**结构化服务器CMS安装文档**

**简介**：如下是针对结构化服务器CMS涉及的Linux(Centos7)运行环境下软件部署。

# 环境配置

## 硬件环境

1台测试物理机，配置如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点ip | 内存 | 硬盘 | cpu | 操作系统 |
| 172.16.3.89 | 256G | 4\*550G | 2CPU\*12核\*2线程 | CentOS-7.1（64位） |

## 软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 版本号 | 下载地址 | 安装路径 | 说明 |
| JDK | 1.8.0\_74 | http://download.oracle.com/otn-pub/java/jdk/8u74-b02/jdk-8u74-linux-x64.tar.gz | /opt/jdk1.8.0\_74 | Java JDK环境 |
| kafka | kafka\_2.11-0.10.2.0 | http://apache.opencas.org/kafka/0.10.2.0/kafka\_0.10.2.0.tgz | /opt/kafka | Kafka服务 |
| KafkaOffsetMonitor | 0.2.0 | https://github.com/quantifind/KafkaOffsetMonitor/releases/download/v0.2.0/KafkaOffsetMonitor-assembly-0.2.0.jar | /opt/kafkaOffsetMonitor | Kafka web监控 |
| tomcat | 8.5 | http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz | /opt/tomcat8 | Tomcat服务 |
| mysql | 5.7 | https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.18-1.el7.x86\_64.rpm-bundle.tar |  | Mysql数据库 |
| elasticsearch | 5.4.3 | https://www.elastic.co/downloads | /opt/elasticsearch-5.4.3 | ES索引库 |
| Elasticsearch Head | 5.x | https://www.elastic.co/downloads | /opt/elasticsearch-head | ES-head插件 |
| node | 6.11 | https://nodejs.org/en/download/ | /opt/node-v6.11.0-linux-x64 | ES-head插件依赖nodeJS |

# 软件环境安装

## 准备工作

### 安装操作系统centos7

安装过程参考相关文档

### 建立hadoop用户

**·说明1：**USERGROUPS\_ENAB存于/etc/login.defs文件中，在系统新安装时默认为yes，如果不确定是否为yes，还是使用方法A。若添加用户组hadoop时提示已经存在则忽略

**·方法A:**

添加用户组：groupadd hadoop

新建用户并添加到组：useradd -g hadoop -m hadoop

**·方法B：**

当USERGROUPS\_ENAB为yes时，也可以直接执行下面一条：

添加用户并自动创建同名的组并添加： useradd -m hadoop

**·说明2：**系统需要添加一个hadoop用户和组，用于安装java的运行环境

·说明3：jdk8、tomcat8、kafka、KafkaOffsetMonitor均切换到hadoop用户安装。mysql5.7切换到root用户安装。

### 创建hadoop的用户密码

Root下执行passwd hadoop （输入新密码，重复密码）

说明：新建立用户后需要用passwd更改用户密码，否则会出现无法登陆或者hadoop身份使用sudo时密码验证不通过的情况。

### 远程用户登录说明

在远程终端登录（ssh）时需要使用su – hadoop来切换用户身份而不是su hadoop，出现一些程序找不到命令的情况往往是后者方式登录。（前者指令可能会被word转义为其它字符，请自行输入这个符号为英文字符的减号）

### Opt目录授权

执行su切换至root用户，执行chown –R hadoop:hadoop /opt/

## 安装JDK8

### 检查

执行java –version

如果输出版本为java version “1.8.xxxx”则表示已经安装JDK8

如果发现已安装JDK8则跳过2.2章

如果没有安装，会提示错误

### 解压

步骤1：执行tar -zxvf jdk-8u74-linux-x64.tar.gz解压文件

步骤2：移动到/opt目录

### 配置环境变量

步骤1：在~/.bash\_profile中加入如下设置：

JAVA\_HOME=/opt/jdk1.8.0\_74

PATH=$JAVA\_HOME/bin:$PATH

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

export JAVA\_HOME

export PATH

export CLASSPATH

步骤:2：执行source ~/.bash\_profile命令使之生效,用java –version验证安装是否成功.

说明1：以hadoop身份在命令行中执行vi ~/.bash\_profile打开编辑进行修改。CentOS下的vi可以按下i后用方向键控制文字游标，ESC退出编辑，:wq回车保存并退出。其它版本的linux在vi编辑上存在差异需参考有关文档操作。

说明2：如果zookeeper、kafka无法启动，可以尝试source ~/.bash\_profile使环境变量重新生效再运行

## 安装tomcat8

### 获取

如果没有tomcat安装包，则前往<https://tomcat.apache.org/>官网下载：

执行wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz

### 解压

步骤1：执行tar zxvf apache-tomcat-8.5.20.tar.gz

步骤2：执行mv apache-tomcat-8.5.20.tar.gz /opt/tomcat8

### 配置

步骤1：

执行

cd /opt/tomcat8

vi bin/catalina.sh

在开始处加入如下行

JAVA\_OPTS="-server -Xms4096m -Xmx4096m -Djava.awt.headless=true"

说明1：JVM参数具体可以根据服务器硬件配置和其它软件使用情况做适当调整，但最大值也不要超过实际物理内存的2/3。

步骤2：

执行vi conf/server.xml

Vi编辑器中执行 :set nu 显示行号

按下i进入insert模式

在77行附近找到<Connector executor="tomcatThreadPool"行

标签中加入：

URIEncoding="UTF-8"

加入后如下：

<Connector executor="tomcatThreadPool"

port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" URIEncoding="UTF-8"/>

### 启动

步骤1：执行sh /opt/tomcat8/bin/startup.sh

步骤2：执行netstat -antp|grep 8080 查看是否有java进程使用了8080。如果有则说明启动成功

### 停止

执行sh /opt/tomcat8/bin/shutdown.sh

或ps查看进程id（tomcat进程名运行为java），执行kill -9 pid (pid为tomcat进程id)

### 部署程序

本地应用程序打包后拷贝到/opt/tomcat8/webapps/ROOT下，然后启动tomcat即可。

在/opt/ 目录下创建mkdir icv\_system ,拷贝ivu的mode 目录到该文件夹下：

（即：/opt/icv\_system/model）

修改cmsService.properties中的

ICV\_SYSTEM=/opt/icv\_system

jdbc、kafka、接口path等参数为实际安装环境IP：port

### 补充说明

为方便浏览器页面部署与管理，可以修改/opt/tomcat8/conf/tomcat-users.xml 中<tomcat-users>标签内加入如下配置：

<tomcat-users>

<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

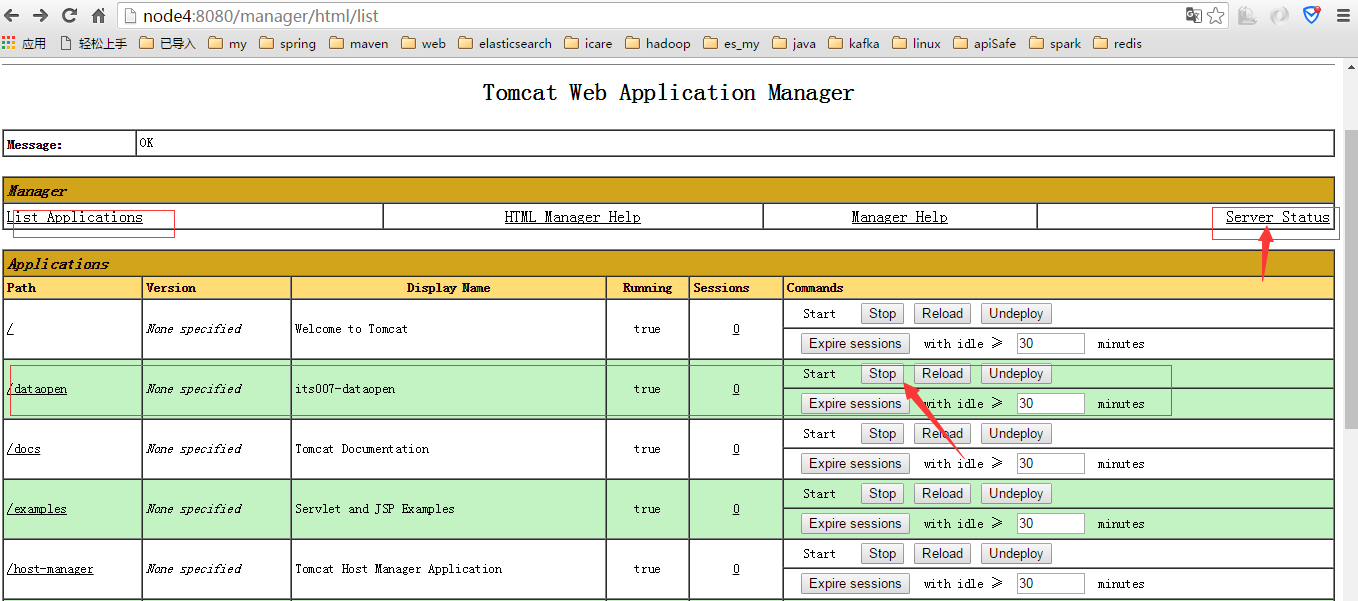
<user username="admin" password="admin" roles="manager-gui"/>

</tomcat-users>

保存重启服务，浏览器登录http://主机地址:8080/manager输入配置的用户/密码(admin/admin)

如果是本机登录ip则为127.0.0.1

登录后可对部署的应用进行管理和查看服务运行情况，如下图：



Tomcat的logs目录下的catalina.out记录了全部的运行日志，然而按天记录的日志也已经有了，这个文件会占用很大的空间且是不必要的，因此可以设置为空

执行rm –f catalina.out删除文件

执行ln –s /dev/null catalina.out建立软连接

那么此文件将保持0字节

服务器运行日志可以在catalina.2017-08-18.log等文件当中查看

## 安装Shell脚本

### 说明

Shell脚本存放在安装包的cmsshells下，脚本system\_commander\_with\_java.sh需要在root权限下开机运行。作用是为tomcat下部署的cms服务提供系统操作接口，而不再需要root权限执行java程序。

### 安装

·切换到root用户

·将cmsshells文件夹放在/opt/下：执行cp cmsshells /opt/cmsshells

·授予脚本可执行权限：chmod +x system\_commander\_with\_java.sh

·添加脚本自启方法有两种（两者等效）：

方法一：执行sh system\_commander\_with\_java.sh启动脚本，脚本会检查自启状态并自动添加自启

方法二：执行chmod +x /etc/rc.d/rc.local为此文件添加可执行权限，

执行vi /etc/rc.d/rc.local编辑文件，

添加条目 nohup sh /opt/cmsshells/system\_commander\_with\_java.sh &

Esc,输入:wq保存退出

·卸载程序时，取消自启删除/etc/rc.d/rc.local内的相关条目并且不再执行脚本即可

### 日志查看

每次脚本启动时都会在脚本目录下的log文件夹内输出日志文件，文件名格式如

system\_commander\_with\_java(2017-08-10----17:40:52).log

其中时间为启动时刻的时间

### 网卡配置锁定

脚本被允许自启、运行期间，网卡配置会被锁定，建议通过本系统的系统设置页面对网络配置进行修改。

## mysql5.7安装

### 配置YUM源

# 下载mysql源安装包

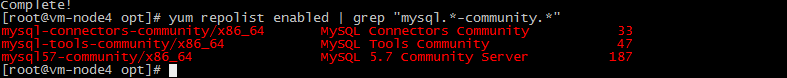
shell> wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

# 安装mysql源

shell> yum localinstall mysql57-community-release-el7-8.noarch.rpm

检查mysql源是否安装成功

shell> yum repolist enabled | grep "mysql.\*-community.\*"

****

出现上图界面安装成功

### 安装MySQL

shell> yum install mysql-community-server

确认安装后会下载大约190M的数据

启动MySQL服务

shell> systemctl start mysqld

查看是否启动成功

netstat -antp |grep 3306 发现已经启动OK

### 设置开机启动

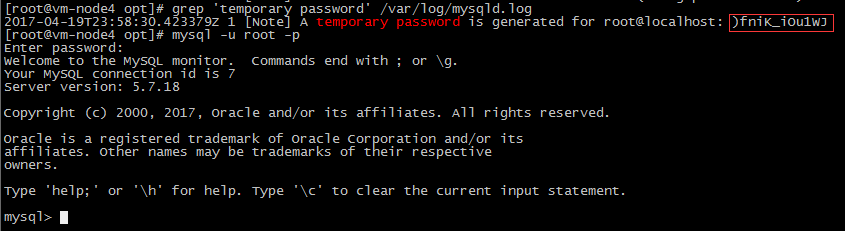
shell> systemctl enable mysqld

shell> systemctl daemon-reload

### 修改root默认密码

mysql安装完成之后，在/var/log/mysqld.log文件中给root生成了一个默认密码。通过下面的方式找到root默认密码，然后登录mysql进行修改：

shell> grep 'temporary password' /var/log/mysqld.log



mysql>ALTER USER 'root'@'localhost' IDENTIFIED BY "123456";

注意：mysql5.7默认安装了密码安全检查插件（validate\_password），默认密码检查策略要求密码必须包含：大小写字母、数字和特殊符号，并且长度不能少于8位。否则会提示ERROR 1819 (HY000): Your password does not satisfy the current policy requirements错误，

在/etc/my.cnf文件添加validate\_password\_policy配置，指定密码策略

添加my.cnf文件中添加如下配置禁用即可：

validate\_password = off

↑添加上述选项之前请保证my.cnf中有插件加载项说明

plugin-load=validate\_password.so

### 添加远程登录用户

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

重新启动mysql服务使配置生效：

systemctl restart mysqld

注意关闭防火墙或开放3306访问端口：

systemctl stop firewalld.service

禁用防火墙

systemctl mask firewalld

### 配置默认编码为utf8

修改/etc/my.cnf配置文件，在[mysqld]下添加编码配置，如下所示：

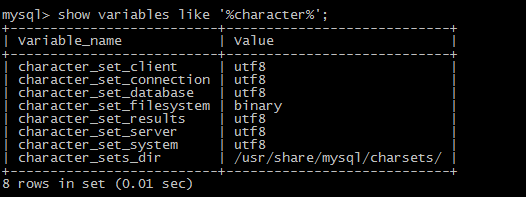
[mysqld]

character\_set\_server=utf8

init\_connect='SET NAMES utf8'

重新启动mysql服务，查看[**数据库**](http://lib.csdn.net/base/14)默认编码如下所示：

show variables like '%character%';



**默认配置文件路径：**

配置文件：/etc/my.cnf 中查看。

数据库文件：/var/lib/mysql/ (如果要专门挂到指定盘，可在初始化数据库前修改/etc/my.cnf中该配置后在初始化)

日志文件：/var/log/mysqld.log

**启动/重启**

systemctl start mysqld

systemctl restart mysqld

systemctl stop mysqld

-------------------------------------------------------------------------------------------------安装完成

### 其他方式安装说明

如果采用rpm安装包安装参考如下：

1.安装前需要先卸载mariadb-lib

rpm -qa|grep mariadb

rpm -e mariadb-libs-5.5.41-2.el7\_0.x86\_64 --nodeps

(rpm -e --nodeps xxxx 强制卸载)

2. 下载安装包：mysql-5.7.18-1.el7.x86\_64.rpm-bundle.tar放到/opt/目录下

解压：tar -xvf mysql-5.7.18-1.el7.x86\_64.rpm-bundle.tar

3. 执行下列安装：（请按照如下顺序安装，有依赖）

rpm -ivh mysql-community-common-5.7.18-1.el7.x86\_64.rpm

rpm -ivh mysql-community-libs-5.7.18-1.el7.x86\_64.rpm

rpm -ivh mysql-community-client-5.7.18-1.el7.x86\_64.rpm

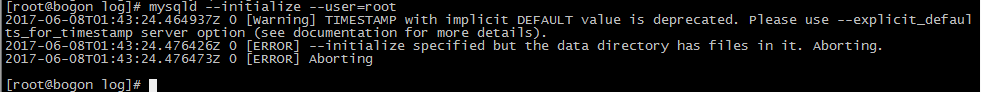
rpm -ivh mysql-community-server-5.7.18-1.el7.x86\_64.rpm

**root用户下执行：mysqld --initialize --user=mysql初始化**

后续安装设置同前2.4.3~2.4.6步骤

**常见错误：**

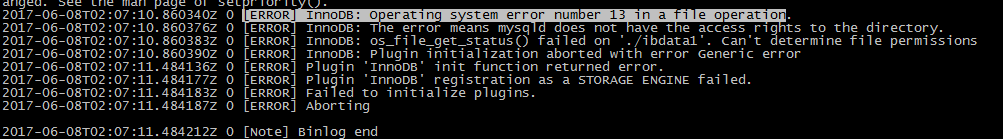
mysql启动初始化等错误明细可以到：/var/log/mysqld.log 查看具体原因解决。



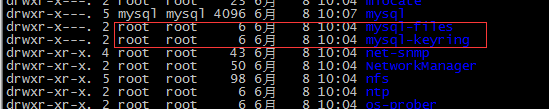
rm -rf /var/lib mysql下已有文件重新执行初始化

启动失败：

systemctl status mysqld.service和journalctl -xe查看

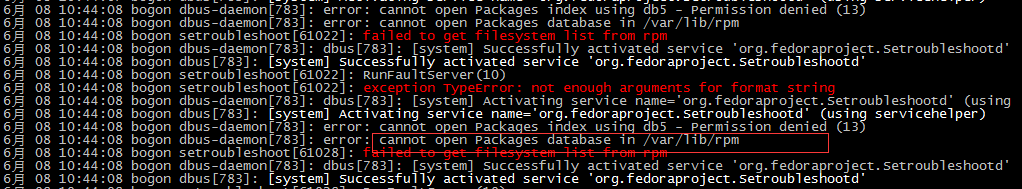


发现权限问题



chown -R mysql:mysql /var/lib/mysql\*

重启mysql任然错误，查看日志



chown -R mysql:mysql /var/lib/rpm

修改配置：

vi /etc/selinux/config

SELINUX=permissive

重启服务reboot后重新启动mysql即可

客户端可以使用SQLyog 等工具连接mysql

### mysql主从复制配置

 今天说一下MySQL的主从复制如何做到！

准备工作：

1. 两个虚拟机：我这里用的是CentOS5.5，IP地址分别是192.168.1.101 和192.168.1.105；
2. 101做主服务器，105做从服务器（**都已经安装相同版本的Mysql,而且都已经建好要相同数据库的结构**）；

好了，现在开始吧，来看看这听起来高大上的主从复制是怎么回事。

原理：mysql要做到主从复制，其实依靠的是二进制日志，即：假设主服务器叫A，从服务器叫B；主从复制就是

　　   B跟着A学，A做什么，B就做什么。那么B怎么同步A的动作呢？现在A有一个日志功能，把自己所做的增删改查的动作

　　   全都记录在日志中，B只需要拿到这份日志，照着日志上面的动作施加到自己身上就可以了。这样就实现了主从复制。

扩展：MYSQL还有一种日志叫做：慢日志

　　   可以设置一个时间，那么所有执行时间超过这个时间的SQL都会被记录下来。这样就可以通过慢日志快速的找到网站中SQL的瓶颈来进行优化。

         大家有时间可以研究一下，这里不多做介绍。

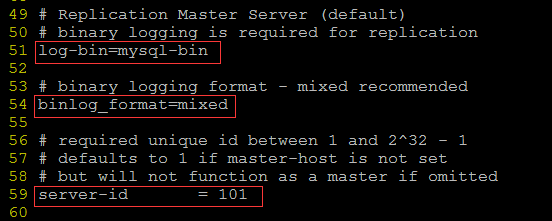
实现步骤：

1.首先修改mysql的配置文件，使其支持二进制日志功能。

打开主服务器的mysql配置文件：my.conf

代码：# vi /etc/my.cnf

加入如下三行代码：



参数解释：log-bin=mysql-bin  //将mysql二进制日志取名为mysql-bin

　　　　   binlog\_format=mixed //二进制日志的格式，有三种：statement/row/mixed,具体分别不多做解释，这里使用mixed

　　　　   server-id=101 //为服务器设置一个独一无二的id便于区分，这里使用ip地址的最后一位充当server-id

配置完成，:wq 保存，重启mysql

重启mysql命令：# service mysqld restart

同样的，进入从服务器，配置从服务器的my.cnf,重复步骤1即可，

唯一的区别是，server-id要改成从服务器的ip尾位，即server-id=105;其他两项是一样的，保存，并重启mySQL；

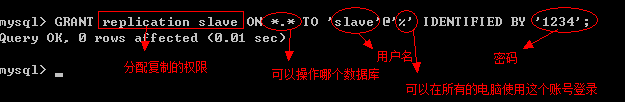
2.在主服务器上为从服务器分配一个账号，就像一把钥匙，从服务器拿着这个钥匙，才能到主服务器上来共享主服务器的日志文件。

进入主服务器的mysql界面，

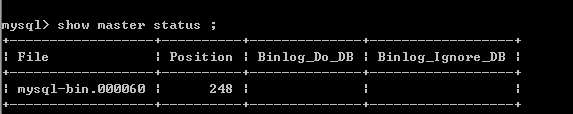
命令： # mysql -u root -p 111111     //我这里mysql账号是root，密码是111111

在mysql操作界面下，输入下面一行命令：

GRANT replication slave ON \*.\* TO 'slave'@'%' IDENTIFIED BY '111111';



3.查看主服务器BIN日志的信息（执行完之后记录下这两值，然后在配置完从服务器之前不要对主服务器进行任何操作，因为每次操作数据库时这两值会发生改变）.



4.设置从服务器

进入从服务器mysql

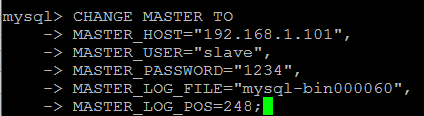
命令： # mysql -u root -p111111

关闭slave（如果你以前配置过主从的话，一定要先关闭）

命令：stop slave;

开始配置：

输入下面代码即可：



参数解释：MASTER\_HOST  :  设置要连接的主服务器的ip地址

　　　　   MASTER\_USER  :  设置要连接的主服务器的用户名

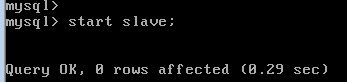
　　　　   MASTER\_PASSWORD  :  设置要连接的主服务器的密码

   　　　　MASTER\_LOG\_FILE  :  设置要连接的主服务器的bin日志的日志名称，即第3步得到的信息

   　　　　MASTER\_LOG\_POS  :  设置要连接的主服务器的bin日志的记录位置，即第3步得到的信息，（这里注意，最后一项不需要加引号。否则配置失败）

先在从服务器配置完成，启动从服务器：

命令： start slave;



5.查看是否配置成功：

命令： show slave status;

https://images2015.cnblogs.com/blog/896490/201603/896490-20160311221434663-456558902.png

上面两项均为yes，说明配置成功，否则，请重复前面的步骤。

ok，到这里MySQL的主从复制就配置完了，其实明白原理的话还是很简单的，

### mysql主主复制配置

#### 实现原理

     主主复制即在两台MySQL主机内都可以变更数据，而且另外一台主机也会做出相应的变更。聪明的你也许已经想到该怎么实现了。对，就是将两个主从复制有机合并起来就好了。只不过在配置的时候我们需要注意一些问题，例如，主键重复，server-id不能重复等等。

#### 配置文件修改

**主服务器（192.168.1.101）**

my.cnf文件中新增如下属性

sync\_binlog = 1

auto\_increment\_increment = 2 #步进值auto\_imcrement。一般有n台主MySQL就填n

auto\_increment\_offset = 1 #起始值。一般填第n台主MySQL。此时为第一台主MySQL

replicate-do-db = lnpcmsdb #要同步的数据库

replicate-ignore-db = mysql,information\_schema #要忽略的数据库

**从服务器（**192.168.1.105**）**

my.cnf文件中新增如下属性

auto-increment-increment = 2 #步进值auto\_imcrement。一般有n台主MySQL就填n

auto-increment-offset = 2 #起始值。一般填第n台主MySQL。此时为第二台主MySQL

replicate-do-db=lnpcmsdb #要同步的数据库

#### 开始构建主主复制

 因为主主复制是两个主从复制组合一起，所以我就接着上面主从复制接着配置。

 第一步：

     在从服务器（192.168.1.105）中创建一个供主服务器（192.168.1.101）主机中可以登录的MySQL用户，授权复制的权限,用户名：slave 密码：123456。

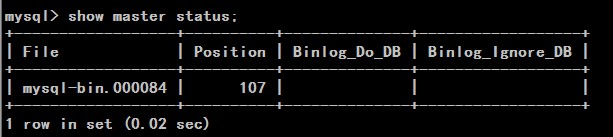
          mysql>GRANT REPLICATION SLAVE ON \*.\* TO ‘slave’@’%’ IDENTIFIED BY ‘1234546’;

           mysql>FLUSH PRIVILEGES;

   第二步：

       在从服务器（192.168.1.105）查看二进制日志名和位置

           mysql>show master status;



  第三步：

在主服务器（192.168.1.101）中，执行

mysql –uroot –p123456;

        mysql>CHANGE MASTER TO

        MASTER\_HOST=’ 192.168.1.105’,

        MASTER\_USER=’slave’,

        MASTER\_PASSWORD=’123456’,

        MASTER\_LOG\_FILE=’mysql-bin.000084’,

        MASTER\_LOG\_POS=107;

#### 测试主主复制

在两台服务器中分别执行

stop slave;

start slave；

show slave status; #查看运行状态

https://images2015.cnblogs.com/blog/896490/201603/896490-20160311221434663-456558902.png

上面两项均为yes，说明配置成功，否则，请重复前面的步骤。

接下来可以通过修改数据库的数据，看两台服务器的数据是否一致来进行验证。

## KAFKA安装

根据产品需求定位，目前kafka采用单机服务方式部署(1 broker /1 partition)，后期根据产品需求可以做集群配置。

### 解压

执行解压tar -zxvf kafka\_2.11-0.10.2.0.tgz

执行 mv kafka\_2.11-0.10.2.0 /opt/kafka\_2.11-0.10.2.0 移动到opt下面

切换工作目录执行cd /opt/kafka\_2.11-0.10.2.0

在该目录下创建data目录:mkdir data

### 配置

步骤1：Kafka目录下server.properties修改内容：

35行 listeners=PLAINTEXT://127.0.0.1:9092

63行log.dirs=/opt/kafka\_2.11-0.10.2.0/data/kafka-logs

步骤2：修改zookeeper.properties

dataDir=/opt/kafka\_2.11-0.10.2.0/data/zookeeper

### 启动zk服务

执行cd /opt/kafka\_2.11-0.10.2.0/bin

执行./zookeeper-server-start.sh ../config/zookeeper.properties >> /dev/null&

### 启动kafka服务

./kafka-server-start.sh ../config/server.properties >> /dev/null&

查看是否启动成功

Jps ，如果看到QuorumPeerMain和Kafka进程，则启动成功



说明：目前单机版kafka较简单，采用kafka自带zk方式安装启动，采用默认配置。后续根据实际情况可做配置参数的调整和优化。

## kafkaOffsetMonitor监控插件安装

### 解压

1.执行解压tar -zxvf kafkaOffsetMonitor.tar.gz

2.执行移动 mv kafkaOffsetMonitor /opt/kafkaOffsetMonitor

### 配置

1.执行cd /opt/kafkaOffsetMonitor 切换目录

1.执行vi start.sh打开编辑，修改需要监听zk地址端口

（使用本机地址127.0.0.1:2181）

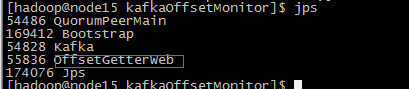
3.port 为kafkaOffsetMonitor监听的web页访问端口

（譬如port：8086 访问web地址为：http://主机地址:8086）

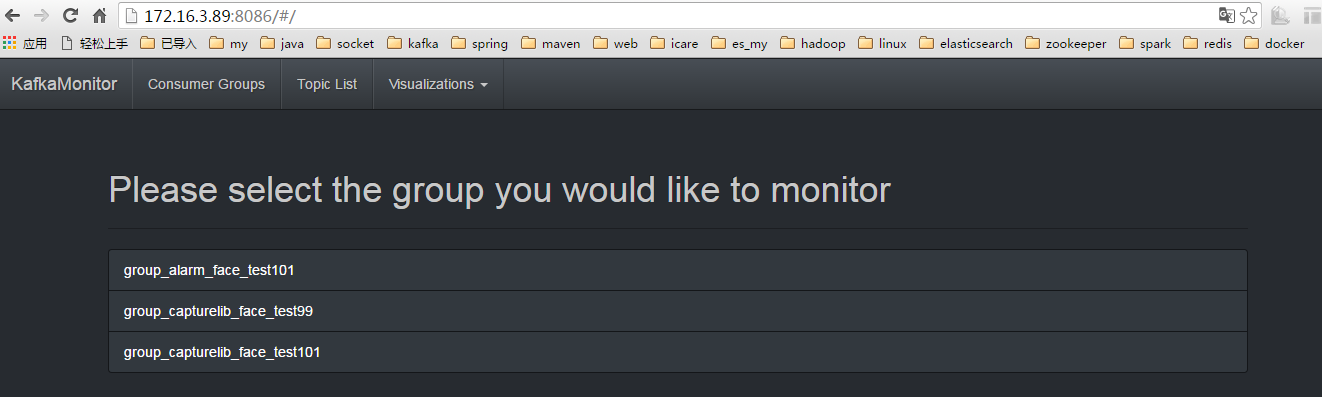
### 启动web监听

1.执行sh start.sh

2.执行jps查看OffsetGetterWeb进程即启动成功



或者浏览器访问到如下地址页面即成功



## Elasticsearch安装

### 系统配置

安装启动ES服务前，先修改如下系统配置, 否则ES不能正常启动

切换至root用户，然后执行以下操作，条目不存在则创建

执行 vi /etc/security/limits.conf

如果文件除了#注释以外没有内容，以下内容可以直接复制粘贴

#修改打开文件的最大数目：

hadoop soft nofile 65536

hadoop hard nofile 65536

#修改最大内存锁定空间：  
hadoop soft memlock unlimited  
hadoop hard memlock unlimited

#修改最大进程数：

hadoop hard nproc 65536

hadoop soft nproc 65536

#修改，把使用swap虚拟内存几率降到最低，并添加max\_map\_count值：

vi /etc/sysctl.conf   
vm.swappiness=0

vm.max\_map\_count=262144

修改后需重启机器生效

### 解压

1. ES官网下载最新稳定版本elasticsearch-5.4.3.tar.gz

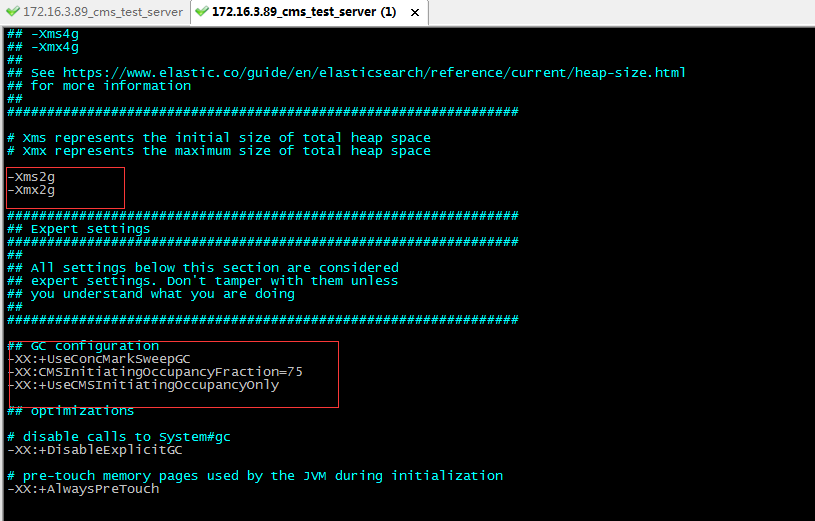
/opt/ 下解压：

tar -zxf elasticsearch-5.4.3.tar.gz

### 修改JVM参数

切换目录 cd /opt/elasticsearch-5.4.3/bin

执行 vi ../config/jvm.options



修改堆内存、最大不超过30G

-Xms6g

-Xmx6g

可根据实际可用内存调整。

如果堆内存超出可用内存可能会导致死机

### 配置

vi ../config/elasticsearch.yml默认都注释掉的，对应行下添加如下配置行，粘贴时注意#号：

# 修改集群名称和node名称：

cluster.name: icare\_cms\_es

node.name: icare\_cms\_es

因ES产生的data索引文件会很大，单独用一块挂载到/data1的盘存放数据和日志文件

#修改数据和日志文件配置：**注意冒号之后有空格**

path.data: /data1/es/data

path.logs: /data1/es/logs

（附：补充说明： 生成环境如果数据量很大，一块盘不满足存储，可以分别挂载多个数据盘，比如data1，data2，data3等，然后把数据配置写到多块盘下，配置如下：

path.data: /data1/es/data, /data2/es/data, /data3/es/data

path.logs: /data1/es/logs

**注意冒号之后有空格**

(附：已挂载硬盘信息查看df -h ) ）

继续修改配置：

#锁定ES内存：

bootstrap.memory\_lock: true

#修改绑定网卡IP，设为0.0.0.0时可以跨域访问

network.host: 0.0.0.0

# 打开http访问9200端口

http.port: 9200

#master节点初始列表，目前只单节点

discovery.zen.ping.unicast.hosts: ["127.0.0.1"]

#有master资格的节点数，这里只能为默认值1

discovery.zen.minimum\_master\_nodes: 1

#是否支持跨域，默认为false

http.cors.enabled: true

#当设置允许跨域，默认为\*,表示支持所有域名

http.cors.allow-origin: /.\*/

### 启动

后台启动（推荐）：bin/elasticsearch -d

前台启动：bin/elasticsearch

(当启动失败可直接前台启动，方便看到启动失败日志，如果后台启动，需到/data1/es/logs/icare\_cms\_es.log中查看日志)

浏览器访问：http://主机地址:9200/ 看到ES信息则OK ！

## Elasticsearch Head插件安装

### 解压node

解压node-v6.11.0-linux-x64.tar.gz到/opt/下:

tar -xzvf node-v6.11.0-linux-x64.tar.gz

### 配置添加环境变量

hadoop> vi ~/.bash\_profile

export NODE\_HOME=/opt/node-v6.11.0-linux-x64

export PATH=$PATH:$NODE\_HOME/bin

使修改配置立刻生效：

source ~/.bash\_profile

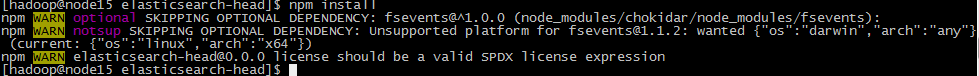
### 解压elasticsearch-head

下载并解压elasticsearch-head.zip到/opt/下

解压：hadoop> unzip elasticsearch-head.zip

cd elasticsearch-head

npm install



### 配置elasticsearch

1.确保elasticsearch-5.4.3/config/elasticsearch.yml中已加入以下内容（2.7.4已经添加）

http.cors.enabled: true

http.cors.allow-origin: "\*"

2. 打开elasticsearch-head/Gruntfile.js，找到下面connect属性，新增hostname: ‘\*’:

connect: {

        server: {

            options: {

                hostname: '\*',

                port: 9100,

                base: '.',

                keepalive: true

            }

        }

}

### 启动es的head插件

cd /opt/elasticsearch-head/node\_modules/grunt/bin/

前台启动：./grunt server

后台启动： ./grunt server >> /dev/null&

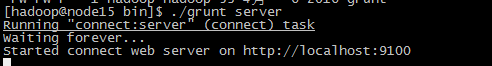
如果发现grunt执行权限无执行权限需先修改执行权限再启动即可：

root> chmod a+x /opt/elasticsearch-head/node\_modules/grunt/bin/grunt

SSH 断开后还是有问题（待解决）：

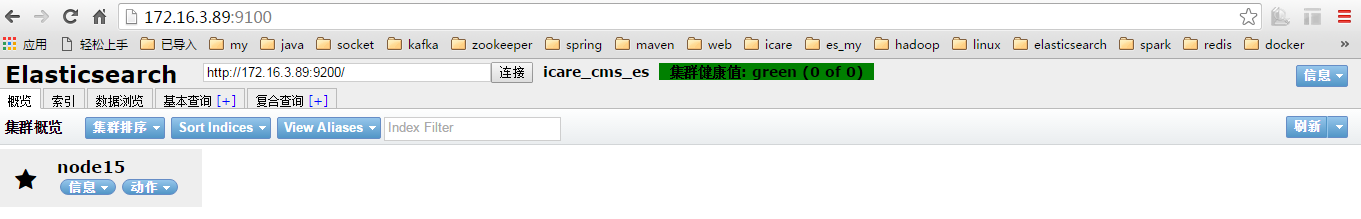
cd /opt/elasticsearch-head/node\_modules/grunt/

nohup bin/grunt server > bin/nohup.out 2>&1 &



(关闭head： 直接kill -9 Pid)

启动后浏览器访问http://主机地址:9100 看到如下页面即OK !

****

## Vsftp（可选安装）

### 创建ftp用户及文件夹

Vsftpd需要一个根目录，在根目录下存放ftp登录用户的文件，可以防止ftp用户访问其home目录以外的目录。

·Root下执行命令 mkdir /ftp

·修改权限 chmod 770 /ftp

Vsftpd使用本地用户登录方式，需要专门用于ftp文件操作的用户

·创建ftp用户组 groupadd ftpuser

·创建用户useradd –d /ftp/ftpuser1 –s /sbin/nologin –g ftpuser ftpuser1

·创建用户的登录密码 passwd ftpuser1

创建多个用户重复以上两个步骤，替换ftpuser1为其它用户名即可

### 检查安装

·执行rpm -qa | grep vsftpd，发现有已经安装则跳过安装步骤直接进行配置

### 执行安装

·执行rpm -i vsftpd-3.0.2-21.el7.x86\_64.rpm

### 修改配置

·vi /etc/vsftpd/vsftpd.conf

修改anonymous\_enable=NO

去除chroot\_local\_user=YES前面的注释

添加use\_localtime=YES

添加local\_root=/ftp

·执行setsebool allow\_ftpd\_full\_access on

### 启动服务

**·启动：systemctl start vsftpd**

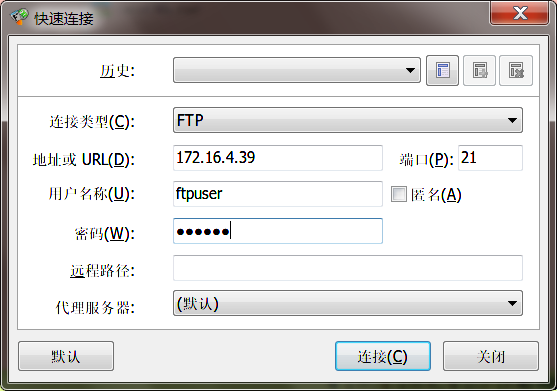
**·重启：systemctl restart vsftpd**

**·停止：systemctl sop vsftpd**

**·开机启动设置：systemctl enable vsftpd**

### 连接ftp

**以flashfxp为例，快速连接填写如下信息，用户名密码为linux中建立的ftp用户信息**



### 问题和解决

#### 错误代码500

**·500 OOPS: failed to open xferlog log file:xxxx**

Log文件没有写入权限，chmod +w xxx，或者在/etc/vsftpd/vsftpd.conf中将xferlog\_file的文件路径改为其它有权限的路径，或者注释掉使用默认值

**·500 OOPS: cannot change directory:xxx**

**这个文件夹不存在**

**如果是连接时直接报错，则用户文件夹不存在，执行mkdir xxx和chown ftpuser:ftpuser xxx为用户建立有权限的文件夹。**

**如果是尝试切换到其它目录时报这个错误，是属于用户没有这个文件夹得操作权限。**

**·500 OOPS: vsftpd: refusing to run with writable root inside chroot()**

ftp的根目录，“其它用户”有写的权限。执行chmod 775 ftp根目录后执行systemctl restart vsftpd重启ftp服务

#### 错误代码553

**·553 ：Could not create file.**

无法创建文件，可能是没有写入权限。

如果确认ftp用户对该文件夹有写入权限，则检查vsftpd的权限。执行sestatus -b| grep ftp，若allow\_ftpd\_full\_access为off则需要执行setsebool allow\_ftpd\_full\_access on开启权限。如果已经是on，则是文件夹权限不足，需要添加写入权限

## 安装nginx

### 解压文件

·执行tar –zxvf nginx-1.12.1.tar.gz

### 安装

·执行cd nginx-1.12.1

·执行yum -y install pcre-devel（已安装则忽略错误

·执行yum install -y zlib-devel（已安装则忽略错误

·执行./configure --prefix=/opt/nginx --with-http\_stub\_status\_module

·执行make

·执行make install

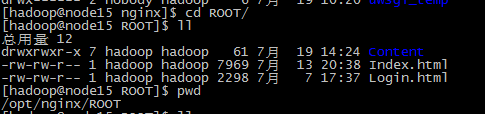
·执行chown root /opt/nginx/sbin/nginx

·执行setcap cap\_net\_bind\_service=+eip /opt/nginx/sbin/nginx

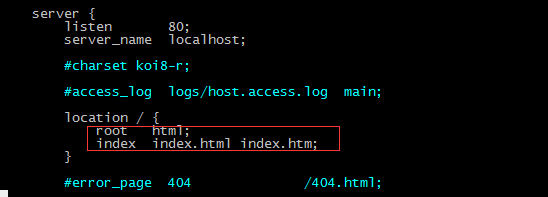
此安装方式需要保证服务器能够连接到公网以获取编译环境，如果无法获取到依赖库那么在执行以上步骤后将压缩包nginx-build.zip中的内容解压到/opt/nginx覆盖

### 部署程序

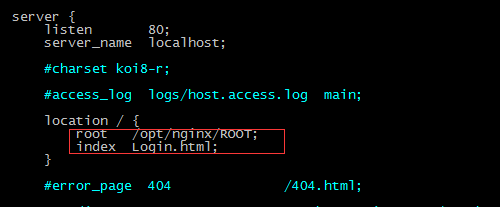
Web页面放入/opt/nginx/ROOT/，使得ROOT的目录结构如下



执行vi conf/nginx.conf



修改为：



即：和实际web访问路径一致

location / {

root /opt/nginx/ROOT;

index Login.html;

}

### 启动

·启动： ./nginx –c /opt/nginx/conf/nginx.conf

·停止： ./nginx -s stop

·重载配置： ./nginx -s reload

·添加开机启动：

执行vi /etc/rc.local

文件最后添加 /opt/nginx/sbin/nginx

\*访问地址为http://本机ip，或者<http://本机ip:80>

\*启动nginx时如果提示80端口没有权限，在root下执行

setcap cap\_net\_bind\_service=+eip nginx

进行授权，nginx需要指定到具体目录

### 问题和解决

部署nginx工程，运行过程中，操作前台查看实时视频时不出现视频，后台报错出现Error：spawn EACCES（类似下图）错误是插件需要运行权限，在root下执行 chmod a+x /opt/nginx/ROOT –R授予执行权限即可



查看nginx安装路径

ps –ef|grep nginx

## 数据脚本初始化

如果数据库和初始化表数据没有创建，第一次部署启动tomcat时候会自动初执行始化sql脚本。

## ES索引与初始化

如果ES索引库和mapping没有，第一次部署启动tomcat的时候，会自动创建索引库和mapping定义。

## 应用程序部署

1. 把部署包ROOT 放到/opt/tomcat8/webapps/下

2. 部署包下面的/opt/tomcat8/webapps/ROOT/WEB-INF/classes/cmsService.properties 配置各个服务连接地址，端口一般不需要改动，ip如果不连接其它主机的服务则全部为127.0.0.1

3. 修改完重新启动tomcat生效。