

平台搭建个人报告_16211112孙浩文

一．整体目标任务说明

1.1 整体目标

使用三台虚拟机搭建Hadoop集群环境，需要配置的组件包括Hadoop、ZooKeeper、HBase、Hive。最终完成Hadoop、HBase完全分布式的部署。

1.2 任务说明

搭建Hadoop集群环境至少需要3个节点（也就是3台服务器设备）。

我们组刚好一共有三位同学：16211109周宏建、16211112孙浩文、16211130董开宇。

组长16211109周宏建的虚拟机作为master节点，负责master节点所有组件的安装和部署。

组员16211112孙浩文的虚拟机作为slave1节点，负责slave1节点所有组件的安装和部署。

组员16211130董开宇的虚拟机作为slave2节点，负责slave2节点所有组件的安装和部署及小组整体报告撰写。

二．实验方案计划分析

2.1 环境说明

拥有的资源为三台虚拟机：

各节点之间局域网连接，可以相互ping通，各节点IP及登录信息如下：

Hostname	IP	User	Password
----------	----	------	----------

16211109-master	10.251.254.87	root	098765
16211112-slave1	10.251.254.252	root	098765
16211130-slave2	10.251.254.250	root	098765

三个节点均使用CentOS 7系统，为了便于维护，集群环境配置项一切涉及密码登录的地方均使用相同用户名、相同用户密码、相同Hadoop、ZooKeeper、HBase、Hive目录结构。

2.2 版本兼容性解决方案

我们采用的各组件版本为助教推荐搭配版本：

JDK : jdk1.8.0_201

Hadoop : Hadoop-2.7.1

Hbase : HBase-1.1.5-bin

Zookeeper : ZooKeeper-3.4.5

Hive : Hive-1.2.1-bin

2.3 计划的各节点角色及服务进程

Hostname	IP	节点角色及进程
16211109-master	10.251.254.87	Hadoop: master(NameNode、SecondaryNameNode、ResourceManager) ZooKeeper: follower(QuorumPeerMain) Hbase: master(HMaster、HRegionServer)
16211112-slave1	10.251.254.252	Hadoop: slave(DataNode、NodeManager) ZooKeeper: follower(QuorumPeerMain) Hbase: slave(HRegionServer)
16211130-slave2	10.251.254.250	Hadoop: slave(DataNode、NodeManager) ZooKeeper: leader(QuorumPeerMain) Hbase: slave(HRegionServer)

2.4 Hadoop集群各组件部署配置顺序

因为Hadoop集群的组件启动顺序与关闭顺序都有着严格的要求：

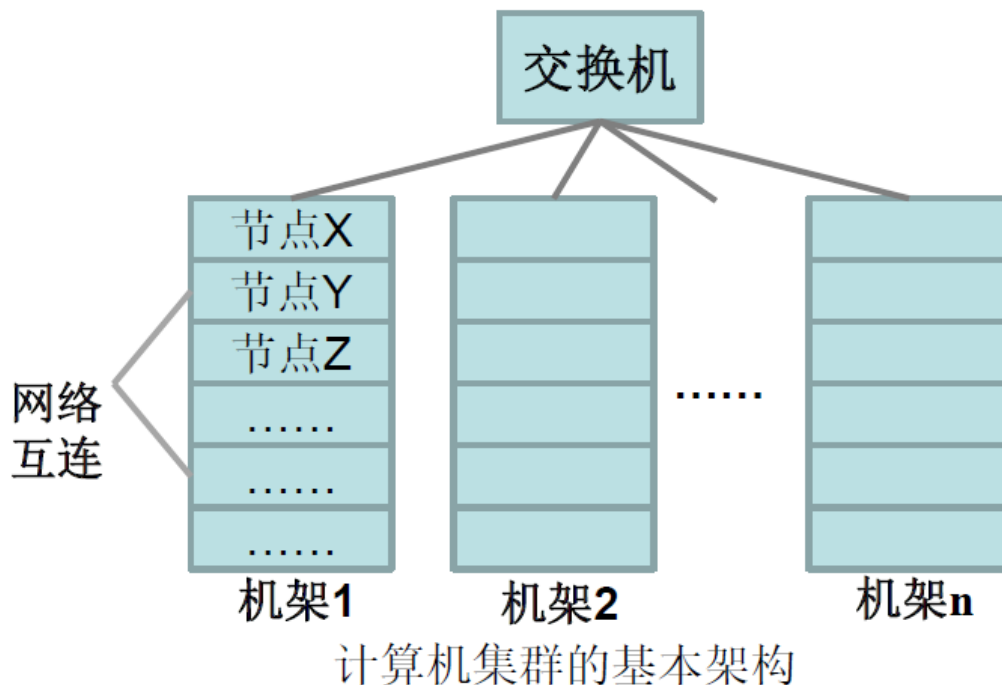
启动顺序：Hadoop->ZooKeeper->HBase->Hive

关闭顺序：Hive->HBase->ZooKeeper->Hadoop

为了最大程度避免配置过程出错，所以我们部署各组件的顺序也是按照启动顺序进行。

2.5 其他需要考虑的事项

通过本学期的大数据课程我们了解到，计算机集群的基本架构如图：



一个机架中运行着多个节点，一个集群由多个机架组成。

同一机架中的节点通过网络互联，不同机架的节点通过交换机互联。

所以我们在选定机器搭建Hadoop集群环境时也要将计算机集群的特点利用起来，所有的节点最好两两分布于不同的机架上。这样的话，如果某机架发生意外情况导致机架内所有节点不可逆损坏，受到影响的节点数量将是一个最低的值，从而可以最大程度得保证更多节点的运行，使集群系统尽最大可能正常提供服务，提高系统的可用性和容错性。

若较多的节点位于同一机架，则该机架若损坏将可能导致过多节点宕机进而整个集群系统彻底不可用，应该极力避免这种情况。

如果有条件的话，甚至可以不局限于物理空间限制，将集群的各个节点分散部署在全国乃至全球各地的机房，这样可以极大提高可用性和可靠性。

三 . 个人任务分工

本人负责Hadoop集群中slave1节点的配置，需要安装配置的组件包括Hadoop、ZooKeeper、HBase、Hive。

四 . 部署前的准备工作

4.1 添加hosts映射关系

在slave1节点上添加hosts映射关系：

```
$ vim /etc/hosts
```

添加内容如下：

```
10.251.254.87 16211109-master
```

```
10.251.254.252 16211112-slave1
```

```
10.251.254.250 16211130-slave2
```

4.2 集群之间ssh无密码登录

CentOS默认安装了ssh，如果没有需要先安装ssh。

集群环境的使用必须通过ssh无密码登陆来执行，本机登陆本机必须无密码登陆，主机与从机之间必须可以双向无密码登陆，从机与从机之间无限制。

4.2.1 开启Authentication免登陆

CentOS默认没有启动ssh无密登录,编辑\$ vim /etc/ssh/sshd_config，去掉以下两行注释，开启Authentication免登陆。

```
# RSAAuthentication yes
```

```
# PubkeyAuthentication yes
```

如果是root用户下进行操作，还要去掉 #PermitRootLogin yes的注释，允许root用户登录。

4.2.2 生成authorized_keys

输入命令，\$ ssh-keygen -t rsa，生成key，一直按回车。

就会在/root/.ssh生成：authorized_keys id_rsa.pub id_rsa 三个文件。

4.2.3 合并公钥到authorized_keys文件

对authorized_keys文件进行\$ chmod 600 authorized_keys操作，使当前用户具有 authorized_keys的读写权限。

拷贝完成后，在slave1机器上进行 service sshd restart 操作，重新启动ssh服务。

输入 ssh 10.251.254.xxx，测试能否无需输入密码连接另外两台机器。

4.3 安装jdk

新建目录：\$ mkdir /root/apps

解压安装：\$ tar -zxfjdk.tar.gz

修改配置文件：\$ vim /etc/profile，配置环境变量

export JAVA_HOME=/root/apps/jdk1.8.0_201

export PATH=\$JAVA_HOME/bin:\$PATH

重新加载配置文件使之生效\$ source /etc/profile

4.4 关闭防火墙和selinux

\$ systemctl stop firewalld.service

\$ systemctl disable firewalld.service

\$ vim /etc/sysconfig/selinux

SELINUX=enforcing 改为 SELINUX=disabled

\$ reboot

五. Hadoop集群安装配置

5.1 安装Hadoop

该步骤由master节点操作，配置完成后拷贝至slave节点

5.2 修改环境变量

修改配置文件：\$ vim /etc/profile，配置环境变量

```
export JAVA_HOME=/root/apps/jdk1.8.0_201
```

```
export HADOOP_HOME=/root/apps/Hadoop
```

```
export HADOOP_INSTALL=$HADOOP_HOME
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_HOME=$HADOOP_HOME
```

```
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export YARN_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export
```

```
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

重新加载配置文件使之生效\$ source /etc/profile

5.3 修改Hadoop配置文件

配置Hadoop的配置文件core-site.xml hdfs-site.xml mapred-site.xml yarn-site.xml

slaves

```
$cd /root/apps/Hadoop/etc/Hadoop
```

```
$vim core-site.xml
```

1.core-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://16211109-master:9000</value>
  </property>
  <property>
    <name>Hadoop.tmp.dir</name>
    <value>file:/root/apps/Hadoop/tmp</value>
  </property>
</configuration>
```

2.hdfs-site.xml

```
<configuration>
```

```

<property>
  <name>dfs.replication</name>
  <value>2</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/root/apps/Hadoop/tmp/dfs/name</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/root/apps/Hadoop/tmp/dfs/data</value>
</property>
<property>
  <name>dfs.namenode.secondary.http-address</name>
  <value>16211109-master:9001</value>
</property>
</configuration>

```

3.mapred-site.xml

```

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

4.yarn-site.xml

```

<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>16211109-master</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.log-aggregation.retain-seconds</name>
    <value>604800</value>
  </property>
</configuration>

```

5.slaves

```

16211112-slave1

```

5.4 各节点查看进程状态

启动成功各节点通过\$ jps查看进程：

```
[root@16211112-slave1 hadoop]# jps
3280 NodeManager
3337 Jps
3181 DataNode
```

六. Zookeeper集群安装配置

6.1 安装Zookeeper

该步骤只在master节点操作，配置完成后scp分发至slave节点

6.2 编辑myid文件

修改slave1节点上myid文件中的数字（slave1为2）

```
$ vim /root/apps/ZooKeeper/data/myid
```

6.3 启动ZooKeeper集群

对slave1节点：

```
$ cd /root/apps/ZooKeeper/bin
```

```
$ ./zkServer.sh start
```

6.4 成功标识

查看状态：

```
$ ./zkServer.sh status
```

```
$ jps
```



```
[root@16211112-slave1 bin]# ./zkServer.sh start
JMX enabled by default
Using config: /root/apps/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@16211112-slave1 bin]# ./zkServer.sh status
JMX enabled by default
Using config: /root/apps/zookeeper/bin/../conf/zoo.cfg
Mode: follower
[root@16211112-slave1 bin]# jps
3280 NodeManager
3779 QuorumPeerMain
3843 Jps
3181 DataNode
```

七. HBase集群安装配置

7.1 安装HBase

该步骤只在master节点操作，配置完成后scp分发至slave节点

7.2 修改配置文件

配置HBase的配置文件HBase-env.sh HBase-site.xml regionservers

```
$cd /root/apps/HBase/conf
```

```
$vim HBase-env.sh
```

1.HBase-env.sh

```
export JAVA_HOME=/root/apps/jdk1.8.0_201
export HBASE_CLASSPATH=/root/apps/Hadoop/etc/Hadoop/
export HBASE_MANAGES_ZK=false
```

2.HBase-site.xml

```
<configuration>
  <property>
    <name>HBase.root.dir</name>
    <value>hdfs://10.251.254.87:9000/HBase</value>
  </property>
  <property>
    <name>HBase.master</name>
    <value>16211109-master</value>
  </property>
  <property>
    <name>HBase.cluster.distributed</name>
    <value>true</value>
  </property>
</configuration>
```

```

</property>
<property>
  <name>HBase.ZooKeeper.property.clientPort</name>
  <value>2181</value>
</property>
<property>
  <name>HBase.ZooKeeper.quorum</name>
  <value>16211109-master,16211112-slave1,16211130-slave2</value>
</property>
<property>
  <name>ZooKeeper.session.timeout</name>
  <value>60000000</value>
</property>
<property>
  <name>dfs.support.append</name>
  <value>true</value>
</property>
</configuration>

```

3.regionServers

16211109-master

16211112-slave1

16211130-slave2

7.3 jps查看各节点进程状态

```

[root@16211112-slave1 logs]# jps
3280 NodeManager
3779 QuorumPeerMain
6295 HRegionServer
6522 Jps
3181 DataNode

```

八. Hive集群安装配置

8.1 安装Hive

进入apps目录 : & cd /root/apps

解压安装 : \$ tar -xvf apache-Hive-1.2.1-bin.tar.gz

\$ mv apache-Hive-1.2.1-bin Hive

8.2 修改环境变量

编辑配置文件 : \$ vim /etc/profile , 修改环境变量

```
export HIVE_HOME=/root/apps/Hive
```

```
export
```

```
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

```
:$HIVE_HOME
```

重新加载配置文件使之生效\$ source /etc/profile

8.3 修改配置文件

进入/root/apps/Hive/conf目录

```
$ cd /root/apps/Hive/conf
```

拷贝Hive-env.sh.template文件为Hive-env.sh

```
$ cp Hive-env.sh.template Hive-env.sh
```

编辑修改：\$ vim Hive-env.sh

```
HADOOP_HOME=/root/apps/Hadoop
```

8.4 启动Hive并测试

```
$ Hive
```

启动后新建一个表测试：

```
hive> create table hivetest (name string, age int)
> ;
OK
Time taken: 1.275 seconds
hive> show tables;
OK
hivetest
Time taken: 0.127 seconds, Fetched: 1 row(s)
```

8.5 安装mysql并配置

解压：\$ tar -xvf mysql-5.7.22-1.el6.x86_64.rpm-bundle.tar

依次执行：

```
$ sudo rpm -ivh --force mysql-community-common-5.7.16-1.el6.x86_64.rpm
```

```
$ yum remove mysql-libs
```

```
$ sudo rpm -ivh --force mysql-community-libs-5.7.16-1.el6.x86_64.rpm
```

```
$ sudo rpm -ivh --force mysql-community-client-5.7.16-1.el6.x86_64.rpm
```

```
$ sudo rpm -ivh --force --nodeps mysql-community-server-5.7.16-1.el6.x86_64.rpm
```

8.6 启动mysql

```
$ service mysqld start
```

在log文件中查找root密码：\$ vim /var/log/mysqld.log

```
2019-07-01T10:21:48.360592Z 1 [Note] A temporary password is generated for
root@localhost: .05em(zeKw1j
```

登录mysql：\$ mysql -u root -p

输入密码：.05em(zeKw1j

修改密码：

```
mysql> set global validate_password_policy=0;
```

```
mysql> set global validate_password_length=4;
```

```
mysql> set password=password('000000');
```

创建Hive用户：

```
mysql> CREATE USER 'Hive'@'%' IDENTIFIED BY '000000';
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'Hive'@'%' IDENTIFIED BY '000000';
```

```
mysql> FLUSH PRIVILEGES;
```

8.7 配置Hive

解压mysql-connector-java：

```
$ tar -zxf mysql-connector-java-5.1.40.tar.gz
```

复制jar包到Hive的lib目录下：

```
$ cp mysql-connector-java-5.1.40-bin.jar /root/apps/Hive/lib
```

进入/root/apps/Hive/conf目录

```
$ cd /root/apps/Hive/conf
```

新建：\$ vim Hive-site.xml

添加如下内容：

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/Hive?
createDatabaseIfNotExist=true</value>
    <description>JDBC connect string for a JDBC metastore</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>Hive</value>
    <description>username to use against metastore database</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>000000</value>
    <description>password to use against metastore database</description>
  </property>
</configuration>

```

8.8 运行Hive实例并测试

\$ Hive

```

hive> show tables;
OK
Time taken: 0.025 seconds

```

九. 实验成果

9.1 Hadoop配置成功

启动成功各节点

通过\$ jps查看进程：

```

[root@16211112-slave1 hadoop]# jps
3280 NodeManager
3337 Jps
3181 DataNode

```

9.2 ZooKeeper配置成功

查看状态：

```
$ ./zkServer.sh status
```

```
$ jps
```

```
[root@16211112-slave1 bin]# ./zkServer.sh start
JMX enabled by default
Using config: /root/apps/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@16211112-slave1 bin]# ./zkServer.sh status
JMX enabled by default
Using config: /root/apps/zookeeper/bin/../conf/zoo.cfg
Mode: follower
[root@16211112-slave1 bin]# jps
3280 NodeManager
3779 QuorumPeerMain
3843 Jps
3181 DataNode
```

打开ZooKeeper客户端无报错

```
[root@16211112-slave1 bin]# ./zkCli.sh
Connecting to localhost:2181
2019-07-03 11:40:49.362 [myid:] - INFO [main:Environment@100] - Client environment:zookeeper.version=3.4.5-1392890, built on 09/30/2012 17:52 GMT
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:host.name=16211112-slave1
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:java.version=1.8.0_201
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:java.vendor=Oracle Corporation
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:java.home=/root/apps/jdk1.8.0_201/jre
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:java.class.path=/root/apps/zookeeper/bin/../build/classes:/root/apps/zookeeper/bin/../build/lib/*:/root/apps/zookeeper/bin/../lib/slf4j-log4j12-1.6.1.jar:/root/apps/zookeeper/bin/../lib/slf4j-api-1.6.1.jar:/root/apps/zookeeper/bin/../lib/netty-3.2.2.Final.jar:/root/apps/zookeeper/bin/../lib/log4j-1.2.15.jar:/root/apps/zookeeper/bin/../lib/jline-0.9.94.jar:/root/apps/zookeeper/bin/../zookeeper-3.4.5.jar:/root/apps/zookeeper/bin/../src/java/lib/*:/root/apps/zookeeper/bin/../conf:
2019-07-03 11:40:49.366 [myid:] - INFO [main:Environment@100] - Client environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib
2019-07-03 11:40:49.367 [myid:] - INFO [main:Environment@100] - Client environment:java.io.tmpdir=/tmp
2019-07-03 11:40:49.367 [myid:] - INFO [main:Environment@100] - Client environment:java.compiler=<NA>
2019-07-03 11:40:49.367 [myid:] - INFO [main:Environment@100] - Client environment:os.name=Linux
2019-07-03 11:40:49.367 [myid:] - INFO [main:Environment@100] - Client environment:os.arch=amd64
2019-07-03 11:40:49.367 [myid:] - INFO [main:Environment@100] - Client environment:os.version=3.10.0-229.el7.x86_64
2019-07-03 11:40:49.368 [myid:] - INFO [main:Environment@100] - Client environment:user.name=root
2019-07-03 11:40:49.368 [myid:] - INFO [main:Environment@100] - Client environment:user.home=/root
2019-07-03 11:40:49.368 [myid:] - INFO [main:Environment@100] - Client environment:user.dir=/root/apps/zookeeper/bin
2019-07-03 11:40:49.369 [myid:] - INFO [main:ZooKeeper@438] - Initiating client connection, connectString=localhost:2181 sessionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@4b85612c
Welcome to ZooKeeper!
```

9.3 HBase安装配置成功

jps查看各节点进程状态

```
[root@16211112-slave1 logs]# jps
3280 NodeManager
3779 QuorumPeerMain
6295 HRegionServer
6522 Jps
3181 DataNode
```

9.4 Hive安装配置成功

运行Hive实例并测试

```
$ Hive
```

```
hive> show tables;
OK
Time taken: 0.025 seconds
```

十. 遇到的问题及解决方案

10.1 scp命令发送时显示“没有那个目录”

解决方案：

对于各个节点：

.ssh目录需要700权限

```
$ sudo chmod 700 ~/.ssh
```

.ssh目录下的authorized_keys文件需要600或644权限

```
$ sudo chmod 644 ~/.ssh/authorized_keys
```

10.2 执行/root/apps/Hadoop/etc/Hadoop/start-all.sh启动Hadoop时

报错：“Error:JAVA_HOME is not set and could not be found”

解决方案：

对于各个节点：

```
$ vim /root/Hadoop/etc/Hadoop/Hadoop-env.sh
```

将 `export JAVA_HOME=$JAVA_HOME`

修改为 `export JAVA_HOME=/root/apps/jdk1.8.0_201`

然后再次执行脚本start-all.sh即可

10.3 执行/root/apps/HBase/bin/start-base.sh后某节点无

HRegionServer

```
[root@16211112-slave1 conf]# jps
3280 NodeManager
3779 QuorumPeerMain
5959 Jps
3181 DataNode
```

```
[root@16211130-slave2 conf]# jps
5376 HRegionServer
3252 QuorumPeerMain
2661 DataNode
5531 Jps
2734 NodeManager
```

由于节点间时间不同步所致：

```
[root@16211109-master hbase]# date
2019年 07月 01日 星期一 17:11:10 CST
```

```
[root@16211130-slave2 conf]# date
2019年 07月 01日 星期一 17:11:10 CST
```

```
[root@16211112-slave1 conf]# date
2019年 07月 01日 星期一 17:13:18 CST
```

解决方案：各节点安装ntp服务使时间与网络同步

```
# yum -y install ntp ntpdate
```

[ntpdate cn.pool.ntp.org](http://ntpdate.cn.pool.ntp.org)

10.4 安装mysql-libs时报错：错误：依赖检测失败：mariadb-libs 被mysql-community-libs-5.7.22-1.el6.x86_64 取代

解决方案：

使用\$ `yum remove mysql-libs`命令删除原本的mysql-libs

重新安装即可

10.5 安装mysql-server时报错：错误：依赖检测失败：libsasl2.so.20 (64bit) 被 mysql-community-server-5.7.22-1.el6.x86_64 需要

解决方案：

安装命令加两个参数即可：

\$ `sudo rpm -ivh mysql-community-server-5.7.22-1.el6.x86_64.rpm --force --nodeps`

10.6 修改mysql密码时报错：

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

解决方案：

先修改密码安全等级：

`mysql> set global validate_password_policy=0;`

`mysql> set global validate_password_length=4;`

再修改密码：

`mysql> set password=password('000000');`