Linux 服务器部署文档

1. 更新 163 yum 源

cd /etc/yum.repos.d/
mv CentOS-Base.repo CentOS-Base.repo.backup
wget http://mirrors.163.com/.help/CentOS6-Base-163.repo

[root@localhost ~]# yum clean all
[root@localhost ~]# yum makecache
[root@localhost ~]# yum update

或者是：http://centos.ustc.edu.cn/

2. 安装组件

// 安装上传下载组件 lrzsz
```
yum -y install lrzsz
```

// centos7 精简安装后使用发现没有 killall 命令
```
yum -y install psmisc
```

psmisc 软件包包含三个帮助管理/proc 目录的程序。

安装下列程序：fuser, killall,pstree 和 pstree.x11(到 pstree 的链接)

fuser 显示使用指定文件或者文件系统的进程的 PID。

killall 杀死某个名字的进程，它向运行指定命令的所有进程发出信号。

pstree 树型显示当前运行的进程。

pstree.x11 与 pstree 功能相同，只是在退出前需要确认。

安装 iptables 防火墙
```
yum install iptables-services
```

3. 安装 jdk

先卸载 open-jdk

java –version
rpm -qa | grep java

rpm -e --nodeps java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86_64
rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86_64

开始安装：
mkdir /usr/local/src/java
rz 上传 jdk tar 包
tar -xvf jdk-7u71-linux-i586.tar.gz

yum install glibc.i686

① vi /etc/profile

② 在末尾行添加

```
#set java environment
JAVA_HOME=/usr/local/src/java/jdk1.7.0_71
CLASSPATH=.:$JAVA_HOME/lib.tools.jar
PATH=$JAVA_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH
```

保存退出
③source /etc/profile 使更改的配置立即生效
④java -version 查看 JDK 版本信息，如果显示出 1.7.0 证明成功

4. 创建 ucenter 用户

一般生成环境是不会使用 root 用户来发布 tomcat 等应用的。

useradd ucenter
passwd ucenter -设置密码为：ucenter

mkdir /ucenter
chown ucenter:ucenter /ucenter/ -R

5. 安装 mysql

安装 mysql 的 percona 分支；
文档：http://www.percona.com/doc/percona-server/5.6/

安装包：Percona-Server-5.6.21-70.0-r688-el6-x86_64-bundle.tar
下载地址：
https://www.percona.com/downloads/Percona-Server-5.6/Percona-Server-5.6.21-70.0/binary/redhat/6/x86_64/Percona-Server-5.6.21-70.0-r688-el6-x86_64-bundle.tar

首先安装 cmake
yum -y install cmake

cd /usr/local/src/
mkdir mysql-percona
cd mysql-percona/
rz 上传安装包

tar -xvf Percona-Server-5.6.21-70.0-r688-el6-x86_64-bundle.tar

rpm -ivh Percona-Server-shared-56-5.6.21-rel70.0.el6.x86_64.rpm
rpm -ivh Percona-Server-client-56-5.6.21-rel70.0.el6.x86_64.rpm
rpm -ivh Percona-Server-server-56-5.6.21-rel70.0.el6.x86_64.rpm

启动：
service mysql start

修改 root 密码：
mysqladmin -u root password "root"

登录：
mysql -uroot -proot

设置远程访问（使用 root 密码）：
grant all privileges on *.* to 'root' @'%' identified by 'root';
flush privileges;

防火墙打开 3306 端口
/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
/etc/rc.d/init.d/iptables save
/etc/init.d/iptables status

5.1. 安装 3307 端口 mysql

mkdir /usr/local/mysql/data -p
mkdir /usr/local/mysql/logs -p
mkdir /usr/local/mysql/etc -p
mkdir /usr/local/mysql/var -p

chown mysql:mysql /usr/local/mysql/ -R

cp /etc/my.cnf /usr/local/mysql/etc/

vi /usr/local/mysql/etc/my.cnf

[mysqld]
port=3307
datadir=/usr/local/mysql/data
socket=/usr/local/mysql/mysql-3307.sock
user=mysql
log_error=/usr/local/mysql/logs/db_error.log
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/usr/local/mysql/logs/db_error.log
pid-file=/usr/local/mysql/mysqld-3307.pid

安装：
/usr/bin/mysql_install_db --defaults-file=/usr/local/mysql/etc/my.cnf --basedir=/usr/ --datadir=/usr/local/mysql/data --user=mysql

启动：
/usr/bin/mysqld_safe --defaults-file=/usr/local/mysql/etc/my.cnf &

修改 root 密码：
mysqladmin -h127.0.0.1 -P3307 -uroot password "root"

修改远程访问：
mysql -uroot -h127.0.0.1 -P3307 -p

grant all privileges on *.* to 'root' @'%' identified by 'root';
flush privileges;

防火墙打开 3307 端口
/sbin/iptables -I INPUT -p tcp --dport 3307 -j ACCEPT
/etc/rc.d/init.d/iptables save
/etc/init.d/iptables status

5.2. 解决 mysql 访问慢的问题

vim /etc/my.cnf

在[mysqld]下面添加：
skip-name-resolve

重启 mysql 服务：
service mysql restart

原因：
mysql 客户端每次访问 db，mysql 就会试图去解析来访问的机器的 hostname，并缓存到 hostname cache，如果这时解析不了，等一段时间会失败，数据才能被取过来。

6. 安装 Redis

yum -y install cpp binutils glibc glibc-kernheaders glibc-common glibc-devel gcc make gcc-c++ libstdc++-devel tcl

mkdir -p /usr/local/src/redis && cd /usr/local/src/redis
wget http://download.redis.io/releases/redis-2.8.17.tar.gz 或者 rz 上传
tar -xvf redis-2.8.17.tar.gz

创建软连接
ln -s /usr/local/src/redis/redis-2.8.17 /usr/local/redis

cd /usr/local/redis
make
make test #这个就不要执行了，需要很长时间
make install

cp redis.conf /etc/
vi /etc/redis.conf
修改如下，默认为 no
daemonize yes

#启动
redis-server /etc/redis.conf
#测试
redis-cli
#密码验证
auth redistest

7. 安装 Nginx

yum -y install gcc-c++
yum -y install pcre pcre-devel
yum -y install zlib zlib-devel
yum -y install openssl openssl-devel

mkdir /usr/local/src/nginx
cd /usr/local/src/nginx

wget http://nginx.org/download/nginx-1.10.2.tar.gz 或 rz 上传
tar -xvf nginx-1.10.2.tar.gz
cd nginx-1.10.2

#安装到/usr/local
mkdir -p /usr/local/nginx

创建 nginx 用户
useradd nginx

./configure --prefix=/usr/local/nginx --user=nginx --group=nginx
make
make install

由于非 root 用户不能占用 80 端口所以使普通用户以 root 身份启动 nginx。

cd /usr/local/nginx/sbin
chown root nginx
chmod u+s nginx

防火墙打开 80 端口
service iptables stop //关闭防火墙

/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT
/etc/rc.d/init.d/iptables save

/etc/init.d/iptables status

启动 nginx
/usr/local/nginx/sbin/nginx

8. 安装 Zookeeper

mkdir /usr/local/src/zookeeper
cd /usr/local/src/zookeeper

wget 或 rz 上传 zookeeper 源码安装文件到/usr/local/src/zookeeper
wget http://mirrors.hust.edu.cn/apache/zookeeper/stable/zookeeper-3.4.12.tar.gz

tar -xzvf zookeeper-3.4.12.tar.gz
cd zookeeper-3.4.12

修改配置文件
cd conf
mv zoo_sample.cfg zoo.cfg

创建软连接
ln -s /usr/local/src/zookeeper/zookeeper-3.4.12 /usr/local/zookeeper

创建 data 目录
mkdir /usr/local/zookeeper/data

启动 zookeeper
./zkServer.sh start

9. 安装 RabbitMQ

9.1. 安装 Erlang

9.1.1. 添加 yum 支持

cd /usr/local/src/
mkdir rabbitmq
cd rabbitmq

wget http://packages.erlang-solutions.com/erlang-solutions-1.0-1.noarch.rpm

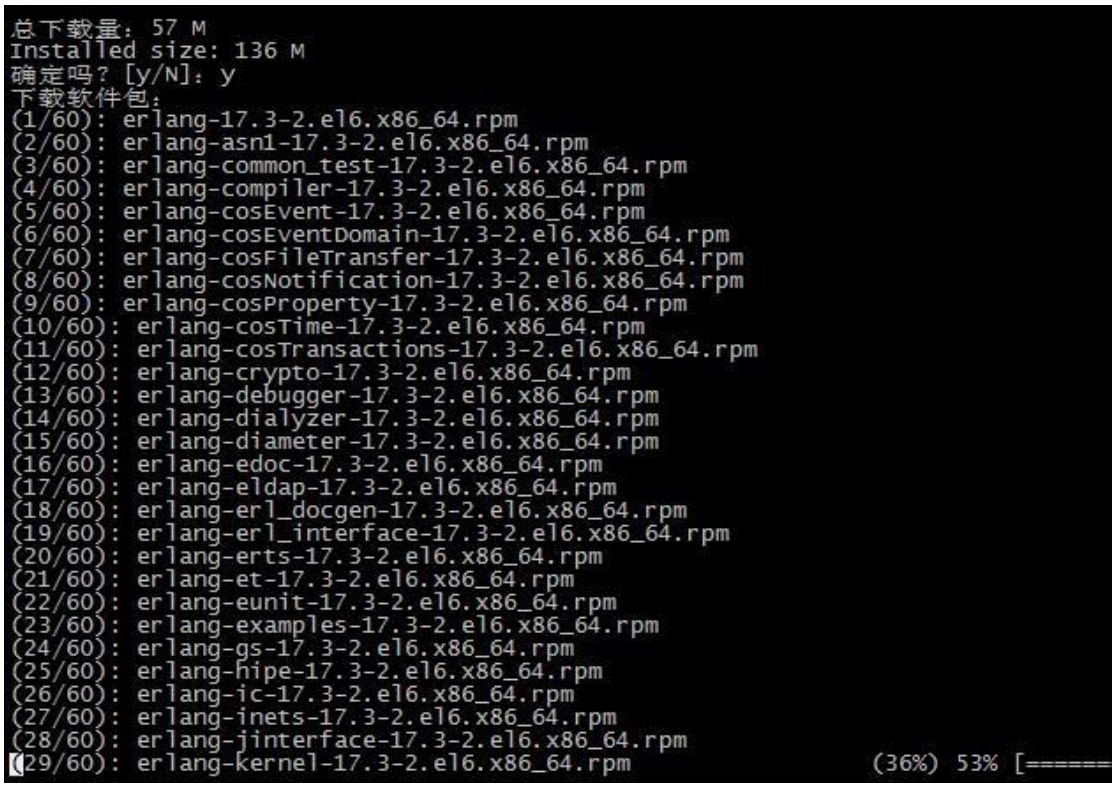
rpm -Uvh erlang-solutions-1.0-1.noarch.rpm

rpm --import http://packages.erlang-solutions.com/rpm/erlang_solutions.asc

sudo yum install erlang

或者：
上传 esl-erlang_17.3-1~centos~6_amd64.rpm
执行 yum install esl-erlang_17.3-1~centos~6_amd64.rpm

上传：esl-erlang-compat-R14B-1.el6.noarch.rpm
yum install esl-erlang-compat-R14B-1.el6.noarch.rpm



9.2. 安装 RabbitMQ

上传 rabbitmq-server-3.4.1-1.noarch.rpm 文件到/usr/local/src/rabbitmq/
安装：
rpm -ivh rabbitmq-server-3.4.1-1.noarch.rpm

9.2.1. 启动、停止

service rabbitmq-server start

service rabbitmq-server stop
service rabbitmq-server restart

9.2.2. 设置开机启动

chkconfig rabbitmq-server on

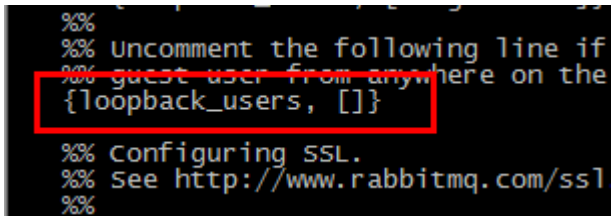
9.2.3. 设置配置文件

cd /etc/rabbitmq
cp /usr/share/doc/rabbitmq-server-3.4.1/rabbitmq.config.example /etc/rabbitmq/

mv rabbitmq.config.example rabbitmq.config

9.2.4. 开启用户远程访问

vi /etc/rabbitmq/rabbitmq.config



注意要去掉后面的逗号。

9.2.5. 开启 web 界面管理工具

rabbitmq-plugins enable rabbitmq_management
service rabbitmq-server restart

9.2.6. 防火墙开放 15672 端口

/sbin/iptables -I INPUT -p tcp --dport 15672 -j ACCEPT
/sbin/iptables -I INPUT -p tcp --dport 5672 -j ACCEPT
/etc/rc.d/init.d/iptables save

10. 安装 solr

切换到 ucenter 用户；

mkdir /ucenter/web/

rz 上传 taotao-solr-4.10.1.tar.gz

tar -xvf taotao-solr-4.10.1.tar.gz

mv example taotao-solr
cd taotao-solr/

java -Dsolr.solr.home=taotao-solr -jar start.jar &

/sbin/iptables -I INPUT -p tcp --dport 8983 -j ACCEPT
/etc/rc.d/init.d/iptables save

11. 程序部署

mkdir /ucenter/web/upload

名称	路径	server port	http port	Connector port
taotao-manage	/ucenter/web/taotao-manage/tomcat-taotao-manage	18005	18080	18109
taotao-web-01	/ucenter/web/taotao-web/tomcat-taotao-web-01	18006	18081	18110
taotao-web-02	/ucenter/web/taotao-web/tomcat-taotao-web-02	18007	18082	18111
taotao-web-03	/ucenter/web/taotao-web/tomcat-taotao-web-03	18008	18083	18112
taotao-sso-01	/ucenter/web/taotao-sso/tomcat-sso-01	18009	18084	18113
taotao-sso-02	/ucenter/web/taotao-sso/tomcat-sso-02	18010	18085	18114
taotao-order	/ucenter/web/taotao-order/tomcat-order	18011	18086	18115

12. Nginx 配置负载均衡

在 http 节点添加：

```
    upstream taotao-manage {
        server 127.0.0.1:18080;
        server 127.0.0.1:18081;
```

}

修改代理指向 upstream
proxy_pass http://tt-manage;

13. 安装 RocketMQ

13.1. 服务器环境

No.	IP	User	Password	Role	Mode
1	192.168.100.24	root		nameServer1,brokerServer1 Master1	Master1
2	192.168.100.25	root		nameServer2,brokerServer2 Master2	Master2

13.2. hosts 添加信息

```
vi /etc/hosts
```

```
192.168.100.24   rocketmq-nameserver1
192.168.100.24   rocketmq-master1
192.168.100.25   rocketmq-nameserver2
192.168.100.25   rocketmq-master2
```

13.3. 上传 RocketMQ 安装包

13.3.1. 上传安装压缩包

```
// 创建安装包存放目录
mkdir -p /usr/local/src/rocketmq && cd /usr/local/src/rocketmq
// 上传 rocketmq 安装包 alibaba-rocketmq-3.2.6.tar.gz
rz
```

13.3.2. 解压安装包

```
## 解压到/usr/local
tar -zxvf alibaba-rocketmq-3.2.6.tar.gz -C /usr/local
```

13.3.3. 重命名(添加版本号)

```
mv alibaba-rocketmq alibaba-rocketmq-3.2.6
```

13.3.4. 创建软连接

```
ln -s alibaba-rocketmq-3.2.6 rocketmq
```

13.4. 创建存储路径

```
mkdir /usr/local/rocketmq/store

mkdir /usr/local/rocketmq/store/commitlog

mkdir /usr/local/rocketmq/store/consumequeue

mkdir /usr/local/rocketmq/store/index

mkdir /usr/local/rocketmq/store/abort
```

```
touch /usr/local/rocketmq/store/checkpoint
```

13.5. RocketMQ 配置文件

13.5.1. 编辑 broker 配置文件

```
vim /usr/local/rocketmq/conf/2m-noslave/broker-a.properties

vim /usr/local/rocketmq/conf/2m-noslave/broker-b.properties
```

13.5.2. broker-a|broker-b 详细配置

```
#所属集群名字
brokerClusterName=rocketmq-cluster
#broker 名字，注意此处不同的配置文件填写的不一样
brokerName=broker-a|broker-b
#0 表示 Master，大于 0 表示 Slave
```

```
brokerId=0
#nameServer 地址，分号分割
namesrvAddr=rocketmq-nameserver1:9876;rocketmq-nameserver2:9876
#在发送消息时，自动创建服务器不存在的 topic，默认创建的队列数
defaultTopicQueueNums=4
#是否允许 Broker 自动创建 Topic，建议线下开启，线上关闭
autoCreateTopicEnable=true
#是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭
autoCreateSubscriptionGroup=true
#Broker 对外服务的监听端口
listenPort=10911
#删除文件时间点，默认凌晨 4 点
deleteWhen=04
#文件保留时间，默认 48 小时
fileReservedTime=120
#commitLog 每个文件的大小默认 1G
mapedFileSizeCommitLog=1073741824
#ConsumeQueue 每个文件默认存 30W 条，根据业务情况调整
mapedFileSizeConsumeQueue=300000
#destroyMapedFileIntervalForcibly=120000
#redeleteHangedFileInterval=120000
#检测物理文件磁盘空间
diskMaxUsedSpaceRatio=88
#存储路径
storePathRootDir=/usr/local/rocketmq/store
#commitLog 存储路径
storePathCommitLog=/usr/local/rocketmq/store/commitlog
#消费队列存储路径存储路径
storePathConsumeQueue=/usr/local/rocketmq/store/consumequeue
#消息索引存储路径
storePathIndex=/usr/local/rocketmq/store/index
#checkpoint 文件存储路径
storeCheckpoint=/usr/local/rocketmq/store/checkpoint
#abort 文件存储路径
abortFile=/usr/local/rocketmq/store/abort
#限制的消息大小
maxMessageSize=65536
#flushCommitLogLeastPages=4
#flushConsumeQueueLeastPages=2
#flushCommitLogThoroughInterval=10000
#flushConsumeQueueThoroughInterval=60000
#Broker 的角色
#- ASYNC_MASTER 异步复制 Master
#- SYNC_MASTER 同步双写 Master
```

```
#- SLAVE
brokerRole=ASYNC_MASTER
#刷盘方式
#- ASYNC_FLUSH  异步刷盘
#- SYNC_FLUSH  同步刷盘
flushDiskType=ASYNC_FLUSH

#checkTransactionMessageEnable=false

#发消息线程池数量
#sendMessageThreadPoolNums=128
#拉消息线程池数量
#pullMessageThreadPoolNums=128
```

13.6.修改日志配置文件【两台服务器】

```
mkdir -p /usr/local/rocketmq/logs
cd /usr/local/rocketmq/conf
sed -i 's#${user.home}#/usr/local/rocketmq#g' *.xml
```

13.7.修改启动脚本

```
vim /usr/local/rocketmq/bin/runbroker.sh
#===========================================================================
# JVM Configuration
#===========================================================================
JAVA_OPT="${JAVA_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:PermSize=128m -XX:MaxPermSize=320m"
```

```
vim /usr/local/rocketmq/bin/runserver.sh
#===========================================================================
# JVM Configuration
#===========================================================================
JAVA_OPT="${JAVA_OPT} -server -Xms1g -Xmx1g -Xmn512m -XX:PermSize=128m -XX:MaxPermSize=320m"
```

13.8.启动 NameServer【两台服务器】

```
cd /usr/local/rocketmq/bin
nohup sh mqnamesrv &
```

注：必须先启动 NameServer，然后再启动 BrokerServer。

13.9.启动 BrokerServer A【192.168.100.24】

```
cd /usr/local/rocketmq/bin
nohup sh mqbroker -c /usr/local/rocketmq/conf/2m-noslave/broker-a.properties >/dev/null 2>&1 &
netstat –ntlp
jps // Java Virtual Machine Process Status Tools
tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/broker.log
tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/namesrv.log
```

13.10. 启动 BrokerServer B【192.168.100.25】

```
cd /usr/local/rocketmq/bin
nohup sh mqbroker -c /usr/local/rocketmq/conf/2m-noslave/broker-b.properties >/dev/null 2>&1 &
netstat -ntlp
jps
tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/broker.log
tail -f -n 500 /usr/local/rocketmq/logs/rocketmqlogs/namesrv.log
```

13.11. Broker 启动问题

- 多网卡服务器环境会出现以下问题

```
com.alibaba.rocketmq.remoting.exception.RemotingConnectException: connect to <10.105.23.114:10911> failed
```

- **解决方案 1**

  可参考 RocketMQ 用户指南的 12 章，broker 配置参数一节。

1) 获取 broker 的默认配置

```
vim /usr/local/rocketmq/conf/2m-noslave/broker-a.properties
```

2) broker 启动时，如何加载配置

```
### Step1：生成 Broker 默认配置模板

sh mqbroker -m > broker.p
```

Step2：修改配置文件，即 broker.p 文件

注：将 **namesrvAddr=127.0.0.1:9876** 等配置信息补全，brokerServer 启动不了的很大部分原因是因为生成的配置文件信息不全造成的。至于为何会造成读取以及生成配置信息丢失的原因，需作更深入的探讨。

Step3：加载修改完善后的配置文件

nohup sh mqbroker -c broker.p

broker 运行过程中，动态修改 broker 的配置，注意：并非所有配置项都支持动态变更。

修改地址为 192.168.1.100:10911 的 Broker 消息保存时间为 24 小时

sh mqadmin updateBrokerConfig -b 192.168.1.100:10911 -k fileReservedTime 24

- **解决方案 2**

1、进入 rocketmq 日志目录：/usr/local/rocketmq/logs/rocketmqlogs

2、查看 store.log 日志文件，检查是否存在异常，出现问题很大程度上是因为缺少必须的数据目录，如：/usr/local/rocketmq/store/checkpoint 等目录不存在；此外，还需查看 broker.log，检查 broker 日志打印是否存在异常。

13.12. 数据清理

```
cd /usr/local/rocketmq/bin

sh mqshutdown namesrv
--等待停止
rm -rf /usr/local/rocketmq/store
mkdir /usr/local/rocketmq/store
mkdir /usr/local/rocketmq/store/commitlog
mkdir /usr/local/rocketmq/store/consumequeue
mkdir /usr/local/rocketmq/store/index
--按照上面步骤重启 NameServer 与 BrokerServer
```

13.13. RocketMQ Console

```
nohup java -jar rocketmq-console-ng-1.0.0.jar &
```