# 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/

# 安装rzsz

yum -y install lrzsz

# 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

# 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

# 安装mysql

安装的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

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

## 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

## 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。

# 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

cd redis-2.8.17

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

# 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-1.7.7.tar.gz> 或 rz上传

tar -xvf nginx-1.7.7.tar.gz

cd nginx-1.7.7

#安装到/ucenter

mkdir -p /ucenter/soft/nginx

./configure --prefix=/ucenter/soft/nginx --user=ucenter --group=ucenter

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 /ucenter/soft/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

# 安装zookeeper

mkdir /usr/local/src/zookeeper

cd /usr/local/src/zookeeper

rz 上传/usr/local/src/zookeeper

tar -xvf zookeeper-3.4.8.tar.gz

cd zookeeper-3.4.8

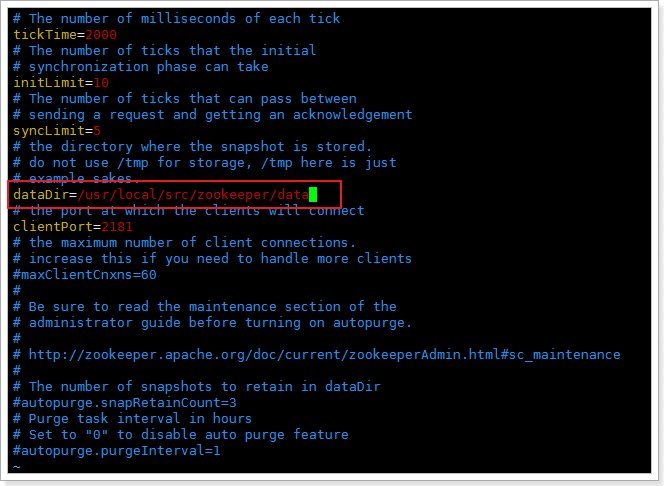
修改配置文件：

cd conf

mv zoo\_sample.cfg zoo.cfg

创建data目录：

mkdir /usr/local/src/zookeeper/data



启动：

cd ../bin

sh zkServer.sh start

## 搭建集群

步骤：

1. 在data目录下创建myid文件写入id值，比如，节点1写入1，节点2写入2等。
2. 在conf下的zoo.cfg文件中设置不同的端口，并且设置server.A=B:C:D的信息
   1. 说明：  
      server.A=B：C：D：其中 A 是一个数字，表示这个是第几号服务器；B 是这个服务器的 ip 地址；C 表示的是这个服务器与集群中的 Leader 服务器交换信息的端口；D 表示的是万一集群中的 Leader 服务器挂了，需要一个端口来重新进行选举，选出一个新的 Leader，而这个端口就是用来执行选举时服务器相互通信的端口。如果是伪集群的配置方式，由于 B 都是一样，所以不同的 Zookeeper 实例通信端口号不能一样，所以要给它们分配不同的端口号。
3. 分别启动各个服务节点即可。

# 安装RabbitMQ

## 安装Erlang

### 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

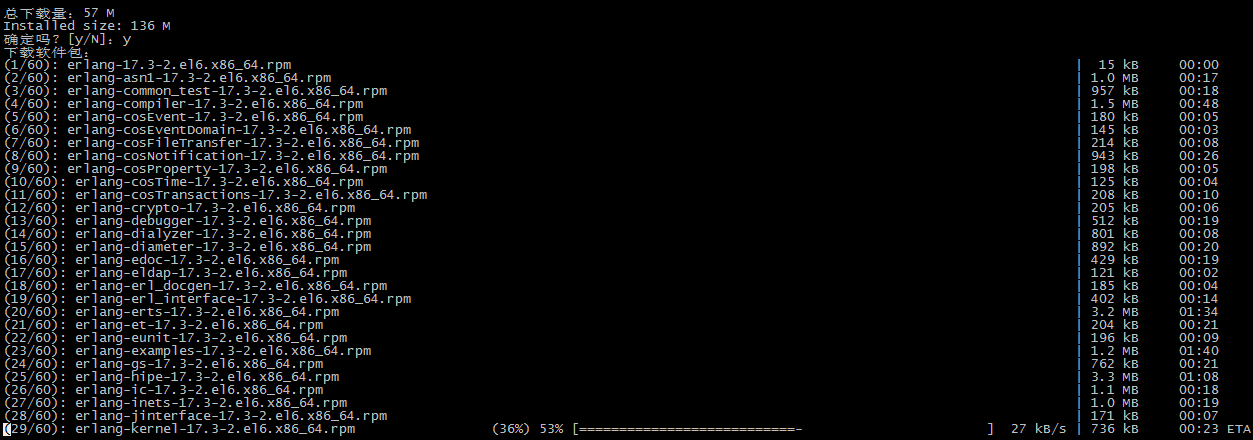
或者：

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



## 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

### 启动、停止

service rabbitmq-server start

service rabbitmq-server stop

service rabbitmq-server restart

### 设置开机启动

chkconfig rabbitmq-server on

### 设置配置文件

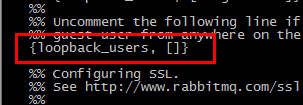
cd /etc/rabbitmq

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

mv rabbitmq.config.example rabbitmq.config

### 开启用户远程访问

vi /etc/rabbitmq/rabbitmq.config



**注意要去掉后面的逗号。**

### 开启web界面管理工具

rabbitmq-plugins enable rabbitmq\_management

service rabbitmq-server restart

### 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

# 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

# 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 |

# 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://taotao-manage>;