# 走一遍

systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

# Disconf优点总结如下：

**部署简单**：同一个上线包，无须改动配置，即可在多个环境中上线   
**部署动态化**：更改配置，无需重新打包或重启，即可实时生效   
**统一管理**：提供web平台，统一管理多个环境多个产品的所有配置

# 安装Disconf需要的环境

需要安装Nginx、Tomcat、Zookeeper、Redis和MySQL。

Nginx：处理静态资源请求、动态请求转发到Tomcat

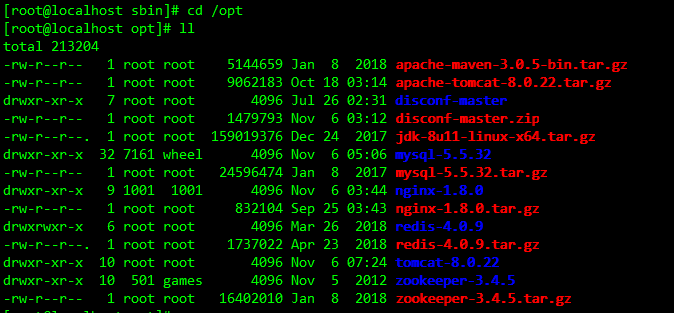
Tomcat：处理Nginx的请求

Zookeeper：管理Disconf配置信息，配置变更通过zk通知

Redis：用户session管理

MySQL：应用管理、用户管理、角色管理、环境管理、配置持久化

我将这些包都存放到了/opt下



**还需要安装jdk,git和maven**

**jdk就不用说了**

git用来拉取百度的disconf

yum install git -y

mavne用来编译他的disconf成war包

wget [http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo](http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo%20-O%20/etc/yum.repos.d/epel-apache-maven.repo)

yum -y install apache-maven

**可能是必要的安装**

yum install -y openssl\*

yum -y install ncurses-devel

# 前提必须要做的一些事情:

1、创建文件夹

（以下的文件夹路径读者可自行设置！）

配置文件所在路径：

/usr/local/disconf/resource

源码所在路径：

/usr/local/disconf/src

War包所在路径：

/usr/local/disconf/war

2、下载源码

进入到disconf/src下，在GitHub上面直接下载源码，利用下面的命令：

cd /usr/local/disconf/src

git clone <https://github.com/knightliao/disconf.git>

3、拷贝配置文件

  将/usr/local/disconf/src/disconf/disconf-web/profile/rd路径下的配置文件拷贝到/usr/local/disconf/resource（要将application-demo.properties文件改名为application.properties），并修改相应的配置文件。

拷贝命令：

cd /usr/local/disconf/src/disconf/disconf-web/profile/rd

cp \* /usr/local/disconf/resource

cd /usr/local/disconf/resource

mv ./application-demo.properties ./application.properties

4:修改相应的配置文件

这些配置文件修改后，都需要重新进行系统构建。

jdbc-mysql.properties(数据库的配置)：

  将mysql的连接地址、用户名和密码改为相应数据库的信息。

|  |
| --- |
| jdbc.driverClassName=com.mysql.jdbc.Driver  jdbc.db\_0.url=jdbc:mysql://192.168.106.17:3306/disconf?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=false  jdbc.db\_0.username=root  jdbc.db\_0.password=root  jdbc.maxPoolSize=20  jdbc.minPoolSize=10  jdbc.initialPoolSize=10  jdbc.idleConnectionTestPeriod=1200  jdbc.maxIdleTime=3600 |

redis-config.properties(配置redis的地址)：

修改相应的host和port，如果redis是集群，可配置相应的节点；如果redis是单机版，也必须配置两个redis-client（否则会报错），两个redis-client配置相同的host和port即可。

|  |
| --- |
| redis.group1.retry.times=2  redis.group1.client1.name=root  redis.group1.client1.host=192.168.106.17  redis.group1.client1.port=6379  redis.group1.client1.timeout=5000  redis.group1.client1.password=root  redis.group1.client2.name=root  redis.group1.client2.host=192.168.106.17  redis.group1.client2.port=6380  redis.group1.client2.timeout=5000  redis.group1.client2.password=root  redis.evictor.delayCheckSeconds=300  redis.evictor.checkPeriodSeconds=30  redis.evictor.failedTimesToBeTickOut=6 |

zoo.properties(zookeeper的地址配置)：

将Zookeeper集群的ip和端口号配置在hosts属性中即可，各节点地址之间用英文逗号隔开，其他不用修改。

|  |
| --- |
| hosts=127.0.0.1:8581,127.0.0.1:8582,127.0.0.1:8583  # zookeeper\u7684\u524D\u7F00\u8DEF\u5F84\u540D  zookeeper\_url\_prefix=/disconf |

日志文件的配置可以不用修改，如需修改参考网上log4j配置文件修改方式修改即可。

5、修改环境变量

修改环境变量，将创建的配置文件的路径和war包的路径配置在环境变量中。

vi /etc/profile

按“i”开启编辑，将下面的内容添加到环境变量的最后面：

ONLINE\_CONFIG\_PATH=/usr/local/disconf/resource

WAR\_ROOT\_PATH=/usr/local/disconf/war

export ONLINE\_CONFIG\_PATH

export WAR\_ROOT\_PATH

按Esc结束编辑，使用“:x”命令退出，利用下面的命令使环境变量生效：

source /etc/profile

6、构建

进入到disconf-web路径下，修改其pom文件：

cd /usr/local/disconf/src/disconf/disconf-web

vi pom.xml

将一下内容加入到pom文件的profiles结点下，如果JDK的版本低于1.8，则此步骤不需要执行：

<profile>

    <id>doclint-java8-disable</id>

    <activation>

      <jdk>[1.8,)</jdk>

    </activation>

    <build>

      <plugins>

        <plugin>

          <groupId>org.apache.maven.plugins</groupId>

          <artifactId>maven-javadoc-plugin</artifactId>

          <configuration>

            <additionalparam>-Xdoclint:none</additionalparam>

          </configuration>

        </plugin>

      </plugins>

    </build>

  </profile>

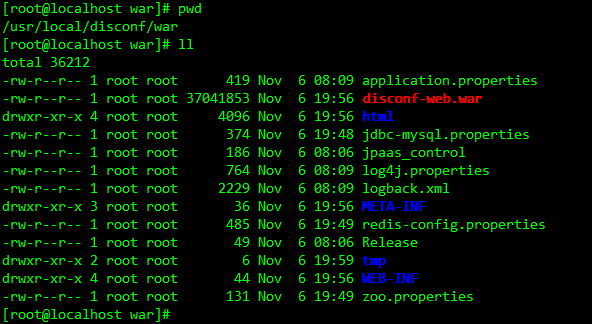


利用deploy/deploy.sh进行构建，构建过程会download很多东西，可能会有些慢，虚拟机要一直保持能上网的状态：

cd /usr/local/disconf/src/disconf/disconf-web

sh deploy/deploy.sh

构建完成后，在war目录下会生成相应的内容，主要的是加粗的那几项：



# 上线前的初始化工作

 在/usr/local/disconf/src/disconf/disconf-web/sql目录下有sql文件，将这些sql文件初始化到数据库中。这四个sql文件的执行是有顺序的，按照sql/readme.md文件中的顺序执行。

0-init\_table.sql

1-init\_data.sql

201512/20151225.sql

20160701/20160701.sql

2、配置Tomcat:

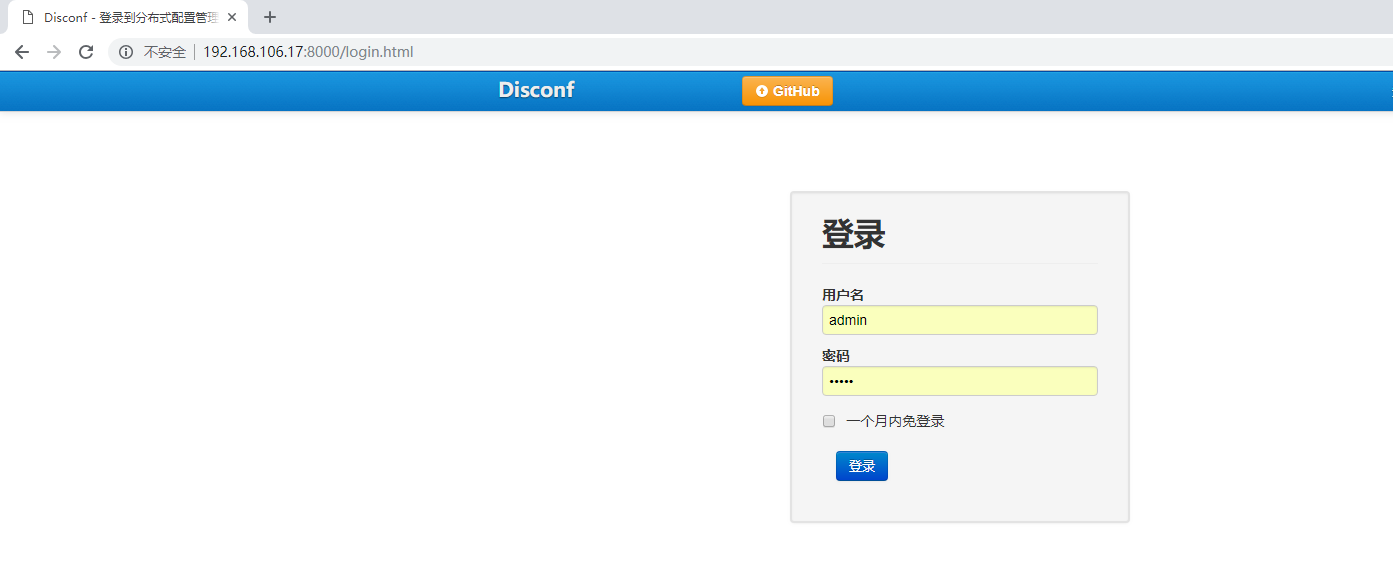
请查阅tomcat安装部分

3、部署前端

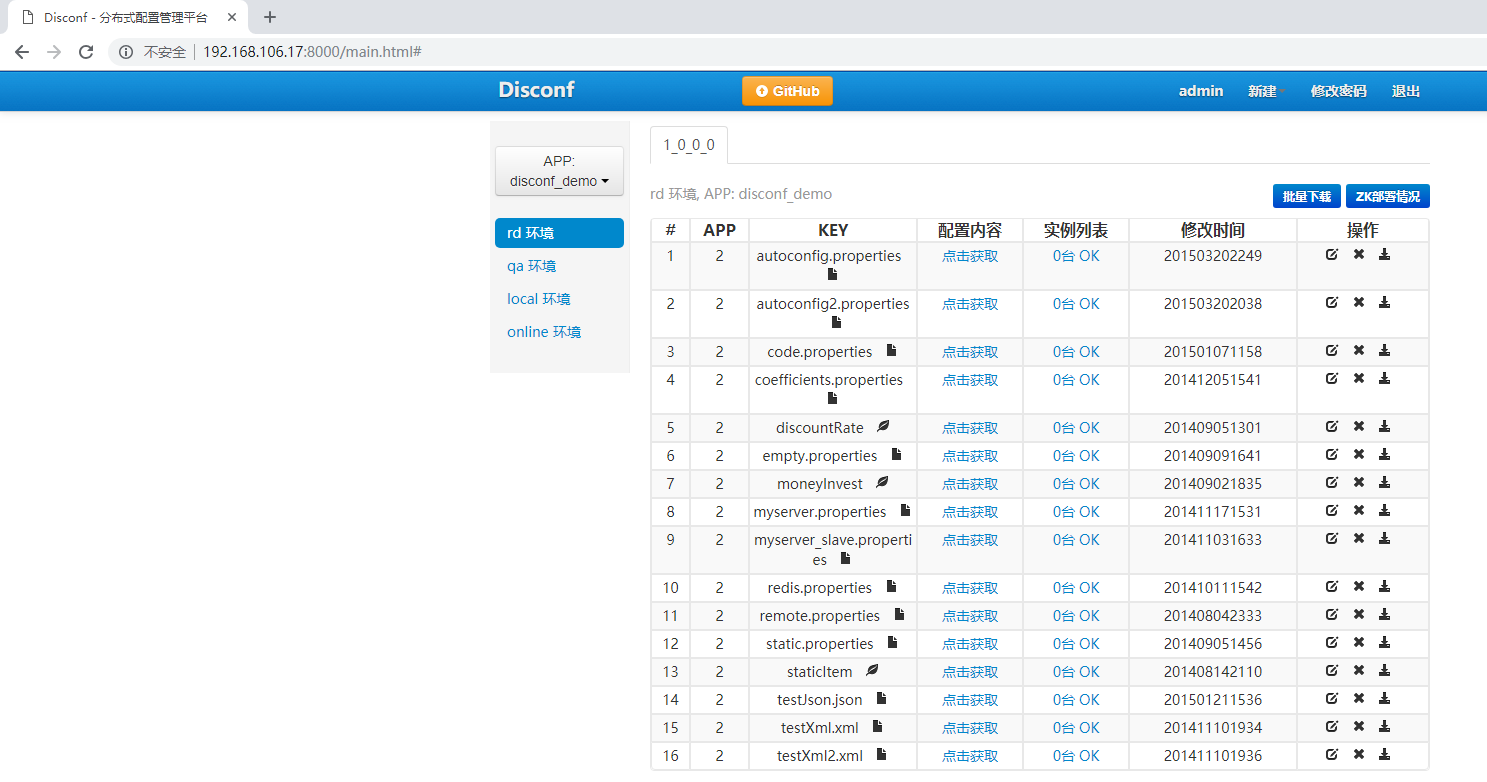
修改nginx.conf文件，将下面的内容添加进去：

请查阅nginx安装部分

# 运行效果



登录页面(默认用户名和密码：admin/admin)：



# 安装nginx

tar -zxvf nginx-1.8.0.tar.gz   
cd nginx-1.8.0

参数设置如下：

./configure --prefix=/usr/local/nginx

make

make install

启动nginx

cd /usr/local/nginx/sbin/

./nginx

退出

./nginx -s stop

退出

./nginx -s quit

重启

./nginx -s quit

./nginx

**将disconf-stable/disconf-web/html复制到/usr/local/disconf/war/**

**修改Nginx代理配置和静态资源目录**

upstream disconf {

server 127.0.0.1:8080; #tomcat服务器的地址

}

server {

listen 8091; #监听端口

server\_name localhost;

index index.html;

location / {

root /usr/local/disconf/war/html; #静态资源目录

index index.html;

}

location ~ ^/(api|export) {

proxy\_pass\_header Server;

proxy\_set\_header Host $http\_host;

proxy\_redirect off;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Scheme $scheme;

proxy\_pass http://disconf;

}

}

# **安装Tomcat**

启动Tomcat

./catalina.sh start

#访问浏览器localhost:8080看是否安装正确?

在配置文件host下添加



**<Context path="" docBase="/usr/local/disconf/war"></Context>**

# **安装Zookeeper**

启动Zookeeper

cd zookeeper-3.4.10/bin

./zkServer start #ps aux | grep zoo验证进程是否起来   
./zkCli.sh #zk客户端

只需要启动server端就行了

# 安装Redis

1、下载Redis源码包，Redis官网

2、解压安装

tar zxvf ../Downloads/redis-3.2.9.tar.gz

cd redis-3.2.9

make

make install #安装后在/usr/local/bin会有redis的脚本

cp redis.conf /etc

vi /etc/redis.conf ##将daemonize no改为daemonize yes

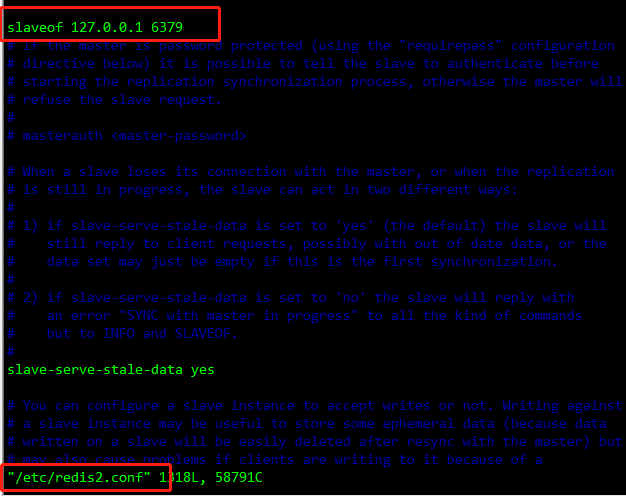
/usr/local/bin/redis-server /etc/redis.conf

ps aux | grep redis #如果启动成功会有redis的进程

3、配置主从

由于Disconf至少需要配置两台redis-client，所以我们可以在一台机器（本机）上安装master/slave模式的redis集群

cp /etc/redis.conf /etc/redis2.conf #redis2启动在6380端口



vi redis2.conf #配置slaveof 127.0.0.1 6379

sudo ./redis-server /etc/redis2.conf

ps aux | grep redis

4、设置主从的shardnam

./redis-cli -p 6379 #redis.conf

CONFIG SET requirepass ‘root’

auth root

./redis.cli -p 6380 #redis2.conf

CONFIG SET requirepass ‘root’

auth root

# 安装MySQL

3、新增 mysql 用户组：

# groupadd mysql

4、新增 mysql 用户，并添加到 mysql 用户组：

# useradd -r -g mysql mysql

5、新建 MySQL 执行文件目录(后面会把编译好的 mysql 程序安装到这个目录)：

# mkdir -p /usr/local/mysql

（-p 参数的作用是：如果最终目录的父目录不存在也会一并创建）

6、新建 MySQL 数据库数据文件目录：

# mkdir -p /home/mysql/data

# mkdir -p /home/mysql/logs

# mkdir -p /home/mysql/temp

（注意：上面的 logs 及 temp 目录是为了以后将 MySQL 的数据文件与执行程序文件分离，

如果你打算设置到不同的路径，注意修改对应的执行命令和数据库初始化脚本。正式生产环

境，建议数据目录和日志目录都使用单独的分区来挂载，不同分区属于不同的磁盘或磁盘

组。）

7、增加 PATH 环境变量搜索路径：

# vi /etc/profile

##在 profile 文件末尾增加两行

# mysql env param

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

export PATH

使 PATH 搜索路径立即生效：

# source /etc/profile

8、安装编译 MySQL 需要的依赖包：

（mysql 从 5.5 版本开始，不再使用./configure 编译，而是使用 cmake 编译器，具体的

cmake 编译参数可以参考 mysql 官网文档

http://dev.mysql.com/doc/refman/5.5/en/source-configuration-options.html，安装基

本依赖包，先用 yum 安装 cmake、automake 、autoconf ，另 MySQL 5.5.x 需要最少安装的

包有：bison,gcc、gcc-c++、ncurses-devel）：

# yum install make cmake gcc gcc-c++ bison bison-devel ncurses ncurses-devel autoconf automake

9、进入/usr/local/src 目录，上传 mysql-5.6.26.tar.gz 源代码到/usr/local/src 目录：

# cd /usr/local/src

10、开始编译安装 mysql-5.6.26：

解压缩源码包：

# tar -zxvf mysql-5.6.26.tar.gz

进入解压缩源码目录：

# cd mysql-5.6.26

使用 cmake 源码安装 mysql（如果你打算安装到不同的路径，注意修改下面语句中

/usr/local/mysql 和/home/mysql/data 路径！）

[root@edu-mysql-01 mysql-5.6.26]# cmake \

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DMYSQL\_UNIX\_ADDR=/usr/local/mysql/mysql.sock \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1 \

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 \

-DWITH\_MEMORY\_STORAGE\_ENGINE=1 \

-DWITH\_READLINE=1 \

-DENABLED\_LOCAL\_INFILE=1 \

-DMYSQL\_DATADIR=/home/mysql/data \

-DMYSQL\_USER=mysql \

-DMYSQL\_TCP\_PORT=3306 \

-DENABLE\_DOWNLOADS=1

上面的这些复制完，回车，然后就开始 cmake 的过程，一般时间不会很长。

配置解释：

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql 设置安装目录

-DMYSQL\_DATADIR=/home/mysql/data 设置数据库存放目录

-DMYSQL\_UNIX\_ADDR=/usr/local/mysql/mysql.sock 设置 UNIX socket 目录

-DMYSQL\_USER=mysql 设置运行用户

-DDEFAULT\_CHARSET=utf8 设置默认字符集，默认 latin1

-DEFAULT\_COLLATION=utf8\_general\_ci 设置默认校对规则，默认 latin1\_general\_ci

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 添加 InnoDB 引擎支持

-DENABLE\_DOWNLOADS=1 自动下载可选文件，比如自动下载谷歌的测试包

-DMYSQL\_TCP\_PORT=3306 设置服务器监听端口，默认 3306

-DSYSCONFDIR=/etc 设置 my.cnf 所在目录，默认为安装目录）

执行过程中会出现：

CMake Error: Problem with tar\_extract\_all(): Invalid argument

CMake Error: Problem extracting tar: /usr/local/src/mysql-5.6.26/source\_downloads/gmock-1.6.0.zip

解决方法：

cd mysql 目录下面会发现有一个 source\_downloads 目录，需要解压 unzip gmock-1.6.0.zip,然

后再重新执行上述配置过程。当然你也可以去掉-DENABLE\_DOWNLOADS=1 这个选项，不编

译谷歌的测试包也没有什么问题，但是之前的某些版本会出现无法编译的问题.

11、cmake 结束后开始编译源码，这一步时间会较长，请耐心等待：

# make

12、安装编译好的程序：

# make install

（注意：如果需要重装 mysql，在/usr/local/src/mysql-5.6.26 在执行下 make install 就可以了，

不需要再 cmake 和 make）

13、清除安装临时文件 ：

# make clean

14、修改 mysql 目录拥有者为 mysql 用户：

# chown -Rf mysql:mysql /usr/local/mysql

# chown -Rf mysql:mysql /home/mysql

15、进入 mysql 执行程序的安装路径：

# cd /usr/local/mysql

16、执行初始化配置脚本，创建系统自带的数据库和表（注意：路径/home/mysql/data 需要换成你自定定义的数据库存放路径）：

# scripts/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql --datadir=/home/mysql/data

Installing MySQL system tables...2015-12-13 15:21:53 0 [Warning] TIMESTAMP with implicit

DEFAULT value is deprecated. Please use --explicit\_defaults\_for\_timestamp server option (see

documentation for more details).

2015-12-13 15:21:53 0 [Note] /usr/local/mysql/bin/mysqld (mysqld 5.6.26) starting as process

17362 ...

2015-12-13 15:21:53 17362 [Note] InnoDB: Using atomics to ref count buffer pool pages

2015-12-13 15:21:53 17362 [Note] InnoDB: The InnoDB memory heap is disabled

2015-12-13 15:21:53 17362 [Note] InnoDB: Mutexes and rw\_locks use GCC atomic builtins

2015-12-13 15:21:53 17362 [Note] InnoDB: Memory barrier is not used

2015-12-13 15:21:53 17362 [Note] InnoDB: Compressed tables use zlib 1.2.3

2015-12-13 15:21:53 17362 [Note] InnoDB: Using CPU crc32 instructions

2015-12-13 15:21:53 17362 [Note] InnoDB: Initializing buffer pool, size = 128.0M

2015-12-13 15:21:53 17362 [Note] InnoDB: Completed initialization of buffer pool

2015-12-13 15:21:53 17362 [Note] InnoDB: The first specified data file ./ibdata1 did not exist: a

new database to be created!

2015-12-13 15:21:53 17362 [Note] InnoDB: Setting file ./ibdata1 size to 12 MB

2015-12-13 15:21:53 17362 [Note] InnoDB: Database physically writes the file full: wait...

2015-12-13 15:21:53 17362 [Note] InnoDB: Setting log file ./ib\_logfile101 size to 48 MB

2015-12-13 15:21:53 17362 [Note] InnoDB: Setting log file ./ib\_logfile1 size to 48 MB

2015-12-13 15:21:53 17362 [Note] InnoDB: Renaming log file ./ib\_logfile101 to ./ib\_logfile0

2015-12-13 15:21:53 17362 [Warning] InnoDB: New log files created, LSN=45781

2015-12-13 15:21:53 17362 [Note] InnoDB: Doublewrite buffer not found: creating new

2015-12-13 15:21:53 17362 [Note] InnoDB: Doublewrite buffer created

2015-12-13 15:21:53 17362 [Note] InnoDB: 128 rollback segment(s) are active.

2015-12-13 15:21:53 17362 [Warning] InnoDB: Creating foreign key constraint system tables.

2015-12-13 15:21:53 17362 [Note] InnoDB: Foreign key constraint system tables created

2015-12-13 15:21:53 17362 [Note] InnoDB: Creating tablespace and datafile system tables.

2015-12-13 15:21:53 17362 [Note] InnoDB: Tablespace and datafile system tables created.

2015-12-13 15:21:53 17362 [Note] InnoDB: Waiting for purge to start

2015-12-13 15:21:53 17362 [Note] InnoDB: 5.6.26 started; log sequence number 0

2015-12-13 15:21:53 17362 [Note] Binlog end

2015-12-13 15:21:53 17362 [Note] InnoDB: FTS optimize thread exiting.

2015-12-13 15:21:53 17362 [Note] InnoDB: Starting shutdown...

2015-12-13 15:21:54 17362 [Note] InnoDB: Shutdown completed; log sequence number 1625977

OK

Filling help tables...2015-12-13 15:21:54 0 [Warning] TIMESTAMP with implicit DEFAULT value is

deprecated. Please use --explicit\_defaults\_for\_timestamp server option (see documentation for

more details).

2015-12-13 15:21:54 0 [Note] /usr/local/mysql/bin/mysqld (mysqld 5.6.26) starting as process

17384 ...

2015-12-13 15:21:54 17384 [Note] InnoDB: Using atomics to ref count buffer pool pages

2015-12-13 15:21:54 17384 [Note] InnoDB: The InnoDB memory heap is disabled

2015-12-13 15:21:54 17384 [Note] InnoDB: Mutexes and rw\_locks use GCC atomic builtins

2015-12-13 15:21:54 17384 [Note] InnoDB: Memory barrier is not used

2015-12-13 15:21:54 17384 [Note] InnoDB: Compressed tables use zlib 1.2.3

2015-12-13 15:21:54 17384 [Note] InnoDB: Using CPU crc32 instructions

2015-12-13 15:21:54 17384 [Note] InnoDB: Initializing buffer pool, size = 128.0M

2015-12-13 15:21:54 17384 [Note] InnoDB: Completed initialization of buffer pool

2015-12-13 15:21:54 17384 [Note] InnoDB: Highest supported file format is Barracuda.

2015-12-13 15:21:54 17384 [Note] InnoDB: 128 rollback segment(s) are active.

2015-12-13 15:21:54 17384 [Note] InnoDB: Waiting for purge to start

2015-12-13 15:21:54 17384 [Note] InnoDB: 5.6.26 started; log sequence number 1625977

2015-12-13 15:21:55 17384 [Note] Binlog end

2015-12-13 15:21:55 17384 [Note] InnoDB: FTS optimize thread exiting.

2015-12-13 15:21:55 17384 [Note] InnoDB: Starting shutdown...

2015-12-13 15:21:56 17384 [Note] InnoDB: Shutdown completed; log sequence number 1625987

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/local/mysql/bin/mysqladmin -u root password 'new-password'

/usr/local/mysql/bin/mysqladmin -u root -h edu-mysql-02 password 'new-password'

Alternatively you can run:

/usr/local/mysql/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/local/mysql/bin/mysqld\_safe &

You can test the MySQL daemon with mysql-test-run.pl

cd mysql-test ; perl mysql-test-run.pl

Please report any problems at http://bugs.mysql.com/

The latest information about MySQL is available on the web at

Support MySQL by buying support/licenses at http://shop.mysql.com

New default config file was created as /usr/local/mysql/my.cnf and

will be used by default by the server when you start it.

You may edit this file to change server settings

WARNING: Default config file /etc/my.cnf exists on the system

This file will be read by default by the MySQL server

If you do not want to use this, either remove it, or use the

--defaults-file argument to mysqld\_safe when starting the server

[root@edu-mysql-01 mysql] # chown -Rf mysql:mysql /usr/local/mysql/my.cnf

19、不需要编辑/etc/my.cnf:

20、复制服务启动脚本：

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

21、启动 MySQL 服务：

# service mysql start

Starting MySQL.. SUCCESS!

（初次启动会在/usr/local/mysql 目录下生成 mysql.sock 文件）

22、设置 MySQL 开机自动启动服务：

# chkconfig mysql on

设置 MySQL 数据库 root 用户的本地登录密码（初始用户没有密码）：

# mysqladmin -u root password 'root'

23、登录并修改 MySQL 用户 root 的密码：

# mysql -uroot -p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.6.26-log Source distribution

Copyright (c) 2000, 2015, 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>

mysql> show databases;

**如果mysql遇到一些问题:**

修改 改 root 用 户 密码：

mysql> update user set Password = password('root') where User='root';

Query OK, 4 rows affected (0.00 sec)

Rows matched: 5 Changed: 4 Warnings: 0

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

允许 root 远程登录 ，设 置远程登录密码 ： www.roncoo.com

mysql> use mysql;

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'root' WITH

GRANT OPTION;

mysql> flush privileges;

mysql> exit;

