# 前言

Tsconfig.msc

本文档所有软件库均为基于源码编译安装

Sudo curl –v –X INPUT 222.249.163.50

# 1.JDK:

1、查看安装java的地方：  
[root@ubuntu:~$](mailto:root@ubuntu:~$) cd /usr/lib/jvm  
[root@ubuntu:~$](mailto:root@ubuntu:~$) ls -l  
[root@ubuntu:~$](mailto:root@ubuntu:~$) rm -fr \*

2、删除原有的JDK   
[root@ubuntu:~$](mailto:root@ubuntu:~$) cd /usr/share/java/  
[root@ubuntu:~$](mailto:root@ubuntu:~$) ls -l  
[root@ubuntu:~$](mailto:root@ubuntu:~$) rm -fr /usr/share/java/\*

3、删去原有链接   
到/usr/bin目录下，把原用的java、javac文件删除:  
[root@ubuntu:~$](mailto:root@ubuntu:~$) cd /usr/bin  
[root@ubuntu:~$](mailto:root@ubuntu:~$) ls -l  
[root@ubuntu:~$](mailto:root@ubuntu:~$) rm -rf java   
[root@ubuntu:~$](mailto:root@ubuntu:~$) rm –rf javac

**三、安装jdk**  
到jdk-6u30-linux-i586.bin目录下，执行如下命令安装jdk  
[root@ubuntu:~$](mailto:root@ubuntu:~$) cd /home  
[root@ubuntu:~$](mailto:root@ubuntu:~$) ls -l  
[root@ubuntu:~$](mailto:root@ubuntu:~$) chmod 755 jdk-6u30-linux-i586.bin  
[root@ubuntu:~$](mailto:root@ubuntu:~$) ./jdk-6u30-linux-i586.bin  
按提示操作，完成安装后在当前目录下会看到jdk1.6.0\_30目录，到此JDK已经安装完成。

**四、配置环境变量:**

1)、修改/etc/profile文件  
执行如下命令  
#vi /etc/profile 或 sudo gedit /etc/profile

在/etc/profile最底部写入如下信息:  
#set java environment

JAVA\_HOME=/home/jdk1.6.0\_30  
PATH=$JAVA\_HOME/bin:$PATH  
CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  
export JAVA\_HOME  
export PATH  
export CLASSPATH

注意： 需要注销系统才能够生效。且永久有效。该文件只针对root用户有效，如果要使非root用户也生效，就要配置.bash\_profile和.bashrc文件。

2)、修改/home/.bash\_profile文件  
/etc/profile 是一个global config file，会影响系统全局用户，如果你只想对single user生效的话，可以修改 #vi ~/.bash\_profile。  
在.bash\_profile文件末尾加入：  
JAVA\_HOME=/home/jdk1.6.0\_30  
PATH=$JAVA\_HOME/bin:$PATH  
CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  
export JAVA\_HOME  
export PATH  
export CLASSPATH

注意：只会对当前shell窗口起作用，重启或者重新打开shell窗口时，环境变量就会丢失。

3)、修改.bashrc文件  
该文件位于当前用户目录下，每个用户都有该文件，通过如下命令可找到：  
[ubuntu@ubuntu:/home/hadoop$](mailto:ubuntu@ubuntu:/home/hadoop$) cd ~  
[ubuntu@ubuntu:~$](mailto:ubuntu@ubuntu:~$) pwd  
在.bashrc文件末尾加入：  
JAVA\_HOME=/home/jdk1.6.0\_30  
PATH=$JAVA\_HOME/bin:$PATH  
CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  
export JAVA\_HOME  
export PATH  
export CLASSPATH

注意：修改后必须使用 source ~/.bashrc 命令使其生效，且永久有效。

重新启动计算机或注销。

# 2.指定tomcat运行时JDK版本

2017年06月21日 10:44:46

阅读数：650

tomcat作为日常开发的web应用服务器，给开发测试带来了很多便利，tomcat的运行依赖JDK的支持，在安装JDK时经常会配置环境变量：JAVA\_HOME、CLASSPAT，且需要添加path变量。把应用部署到tomcat下时大多数不需要修改运行环境，但是有时候需要修改JDK的版本，从可以修改tomcat运行时的JDK版本，我们可以知道使用tomcat时不时必须要配置JDK环境变量，因为我们可以通过修改tomcat的文件，指定tomcat运行时的JDK版本。

下面说明如何修改tomcat运行时JDK版本

windows

1、解压下载的tomcat;

2、找到bin下的setclasspath.bat文件；在文件的开始出添加如下代码：

set JAVA\_HOME=D:\Program Files\Java\jdk7\jdk1.7.0\_51

set JRE\_HOME=D:\Program Files\Java\jdk7\jre7

上面的意思是设定JAVA\_HOME和JRE\_HOME的路径；

通过这里我们可以看出可以不设置JDK的环境变量；

linux

在bin下找到setclasspath.sh，在文件的最开始加入如下代码：

set JAVA\_HOME=/home/jdk/Java\jdk7\jdk1.7.0\_51

set JRE\_HOME=/home/jdk/Java\jdk7\jre7

修改了setclasspath文件之后，tomcat在启动时便使用设定的JDK。

但是为什么这样设置之后就可以呢？

我们都知道启动tomcat可以通过运行bin下的startup.bat，startup.bat会调用catalina.bat文件，而catalina.bat会调用setclasspath.bat文件来获取JAVA\_HOME和JRE\_HOME这两个环境变量的值，因此若要在tomcat启动时指向特定的JDK，则需在setclasspath.bat文件的开头处加上JAVA\_HOME和JRE\_HOME。

基于上面的运行方式，还有第二种修改方式，如下：

1、修改tomcat/bin/catalina.bat，增加 set JAVA\_HOME=D:\Program Files\Java\jdk7\jdk1.7.0\_51

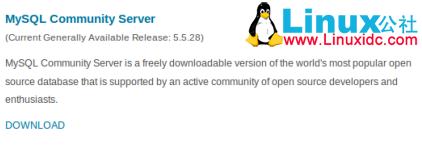
2、修改tomcat/bin/setclasspath.bat，同样增加

set JAVA\_HOME=D:\Program Files\Java\jdk7\jdk1.7.0\_51

set JRE\_HOME=D:\Program Files\Java\jdk7\jre7

# 3-1.MySql:

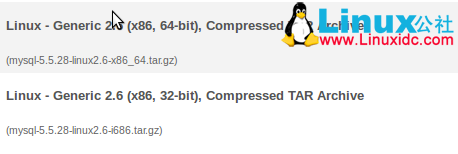
1.在浏览器中输入http://www.mysql.com/downloads/，进入mysql的下载中心，在这里有使用mysql开发的一些工具，包括mysql的驱动，数据库以及图形开发工具MySQL Workbench等。在这里我们选择MySQL Community Server链接，进入mysql 数据库服务器的下载：



如上，点击“MySQL Community Server”链接进入下载页面，接着选择myql的平台，选择Linux - Generic平台：



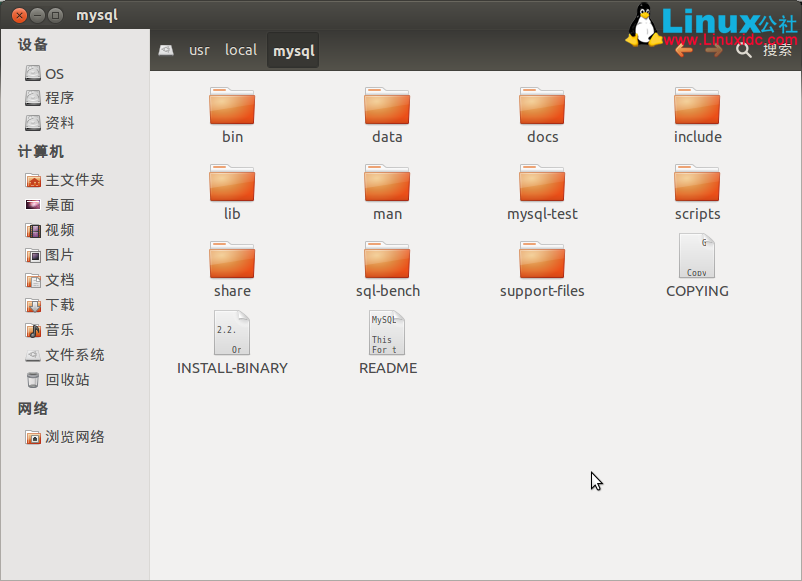
然后在其中选择二进制tar.gz文件形式的文件进行下载：



选择32位或64位的下载，这里选择32位的，下载后得到文件mysql-5.5.28-linux2.6-i686.tar.gz，然后将其解压，并重命名为mysql，使用mv命令将其移到/usr/local目录下：

sudo mv ~/下载/mysql  /usr/local

则得到的目录结构如下：



将上面的做好了后，我们现在就可以进入mysql的安装了，根据INSTALL-BINARY文件的描述,可知mysql默认的安装目录就是在/usr/local/mysql，这就是上面为什么我们要将其移动在/uer/local下的原因；如果在你的机器上以前安装有老板本的mysql，需要先将它的文件删除，同时注意删除老板本的etc/my.cnf文件和/etc/mysql目录，这两个文件控制的是mysql的一些配置属性。

按上面的记叙我可以知，先要创建的一个名为mysql的用户组和用户，来承载mysql数据库的运行,使用如下命令：

创建用户组：

sudo groupadd mysql

在创建的用户组中创建一个用户：

sudo useradd -r -g mysql mysql

这里使用sudo命令是确保以root权限执行此命令，如果你登入本机的用户是root用户，则直接的使用groupadd和useradd命令。

题外话：对应删除用户组及用户的命令是groupdel和userdel。

接着进入mysql目录，修改mysql目录的拥有者,为mysql用户：

进入目录：

cd /usr/local/mysql

修改目录的拥有者：

sudo chown -R mysql .

sudo chgrp -R mysql .

这里的点“.”代表的就是当前目录，选项-R表示递归当前目录及其子目录。

现在真正的进入主题，安装mysql，执行命令：

sudo scripts/mysql\_install\_db --user=mysql

其实，这一步正真的目地就是生成一些mysql数据库运行的系统数据库。

注意：在[Ubuntu](http://www.linuxidc.com/topicnews.aspx?tid=2) 12.04下安装mysql 5.5.28版本执行此命令时，会提示如下错误的信息：

./bin/mysqld: error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory

这说明还要安装一个libaio的依赖库，执行如下命令：

sudo apt-get install libaio-dev

安装完libaio后，继续执行“sudo  scripts/mysql\_install\_db --user=mysql”命令来进行安装。

执行完上面的命令后，其实就已经完成了mysql的安装，但为了数据库的安全，可以将mysql目录的拥有者改为root用户，并将生成的系统依赖数据赋给mysql用户,执行如下命令：

chown -R root .

chown -R mysql data

安装好mysql后，就可以试着启动它，使用如下命令：

sudo ./support-files/mysql.server start

同样重启和停止，只需要将上面命令的start改为restart或stop。

启动完mysql后，我们接着可以测试一下，使用“./bin/mysql”命令来进入mysql数据库的控制台，执行SQL命令。

为了数据库的安全我们需要为数据库访问设置密码，可以执行如下命令，将mysql的配置文件安装到/etc目录下：

sudo cp support-files/my-medium.cnf /etc/my.cnf

然后更改/etc/my.cnf文件，在[client]段下，取消password注释，并将your\_password改为你的密码。

这样设置好后，进入mysql的控制台，则需要使用如下命令：

sudo ./bin/mysql -uroot -p

最后：

上面介绍的这么多，总结mysql的安装实际上只需执行“sudo ./scripts/mysql\_install\_db --user=系统用户名“这一条命令，这里设置的系统用户可以直接的使用当前登入系统用户，在linux中查看当前用户的命令是:

who am i

同时，如果不想将mysql安装到/usr/local/mysql下，可以使用如下命令将你指定的目录与/usr/local/mysql目录关联：

ln -s <mysql目录> /usr/local/mysql

或修改<mysql目录>/support-files/mysql.server文件，为其指定basedir和datadir两个参数，如：

basedir=<mysql目录>

datadir=<mysql目录>/data

在启动mysql时，还可以使用如下命令：

sudo ./bin/mysqld\_safe --user=mysql &

启动后，修改密码也可以使用如下命令：

sudo ./bin/mysqladmin -u root -p password '新密码'

提示输入旧数据库的密码后,密码才变更为新密码。

为了确保数据库中支持中文，可以修改/etc/my.cnf文件：

在[client]段下添加

default-character-set = utf8

在[mysqld]段下添加

character\_set\_server = utf8

安装好mysql后，就可以安装mysql的图形化客服端MySQL Workbench了，可以到http://www.mysql.com/downloads/workbench/下选择合适的版本下载，在上面有[Ubuntu](http://www.linuxidc.com/topicnews.aspx?tid=2" \t "_blank" \o "Ubuntu)下的版本，下载后的文件为BED格式，双击后可以直接安装十分方便。

如果在安装好MySQL Workbench后，出现运行mysql.server start无法启动的情况，可以查看一下，是否存在/etc/mysql目录，如果有要将其删除；同时还要检查/etc/my.cnf文件的内容。

更多Ubuntu相关信息见[Ubuntu](http://www.linuxidc.com/topicnews.aspx?tid=2" \t "_self" \o "Ubuntu) 专题页面 <http://www.linuxidc.com/topicnews.aspx?tid=2>

# 3-2.MySql:

首先，下载二进制版本的mysql包: <http://cdn.mysql.com/Downloads/MySQL-5.6/mysql-5.6.10-linux-glibc2.5-x86_64.tar.gz>

因为Ubuntu Server上 wget比较慢，所以我是直接在windows用迅雷下载好，然后WinSCP上传到服务器的，看各自喜好了。

上正题。

包放在~/Download目录下，全部安装命令如下：

1、解压tar.gz

tar –xzf mysql-5.6.10-linux-glibc2.5-x86\_64.tar.gz

2、重命名解压的文件夹

mv mysql-5.6.10-linux-glibc2.5-x86\_64 mysql

3、将mysql文件夹移动到/usr/local目录下

sudo mv mysql /usr/local

4、进入mysql目录

cd /usr/local/mysql

5、增加mysql用户组

sudo groupadd mysql

6、增加mysql用户

sudo useradd -r -g mysql mysql

7、将mysql文件夹own及grp变更为mysql

sudo chown -R mysql .

sudo chgrp -R mysql .

8、执行mysql安装脚本

sudo scripts/mysql\_install\_db --user=mysql

（若未安装libaio包，会有一个报错提示，安装libaio-dev后，再运行脚本即可）

sudo apt-get install libaio-dev

9、将目录权限变更回来，仅保留data目录为mysql用户

sudo chown -R root .

10、将mysql配置文件拷贝到etc目录(全局配置)

注意：5.6版本的默认配置文件名称由原先的my-medium变更为了my-default。

sudo cp support-files/my-default.cnf /etc/my.cnf

11、启动mysql

sudo bin/mysqld\_safe --user=mysql &

12、初始化mysql root用户密码

sudo bin/mysqladmin -u root password '密码文字'

13、复制mysql.server脚本到/etc/init.d(初始化服务，有些人喜欢改成mysqld，在这里改就可以)

sudo cp support-files/mysql.server /etc/init.d/mysql.server

14、查看mysql运行状态

sudo service mysql.server status  
如果运行正常，会显示 MySQL running。

如果显示 not running，应该是前面没有启动服务，可直接用service mysql.server start启动

sudo service mysql.server [status|start|stop]

15、让mysql开机启动[defaults]，取消开机启动[remove]

sudo update-rc.d -f mysql.server defaults  [remove]

16、将mysql/bin/mysql命令加入到用户命令中，或将mysql/bin目录加入path

加入用户命令：

sudo ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql

加入环境变量：

export PATH=$PATH:/usr/local/mysql/bin

17、允许root用户远程登录

1>进入mysql: mysql –u root –p

2>改变数据库: use mysql;

3>从任意主机登录: grant all privileges on \*.\* to root@"%" identified by "密码文字" with grant option;

4>从指定主机登录: grant all privileges on \*.\* to root@"192.168.1.101" identified by "passw0rd" with grant option;

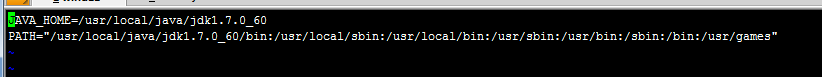
5>授权生效: flush privileges;

6>查看host为%授权是否添加: select \* from user;

18、找个客户端去试试吧。

# 3.Tomcat

需要在etc/environment添加：



# 4.Apache

1. 首先要安装apr apr-utils pcre, 分别用源码编译安装【./configure;make;make install;】

2.在执行./configure时，出现报错信息configure: error: no acceptable C compiler found in $PATH

解决方法： sudo apt-get install gcc【安装gcc编译套件】

3.若提示没有make命令 则要安装：

Sudo apt-get install make

安装APR(Apache Portable Runtime )  
下载：[http://apr.apache.org/download.cgi](http://apr.apache.org/download.cgi" \t "_blank)

#cd /tmp/52lamp/ //源码存放位置  
#tar -zxvf apr-1.4.2.tar.gz //unzip -o apr-1.4.2.zip  
#cd apr-1.4.2  
#./configure  
#make  
#make install

下载：http://download.chinaunix.net/download/0001000/472.shtml  
#tar -zxvf apr-util-1.3.9.tar.gz  
#cd apr-util-1.3.9  
#./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr  
#make  
#make install

./configure仍提示APR-util not found，增加--with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util后出现  
configure: error: pcre-config for libpcre not found. PCRE is required and available from <http://pcre.org/>

#./configure –help | grep pcre  
--with-pcre=PATH        Use external PCRE library

下载：[http://sourceforge.net/projects/pcre](http://sourceforge.net/projects/pcre" \t "_blank)  
#unzip -o pcre-8.10.zip  
#cd pcre-8.10  
#./configure --prefix=/usr/local/pcre  
#make  
#make install

安装pcre报错“You need a C++ compiler for C++ support.”解决办法

解决办法：

执行sudo apt-get install build-essential

Ubuntu默认并不提供C/C＋＋的编译环境，因此还需要通过上述命令进行杆状即可。

继续安装Apache/httpd，./configure 时加上参数 --with-apr=/usr/local/apr/ --with-apr-util=/usr/local/apr-util/ --with-pcre=/usr/local/pcre，这个问题就解决了

启动，重启和停止  
/usr/local/apache/bin/apachectl -k start     （不加参数k也行）   
/usr/local/apache/bin/apachectl -k restart  
/usr/local/apache/bin/apachectl -k stop

复制Apache启动文件到usr/sbin里面  前面为你安装的apache的目录。

#cp /usr/local/apache/bin/apachectl   /usr/sbin/

启动Apache时就可以简单的输入，而不用输入很长一段了。

 sudo apachectl start 就可以启动了

设置Apache开机自启动

~~#vi /etc/init.d//rc.d/rc.local~~

~~增加一行 /sbin/apachectl start~~

# 5.整合apache+tomcat

解压安装

1. tar -zxvf tomcat-connectors-1.2.40-src.tar.gz
2. cd tomcat-connectors-1.2.40-src
3. cd **native**
4. ./configure --with-apxs=/usr/local/apache2/bin/apxs  --with-java-home=/usr/jdk1.6
5. make
6. make install
7. cp apache-2.0/mod\_jk.so /usr/local/apache2/modules/

**/usr/local/apache2/conf/httpd.conf**  
增加  
**include /usr/local/apache2/conf/mod-jk.conf**

新建mod-jk.conf配置文件

**Java代码  [收藏代码](javascript:void())**

1. #加载mod\_jk模块，注意名称与modules目录下的文件名保持一致.
2. LoadModule jk\_module /usr/local/apache2/modules/mod\_jk.so
4. #指定workers.properties文件目录.
5. JkWorkersFile /usr/local/apache2/conf/workers.properties
7. #指定哪些请求由tomcat处理,controller为workers.properties文件里指定的负载控制器.
8. JkMount /\* controller
9. #JkMount /\*.html controller
10. #JkMount /\*.jsp controller
11. #JkMount /\*.**do** controller
12. #JkMount /servlet/\* controller
13. #JkUnMount /index.html controller

新建workers.properties配置文件

**Java代码  [收藏代码](javascript:void())**

1. worker.list = controller      #不指定报错Could not find a worker **for** worker name=controller
2. #========tomcat1========
3. worker.tomcat1.port=8019
4. worker.tomcat1.host=localhost
5. worker.tomcat1.type=ajp13
6. worker.tomcat1.lbfactor=1
7. #========tomcat2========
8. worker.tomcat2.port=8029
9. worker.tomcat2.host=localhost
10. worker.tomcat2.type=ajp13
11. worker.tomcat2.lbfactor=1
12. #========tomcat3========
13. worker.tomcat3.port=8039
14. worker.tomcat3.host=localhost
15. worker.tomcat3.type=ajp13
16. worker.tomcat3.lbfactor=3
17. #========tomcat4========
18. worker.tomcat4.port=8049      #指定tomcat2服务器AJP的端口,默认为8009
19. worker.tomcat4.host=localhost #指定tomcat2服务器IP或域名
20. worker.tomcat4.type=ajp13     #指定tomcat2与apache AJP通讯协议
21. worker.tomcat4.lbfactor=3     #指定负载平衡因数,只有启用了负载平衡才有用.
23. #========controller负载平衡控制器========
24. worker.controller.type=lb                          #指定controller类型
25. worker.controller.balanced\_workers=tomcat1,tomcat2,tomcat3,tomcat4 #指定负载平衡的tomcat
26. worker.controller.sticky\_session=**true**              #指定是否粘性session
27. worker.controller.sticky\_session\_force=**false**
28. worker.connection\_pool\_size=3000
29. worker.connection\_pool\_minsize=50
30. worker.connection\_pool\_timeout=50000
31. # session配置说明:
32. #当sticky\_session,sticky\_session\_force都为**true**时不复制session,
33. #sticky\_session\_force=**false**指集群中某台服务器多次请求没有响应,则转发到其它服务器处理,
34. #sticky\_session=**false**不使用粘性session,同时配置不复制session时,注意转发请求后可能会找不到原来的session.

集群设置：

每个tomcat中conf中server.xml设置如下：

<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">

<!--For clustering, please take a look at documentation at:

/docs/cluster-howto.html (simple how to)

/docs/config/cluster.html (reference documentation) -->

<!--<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>-->

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"

channelSendOptions="8">

<Manager className="org.apache.catalina.ha.session.DeltaManager"

expireSessionsOnShutdown="false"

notifyListenersOnReplication="true"/>

<Channel className="org.apache.catalina.tribes.group.GroupChannel">

<Membership className="org.apache.catalina.tribes.membership.McastService"

mcastBindAddress="192.168.159.142" ---本机地址

address="224.0.0.1" ------组播地址

port="45564"

frequency="500"

dropTime="3000"/>

<Receiver className="org.apache.catalina.tribes.transport.nio.NioReceiver"

address="192.168.159.142" ---本机地址

port="4000"

autoBind="100"

selectorTimeout="5000"

maxThreads="6"/>

<Sender className="org.apache.catalina.tribes.transport.ReplicationTransmitter">

<Transport className="org.apache.catalina.tribes.transport.nio.PooledParallelSender"/>

</Sender>

<Interceptor className="org.apache.catalina.tribes.group.interceptors.TcpFailureDetector"/>

<Interceptor className="org.apache.catalina.tribes.group.interceptors.MessageDispatch15Interceptor"/>

</Channel>

<Valve className="org.apache.catalina.ha.tcp.ReplicationValve"

filter=""/>

<Valve className="org.apache.catalina.ha.session.JvmRouteBinderValve"/>

<Deployer className="org.apache.catalina.ha.deploy.FarmWarDeployer"

tempDir="/tmp/war-temp/"

deployDir="/tmp/war-deploy/"

watchDir="/tmp/war-listen/"

watchEnabled="false"/>

<ClusterListener className="org.apache.catalina.ha.session.JvmRouteSessionIDBinderListener"/>

<ClusterListener className="org.apache.catalina.ha.session.ClusterSessionListener"/>

</Cluster>

备注：

1.开启组播地址：

Route add –net 224.0.0.0 netmask 240.0.0.0 dev eth0

2.tomcat启动参数：在catalina.sh添加：

JAVA\_OPTS="$JAVA\_OPTS -Djava.net.preferIPv4Stack=true"

3.在/etc/hosts中添加192.168.159.142 wind11 【wind11为hostname】

修改web应用里面WEB-INF目录下的web.xml文件，加入标签  
<distributable/>  
直接加在</web-app>之前就可以了  
做tomcat集群必须需要这一步，否则用户的session就无法正常使用。

<%@ page contentType="text/html; charset=GBK" %>

<%@ page import="java.util.\*" %>

<html><head><title>Cluster App Test</title></head>

<body>

Server Info:

<%

out.println(request.getLocalAddr() + " : " + request.getLocalPort()+"<br>");%>

<%

out.println("<br> ID " + session.getId()+"<br>");

// 如果有新的 Session 属性设置

String dataName = request.getParameter("dataName");

if (dataName != null && dataName.length() > 0) {

String dataValue = request.getParameter("dataValue");

session.setAttribute(dataName, dataValue);

}

out.println("<b>Session 列表</b><br>");

System.out.println("============================");

Enumeration e = session.getAttributeNames();

while (e.hasMoreElements()) {

String name = (String)e.nextElement();

String value = session.getAttribute(name).toString();

out.println( name + " = " + value+"<br>");

System.out.println( name + " = " + value);

}

%>

<form action="index.jsp" method="POST">

名称:<input type=text size=20 name="dataName">

<br>

值:<input type=text size=20 name="dataValue">

<br>

<input type=submit>

</form>

</body>

</html>

# 6.端口

自打2.4版本以后的Linux内核中， 提供了一个非常优秀的防火墙工具。这个工具可以对出入服务的网络数据进行分割、过滤、转发等等细微的控制，进而实现诸如防火墙、NAT等功能。  
一般来说， 我们会使用名气比较的大iptables等程序对这个防火墙的规则进行管理。iptables可以灵活的定义防火墙规则， 功能非常强大。但是由此产生的副作用便是配置过于复杂。一向以简单易用著称Ubuntu在它的发行版中，附带了一个相对iptables简单很多的防火墙配置工具：ufw。

ufw默认是没有启用的。也就是说， ubuntu中的端口默认都是开放的。使用如下命令启动ufw：  
$sudo ufw default deny  
$sudo ufw enable  
通过第一命令，我们设置默认的规则为allow， 这样除非指明打开的端口， 否则所有端口默认都是关闭的。第二个命令则启动了ufw。如果下次重新启动机器， ufw也会自动启动。  
对于大部分防火墙操作来说， 其实无非就是的打开关闭端口。如果要打开SSH服务器的22端口， 我们可以这样：  
$sudo ufw allow 22  
由于在/etc/services中， 22端口对应的服务名是ssh。所以下面的命令是一样的：  
$sudo ufw allow ssh  
现在可以通过下面命令来查看防火墙的状态了：  
$sudo ufw status  
Firewall loaded

To Action From  
– —— —-  
22:tcp ALLOW Anywhere  
22:udp ALLOW Anywhere  
我们可以看到， 22端口的tcp和udp协议都打开了。  
删除已经添加过的规则：  
$sudo ufw delete allow 22  
只打开使用tcp/ip协议的22端口：  
$sudo ufw allow 22/tcp  
打开来自192.168.0.1的tcp请求的80端口：  
$sudo ufw allow proto tcp from 192.168.0.1 to any port 22  
要关系防火墙：  
$sudu ufw disable

ubuntu下的ufw防火墙配置  
2009-06-26 23:47  
UFW防火墙是一个主机端的iptables类防火墙配置工具。这个工具的目的是提供给用户一个可以轻松驾驭的界面，就像包集成和动态检测开放的端口一样。  
在Ubuntu中安装UFW：  
目前这个包存在于Ubuntu 8.04的库中。  
sudo apt-get install ufw  
上面这行命令将把软件安装到您系统中。  
开启/关闭防火墙 (默认设置是’disable’)  
# ufw enable|disable  
转换日志状态  
# ufw logging on|off  
设置默认策略 (比如 “mostly open” vs “mostly closed”)  
# ufw default allow|deny  
许 可或者屏蔽某些入埠的包 (可以在“status” 中查看到服务列表［见后文］)。可以用“协议：端口”的方式指定一个存在于/etc/services中的服务名称，也可以通过包的meta-data。 ‘allow’ 参数将把条目加入 /etc/ufw/maps ，而 ‘deny’ 则相反。基本语法如下：  
# ufw allow|deny [service]  
显示防火墙和端口的侦听状态，参见 /var/lib/ufw/maps。括号中的数字将不会被显示出来。  
# ufw status  
［注意：上文中虽然没有使用 sudo，但是命令提示符号都是“#”。所以……你知道啥意思了哈。原文如此。──译者注］  
UFW 使用范例：  
允许 53 端口  
$ sudo ufw allow 53  
禁用 53 端口  
$ sudo ufw delete allow 53  
允许 80 端口  
$ sudo ufw allow 80/tcp  
禁用 80 端口  
$ sudo ufw delete allow 80/tcp  
允许 smtp 端口  
$ sudo ufw allow smtp  
删除 smtp 端口的许可  
$ sudo ufw delete allow smtp  
允许某特定 IP  
$ sudo ufw allow from 192.168.254.254  
删除上面的规则  
$ sudo ufw delete allow from 192.168.254.254  
——————————————  
我自己还用7.10呢，所以翻译的过程中上面步骤没经过试验。  
Ubuntu的名字都很别嘴，一直记不住：  
\* Ubuntu 6.06 LTS (Dapper Drake)  
\* Ubuntu 6.10 (Edgy Eft)  
\* Ubuntu 7.04 (Feisty Fawn)  
\* Ubuntu 7.10 (Gutsy Gibbon)  
\* Ubuntu 8.04 (Hardy Heron)

ubuntu 防火墙  
2010-01-14 11:02  
ufw是Ubuntu下的一个简易的防火墙配置工具，底层还是调用iptables来处理的，虽然功能较简单，但对桌面型应用来说比较实用，基本常用功能都有，使用也较为容易。  
==鱼漂(admin.net#163.com)原创，转载请注明==  
==http://www.eit.name==

1.安装  
sudo apt-get install ufw

2.启用  
sudo ufw enable  
sudo ufw default deny  
运行以上两条命令后，开启了防火墙，并在系统启动时自动开启。  
关闭所有外部对本机的访问，但本机访问外部正常。

3.开启/禁用  
sudo ufw allow|deny [service]  
打开或关闭某个端口，例如：  
sudo ufw allow smtp　允许所有的外部IP访问本机的25/tcp (smtp)端口  
sudo ufw allow 22/tcp 允许所有的外部IP访问本机的22/tcp (ssh)端口  
sudo ufw allow 53 允许外部访问53端口(tcp/udp)  
sudo ufw allow from 192.168.1.100 允许此IP访问所有的本机端口  
sudo ufw allow proto udp 192.168.0.1 port 53 to 192.168.0.2 port 53  
sudo ufw deny smtp 禁止外部访问smtp服务  
sudo ufw delete allow smtp 删除上面建立的某条规则

4.查看防火墙状态  
sudo ufw status

一般用户，只需如下设置：  
sudo apt-get install ufw  
sudo ufw enable  
sudo default deny  
以上三条命令已经足够安全了，如果你需要开放某些服务，再使用sudo ufw allow开启。

Ubuntu防火墙 UFW 设置简介  
2010-03-03 16:27  
1.安装  
sudo apt-get install ufw  
2.启用  
sudo ufw enable  
sudo ufw default deny  
运行以上两条命令后，开启了防火墙，并在系统启动时自动开启。关闭所有外部对本机的访问，但本机访问外部正常。  
3.开启/禁用  
sudo ufw allow|deny [service]  
打开或关闭某个端口，例如：  
sudo ufw allow smtp　允许所有的外部IP访问本机的25/tcp (smtp)端口  
sudo ufw allow 22/tcp 允许所有的外部IP访问本机的22/tcp (ssh)端口  
sudo ufw allow 53 允许外部访问53端口(tcp/udp)  
sudo ufw allow from 192.168.1.100 允许此IP访问所有的本机端口  
sudo ufw allow proto udp 192.168.0.1 port 53 to 192.168.0.2 port 53  
sudo ufw deny smtp 禁止外部访问smtp服务  
sudo ufw delete allow smtp 删除上面建立的某条规则  
4.查看防火墙状态  
sudo ufw status  
一般用户，只需如下设置：  
sudo apt-get install ufw  
sudo ufw enable  
sudo ufw default deny  
以上三条命令已经足够安全了，如果你需要开放某些服务，再使用sudo ufw allow开启。  
开启/关闭防火墙 (默认设置是’disable’)  
sudo ufw enable|disable  
转换日志状态  
sudo ufw logging on|off  
设置默认策略 (比如 “mostly open” vs “mostly closed”)  
sudo ufw default allow|deny  
许可或者屏蔽端口 (可以在“status” 中查看到服务列表)。可以用“协议：端口”的方式指定一个存在于/etc/services中的服务名称，也可以通过包的meta-data。 ‘allow’ 参数将把条目加入 /etc/ufw/maps ，而 ‘deny’ 则相反。基本语法如下：  
sudo ufw allow|deny [service]  
显示防火墙和端口的侦听状态，参见 /var/lib/ufw/maps。括号中的数字将不会被显示出来。  
sudo ufw status  
UFW 使用范例：  
允许 53 端口  
$ sudo ufw allow 53  
禁用 53 端口  
$ sudo ufw delete allow 53  
允许 80 端口  
$ sudo ufw allow 80/tcp  
禁用 80 端口  
$ sudo ufw delete allow 80/tcp  
允许 smtp 端口  
$ sudo ufw allow smtp  
删除 smtp 端口的许可  
$ sudo ufw delete allow smtp  
允许某特定 IP  
$ sudo ufw allow from 192.168.254.254  
删除上面的规则  
$ sudo ufw delete allow from 192.168.254.254

vi /etc/sysconfig/iptables   
  
-A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT（允许80端口通过防火墙）   
-A INPUT -m state –state NEW -m tcp -p tcp –dport 3306 -j ACCEPT（允许3306端口通过防火墙）   
特别提示：很多网友把这两条规则添加到防火墙配置的最后一行，导致防火墙启动失败，正确的应该是添加到默认的22端口这条规则的下面   
  
添加好之后防火墙规则如下所示： 

复制代码

代码如下:

# Firewall configuration written by system-config-firewall   
# Manual customization of this file is not recommended.   
\*filter   
:INPUT ACCEPT [0:0]   
:FORWARD ACCEPT [0:0]   
:OUTPUT ACCEPT [0:0]   
-A INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT   
-A INPUT -p icmp -j ACCEPT   
-A INPUT -i lo -j ACCEPT   
-A INPUT -m state –state NEW -m tcp -p tcp –dport 22 -j ACCEPT   
-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 3306 -j ACCEPT   
-A INPUT -j REJECT –reject-with icmp-host-prohibited   
-A FORWARD -j REJECT –reject-with icmp-host-prohibited   
COMMIT

/etc/init.d/iptables restart   
#最后重启防火墙使配置生效

Ubuntu中iptables：

sudo iptables -I INPUT -p tcp --dport 6379 -j ACCEPT

## [linux/ubuntu 端口开放](http://www.cnblogs.com/toSeeMyDream/p/6971757.html)

*在ubuntu下面开放端口好像主要有两种方法，一种是ubuntu自带的防火墙，一种是iptables，这里我们主要使用iptables。本文的系统版本为ubuntu14.04和ubuntu16.04*

## ****安装iptables****

一般情况下，ubuntu安装好的时候，iptables会被安装上，没有安装上也没啥，一条命令就可以安装了。

|  |  |
| --- | --- |
| 1 | apt-get install iptables |

安装好了之后就可以添加规则了

|  |  |
| --- | --- |
| 1 | iptables -I INPUT -p tcp --dport 80 -j ACCEPT |

然后可以直接保存了

|  |  |
| --- | --- |
| 1 | iptables-save |

## ****持久化iptables****

上文那样做只是暂时保存了开放端口的信息，如果服务器重启，保存的信息就掉了，所以我们需要持久化一下端口信息，这里我们使用 iptables-persistent

### 安装iptables-persistent

|  |  |
| --- | --- |
| 1 | apt-get install iptables-persistent |

### 持久化信息

|  |  |
| --- | --- |
| 1 | service iptables-persistent save |

## 开放特定端口

#对1和102开放22端口

-A INPUT -p tcp -s 192.168.137.1,192.168.137.102 --dport 22 -j ACCEPT

#对所有IP禁止访问22端口

-A INPUT -p tcp --dport 22 -j REJECT

# 7添加字体

1. 首先获得一套“微软雅黑”字体库（Google一下一大把），包含两个文件msyh.ttf（普通）、msyhbd.ttf（加粗）；  
2. 在/usr/share/fonts目录下建立一个子目录，例如win，命令如下：  
    # mkdir /usr/share/fonts/win  
3. 将msyh.ttf和msyhbd.ttf复制到该目录下，例如这两个文件放在/root/Desktop下，使用命令：  
    # cd /root/Desktop  
    # cp msyh.ttf msyhbd.ttf  /usr/share/fonts/win/  
4. 建立字体索引信息，更新字体缓存：  
    # cd /usr/share/fonts/win  
    # mkfontscale**（如果提示 mkfontscale: command not found，则需要安装# sudo apt-get install ttf-mscorefonts-installer**)  
    # mkfontdir  
    # fc-cache -fv（如果提示 fc-cache: command not found，则需要安装# sudo apt-get install fontconfig)  
    至此，字体已经安装完毕！

[Ubuntu字符集设置](http://blog.chinaunix.net/uid-23381466-id-58679.html)

分类： LINUX

2010-03-26 11:41:13

文件从window上传到ubuntu后会显示乱码，原因是因为ubuntu环境设置默认是utf8，windows默认都是GBK，对拷的文件转化后就会乱码。只好手动将ubuntu环境设置为GBK   
1、首先设置sudo gedit /var/lib/locales/supported.d/local   
添加一行 zh\_CN.GBK GBK   
2、sudo locale-gen   
3、修改/etc/environment   
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11:/usr/games"  
LANGUAGE="zh\_CN:zh:en\_US:en"  
GST\_ID3\_TAG\_ENCODING=GBK  
LANG=zh\_CN.GBK  
LC\_CTYLE=zh\_CN.GBK  
CLASSPATH="/usr/lib/jvm/java-1.5.0-sun/lib"  
JAVA\_HOME="/usr/lib/jvm/java-1.5.0-sun"  
LC\_ALL="zh\_CN.GBK"  
然后重启X即可

# 8.Ubuntu14.04配置固定IP

1. 首先用root用户登陆，然后输入你root的密码。如下图：
2. 然后编辑interfaces文件，该文件位于/etc/network/下，执行如下命令：

vim /etc/network/interfaces，回车，如图：

1. 添加：

# interfaces(5) file used by ifup(8) and ifdown(8)auto loiface lo inet loopback这以下的是为添加上去该文件上的内容：# The primary network interfaceauto eth0iface eth0 inet staticaddress 192.168.1.100netmask 255.255.255.0gateway 192.168.1.1

如图：

1. 添加完毕后，保存退出，接下来就是配置DNS解析了，DNS解析文件再/etc/resolv.conf。执行：vim /etc/resolv.conf，如下图：
2. 再该文件下添加：

nameserver 192.168.1.1

nameserver 220.170.64.68

如图：

1. 改完上面，如果重启的话，还是会变为原来的样子，所以要让其永久改变，因此，执行： vim /etc/resolvconf/resolv.conf.d/base，如图：
2. 在里面添加

nameserver 192.168.1.1nameserver 220.170.64.68

如图：

1. 然后保存退出，到这里基本就是完成了，然后还是不能连上网，要将/etc/NetworkManager/目录下的system-connections删除，或者移动到其他目录下。
2. 最后执行/etc/NetworkManager/就可以连上网了。

# [Ubuntu 14.04 配置iptables防火墙](http://www.cnblogs.com/general0878/p/5757377.html)

一．

Ubuntu默认安装是没有开启任何防火墙的，为了服务器的安全，建议大家安装启用防火墙设置，这里推荐使用iptables防火墙.如果mysql启本地使用,可以不用打开3306端口.

#**whereis iptables**#查看系统是否安装防火墙可以看到:

iptables: /sbin/iptables /usr/share/iptables /usr/share/man/man8/iptables.8.gz #表示已经安装iptables

apt-get install iptables #如果默认没有安装，请运行此命令安装防火墙

# **iptables -L** #查看防火墙配置信息，显示如下:

Chain INPUT (policy ACCEPT)

target prot opt source destination

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

# **vi /etc/iptables/rules.v4**

添加以下内容(备注:80是指web服务器端口,3306是指MySQL数据库链接端口,22是指SSH远程管理端口.)

\*filter

:INPUT DROP [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:syn-flood - [0:0]

-A INPUT -i lo -j ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT

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

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

-A INPUT -p icmp -m limit --limit 100/sec --limit-burst 100 -j ACCEPT

-A INPUT -p icmp -m limit --limit 1/s --limit-burst 10 -j ACCEPT

-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j syn-flood

-A INPUT -j REJECT --reject-with icmp-host-prohibited

-A syn-flood -p tcp -m limit --limit 3/sec --limit-burst 6 -j RETURN

-A syn-flood -j REJECT --reject-with icmp-port-unreachable

COMMIT

# **iptables-restore < /etc/iptables/rules.v4** #使防火墙规则生效

#**vi /etc/network/if-pre-up.d/iptables** #创建文件，添加以下内容，使防火墙开机启动

#!/bin/bash

iptables-restore < /etc/iptables/rules.v4

# **chmod +x /etc/network/if-pre-up.d/iptables**#添加执行权限

# **iptables -L -n**查看规则是否生效.

二

可以通过来自<http://www.cnblogs.com/general0878/p/5757377.html>的方法去实现：

1、查看系统是否安装防火墙

sudo whereis iptables

出现如上提示表示已经安装iptables，如果没有安装，可以通过以下命令安装

sudo apt-get install iptables

2、查看防火墙的配置信息

sudo iptables -L

3、新建规则文件

mkdir /etc/iptables #先新建目录，本身无此目录

vi /etc/iptables/rules.v4

两条合起来的命令可以简化成以下写法

mkdir /etc/iptables & vi /etc/iptables/rules.v4

添加以下内容（备注：80是指web服务器端口，3306是指MySQL数据库链接端口，22是指SSH远程管理端口）

\*filter

:INPUT DROP [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:syn-flood - [0:0]

-A INPUT -i lo -j ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT

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

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

-A INPUT -p icmp -m limit --limit 100/sec --limit-burst 100 -j ACCEPT

-A INPUT -p icmp -m limit --limit 1/s --limit-burst 10 -j ACCEPT

-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j syn-flood

-A INPUT -j REJECT --reject-with icmp-host-prohibited

-A syn-flood -p tcp -m limit --limit 3/sec --limit-burst 6 -j RETURN

-A syn-flood -j REJECT --reject-with icmp-port-unreachable

COMMIT

4、使防火墙生效

iptables-restore < /etc/iptables/rules.v4

5、创建文件，添加以下内容，使防火墙开机启动

vi /etc/network/if-pre-up.d/iptables

#!/bin/bash

iptables-restore < /etc/iptables/rules.v4

6、添加执行权限

chmod +x /etc/network/if-pre-up.d/iptables

7、查看规则是否生效

iptables -L -n

通过以上方法可以在Ubuntu上正常通过，并且可以清晰的知道其依赖关系。

# 怎样让Ubuntu14.04中的Apache2.4列出文件和目录

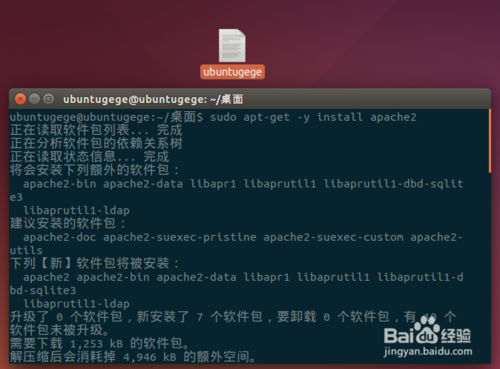
这个功能在Ubuntu 12.04 LTS中Apache 2.2中很好实现，但Ubuntu升级到14.04，Apache也跟着升级到了2.4，配置方便有了很大的不同，之前Apache2.2中的配置在apache2.4中很多都不生效了。2

## 方法/步骤

1. 1

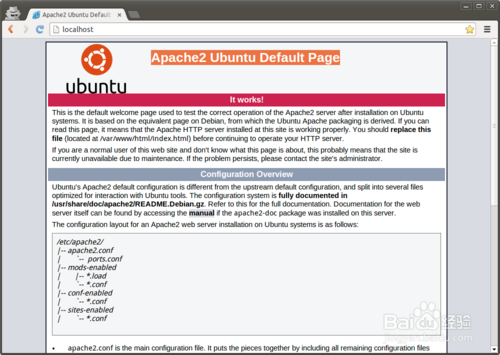
首先安装Apache，命令如下：

sudo apt-get -y install apache2

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=1)

1. 2

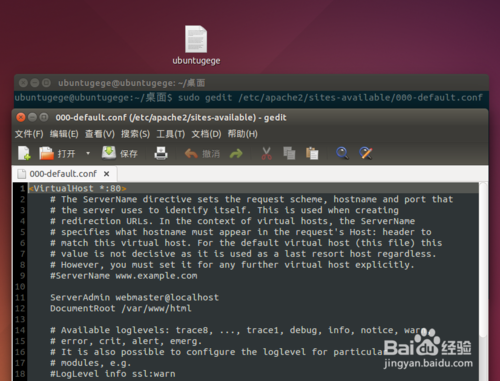
安装完成后，浏览器中打开 http://127.0.0.1或http://localhost/，如果看到如下apache的默认页面就表示安装成功了。

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=2)

1. 3

这时用geidt以管理员权限/身份编辑默认站点的配置文件000-default.conf。

sudo gedit /etc/apache2/sites-available/000-default.conf

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=3)

1. 4

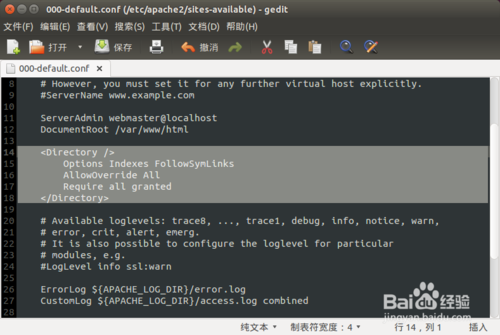
在<VirtualHost \*:80>中添加以下这内容（DocumentRoot /var/www/html这一行后）：

<Directory />

Options Indexes FollowSymLinksAllowOverride All

Require all granted

</Directory>

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=4)

1. 5

把/var/www目录修改为当前用户所有，方便有权限直接在www目录中修改或添加文件。

sudo chown 1000:1000 /var/www -R

同时把默认的index.html重命名为index\_.html，来测试是否能正常列出目录。

cd /var/www/html

mv index.html index\_.html

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=5)

1. 6

记得要重启apache的服务，让配置生效。

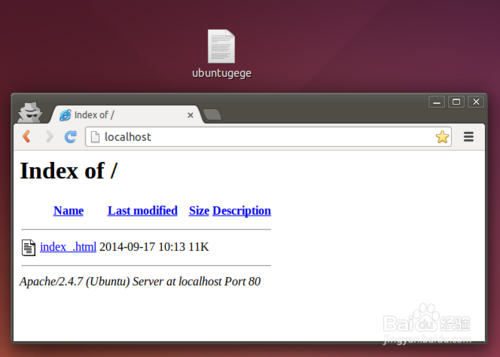
sudo service apache2 restart

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=6)

1. 7

这时再用浏览器打开 http://127.0.0.1或http://localhost/，看到目录已成功列出。

**当然这是在本地开发环境配置中才这样做的，在生产服务器上请谨慎配置！**

[](http://jingyan.baidu.com/album/fa4125acb82d9b28ac709215.html?picindex=7)

# [使用TortoiseGit处理代码冲突](http://www.cnblogs.com/jason-beijing/p/5718190.html)

场景一

user0 有新提交

user1 没有pull -> 写新代码 -> pull -> 提示有冲突

解决办法一

-> stash save(把自己的代码隐藏存起来) -> 重新pull -> stash pop(把存起来的隐藏的代码取回来 ) -> 代码文件会显示冲突 -> 右键选择edit conficts，解决后点击编辑页面的 mark as resolved ->  commit&push

解决办法二(尽量少使用，这种方法的优点是在在原编辑器里处理冲突，代码逻辑看得更清楚一些)

-> stash save(把自己的代码隐藏存起来) -> 重新pull -> stash pop(把存起来的隐藏的代码取回来 ) -> 代码文件会显示冲突 -> 右键选择resolve conflict -> 打开文件解决冲突 -> commit&push

场景二

user0 有新提交

user1 没有pull -> 写新代码 -> commit&push -> 提示有冲突

解决办法一

-> pull -> 代码文件会显示冲突 -> 右键选择edit conficts，解决后点击编辑页面的 mark as resolved ->  commit&push

# Ubuntu16.04完全卸载Mysql 5.7

sudo apt purge mysql-\*

sudo rm -rf /etc/mysql/ /var/lib/mysql

sudo apt autoremove

sudo apt autoclean

**二、通过下载离线安装包DEB Bundle进行安装**

说明：参考官方安装文档，参考：<https://dev.mysql.com/doc/refman/5.7/en/linux-installation-debian.html>

1、下载：

其实DEB Bundle类型就是离线deb安装包，把所有软件打包进去了。

登录站点：<https://dev.mysql.com/downloads/mysql/>

选择系统：

选择版本，在最下方选择16.04：

在列表上会有DEB Package的，这个其实就是deb文件，不过也是在线安装的形式，所以文件很小，不建议选择。

2、解压

tar xvf mysql-server\_5.7.18-1ubuntu16.04\_amd64.deb-bundle.tar

解压出来的文件如下：

3、安装

说明：以下命令是官网提供的

#libaio 如果系统中尚未存在库，则 可能需要安装该库：

sudo apt-get install libaio1

#使用以下命令预配置MySQL服务器软件包：

sudo dpkg-preconfigure mysql-community-server\_\*.deb

#提示：将被要求为root用户提供您的MySQL安装密码。

#对于MySQL服务器的基本安装，请安装数据库公用文件包，客户端包，客户端元包，服务器包和服务器元包（按此顺序）; 可以使用单个命令来执行此操作：

#注意：下面这条命令不能直接运行，应该拆开来按中括号里面以逗号分开的顺序进行安装，比如：

#sudo dpkg -i mysql-{common,community-client,client,community-server,server}\_\*.deb

sudo dpkg -i mysql-common\_\*.deb

sudo dpkg -i mysql-community-client\_\*.deb

sudo dpkg -i mysql-client\_\*.deb

sudo dpkg -i mysql-community-server\_\*.deb

sudo dpkg -i mysql-server\_\*.deb

#如果中途被dpkg警告未满足的依赖关系 ，可以使用apt-get来修复它们，然后再运行中断的命令 ：

sudo apt-get -f install

 安装完成后和第一种方式效果上是一致的。

**三、通过tar.gz压缩包进行安装**

以下教程官网已经提供，参考：<https://dev.mysql.com/doc/refman/5.7/en/binary-installation.html>

说明：这种方式是需要进行后期处理，比如开机启动服务，命令行环境变量，以及配置文件设置等。

1、下载：

 站点：[https://dev.mysql.com/downloads/mysql/](https://dev.mysql.com/downloads/mysql/" \t "_blank)

2、解压

tar zxvf mysql-5.7.18-linux-glibc2.5-x86\_64.tar.gz

sudo mv mysql-5.7.18-linux-glibc2.5-x86\_64 /usr/local

sudo ln -s /usr/local/mysql-5.7.18-linux-glibc2.5-x86\_64/ /usr/local/mysql

3、安装

由于MySQL运行需要libaio1库，所以需要运行以下命令进行安装：

sudo apt-get install libaio1

#添加用户组

sudo groupadd mysql

#添加用户，这个用户是不能登录的

sudo useradd -r -g mysql -s /bin/false mysql

#进入文件目录，mysql是链接

cd /usr/local/mysql

#新建文件夹

sudo mkdir mysql-files

#修改文件夹的权限

sudo chmod 750 mysql-files

sudo chown -R mysql .

sudo chgrp -R mysql .

#安装初始化，注意：此部最后一行会有一个初始化密码，用于root账号的首次登录

sudo bin/mysqld --initialize --user=mysql

#生成证书

sudo bin/mysql\_ssl\_rsa\_setup

#把权限修改回来

sudo chown -R root .

sudo chown -R mysql data mysql-files

4、启动

#启动在后台

sudo bin/mysqld\_safe --user=mysql &

5、登录测试并修改root密码

/usr/local/mysql/bin/mysql -uroot -p

提示：密码在安装初始化时最后一行的信息，里面有括号和特殊字符。

#修改root密码，每一个分号直接回车

mysql> SET PASSWORD = PASSWORD('新密码');

mysql> ALTER USER 'root'@'localhost' PASSWORD EXPIRE NEVER;

mysql> flush privileges;

#增加一个'root'@'%'账号实现远程登录

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

6、配置服务自动启动

#复制服务文件到/etc/init.d

sudo cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql.server

安装sysv-rc-conf

sudo apt-get install sysv-rc-conf

启动sysv-rc-conf

sudo sysv-rc-conf

按空格键出现X，然后按Q退出。重启即可。

服务的相关操作命令：

#服务状态

sudo service mysql.server status

#服务启动

sudo service mysql.server start

#服务停止

sudo service mysql.server stop

7、配置环境变量

在～/.profile文件的最下方加入

export PATH=$PATH:/usr/local/mysql/bin

# Ubuntu环境下挂载新硬盘

[TurkeyMiami](http://blog.51cto.com/12348890)0人评论[1362人阅读2018-03-29 10:44:18](javascript:;)

一、硬盘分区 | Hard disk add new partition

**1、显示硬盘及所属分区情况。在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | sudo fdisk -l |

**显示当前的硬盘及所属分区的情况。如下图所示：**  
**系统提示：DIsk /dev/sdb doesn't contain a valid partition table。**

**２、对硬盘进行分区。在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | sudo fdisk /dev/sdb |

如下图所示：  
在Command (m for help)提示符后面输入m显示一个帮助菜单。

**在Command (m for help)提示符后面输入n，执行 add a new partition 指令给硬盘增加一个新分区。  
 出现Command action时，输入e，指定分区为扩展分区（extended）。  
 出现Partition number(1-4)时，输入１表示只分一个区。  
 后续指定起启柱面（cylinder）号完成分区。**

**在Command (m for help)提示符后面输入p，显示分区表。  
系统提示如下：  
Device Boot                 Start                End                   Blocks          Id             System  
/dev/sdb1                           1            26108           209712478+           5          Extended**

**在Command (m for help)提示符后面输入w，保存分区表。  
系统提示：The partition table has been altered!**

**在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | sudo fdisk -l |

如下图所示：  
系统已经识别了硬盘 /dev/sdb 的分区。

二、硬盘格式化 | Format hard disk

**1、显示硬盘及所属分区情况。在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | sudo mkfs -t ext4 /dev/sdb |

说明：  
-t ext4 表示将分区格式化成ext4文件系统类型。

**三、挂载硬盘分区 | Mount hard disk partition**

**1、显示硬盘挂载情况。在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | df -l |

新硬盘分区没有挂载，无法进入和查看。  
  
**在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | sudo mount -t ext4 /dev/sdb /devdata |

**再次在终端窗口中输入如下命令：**

|  |  |
| --- | --- |
| 1 | df -l |

新硬盘分区已经挂载，如下图最下面的红色方框内容。

**２、配置硬盘在系统启动自动挂载。在文件 /etc/fstab 中加入如下配置：**

|  |  |
| --- | --- |
| 1 | /dev/sdb     /devdata    ext4     defaults       0 0 |

**至此ubuntu硬盘的挂载就完成了**

# Oracle创建数据库

//创建临时表空间

create temporary tablespace chyedu\_temp

tempfile 'E:\app\Administrator\virtual\oradata\orcl\chyedu\_tmp.dbf'

size 32m

autoextend on

next 32m maxsize 2048m

extent management local;

//创建数据表空间

create tablespace chyeduspace datafile 'E:\app\Administrator\virtual\oradata\orcl\chyedu.dbf'

size 32m

autoextend on

next 32m maxsize 2048m

extent management local;

//创建用户并指定表空间

create user chyedu identified by chyedu

default tablespace chyeduspace

temporary tablespace chyedu\_temp;

//给用户授予权限

grant connect,resource to chyedu;

grant create any sequence to chyedu;

grant create any table to chyedu;

grant delete any table to chyedu;

grant insert any table to chyedu;

grant select any table to chyedu;

grant unlimited tablespace to chyedu;

grant execute any procedure to chyedu;

grant update any table to chyedu;

grant create any view to chyedu;

# [使用Oracle SQL Developer迁移MySQL至Oracle数据库](http://www.cnblogs.com/xusweeter/p/6512521.html)

Oracle SQL Developer是Oracle官方出品的数据库管理工具。本文使用Oracle SQL Developer执行从MySQL迁移至Oracle数据库的操作。

2017年3月6日

操作步骤：

**1、在Oracle数据库中创建移植资料档案库用户**

资料档案库用户不是用来接收MySQL迁移数据的用户，而是进行数据迁移配置的用户。

Oracle创建用户命令如下：

create user migrater identified by migrater;

alter user migrater account unlock;

grant dba to migrater;

grant connect, resource to migrater;

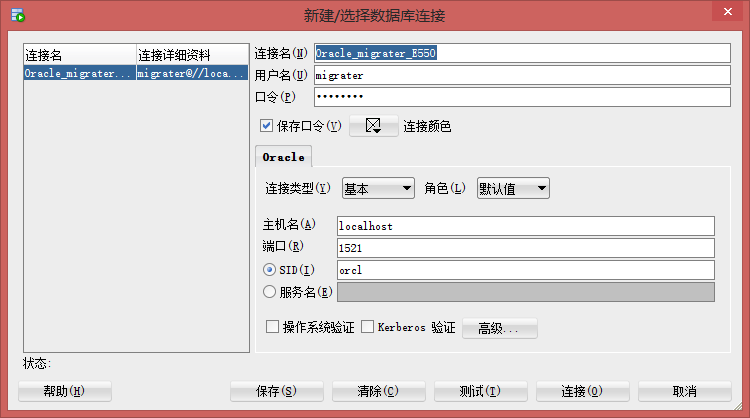
grant create session to migrater;

commit;

备注：请赋予该用户DBA权限。

**2、在SQL Developer中新建Oracle连接**

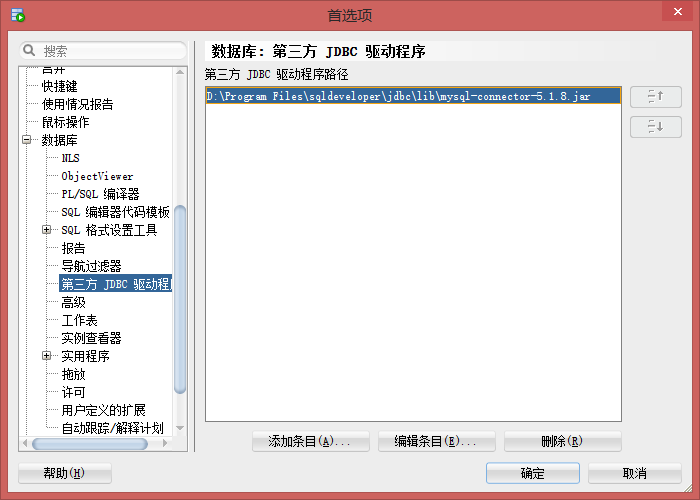
SQL Developer中创建Oracle的连接比较简单，如下图所示：

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230400594-768418103.png)

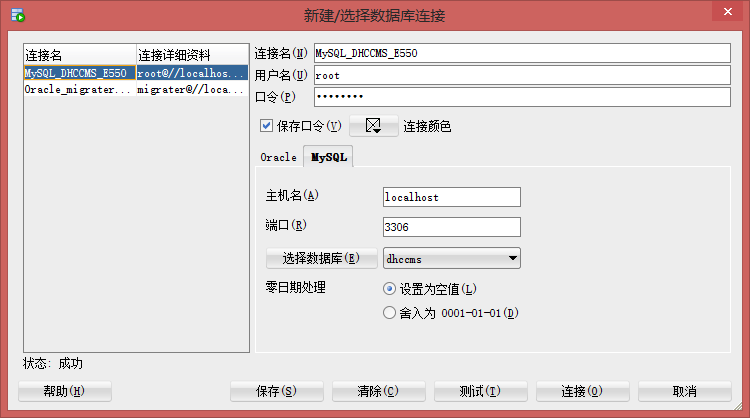
**3、添加第三方JAR包（MySQL）**  
使用SQL Developer建立MySQL的连接，需要添加MySQL的驱动包，添加方法如下：

点击“工具-》首选项”

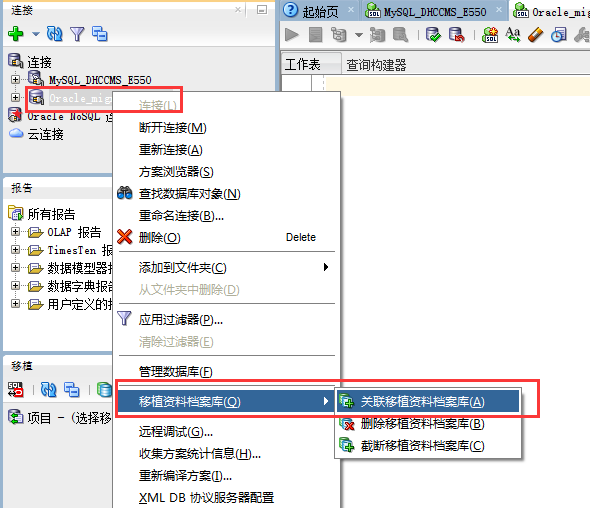
在首选项面板中选择左侧“数据库-》第三方JDBC驱动程序”，点击下方的添加条目按钮选择MySQL的驱动jar包。

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230401875-2097856780.png)

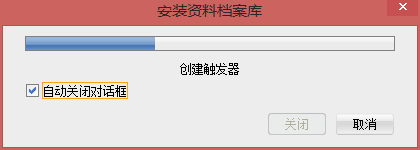
**4、新建MySQL连接**  
添加好MySQL的第三方驱动后，在新建连接页面中即可出现MySQL的配置项。

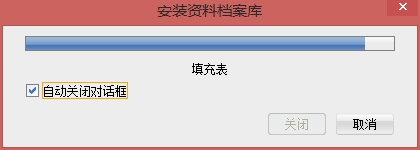
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230403000-1230813775.png)

**5、关联移植资料档案库**  
执行迁移数据库之前需要关联移植资料档案库，操作如下：

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230404000-546646598.png)

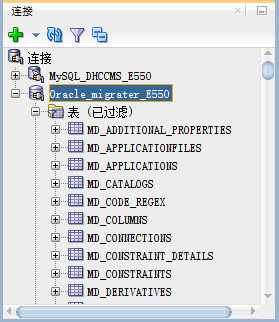
安装过程：

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230404859-835219664.png)

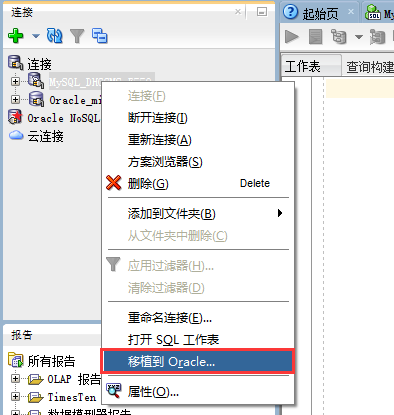
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230405531-41239453.png)

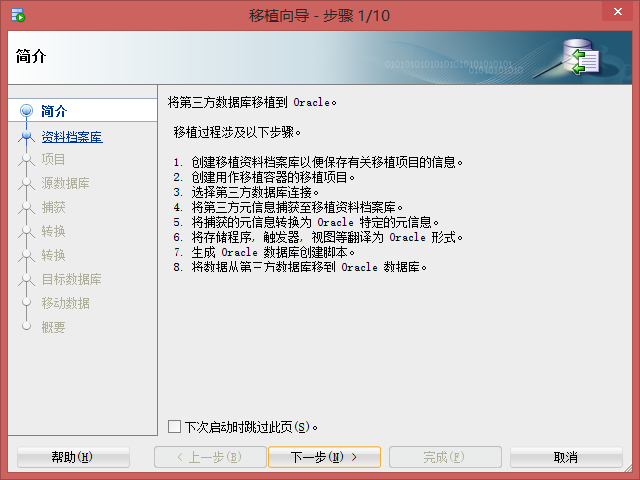
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230406375-243666962.png)

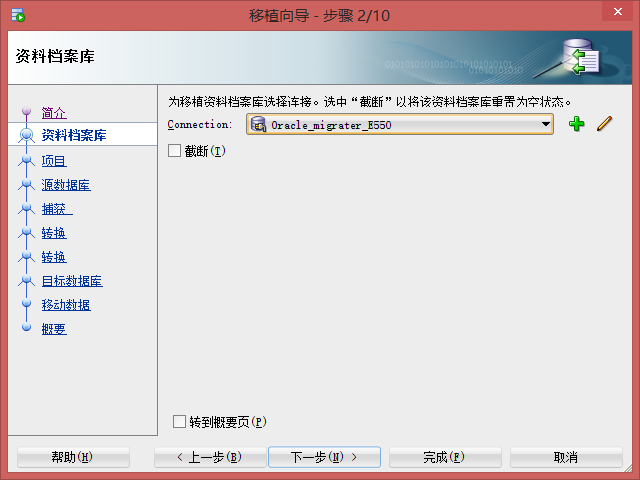
关联过后新建的Oracle数据库中增加很多表：

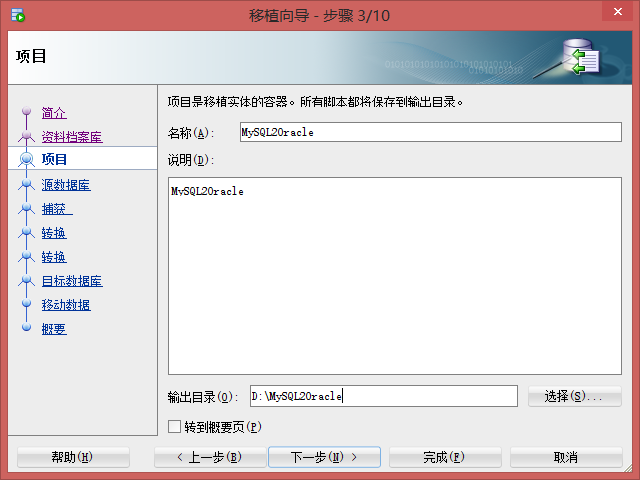
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230407328-854763618.png)

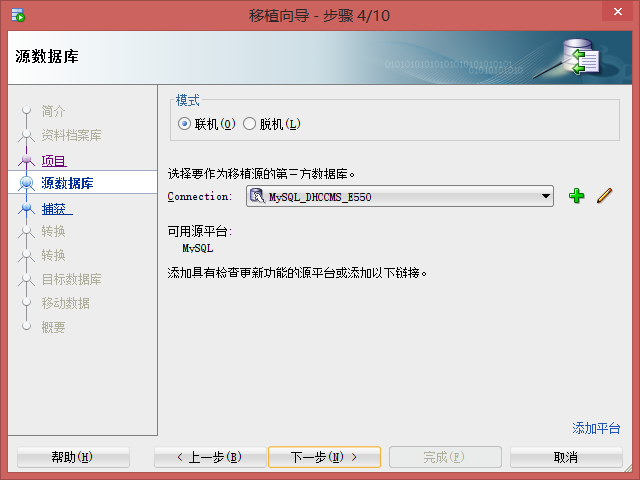
**6、移植MySQL数据库至Oracle数据库**  
在需要数据迁移的MySQL数据库连接上右键，选择“移植到Oracle…”菜单。

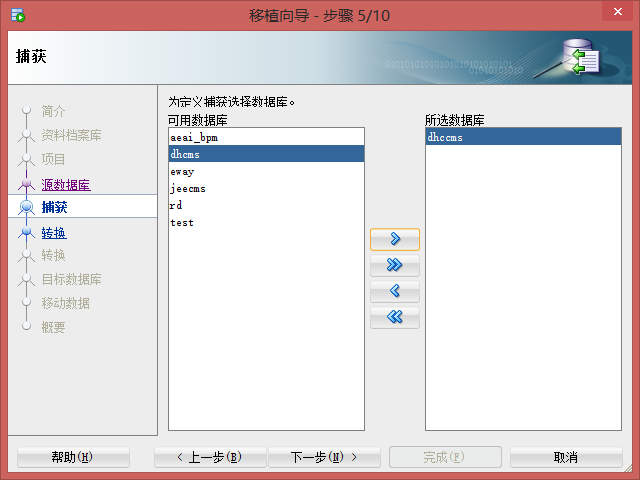
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230409063-1948601255.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230410047-866452466.png)

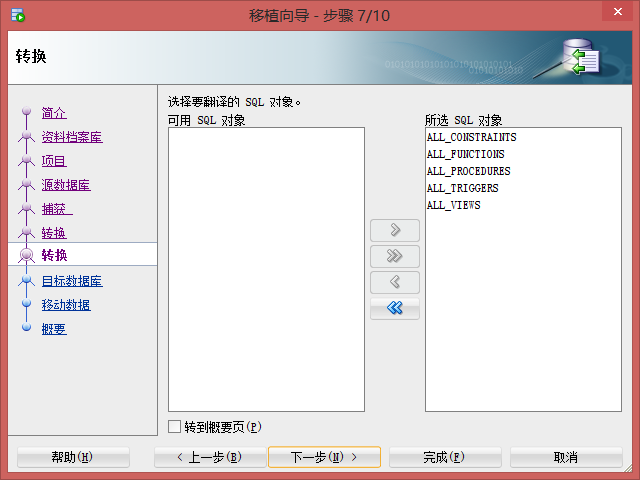
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230411391-1670603251.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230412297-2068662555.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230413594-771977687.png)

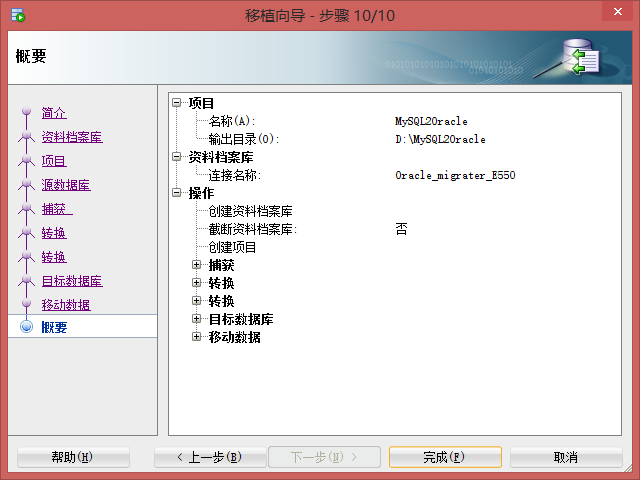
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230415016-1298395935.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230415844-1981180903.png)

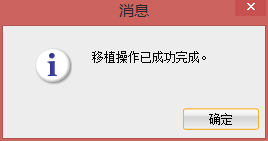
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230416766-606425582.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230417766-1339270029.png)

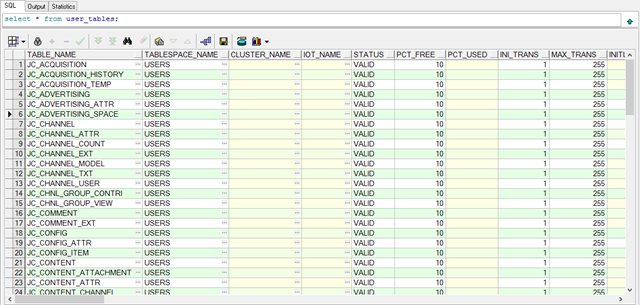
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230418844-2076004605.png)

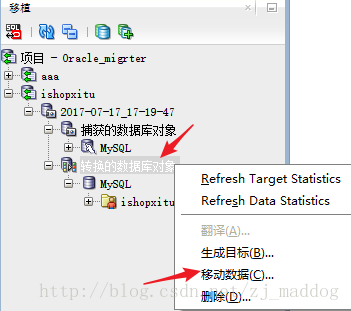
[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230419891-1906512445.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230420766-1444571564.png)

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230421531-604913281.png)

操作完成后会在Oracle数据库中创建和MySQL数据库同名的用户，使用该用户连接数据库即可查看所有迁移的表。

[](http://images2015.cnblogs.com/blog/719294/201703/719294-20170306230422438-1430499879.png)

目前已经完成了数据库中表的转换。   
然后再是数据库中的数据移植，看下图：   


点击移动数据后，会将目标数据库中的数据转移到Oracle中，这样我们就可以在Oracle中看到数据库的表与数据了

# Ubuntu16.04 Docker 安装

安装必要条件

64位的系统

可以使用sudo的用户

 安装

**第一步：设置包管理器仓库**

在Ubuntu上设置Docker仓库。 lsb\_release -cs 可以显示你的 Ubuntu 版本，比如 xenial 或者 trusty。

设置完成后，更新包管理器。

sudo apt-get -y install apt-transport-https ca-certificates

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-get install software-properties-common

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get udate

**第二步：安装Docker**

sudo apt-get -y install docker-ce

**第三步：测试**

sudo docker run hello-world

### 1、获取最新版本的 Docker 安装包

runoob@runoob:~$ wget -qO- https://get.docker.com/ | sh

### Install using the repository

Before you install Docker CE for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

#### SET UP THE REPOSITORY

1. Update the apt package index:
2. $ sudo apt-get update
3. Install packages to allow apt to use a repository over HTTPS:
4. $ sudo apt-get install \
5. apt-transport-https \
6. ca-certificates \
7. curl \
8. software-properties-common
9. Add Docker’s official GPG key:
10. $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Verify that you now have the key with the fingerprint9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

$ sudo apt-key fingerprint 0EBFCD88

pub 4096R/0EBFCD88 2017-02-22

Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

uid Docker Release (CE deb) <docker@docker.com>

sub 4096R/F273FCD8 2017-02-22

1. Use the following command to set up the **stable** repository. You always need the **stable**repository, even if you want to install builds from the **edge** or **test** repositories as well. To add the **edge** or **test** repository, add the word edge or test (or both) after the word stable in the commands below.

**Note**: The lsb\_release -cs sub-command below returns the name of your Ubuntu distribution, such as xenial. Sometimes, in a distribution like Linux Mint, you might need to change $(lsb\_release -cs) to your parent Ubuntu distribution. For example, if you are using Linux Mint Rafaela, you could use trusty.

* + x86\_64 / amd64
  + armhf
  + IBM Power (ppc64le)
  + IBM Z (s390x)

$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

**Note**: Starting with Docker 17.06, stable releases are also pushed to the **edge** and **test**repositories.

[Learn about **stable** and **edge** channels](https://docs.docker.com/install/).

#### INSTALL DOCKER CE

1. Update the apt package index.
2. $ sudo apt-get update
3. Install the latest version of Docker CE, or go to the next step to install a specific version:
4. $ sudo apt-get install docker-ce

**Got multiple Docker repositories?**

If you have multiple Docker repositories enabled, installing or updating without specifying a version in the apt-get install or apt-get update command always installs the highest possible version, which may not be appropriate for your stability needs.

1. To install a specific version of Docker CE, list the available versions in the repo, then select and install:

a. List the versions available in your repo:

$ apt-cache madison docker-ce

docker-ce | 18.03.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages

b. Install a specific version by its fully qualified package name, which is package name (docker-ce) “=” version string (2nd column), for example, docker-ce=18.03.0~ce-0~ubuntu.

$ sudo apt-get install docker-ce=<VERSION>

The Docker daemon starts automatically.

1. Verify that Docker CE is installed correctly by running the hello-world image.
2. $ sudo docker run hello-world

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

Docker CE is installed and running. The docker group is created but no users are added to it. You need to use sudo to run Docker commands. Continue to [Linux postinstall](https://docs.docker.com/install/linux/linux-postinstall/) to allow non-privileged users to run Docker commands and for other optional configuration steps.

#### UPGRADE DOCKER CE

To upgrade Docker CE, first run sudo apt-get update, then follow the [installation instructions](https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-docker), choosing the new version you want to install.

# 登录阿里云docker registry:

$ sudo docker login --username=hzu\_ky\_17@126.com registry.cn-hangzhou.aliyuncs.com

登录registry的用户名是您的阿里云账号全名，密码是您开通服务时设置的密码。

你可以在镜像管理首页点击右上角按钮修改docker login密码。

**从registry中拉取镜像：**

$ sudo docker pull registry.cn-hangzhou.aliyuncs.com/wind-starscube/starscube-showdoc:[镜像版本号]

**将镜像推送到registry：**

$ sudo docker login --username= registry.cn-hangzhou.aliyuncs.com

$ sudo docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/wind-starscube/starscube-showdoc:[镜像版本号]

$ sudo docker push registry.cn-hangzhou.aliyuncs.com/wind-starscube/starscube-showdoc:[镜像版本号]

镜像名称 portainer/portainer

一、启动portainer 容器

直接执行 自动下载docker镜像

6.1单节点运行方式  
docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer  
-d 参数以detach方式运行  
-p 将容器工作端口映射至物理机端口 HOST端口:容器端口  
-v 将容器目录挂载(映射)到物理机对应位置,这里指定通讯方式为sock  
portainer/portainer 为镜像名称

6.2多节点： portainner 也支持用TCP通讯

启动命令  
 docker run -d -p 9000:9000 portainer/portainer -H tcp://<REMOTE\_HOST>:<REMOTE\_PORT>

其中REMOTE\_HOST是slave ip，REMOTE\_PORT是slave 端口

二、访问

http://宿主机IP:9000

进入页面后设置用户名和密码

然后选择 local 或者remote,这里使用单节点 local

运行下边的命令来安装 Compose：

curl -L https://github.com/docker/compose/releases/download/1.3.1/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

注意：如果你在安装的时候出现了 “Permission denied” 的错误信息，这说明你的 /usr/local/bin 目录是不可写的，你需要使用超级用户来安装。运行 sudo -i , 然后运行上边的两个命令，然后 exit 退出。

启动禅道：

sudo docker run -d -p 8081:80 -p 3307:3306 \

-e USER="root" -e PASSWD="password" \

-e BIND\_ADDRESS="false" \

-e SMTP\_HOST="163.177.90.125 smtp.exmail.qq.com" \

-v /data/zbox/:/opt/zbox/ \

--name zentao-server \

registry.cn-hangzhou.aliyuncs.com/wind-starscube/starscube-zentao:latest

# 开启nginx的ssl模块

1.the "ssl" parameter requires ngx\_http\_ssl\_module in /usr/local/nginx/conf/nginx.conf:37

原因是nginx缺少http\_ssl\_module模块，编译安装时带上--with-http\_ssl\_module配置就可以了

2.如果已经安装过nginx,想要添加模块看下面

1)切换到nginx源码包

cd /usr/local/src/nginx-1.11.3

2)查看ngixn原有的模块

/usr/local/nginx/sbin/nginx -V

3)重新配置

./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module

4)重新编译，不需要make install安装。否则会覆盖

make

5)备份原有已经安装好的nginx

cp /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.bak

6)将刚刚编译好的nginx覆盖掉原来的nginx(ngixn必须停止)

cp ./objs/nginx /usr/local/nginx/sbin/

这时，会提示是否覆盖，请输入yes，直接回车默认不覆盖

7)启动nginx，查看nginx模块，发现已经添加

/usr/local/nginx/sbin/nginx -V

# Hadoop常用hdfs命令：

1.创建目录

bin/hdfs dfs -mkdir /user

bin/hdfs dfs -mkdir /user/<username>

在HDFS中创建一个名为path的目录，如果它的上级目录不存在，也会被创建，如同linux中的mkidr –p。

bin/hdfs dfs -mkdir -p /usr/file

2.将本地文件或目录（eg:/home/grid/redis-2.8.12.tar.gz)上传到HDFS中的路径( /usr/file)。

bin/hdfs dfs -put /home/grid/redis-2.8.12.tar.gz /usr/file

bin/hdfs dfs -put etc/hadoop input

3.将文件或目录从HDFS中的路径(/usr/file/redis-2.8.12.tar.gz)拷贝到本地文件路径(/usr/local)。

bin/hdfs dfs -get /usr/file/redis-2.8.12.tar.gz /usr/local

4.查看目录下内容，包括文件名，权限，所有者，大小和修改时间

bin/hdfs dfs -ls /usr/file

5.与ls相似，但递归地显示子目录下的内容。

bin/hdfs dfs -ls -R /usr/file

6.显示path下所有文件磁盘使用情况下，用字节大小表示，文件名用完整的HDFS协议前缀表示。

bin/hdfs dfs -du /usr/file

7.与-du相似，但它还显示全部文件或目录磁盘使用情况

bin/hdfs dfs -du -s /usr/file

8.在HDFS中，将文件或目录从HDFS的源路径移动到目标路径。

bin/hdfs dfs -mv /usr/file/redis-2.8.12.tar.gz /usr

9.在HDFS中，将/usr/redis-2.8.12.tar.gz文件或目录复制到/usr/file

bin/hdfs dfs -cp /usr/redis-2.8.12.tar.gz /usr/file

10.删除一个文件或目录

bin/hdfs dfs -rm -skipTrash /usr/redis-2.8.12.tar.gz

删除一个文件或递归删除目录

bin/hdfs dfs -rmr -skipTrash /usr/redis-2.8.12.tar.gz

11.显示文件内容到标准输出上。

bin/hdfs dfs -cat /usr/file/test.txt

12.创建一个文件。时间戳为当前时间，如果文件本就存在就失败，除非原文件长充为0。

bin/hdfs dfs -touchz /usr/file/test.txt

13.显示文件所占块数(%b)，文件名(%n)，块大小(%n)，复制数(%r)，修改时间(%y%Y)。

bin/hdfs dfs -stat /usr/file/file

14.显示文件最后的1KB内容到标准输出。

bin/hdfs dfs -tail /usr/file/test.txt

15.显示cmd命令的使用信息，你需要把命令的“-”去掉

bin/hdfs dfs -help tail

16.统计文件(夹)数量

bin/hdfs dfs -count /usr

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.0.jar grep input output 'dfs[a-z.]+¡®

bin/hdfs dfs -get output output

# 挂载

sudo mount -t nfs -o rw 192.168.10.210:/home/flyrish/nfs\_data /data

# nohup

1. nohup command > myout.file 2>&1 &

在上面的例子中，0 – stdin (standard input)，1 – stdout (standard output)，2 – stderr (standard error) ；

2>&1是将标准错误（2）重定向到标准输出（&1），标准输出（&1）再被重定向输入到myout.file文件中。

# Redis批量删除

redis-cli -h 12.132.30.21 -p 6379 -a 2016 -n 4 keys "ecard\*" | xargs redis-cli -h 12.132.30.21 -p 6379 -a 2016 -n 4 del

说明：  
-h:redis所在服务器的ip；  
-p：redis端口号  
-a:redis连接密码  
-n:选择redis中的哪一个数据库  
keys:正则匹配所要找的key

该命令不能在redis客户端里面去执行，直接在服务器里面执行即可。

Mysql的dump--set-gtid-purged=off

# Ubuntu开机启动配置：

**说明：**常用的Linux启动项就是在/etc/rc.local的exit 0语句之间添加启动脚本，这里说下另一种方法，使用update-rc.d命令添加/禁止开机启动项，挺好用的。

## 使用

#将脚本添加开机自启，先放入/etc/init.d文件夹，并给予可执行权限

mv xx.sh /etc/init.d

chmod +x /etc/init.d/xx.sh

#设置开机自启

update-rc.d xx.sh defaults

#删除开机自启

update-rc.d -f xx.sh remove

#如果你想禁止程序开机自启，比如apache2、nginx

update-rc.d -f apache2 remove

update-rc.d -f nginx remove

对于有时候不想卸载某些程序，又不想开机启动的，这种办法挺有用的，还可以节省CPU等资源的利用。

Ubuntu18.04如下操作：

1、建立rc-local.service文件

|  |  |
| --- | --- |
| 1 | sudo vi /etc/systemd/system/rc-local.service |

2、将下列内容复制进rc-local.service文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | [Unit]  Description=/etc/rc.local Compatibility  ConditionPathExists=/etc/rc.local    [Service]  Type=forking  ExecStart=/etc/rc.local start  TimeoutSec=0  StandardOutput=tty  RemainAfterExit=yes  SysVStartPriority=99    [Install]  WantedBy=multi-user.target |

3、创建文件rc.local

|  |  |
| --- | --- |
| 1 | sudo vi /etc/rc.local |

4、将下列内容复制进rc.local文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | #!/bin/sh -e  #  # rc.local  #  # This script is executed at the end of each multiuser runlevel.  # Make sure that the script will "exit 0" on success or any other  # value on error.  #  # In order to enable or disable this script just change the execution  # bits.  #  # By default this script does nothing.  echo "看到这行字，说明添加自启动脚本成功。" > /usr/local/test.log  exit 0 |

5、给rc.local加上权限

|  |  |
| --- | --- |
| 1 | sudo chmod +x /etc/rc.local |

6、启用服务

|  |  |
| --- | --- |
| 1 | sudo systemctl enable rc-local |

7、启动服务并检查状态

|  |  |
| --- | --- |
| 1  2 | sudo systemctl start rc-local.service  sudo systemctl status rc-local.service |

8、重启并检查test.log文件

|  |  |
| --- | --- |
| 1 | cat /usr/local/test.log |

# Windows查看已安装补丁

在打开的DOS命令窗口中，输入以下命令：  
wmic qfe list full /format:htable >C:\Temp\hotfixes.htm