

## 修改hostname

之前安装Centos时候，没有怎么注意，将自己的名字设置为hostname，总感觉好别扭，很不习惯，如是就想把hostname改一下，用到hostname命令。  
首先用hostname命令查看当前的主机名，核对一下当前的主机名。

```
[root@shiwei anker]# hostname  
shiwei
```

修改hostname过程：

(1) 修改/etc/hosts文件，将包含当前的hostname改为新的hostname。我将shiwei改为rabbit。

补充一下：

/etc这个目录用来存放所有的系统管理所需要的配置文件和子目录。进入后可以看到很多配置文件

/etc/hosts文件的作用相当于DNS，提供IP地址hostname的对应。

(2) 修改/etc/sysconfig/network文件，将包含当前的hostname改为新的hostname。我将shiwei改为rabbit。

```
1 NETWORKING=yes  
2 HOSTNAME=rabbit
```

(3) 使用命令hostname更改一下，命令为：hostname 新的主机名，然后用hostname名称查看是否修改成功。

```
[root@shiwei anker]# hostname rabbit  
[root@shiwei anker]# hostname  
rabbit
```

重新登录shell，发现hostname名字已经改变：

```
[anker@rabbit ~]$
```

## 修改服务器密码

passwd

## 安装jdk

1、首先先到<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>下载JDK，我下载的版本是jdk-8u181-linux-x64.tar.gz

## 2、配置目录

先新建一个目录：

```
cd /usr/local
```

```
mkdir java
```

```
cd java
```

```
mkdir jdk
```

```
cd jdk
```

## 3、下载完jdk后，通过Xftp 6上传此压缩包到云服务器上的指定目录下解压

```
cd /usr/local/java/jdk
```

## 4、配置环境变量

输入：vim/etc/profile

文件最后添加：

```
#set java environment
```

```
export JAVA_HOME=/usr/local/java/jdk/jdk1.8.0_181
```

```
export JRE_HOME=/usr/local/java/jdk/jdk1.8.0_181/jre
```

```
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
```

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

## 5、保存(:wq)后执行：

```
source /etc/profile
```

## 6、验证安装

```
java -version
```

```
java version "1.8.0_181"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

## 安装Tomcat

1、首先先到<http://tomcat.apache.org>下载Tomcat(apache-tomcat-8.5.24)

## 2、配置目录

```
cd /usr/local/java
```

```
mkdir tomcat
```

建好后再次用Xftp6传到当前的/usr/local/java/tomcat文件夹下解压

```
tar -zxvf apache-tomcat-8.5.24.tar.gz
```

此时tomcat文件夹下生成一个apache-tomcat-8.5.24文件夹

### 3、配置环境

进入到上面的apache-tomcat-8.5.24的bin文件夹下：

```
cd /usr/local/java/tomcat/tomcat8/bin
```

修改apache-tomcat-8.5.24文件夹名为tomcat8：

```
mv apache-tomcat-8.5.24 tomcat8
```

### 4、写入配置

```
vim setclasspath.sh
```

在文件最后写入

```
export JAVA_HOME=/usr/local/java/jdk/jdk1.8.0_181
```

```
export JRE_HOME=/usr/local/java/jdk/jdk1.8.0_181/jre
```

### 5、保存(:wq)后验证执行

在 cd /usr/local/java/tomcat/tomcat8/bin目录下执行

```
./startup.sh
```

```
[root@wennerfamilies_fukeban tomcat8]# ./bin/startup.sh
Using CATALINA_BASE:   /usr/local/java/tomcat/tomcat8
Using CATALINA_HOME:   /usr/local/java/tomcat/tomcat8
Using CATALINA_TMPDIR: /usr/local/java/tomcat/tomcat8/temp
Using JRE_HOME:         /usr/local/java/jdk/jdk1.8.0_181
Using CLASSPATH:        /usr/local/java/tomcat/tomcat8/bin/bootstrap.jar:/usr/local/java/tomcat/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

### 6、开放端口

启动好服务后，我们还需要开放 8080 端口。注：HTTP 协议默认使用 80 端口，而 Tomcat 默认使用 8080 端口。

```
/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT  开放8080端口
```

```
/etc/rc.d/init.d/iptables save  保存设置
```

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

```
[root@wennerFamles_fukeban bin]# /sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
[root@wennerFamles_fukeban bin]# /etc/rc.d/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@wennerFamles_fukeban bin]# /etc/rc.d/init.d/iptables restart
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Unloading modules: [ OK ]
iptables: Applying firewall rules: [ OK ]
```

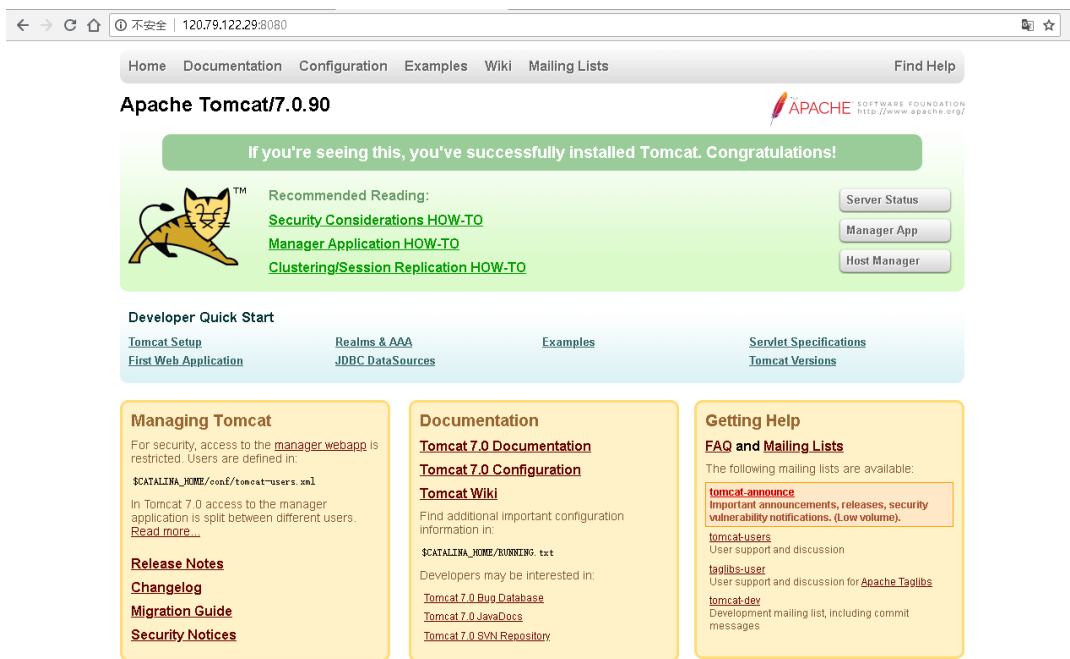
## 7、测试

上面的配置完成后，检查一下 Tomcat 的运行情况。执行命令：**ps -ef | grep tomcat**

```
[root@wennerFamles_fukeban bin]# ps -ef|grep tomcat
root      5414      1  0 Aug06 ?        00:00:26 /usr/local/java/jdk1.8.0_181/bin/java -Djava.util.logging.config.file=/usr/local/java/to
mcat/apache-tomcat-7.0.90/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKey
Size=2048 -Dignore.endorsed.dirs= -classpath /usr/local/java/tomcat/apache-tomcat-7.0.90/bin/bootstrap.jar:/usr/local/java/tomcat/apache-tom
cat-7.0.90/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/java/tomcat/apache-tomcat-7.0.90 -Dcatalina.home=/usr/local/java/tomcat/apache-tom
cat-7.0.90 -Djava.io.tmpdir=/usr/local/java/tomcat/apache-tomcat-7.0.90/temp org.apache.catalina.startup.Bootstrap start
root      6342  6157  0 10:46 pts/0    00:00:00 grep tomcat
```

接着，打开浏览器，测试下。本地机器测试则在地址栏上输入

<http://localhost:8080>；若是远程机器，则输入 <http://远程服务器IP:8080> 来查看结果。



## 安装mysql

**yum install mysql-server**

```

[root@wennerFamles_fukeban ~]# yum install mysql-server
Loaded plugins: fastestmirror
Setting up Install Process
Determining fastest mirrors
base                                                    | 3.7 kB    00:00
base/primary_db                                         | 4.7 MB    00:00
epel                                                    | 3.2 kB    00:00
epel/primary                                             | 3.2 MB    00:00
epel                                                    | 3.2 MB    12517/12517
extras                                                  | 3.4 kB    00:00
extras/primary_db                                       | 25 kB     00:00
updates                                                 | 3.4 kB    00:00
updates/primary_db                                     | 667 kB    00:00
Resolving Dependencies
--> Running transaction check
--> Package mysql-server.x86_64 0:5.1.73-8.el6_8 will be installed
--> Processing Dependency: mysql = 5.1.73-8.el6_8 for package: mysql-server-5.1.73-8.el6_8.x86_64
--> Processing Dependency: perl-DBI for package: mysql-server-5.1.73-8.el6_8.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: mysql-server-5.1.73-8.el6_8.x86_64
--> Processing Dependency: perl(DBI) for package: mysql-server-5.1.73-8.el6_8.x86_64
--> Running transaction check
--> Package mysql.x86_64 0:5.1.73-8.el6_8 will be installed
--> Package perl-DBD-MySQL.x86_64 0:4.013-3.el6 will be installed
--> Package perl-DBI.x86_64 0:1.609-4.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

```

## 查看安装

我们可以使用 rpm 命令来查看 mysql 的安装情况

**rpm -qa | grep mysql\***

**whereis mysql**

```

[root@wennerFamles_fukeban ~]# rpm -qa|grep mysql*
mysql-5.1.73-8.el6_8.x86_64
mysql-libs-5.1.73-8.el6_8.x86_64
mysql-server-5.1.73-8.el6_8.x86_64
[root@wennerFamles_fukeban ~]# whereis mysql
mysql: /usr/bin/mysql /usr/lib64/mysql /usr/share/mysql /usr/share/man/man1/mysql.1.gz

```

## 启动服务

安装好了，那我们就试着启动一下 mysql 服务好了：执行命令

**service mysqld start**

注意：是 **mysqld**，而不是 **mysql**。

```
[root@WennerFamles_fukeban ~]# service mysqld start
Initializing MySQL database: WARNING: The host 'WennerFamles_fukeban' could not be looked up with resolveip.
This probably means that your libc libraries are not 100 % compatible
with this binary MySQL version. The MySQL daemon, mysqld, should work
normally with the exception that host name resolving will not work.
This means that you should use IP addresses instead of hostnames
when specifying MySQL privileges !
Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h WennerFamles_fukeban password 'new-password'

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

Starting mysqld: [ OK ]
```

## 测试mysql

mysql 的服务已经启动好了，那么我们来测试一下，它是否能正确运行了？启动好后，我们来进入 mysql 中试着操作一下。执行命令：**mysql**

```
[root@WennerFamles_fukeban ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \q.
Your MySQL connection id is 2
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> \q
Bye
```

注释：\q 命令指关闭连接，退出 mysql

## 配置mysql

正确安装后，我们需要对 mysql 进行一些测试，方便我们日后的工作。

### 开机自启

我们首先来设置 mysql 开机自启。为什么要设置呢？因为每次关机重启后，mysql 服务都会被停止，处于 stop 状态，这时要使用 mysql，就得执行 `service mysqld start` 命令，比较繁琐，所以设置一下开机自启，可以减少麻烦。

我们利用 `chkconfig` 命令来设置 mysql 开机自启：执行命令

### `chkconfig mysqld on`

```
[root@WennerFamles_fukeban ~]# chkconfig mysqld on
[root@WennerFamles_fukeban ~]# chkconfig list
[root@WennerFamles_fukeban ~]# chkconfig --list
```

aegis	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
agentwatch	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
auditd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
blk-availability	0:off	1:on	2:on	3:on	4:on	5:on	6:off	6:off
crond	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
ecs_mq-service	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
eni-service	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
ip6tables	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
iptables	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
irqbalance	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
iscsi	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
iscsid	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
lvm2-monitor	0:off	1:on	2:on	3:on	4:on	5:on	6:off	
mdmonitor	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
multipathd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
mysqld	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
netconsole	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
netfs	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
network	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
nscd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
ntpd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
ntpddate	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
postfix	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
rdisc	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
restorecond	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
rsyslog	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
saslauthd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
sshd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
sysstat	0:off	1:on	2:on	3:on	4:on	5:on	6:off	
udev-post	0:off	1:on	2:on	3:on	4:on	5:on	6:off	

通过 `chkconfig --list` 命令可以看到各服务的情况，从中可以看到 mysql 的情况：运行级别为2、3、4、5的情况下都是on（开启）的状态。

1. [各状态的解释]
2. 等级0表示：表示关机
3. 等级1表示：单用户模式
4. 等级2表示：无网络连接的多用户命令行模式
5. 等级3表示：有网络连接的多用户命令行模式
6. 等级4表示：不可用
7. 等级5表示：带图形界面的多用户模式
8. 等级6表示：重新启动

### 开启 3306 端口

对于 mysql，我们知道一般都是使用 3306 端口。在我们 Java 程序的数据库连接中，可以看到 jdbc:mysql://localhost:3306/dbname 的语句。因此，在 Linux 下，我们需要开放 3306 端口，以便本机和外部访问 mysql。执行命令：

```
[root@VM_19_120_centos bin]# /sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT # 开放 3306 端口
[root@VM_19_120_centos bin]# /etc/rc.d/init.d/iptables save # 保存配置
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@VM_19_120_centos bin]#
```

### 修改密码

一般而言，我们都会启动密码认证。因此我们给用户设置一下密码。进入 mysql，执行命令：**update user set password=password('密码') where user='用户名'**；以此来给用户设置密码。



```

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;

+-----+
| Tables_in_mysql |
+-----+
| columns_priv     |
| db               |
| event            |
| func             |
| general_log      |
| help_category    |
| help_keyword     |
| help_relation    |
| help_topic       |
| innodb_index_stats |
| innodb_table_stats |
| ndb_binlog_index |
| plugin           |
| proc             |
| procs_priv       |
| proxies_priv     |
| servers          |
| slave_master_info |
| slave_relay_log_info |
| slave_worker_info |
| slow_log         |
| tables_priv      |
| time_zone        |
| time_zone_leap_second |
| time_zone_name   |
| time_zone_transition |
| time_zone_transition_type |
| user             |
+-----+
28 rows in set (0.00 sec)

mysql> update user set password=password('root') where user='root';
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0

```

## 配置远程登录

有一种需要，叫远程登录mysql。因此，我们也配置一下 mysql 的远程登录。该设置，同样是需要先进入 mysql 数据库，然后再执行命令 grant all privileges on \*.\* to 'root'@'%' identified by 'root' with grant option;

```

mysql> grant all privileges on *.* to 'root'@'%' identified by 'root' with grant option;
Query OK, 0 rows affected (0.00 sec)

```

## 命令详解：

# 将所有权限赋予给 root 用户，允许其进行远程登录

grant all privileges on \*.\* to 'root'@'%' identified by 'root' with grant option;

参数解释：

① all privileges：赋予所有的权限给指定用户，也可以替换为赋予某一具体的

权限

② `*`: 所赋予的权限是针对全部数据库的全部表。若指定某一具体数据库的具体表，如:thinkphp\_user

③ `root`: 表示给root用户授权

④ `'%'`: 表示不限制连接的 IP。若想只允许某特定的 IP 进行远程登录，可以改为具体 IP。

⑤ `by 'root'`: root 用户的密码

设置好远程登录后，我们测试一下是否可以远程登录。在本地机器执行远程登录命令如下：

# 连接时需要验证密码

`mysql -h 远程机器IP地址 -P 3306 -u 用户名 -p`

# 在登录的时候直接写上密码，p 和密码不能分开

`mysql -h 远程机器IP地址 -P 3306 -u 用户名 -p密码`

`mysql -h 47.104.223.47 -P 3306 -u root -proot`