]# mkdir -p /fastdfs/storage
```

2.1.6.3.4. 防火墙中打开存储器端口（默认为 23000）

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

添加如下端口行：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT
```

重启防火墙：

```
[root@localhost ~]# service iptables restart
```

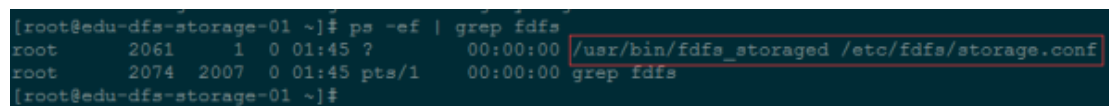
2.1.6.3.5. 启动 Storage

```
[root@localhost ~]# /etc/init.d/fdfs_storaged start
```

（初次成功启动，会在/fastdfs/storage 目录下创建 data、logs 两个目录）

查看 FastDFS Storage 是否已成功启动

```
[root@localhost ~]# ps -ef | grep fdfs
```



```
[root@edu-fdfs-storage-01 ~]# ps -ef | grep fdfs
root      2061      1  0 01:45 ?        00:00:00 /usr/bin/fdfs_storaged /etc/fdfs/storage.conf
root      2074    2007  0 01:45 pts/1    00:00:00 grep fdfs
[root@edu-fdfs-storage-01 ~]#
```

2.1.6.3.6. 关闭 Storage

```
[root@localhost ~]# /etc/init.d/fdfs_storaged stop
```

2.1.6.3.7. 设置 FastDFS 存储器开机启动

```
[root@localhost ~]# vi /etc/rc.d/rc.local
```

添加：

```
## FastDFS Storage
```

```
[root@localhost ~] /etc/init.d/fdfs_storaged start
```

2.1.6.4. 文件上传测试

2.1.6.4.1. 修改 Tracker 服务器中的客户端配置文件

```
[root@localhost ~]# cp /etc/fdfs/client.conf.sample /etc/fdfs/client.conf
```

```
[root@localhost ~]# vi /etc/fdfs/client.conf
```

```
base_path=/fastdfs/tracker
```

```
tracker_server=192.168.4.121:22122
```

2.1.6.4.2. 执行如下文件上传命令

```
[root@localhost ~]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf
```

```
/usr/local/src/FastDFS_v5.05.tar.gz
```

```
返回 ID 号：group1/M00/00/00/wKgEfVUYNYeAb7XFAAVFOL7FJU4.tar.gz
```

(能返回以上文件 ID，说明文件上传成功)

2.1.6.5. 在每个存储节点上安装 nginx

2.1.6.5.1. fastdfs-nginx-module 作用说明

FastDFS 通过 Tracker 服务器,将文件放在 Storage 服务器存储,但是同组存储服务器之间需要进入文件复制,有同步延迟的问题。假设 Tracker 服务器将文件上传到了 192.168.4.125,上传成功后文件 ID 已经返回给客户端。此时 FastDFS 存储集群机制会将这个文件同步到同组存储 192.168.4.126,在文件还没有复制完成的情况下,客户端如果用这个文件 ID 在 192.168.4.126 上取文件,就会出现文件无法访问的错误。而 fastdfs-nginx-module 可以重定向文件连接到源服务器取文件,避免客户端由于复制延迟导致的文件无法访问错误。(解压后的 fastdfs-nginx-module 在 nginx 安装时使用)

2.1.6.5.2. 上传 fastdfs-nginx-module_v1.16.tar.gz 到 /usr/local/src, 解压:

```
[root@localhost ~]# cd /usr/local/src/
```

```
[root@localhost ~]# tar -zxvf fastdfs-nginx-module_v1.16.tar.gz
```

2.1.6.5.3. 修改 fastdfs-nginx-module 的 config 配置文件

```
[root@localhost ~]# cd fastdfs-nginx-module/src
```

```
[root@localhost ~]# vi config
```

```
CORE_INCS="$CORE_INCS /usr/local/include/fastdfs
```

```
/usr/local/include/fastcommon/"
```

修改为:

```
CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
```

(注意 : 这个路径修改是很重要的 , 不然在 nginx 编译的时候会报错的)

2.1.6.5.4. Nginx 安装

具体查看 2.1.5.相关内容, 已安装可忽略。

2.1.6.5.5. 编译 Nginx (添加 fastdfs-nginx-module 模块)

(1) 进入 Nginx 安装目录 :

```
[root@localhost ~]# cd /opt/nginx-1.13.12
```

(2) 添加 fastdfs-nginx-module 模块

```
[root@localhost ~]# ./configure \
--prefix=/opt/nginx-1.13.12 \
--http-log-path=/var/log/nginx/access.log \
--error-log-path=/var/log/nginx/error.log \
--pid-path=/var/run/nginx.pid \
--lock-path=/var/lock/subsys/nginx.lock \
--user=nginx --group=nginx \
--with-pcre --with-http_v2_module \
--with-http_ssl_module \
--with-http_gzip_static_module \
--with-http_stub_status_module \
--with-http_realip_module \
--add-module=/usr/local/fastdfs-nginx-module/src
```

(3) 重新编译 Nginx

```
[root@localhost ~]# make
```

执行 make 命令 (注意只执行 make , 不要执行 install , 因为不需要重新安装 , 只需要生存新的 nginx 启动项即可)

2.1.6.5.6. 复制 fastdfs-nginx-module 源码中的配置文件到 /etc/fdfs 目录，并修改

```
[root@localhost ~]# cp /usr/local/src/fastdfs-nginx-  
module/src/mod_fastdfs.conf /etc/fdfs/
```

```
[root@localhost ~]# vi /etc/fdfs/mod_fastdfs.conf
```

修改以下配置：

```
connect_timeout=10
```

```
base_path=/tmp
```

```
tracker_server=192.168.4.121:22122
```

```
storage_server_port=23000
```

```
group_name=group1
```

```
url_have_group_name = true
```

```
store_path0=/fastdfs/storage
```

2.1.6.5.7. 复制 FastDFS 的部分配置文件到/etc/fdfs 目录

```
[root@localhost ~]# cd /usr/local/src/FastDFS/conf
```

```
[root@localhost ~]# cp http.conf mime.types /etc/fdfs/
```

2.1.6.5.8. 在/fastdfs/storage 文件存储目录下创建软连接,将其链接到实际存放数据的目录

```
[root@localhost ~]# ln -s /fastdfs/storage/data/ /fastdfs/storage/data/M00
```

2.1.6.5.9. 配置 Nginx , 简洁版 nginx 配置样例

```
user root;

worker_processes 1;

events {

    worker_connections 1024;

}

http {

    include mime.types;

    default_type application/octet-stream;

    sendfile on;

    keepalive_timeout 65;

    server {

        listen 8888;

        server_name localhost;

        location ~ /group([0-9])/M00 {

            #alias /fastdfs/storage/data;

            ngx_fastdfs_module;

        }

        error_page 500 502 503 504 /50x.html;

        location = /50x.html {

            root html;

        }

    }

}
```

```
}  
  
}
```

注意、说明：

A、8888 端口值是要与/etc/fdfs/storage.conf 中的 http.server_port=8888 相对应，

因为 http.server_port 默认为 8888,如果想改成 80，则要对应修改过来。

B、Storage 对应多个 group 的情况下，访问路径带 group 名，如 /group1/M00/00/00/xxx，

对应的 Nginx 配置为：

```
location ~/group([0-9])/M00 {  
  
    ngx_fastdfs_module;  
  
}
```

C、如查下载时如发现老报 404 ,将 nginx.conf 第一行 user nobody 修改为 user root 后重新启动。

2.1.6.5.10. 防火墙中打开 Nginx 的 8888 端口

```
[root@localhost ~]# vi /etc/sysconfig/iptables
```

添加：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8888 -j ACCEPT
```

```
[root@localhost ~]# systemctl enable iptables.service
```

2.1.6.5.11. 启动 Nginx

```
[root@localhost ~]# /opt/nginx/sbin/nginx
```

```
ngx_http_fastdfs_set pid=xxx
```

(重启 Nginx 的命令为 : `/opt/nginx/sbin/nginx -s reload`)

2.1.6.5.12. 通过浏览器访问测试时上传的文件

<http://192.168.4.125:8888/group1/M00/00/00/wKgEfVUYNYeAb7XFAAVFOL7FJU4.tar.gz>

注意 : 千万不要使用 `kill -9` 命令强杀 `FastDFS` 进程 , 否则可能会导致 `binlog` 。

2.2. 数据 SQL 脚本

2.2.1. 数据库 PDM 设计

SVN 地址 :

<http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/model/MCP.pdm>

<http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/model/MCP.rtf>

2.2.2. 系统管理 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_sys.sql

2.2.3. 机构管理 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_org.sql

2.2.4. 统一编码 SQL 脚本

2.2.5. 商品管理 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_goods.sql

2.2.6. 采购供货 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_core.sql

2.2.7. 证照管理 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_license.sql

2.2.8. 退货管理 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_refunds.sql

2.2.9. 骨科跟台 SQL 脚本

2.2.10. 视图信息 SQL 脚本

http://192.168.1.12:9005/svn/ebigsvn/EbigSupplyChain/DevProjects/mcp/v1.1/code/back-stage/mcp/mcp-document/库表设计/sql/hdi_view_table.sql

2.3. 项目打包命令

2.3.1. 前端打包命令

(1) 多环境配置

开发环境：dev；

测试环境：test；

生成环境：prod；

(2) maven 打包命令

npm run build:dev

2.3.2. 后端打包命令

目前采用 Maven 命令收到打包，后续集成自动化部署方案再补充。

(1) 多环境配置

开发环境：dev；

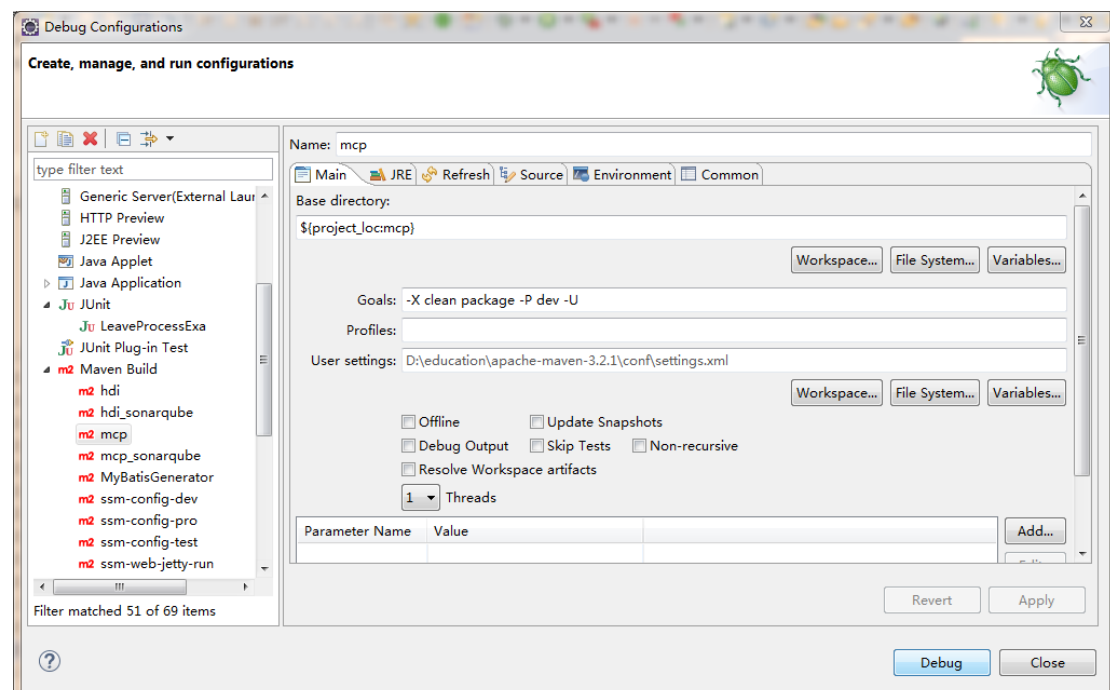
测试环境：test；

生成环境：prod；

(2) maven 打包命令

mvn -X clean package -P dev -U

(3) Eclipse 配置 maven 命令



2.4. 项目运行命令

2.4.1. 前端开发环境部署（使用普通用户进行操作）

- (1) 进入前端目录：

```
[hscm@hscm1 ~]$ /home/hscm/mcp/front-server
```

- (2) 解压命令：

```
[hscm@hscm1 ~]$ unzip mcp-dev.zip
```

2.4.2. 后端开发环境部署（使用普通用户进行操作）

- (1) 进入部署目录：

```
[hscm@hscm1 ~]$ /home/hscm/mcp/back-server
```

- (2) 查看端口：

```
[hscm@hscm1 ~]$ netstat -lnp | grep 2367
```

- (3) 关闭应用：

```
[hscm@hscm1 ~]$ kill -9 进程号
```

- (4) 启动应用：

```
[hscm@hscm1 ~]$ nohup java -jar mcp-server-admin-
```

1.0.TRUNK_20190909111754.jar &

2.4.3. Nginx 配置（使用 root 用户进行操作）

- (1) 进入Nginx配置目录：

```
[root@localhost ~]# /opt/nginx/conf
```

- (2) 配置添加内容：

```
[root@localhost ~]# vi nginx.conf
```

添加内容：

```
# mcp-front-server

server {

    listen      80;

    #    listen      somename:8080;

    server_name  www.xxx.com;


    proxy_http_version 1.1;


    gzip on;

    gzip_min_length 1k;

    gzip_buffers 4 16k;

    #gzip_http_version 1.0;

    gzip_comp_level 8;

    gzip_types text/plain application/javascript application/x-javascript
text/css application/xml text/javascript application/x-httpd-php image/jpeg
image/gif image/png;

    gzip_vary off;

    gzip_disable "MSIE [1-6]\.";


    location / {
```

```
        root    /home/hscm/mcp/front-server;

        index   index.html index.htm;

        try_files $uri $uri/ /index.html;

    }

}
```

(3) 保存退出 :

```
[hscm@hscm1 ~]# x
```

(4) 重启 Nginx

```
[root@localhost ~]# ./opt/nginx/sbin/nginx -s reload
```