查看Linux版本

[root@bogon opt]# cat /etc/redhat-release

CentOS release 6.5 (Final)

[root@bogon opt]# cat /proc/version

Linux version 2.6.32-431.el6.x86\_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7 20120313 (Red Hat 4.4.7-4) (GCC) ) #1 SMP Fri Nov 22 03:15:09 UTC 2013

修改主机名

修改系统网络配置文件

首先备份待修改的文件：

cp /etc/sysconfig/network /etc/sysconfig/network.'date +%Y%m%d.%H%M%S'

然后开始修改：

vi /etc/sysconfig/network

输入i，进入编辑模式，修改HOSTNAME为我们想要的名字 comex01-ct65

最后，wq 保存文件

修改hosts

首先备份待修改的文件：

cp /etc/hosts /etc/hosts.'date +%Y%m%d.%H%M%S'

然后开始修改：

vi /etc/hosts

在127.0.0.1后添加我们要修改的主机名(注意之后要家空格)

最后，wq 保存文件

使用hostname修改当前主机名

执行了上述两步操作后，需要重启才能生效，但是有时候我们不希望重启，就可以直接执行下述命令强制使其生效

hostname comex01-ct65

然后退出重新登录即可： exit

CentOS6.5本地防火墙命令

查看防火墙状态： service iptables status

关闭防火墙： service iptables stop

启动防火墙： service iptables start

重启防火墙： service iptables restart

关闭防火墙开启自启： chkconfig iptables off

检查是否关闭成功： chkconfig --list iptables 全部为off即表示禁用开机自启成功

修改防火墙：vim /etc/sysconfig/iptables

添加端口允许访问（REJECT前）：-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT

JDK1.8安装

在/opt/software/下下载JDK：wget --no-check-certificate --no-cookies --header "Cookie:oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz>

创建目录/usr/local/java/并进入,将jdk压缩包复制到该目录并解压：cp /opt/software/jdk-8u161-linux-x64.tar.gz jdk-8u161-linux-x64.tar.gz; tar -xvf jdk-8u161-linux-x64.tar.gz

删除/usr/local/java/下jdk压缩包，配置JDK全局环境变量：vim /etc/profile在最后追加如下信息：

JAVA\_HOME=/usr/local/java/jdk1.8.0\_161

JRE\_HOME=/usr/local/java/jdk1.8.0\_161/jre

CLASSPATH=.:$JAVA\_HOME/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

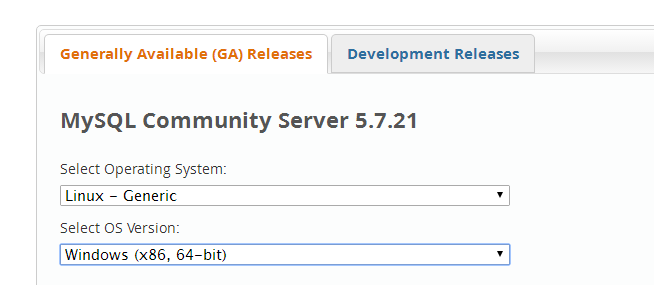
export JAVA\_HOME JRE\_HOME CLASSPATH PATH

使环境变量生效

source /etc/profile

MySQL安装（解压方式）

下载安装包到/opt/software/：wget --no-check-certificate --no-cookies --header "Cookie:oraclelicense=accept-securebackup-cookie" <https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.21-linux-glibc2.12-x86_64.tar.gz>



卸载老版本MySQL

|  |
| --- |
| rpm -qa | grep mysql #查看该操作系统上是否已经安装了mysql数据库  rpm -e mysql #普通删除  rpm -e --nodeps mysql #强力删除  yum -y remove php-mysql  yum -y remove mysql-server mysql |

禁用SeLinux

|  |
| --- |
| setenforce 0 |

安装依赖库

|  |
| --- |
| yum install -y gcc gcc-c++ kernel-devel ntp vim-enhanced flex bison autoconf make automake bzip2-devel ncurses-devel zlib-devel libjpeg-devel libpng-devel libtiff-devel freetype-devel libXpm-devel gettext-devel  pam-devel libtool libtool-ltdl openssl openssl-devel fontconfig-devel libxml2-devel curl-devel  libicu libicu-devel libmcrypt libmcrypt-devel libmhash libmhash-devel pcre-devel libtool-libs gd file patch mlocate diffutils readline-devel glibc-devel glib2-devel libcap-devel  yum install -y libaio |

移动到安装目录

|  |
| --- |
| sudo mkdir /usr/local/mysql  cd /opt/software/  tar -xzvf ./mysql-5.5.49-linux2.6-x86\_64.tar.gz  sudo mv ./mysql-5.5.49-linux2.6-x86\_64/\* /usr/local/mysql |

新建用户

|  |
| --- |
| sudo groupadd mysql  sudo useradd -g mysql mysql   #不让mysql用户直接登录 -s /usr/sbin/nologin  sudo chown -R mysql:mysql /usr/local/mysql/ |

创建data目录

|  |
| --- |
| mkdir data  sudo chown -R mysql:mysql /usr/local/mysql/data/ |

初始化mysql

|  |
| --- |
| [root@localhost local]# /usr/local/mysql/bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql5721 --datadir=/usr/local/mysql5721/data  2018-02-07T13:23:04.166038Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit\_defaults\_for\_timestamp server option (see documentation for more details).  2018-02-07T13:23:04.524230Z 0 [Warning] InnoDB: New log files created, LSN=45790  2018-02-07T13:23:04.577914Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.  2018-02-07T13:23:04.640428Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 074ba41d-0c0a-11e8-a94e-000c296c285d.  2018-02-07T13:23:04.641762Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid\_executed' cannot be opened.  2018-02-07T13:23:04.643137Z 1 [Note] A temporary password is generated for root@localhost: n1e+Y1s2g?-# |

启动MySQL

|  |
| --- |
| [root@localhost local]# /usr/local/mysql5721/bin/mysqld\_safe --user=mysql & |

设置自定义配置文件

|  |
| --- |
| 在mysql根路径下，创建my.cnf文件，文件内容：  [mysqld]  basedir = /usr/local/mysql5721  datadir = /usr/local/mysql5721/data  lower\_case\_table\_names=1  default-storage-engine=INNODB  character\_set\_server=utf8  init\_connect='SET NAMES utf8' |

用初始密码登录MySQL并设置密码

|  |
| --- |
| [root@localhost mysql5721]# chown -R root ./\*  [root@localhost mysql5721]# chown -R mysql data  [root@localhost mysql5721]# /usr/local/mysql5721/bin/mysqladmin -uroot -p password  #使用初始化MySQL时生成的初始密码，设置新密码  [root@localhost mysql5721]# /usr/local/mysql5721/bin/mysql -u root -p |

配置开机自动启动

|  |
| --- |
| cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld  chmod +x /etc/init.d/mysqld  chkconfig --add mysqld  chkconfig --level 2345 mysqld on  chkconfig --list mysqld #查看是否已应用上 |

关闭MySQL

|  |
| --- |
| [root@localhost mysql5721]# /usr/local/mysql5721/bin/mysqladmin -uroot -p shutdown |

设置MySQL自启动

|  |
| --- |
| [root@localhost mysql5721]# cp /usr/local/mysql5721/support-files/mysql.server /etc/rc.d/init.d/mysqld  [root@localhost mysql5721]# chmod +x /etc/rc.d/init.d/mysqld  [root@localhost mysql5721]# chkconfig --add mysqld  [root@localhost mysql5721]# chkconfig --list mysqld |

以后可以使用service命令控制mysql的启动和停止，命令为：service mysqld start和service mysqld stop，有可能执行的时候会报错，这是因为mysql默认安装在/usr/local目录下，如果安装在/usr/local目录下则会正常启动和关闭，不会报错。但是本次安装在自定义的/data目录下，此时需要修改/etc/init.d/mysqld文件，根据实际情况设置

basedir=/user/local/mysql5721、datadir=/user/local/mysql5721/data，保存后退出

设置环境变量

|  |
| --- |
| vim /etc/profile  最后添加  PATH=/data/mysql/bin:/data/mysql/lib:$PATH  export PATH  设置环境变量立即生效  # source /etc/profile |

设置远程主机登录

|  |
| --- |
| mysql> grant all privileges on \*.\* to 'root'@'%' identified by '123456' with grant option; |

修改root密码

|  |
| --- |
| /usr/local/mysql/bin/mysqladmin -u root -p password "123456" |

配置防火墙，开启3306端口

|  |
| --- |
| vim /etc/sysconfig/iptables |

添加如下规则到22端口这条规则的下面即可

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

重启防火墙

|  |
| --- |
| /etc/init.d/iptables restart  # 或  service iptables restart |

启动MySQL

|  |
| --- |
| service mysqld start  # 或 /etc/init.d/mysqld start |

查看是否有mysql进程

|  |
| --- |
| ps -ef | grep mysql |

查看是否有mysql端口

|  |
| --- |
| netstat -ntlp | grep 3306 |

测试mysql,mysqladmin,mysqldump命令是否能正常使用

读取MySQL的版本信息

|  |
| --- |
| mysqladmin version |

MySQL安装（yum方式）

检测系统是否已经安装过mysql或其依赖，若已装过要先将其删除，否则第4步使用yum安装时会报错

|  |
| --- |
| # yum list installed | grep mysql  mysql-libs.i686 5.1.71-1.el6 @anaconda-CentOS-201311271240.i386/6.5  # yum -y remove mysql-libs.i686 |

从mysql的官网下载mysql57-community-release-el6-5.noarch.rpm（注意这里的el6-5即适配RHEL6.5的版本，如果下载了其它版本后面的安装过程中可能会报错）

|  |
| --- |
| wget dev.mysql.com/get/mysql-community-release-el6-5.noarch.rpm |

安装第一步下载的rpm文件

|  |
| --- |
| yum install mysql-community-release-el6-5.noarch.rpm |

安装成功后，我们可以看到/etc/yum.repos.d/目录下增加了以下两个文件

|  |
| --- |
| # ls /etc/yum.repos.d  mysql-community-source.repo  mysql-community.repo |

查看mysql57的安装源是否可用，如不可用请自行修改配置文件（/etc/yum.repos.d/mysql-community.repo）使mysql57下面的enable=1，若有mysql其它版本的安装源可用，也请自行修改配置文件使其enable=0

|  |
| --- |
| # yum repolist enabled | grep mysql  mysql-connectors-community MySQL Connectors Community 13  mysql-tools-community MySQL Tools Community 18  mysql57-community-dmr MySQL 5.7 Community Server Development Milesto 65 |

使用yum安装mysql

|  |
| --- |
|  |

启动mysql服务

|  |
| --- |
| service mysqld start |

查看root密码

|  |
| --- |
| grep "password" /var/log/mysqld.log  2016-08-10T15:03:02.210317Z 1 [Note] A temporary password is generated for root@localhost: AYB(&-3Cz-rW |

现在必须立刻修改密码，不然会报错

|  |
| --- |
| ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement. |

修改密码（如果在此步报错ERROR 1819，请向下翻查看原因及解决方法）

|  |
| --- |
| mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpass'); |

查看mysqld是否开机自启动，并设置为开机自启动

|  |
| --- |
| chkconfig --list | grep mysqld  chkconfig mysqld on |

修改字符集为UTF-8

|  |
| --- |
| vim /etc/my.cnf |

在[mysqld]部分添加

|  |
| --- |
| character-set-server=utf8 |

在文件末尾新增[client]段，并在[client]段添加

|  |
| --- |
| default-character-set=utf8 |

修改好之后重启mysqld服务

|  |
| --- |
| service mysqld restart |

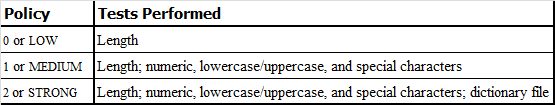
查看修改结果

|  |
| --- |
| mysql> show variables like "%character%";  +--------------------------+----------------------------+  | Variable\_name | Value |  +--------------------------+----------------------------+  | character\_set\_client | utf8 |  | character\_set\_connection | utf8 |  | character\_set\_database | utf8 |  | character\_set\_filesystem | binary |  | character\_set\_results | utf8 |  | character\_set\_server | utf8 |  | character\_set\_system | utf8 |  | character\_sets\_dir | /usr/share/mysql/charsets/ |  +--------------------------+----------------------------+  rows in set (0.00 sec) |

注：在修改密码步骤，若设置的密码为简单密码，可能会出现如下错误

|  |
| --- |
| ERROR **1819** (HY000): Your password does not satisfy the current policy requirements |

这一错误其实与validate\_password\_policy值的设置有关：



validate\_password\_policy值默认为1，即MEDIUM，所以刚开始设置的密码必须符合长度要求，且必须含有数字，小写或大写字母，特殊字符 如果我们只是做为测试用而不需要如此复杂的密码，可使用如下方式修改validate\_password\_policy值

|  |
| --- |
| mysql> set global validate\_password\_policy=**0**;  Query OK, **0** rows affected (**0.00** sec) |

这样，对密码要求就只有长度了，而密码的最小长度由validate\_password\_length值决定 validate\_password\_length参数默认为8，它有最小值的限制，最小值为：

|  |
| --- |
| validate\_password\_number\_count+ validate\_password\_special\_char\_count+ (**2** \* validate\_password\_mixed\_case\_count) |

其中，validate\_password\_number\_count指定了密码中数字的长度，validate\_password\_special\_char\_count指定了密码中特殊字符的长度，validate\_password\_mixed\_case\_count指定了密码中大小字母的长度。这些参数的默认值均为1，所以validate\_password\_length最小值为4，如果显性指定validate\_password\_length的值小于4，尽管不会报错，但validate\_password\_length的值将设为4 设置validate\_password\_length的值：

|  |
| --- |
| mysql> set global validate\_password\_length=**4**;  Query OK, **0** rows affected (**0.00** sec) |

如果修改了validate\_password\_number\_count，validate\_password\_special\_char\_count，validate\_password\_mixed\_case\_count中任何一个值，则validate\_password\_length将进行动态修改。

Nginx安装

下载Nginx

wget <http://nginx.org/download/nginx-1.13.8.tar.gz>

安装必须环境

[root@localhost software]# yum install gcc-c++

[root@localhost software]# yum -y install pcre\*

[root@localhost software]# yum -y install openssl\*

解压并进入解压目录

[root@localhost local]# cd nginx/

设置安装路径

[root@localhost nginx]# ./configure --prefix=/usr/local/nginx1138

编译并安装

[root@localhost nginx]# make

[root@localhost nginx]# make install

开启防火墙的80端口和443端口

/etc/sysconfig/iptables添加端口

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

-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT

启动Nginx

[root@localhost nginx1138]# cd /usr/local/nginx1138/sbin/

[root@localhost sbin]# ./nginx

重启Nginx

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

关闭Nginx

[root@localhost sbin]# kill -9

或者

[root@localhost sbin]# killall -HUP nginx

配置Nginx配置

vim /usr/local/nginx1138/conf/nginx.conf

Maven安装

Redis安装

安装tcl工具

[root@localhost software]# yum -y install tcl

下载Redis

wget http://download.redis.io/releases/redis-4.0.8.tar.gz

加压并进入目录

[root@localhost software]# tar -zxvf redis-4.0.8.tar.gz

编译安装到指定目录

[root@localhost software]# cd redis-4.0.8

[root@localhost software]# make

[root@localhost redis-4.0.8]# mkdir /usr/local/redis408

[root@localhost redis-4.0.8]# make PREFIX=/usr/local/redis408/ install

测试Redis

[root@localhost redis-4.0.8]# cd src

[root@localhost src]# make test

配置Redis

[root@localhost redis-4.0.8]# cp redis.conf /usr/local/redis408/redis.conf

[root@localhost redis-4.0.8]# vim /usr/local/redis408/redis.conf

我只修改了如下两项：

daemonize yes #redis将以守护进程的方式运行，默认为no会暂用你的终端

timeout 300​ #当 客户端闲置多长时间后关闭连接，如果指定为0，表示关闭该功能

设置自启动

vim /etc/init.d/redis

|  |
| --- |
| #!/bin/sh  #  # redis Startup script for Redis Server  #  # chkconfig: - 80 12  # description: Redis is an open source, advanced key-value store.  #  # processname: redis-server  # config: /etc/redis.conf  # pidfile: /var/run/redis.pid  source /etc/init.d/functions  BIN="/usr/local/redis/bin"  CONFIG="/usr/local/redis/redis.conf"  PIDFILE="/var/run/redis.pid"  ### Read configuration  [ -r "$SYSCONFIG" ] && source "$SYSCONFIG"  RETVAL=0  prog="redis-server"  desc="Redis Server"  start() {  if [ -e $PIDFILE ];then  echo "$desc already running...."  exit 1  fi  echo -n $"Starting $desc: "  daemon $BIN/$prog $CONFIG  RETVAL=$?  echo  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog  return $RETVAL  }  stop() {  echo -n $"Stop $desc: "  killproc $prog  RETVAL=$?  echo  [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog $PIDFILE  return $RETVAL  }  restart() {  stop  start  }  case "$1" in  start)  start  ;;  stop)  stop  ;;  restart)  restart  ;;  condrestart)  [ -e /var/lock/subsys/$prog ] && restart  RETVAL=$?  ;;  status)  status $prog  RETVAL=$?  ;;  \*)  echo $"Usage: $0 {start|stop|restart|condrestart|status}"  RETVAL=1  esac  exit $RETVAL |

启动和关闭

service redis start;

service redis stop;

或

[root@localhost bin]# ./redis-server ../redis.conf

kill -9 xxxx

使用Redis

[root@localhost bin]# ./redis-cli

127.0.0.1:6379>

MongoDB安装

下载MongoDB：<https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.6.2.tgz>

解压

[root@localhost software]# tar -zxvf mongodb-linux-x86\_64-3.6.2.tgz

移动到目标地址

[root@localhost software]# mv mongodb-linux-x86\_64-3.6.2 /usr/local/mongodb362

创建数据目录及日志目录

[root@localhost software]# cd /usr/local/mongodb362/

[root@localhost mongodb362]# mkdir mongo\_data

[root@localhost mongodb362]# mkdir mongo\_log

启动

[root@localhost bin]# ./mongod --port 27017 --fork --dbpath=/usr/local/mongodb362/mongo\_data/ --logpath=/usr/local/mongodb362/mongo\_log/mongodb.log --logappend

RabbitMQ安装

安装编译工具

yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel openssl-devel

安装Erlang

wget <http://erlang.org/download/otp_src_20.2.tar.gz>

tar -zxvf otp\_src\_20.2.tar.gz

cd otp\_src\_20.2

./configure --prefix=/usr/local/erlang --with-ssl -enable-threads -enable-smmp-support -enable-kernel-poll --enable-hipe --without-javac

make && make install

配置Erlang环境

|  |
| --- |
| vim /etc/profile  #在文件末尾添加下面代码 'ERLANG\_HOME'等于上一步'--prefix'指定的目录  ERLANG\_HOME=/usr/local/erlang  PATH=$ERLANG\_HOME/bin:$PATH  export ERLANG\_HOME  export PATH  #使环境变量生效  source /etc/profile  #输入命令检验是否安装成功  erl  #如下输出表示安装成功  [root@bogon otp\_src\_20.2]# erl  Erlang/OTP 20 [erts-9.2] [source] [64-bit] [smp:2:2] [ds:2:2:10] [async-threads:10] [hipe] [kernel-poll:false]  Eshell V9.2 (abort with ^G)  1> |

下载RabbitMQ：<http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.15/rabbitmq-server-generic-unix-3.6.15.tar.xz>

安装xz解压工具

下载xz包<https://tukaani.org/xz/xz-5.2.3.tar.bz2>

$tar -jxvf xz-xxx.tar.bz2

$ ./configure --prefix=/opt/gnu/xz
$ make
$ sudo make install

$ln -s /opt/gnu/xz/bin/xz /bin/xz

安装

RabbitMQ3.6版本无需make、make install 解压就可以用

|  |
| --- |
| #解压rabbitmq，官方给的包是xz压缩包，所以需要使用xz命令  xz -d rabbitmq-server-generic-unix-3.6.1.tar.xz  #xz解压后得到.tar包，再用tar命令解压  tar -xvf rabbitmq-server-generic-unix-3.6.1.tar  #移动目录 看个人喜好  cp -rf ./rabbitmq\_server-3.6.1 /usr/local/  cd /usr/local/  #修改文件夹名  mv rabbitmq\_server-3.6.1 rabbitmq-3.6.1  #开启管理页面插件  cd ./rabbitmq-3.6.1/sbin/  ./rabbitmq-plugins enable rabbitmq\_management |

启动

|  |
| --- |
| #启动命令，该命令ctrl+c后会关闭服务  ./rabbitmq-server  #在后台启动Rabbit  ./rabbitmq-server -detached  #关闭服务  ./rabbitmqctl stop  #关闭服务(kill) 找到rabbitmq服务的pid [不推荐]  ps -ef|grep rabbitmq  kill -9 \*\*\*\* |

添加管理员账号

|  |
| --- |
| #进入RabbitMQ安装目录  cd /usr/local/rabbitmq-3.6.1/sbin  #添加用户  #rabbitmqctl add\_user Username Password  ./rabbitmqctl add\_user rabbitadmin 123456  #分配用户标签  #rabbitmqctl set\_user\_tags User Tag  #[administrator]:管理员标签  ./rabbitmqctl set\_user\_tags rabbitadmin administrator |

登录管理界面，浏览器输入地址：http://服务器IP地址:15672/，输入第4部添加的账号密码登录

安装完成之后如果机器有iptables，需要开放端口

|  |
| --- |
| RabbitMQ常用端口（摘自网络）  **4369** (epmd),  **25672** (Erlang distribution)  **5672**, **5671** (AMQP 0-9-1 without and with TLS)  **15672** (if management plugin is enabled)  **61613**, **61614** (if STOMP is enabled)  **1883**, **8883** (if MQTT is enabled) |

安装Zookeeper

下载Zookeeper，推荐稳定版本

|  |
| --- |
| wget http://mirrors.shu.edu.cn/apache/zookeeper/stable/zookeeper-3.4.10.tar.gz |

是

安装Dubbo

下载Dubbo，目前如果需要最新版本需要自行到git上下载源码进行编译

|  |
| --- |
| wget https://codeload.github.com/apache/incubator-dubbo/legacy.tar.gz/master |

安装

安装

安装

安装

安装

安装

安装

安装

安装