结算图标.JPG中国证券登记结算有限责任公司总部项目文档

上线操作手册

项目名称: **内部评级体系建设**

编制单位: **中证信用增进股份有限公司**

编制部门: **研发部**

编制人 : **王振，尚强强**

日 期 : **2017年6月16日**

目录

[1. 说明 4](#_Toc490835711)

[1.1 部署顺序 4](#_Toc490835712)

[1.2 数据初始化 4](#_Toc490835713)

[2. 数据库部署 4](#_Toc490835714)

[2.1 操作系统内核参数、资源限制配置 4](#_Toc490835715)

[2.2 编译安装PostgreSQL 9.4.10 5](#_Toc490835716)

[2.3 创建数据库管理用户 6](#_Toc490835717)

[2.4 导入数据库文件 6](#_Toc490835718)

[2.5 修改数据库管理用户的环境变量 6](#_Toc490835719)

[2.6 修改数据库参数 7](#_Toc490835720)

[2.7 启动数据库 7](#_Toc490835721)

[3. 内评系统数据中转服务器上创建FTP用户 7](#_Toc490835722)

[4. 结算外网服务器上部署数据拉取程序 8](#_Toc490835723)

[4.1 安装JDK 8](#_Toc490835724)

[4.2 生成SSH密钥 9](#_Toc490835725)

[4.3 部署中证数据文件拉取程序 10](#_Toc490835726)

[5. 数据同步服务器上部署数据同步程序 11](#_Toc490835727)

[5.1 安装JDK 11](#_Toc490835728)

[5.2 创建一个用户用于接收数据仓库导出的可回购债券清单 11](#_Toc490835729)

[5.3 创建一个用户用于部署同步程序 11](#_Toc490835730)

[5.4 安装PostgreSql客户端 11](#_Toc490835731)

[5.5 部署数据同步程序 12](#_Toc490835732)

[5.6 调度配置 14](#_Toc490835733)

[6. 应用程序部署 14](#_Toc490835734)

[6.1 环境准备 14](#_Toc490835735)

[6.1.1 安装python-devel 14](#_Toc490835736)

[6.1.2 安装c++编译器 15](#_Toc490835737)

[6.1.3 安装zlib 15](#_Toc490835738)

[6.1.4 安装libxml2 15](#_Toc490835739)

[6.1.5 安装 pcre 16](#_Toc490835740)

[6.1.6 安装nginx 16](#_Toc490835741)

[6.1.7 安装JDK 8 17](#_Toc490835742)

[6.2 安装和启动redis 17](#_Toc490835743)

[6.3 配置nginx 17](#_Toc490835744)

[6.4 后台部署 18](#_Toc490835745)

[6.5 前台部署 18](#_Toc490835746)

[6.6 清除缓存 19](#_Toc490835747)

[6.7 常见错误日志 19](#_Toc490835748)

[7. 备份数据库上部署数据库备份脚本 19](#_Toc490835749)

[7.1 安装JDK 19](#_Toc490835750)

[7.2 安装PostgreSQL客户端 20](#_Toc490835751)

[7.3 部署数据库备份脚本 21](#_Toc490835752)

[7.4 修改数据库服务器的访问权限 21](#_Toc490835753)

[7.5 调度配置 21](#_Toc490835754)

# 说明

## 部署顺序

请按照以下顺序进行部署：

1. 数据库
2. 数据拉取程序
3. 数据同步程序
4. 应用程序
5. 数据库备份脚本

## 数据初始化

数据库在部署之前可根据客户的需要，对数据做一定的初始化操作，以适应客户的数据定制化需求。

初始化内容包括角色、用户、权限、敞口负责人、组合和预设规则，以及评级相关的指标、字段、模型和主标尺。

初始化的数据将以数据包的形式在部署时进行导入。

# 数据库部署

## 操作系统内核参数、资源限制配置

这一步为推荐配置，可根据实际情况进行调整

[root@CM-DB ~]# vi /etc/sysctl.conf

添加或修改配置

kernel.shmmni = 4096

kernel.shmmax = 4398046511104

kernel.shmall = 1073741824

kernel.sem = 500 5120000 2500 9000

fs.aio-max-nr = 1048576

fs.file-max = 6815744

net.ipv4.ip\_local\_port\_range = 9000 65500

net.core.rmem\_default = 262144

net.core.rmem\_max = 4194304

net.core.wmem\_default = 262144

net.core.wmem\_max = 1048586

net.core.somaxconn = 4096

net.ipv4.tcp\_max\_syn\_backlog = 4096

net.ipv4.tcp\_keepalive\_intvl = 20

net.ipv4.tcp\_keepalive\_probes = 3

net.ipv4.tcp\_keepalive\_time = 60

net.ipv4.tcp\_mem = 8388608 12582912 16777216

net.ipv4.tcp\_fin\_timeout = 5

net.ipv4.tcp\_synack\_retries = 2

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_timestamps = 1

net.ipv4.tcp\_tw\_recycle = 0

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_max\_tw\_buckets = 262144

net.ipv4.tcp\_rmem = 8192 87380 16777216

net.ipv4.tcp\_wmem = 8192 65536 16777216

net.nf\_conntrack\_max = 1200000

net.netfilter.nf\_conntrack\_max = 1200000

vm.dirty\_background\_bytes = 409600000

vm.dirty\_expire\_centisecs = 3000

vm.dirty\_ratio = 95

vm.dirty\_writeback\_centisecs = 100

vm.min\_free\_kbytes = 2097152

vm.mmap\_min\_addr = 65536

vm.overcommit\_memory = 0

vm.overcommit\_ratio = 90

vm.zone\_reclaim\_mode = 0

[root@localhost ~]# sysctl -p

[root@CM-DB ~]$ vi /etc/security/limits.conf

\* soft nofile 131072

\* hard nofile 131072

\* soft nproc 131072

\* hard nproc 131072

\* soft core unlimited

\* hard core unlimited

\* soft memlock 50000000

\* hard memlock 50000000

## 编译安装PostgreSQL 9.4.10

先安装依赖

[root@CM-DB ~]# yum install -y perl-ExtUtils-Embed zlib-devel openssl-devel pam-devel libxml2-devel libxslt-devel tcl-devel python-devel readline-devel gcc dos2unix

下载源码（如果无法访问外网需要在能联网的电脑上下载后上传至服务器）

[root@CM-DB ~]# wget https://ftp.postgresql.org/pub/source/v9.4.10/postgresql-9.4.10.tar.bz2

[root@CM-DB ~]# tar xf postgresql-9.4.10.tar.bz2

[root@CM-DB ~]# cd postgresql-9.4.10/

开始编译

[root@CM-DB postgresql-9.4.10]# ./configure --prefix=/usr/local/pgsql-9.4.10 --with-tcl --with-perl --with-python --with-openssl --with-pam --without-ldap --with-libxml --with-libxslt --enable-thread-safety --with-segsize=10

[root@CM-DB postgresql-9.4.10]# gmake world

[root@CM-DB postgresql-9.4.10]# gmake install-world

[root@CM-DB postgresql-9.4.10]# chmod o+rx /usr/local/pgsql-9.4.10

[root@CM-DB postgresql-9.4.10]# chown -R postgres:postgres /usr/local/pgsql-9.4.10

## 创建数据库管理用户

[root@CM-DB postgresql-9.4.10]# useradd postgres

[root@CM-DB postgresql-9.4.10]# passwd postgres

## 导入数据库文件

从提供的版本包中获取数据库文件压缩包，上传至服务器并解压到相应的位置:

[root@CM-DB ~]# mkdir /CMDB/data

[root@CM-DB ~]# cd /CMDB/data

[root@CM-DB ~]# mkdir postgres

[root@CM-DB ~]# unzip cs\_master.zip

得到一个目录/CMDB/data/postgres/cs\_master，这个目录就是数据库实例的位置

将该目录更改所有者为数据库管理用户

[root@CM-DB ~]# chown -R postgres:postgres /CMDB/data/postgres/cs\_master

[root@CM-DB ~]# chmod -R o+rx /CMDB/data

[root@CM-DB ~]# chmod 700 /CMDB/data/postgres/cs\_master

## 修改数据库管理用户的环境变量

[root@CM-DB ~]# su – postgres

[postgres@CM-DB ~]$ vi .bash\_profile

加入

export PGPORT=3432

**export PGDATA=/CMDB/data/postgres/cs\_master #替换成上面解压得到的目录**

export LANG=en\_US.utf8

export PGHOME=/usr/local/pgsql-9.4.10

export LD\_LIBRARY\_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD\_LIBRARY\_PATH

export PATH=$PGHOME/bin:$PATH:.

export MANPATH=$PGHOME/share/man:$MANPATH

export PGHOST=localhost

export PGUSER=postgres

[postgres@CM-DB ~]$ . .bash\_profile

## 修改数据库参数

**参数文件也可以基于UAT环境的配置文件进行改动**

[postgres@CM-DB ~]$ cd $PGDATA

[postgres@CM-DB cs\_master]$ vi postgresql.conf

listen\_addresses = '\*' #监听地址，\*表示全部监听

port = 3432 #监听端口

max\_connection = 1000 #最大连接数

shared\_buffers = 8G **#建议是系统内存的四分之一**

[postgres@CM-DB cs\_master]$ vi pg\_hba.conf

添加或修改以下配置（注意顺序不可变）

local all all trust #本地所有用户连接所有数据库免密码登录

host all all 0.0.0.0/0 md5 #远程所有主机连接所有数据库都需要通过密码登录

host all cs\_master\_stg ::1/128 md5 #本地客户端cs\_master\_stg用户进行远程连接所有数据库时需要通过密码登录

host all all ::1/128 trust #本地客户端所有用户进行远程连接所有数据库时免密码登录

## 启动数据库

[postgres@CM-DB cs\_master]$ pg\_ctl start

# 内评系统数据中转服务器上创建FTP用户

创建一个用于下载的FTP用户，用户名为cmuserget，目录为/CMSWT/cmdir

创建一个用于上传的FTP用户，用户名为cmuserput，目录为/CMSWT/cmdir

groupadd cmuser

usermod -g cmuser cmuserput

usermod -g cmuser cmuserget

chgrp cmuser -R /CMSWT/cmdir

chmod g+w -R /CMSWT/cmdir

chmod g+w -R /CMSWT/cmdir/edw\_data

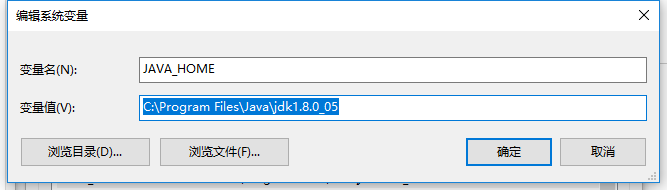
# 结算外网服务器上部署数据拉取程序

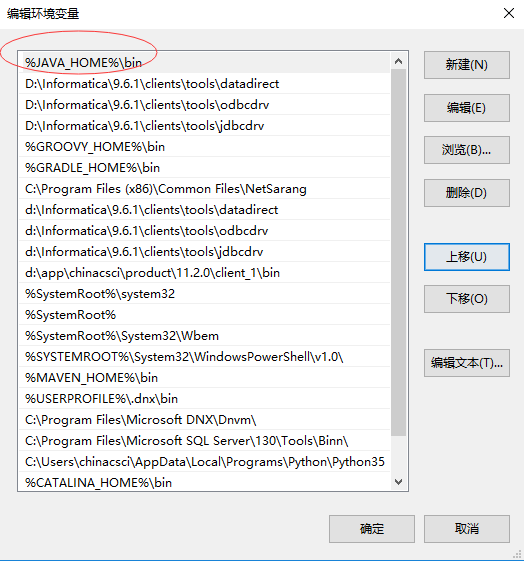
## 安装JDK

需要JDK8及以上版本

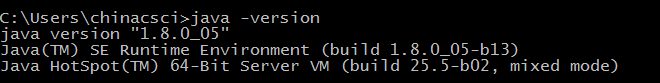
下载安装包并安装到合适的位置

设置环境变量JAVA\_HOME指向刚刚安装的JDK路径，并将%JAVA\_HOME%\bin加入环境变量PATH





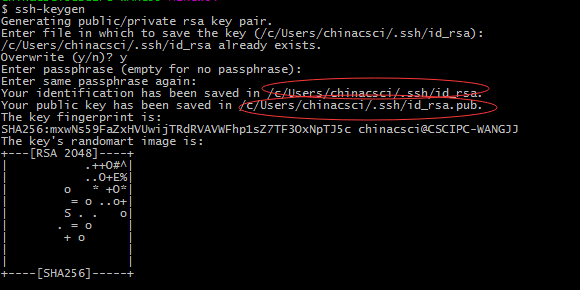
测试安装是否成功



## 生成SSH密钥

使用任意工具生成一对SSH密钥，本文以git为例

打开git bash，执行ssh-keygen，然后一路回车，可以得到一对SSH密钥



第一个红圈是私钥，第二个红圈是公钥

将红圈中所示的SSH公钥提供给中证工程师，中证工程师会将SSH公钥加入到信任列表中。

## 部署中证数据文件拉取程序

解压MasterDataPull.zip解压至合适的位置（推荐D:/）

修改配置文件conf/Application.properties

* sftp.prvkey：上一步中生成的私钥的存放路径
* sftp.localDirectory：建议D:/creditmaster/incoming（临时保存从中证sftp服务器上拉取下来的文件的存放路径）
* sftp.interval：300（拉取间隔时间，单位为秒）
* sftp.host：121.13.219.102（中证数据服务器地址）
* mds.jdbc.url：jdbc:postgresql://121.13.219.103:3432/cs\_master\_ds
* ftp.username：cmuserput（内评系统数据中转服务器上传用户名）
* ftp.password： 内评系统数据中转服务器上传用户密码
* ftp.host：10.129.64.127（内评系统数据中转服务器地址）
* ftp.port：21（内评系统数据中转服务器端口）
* ftp.remoteDirectory：/CMSWT/cmdir/cscs\_data（中证数据文件上传至内评系统数据中转服务器时存放的位置，需要提前创建，并保证ftp上传用户、ftp下载用户对该目录有读写权限）
* monitor.url：<http://172.30.0.10:52090/ultraChinaClearService/services/NMSService>（监控服务webservice的调用地址）
* monitor.targetNamespace：<http://ws.accredit.ultrapower.com/>（监控服务的namespace）
* monitor.host：10.129.24.16（外网服务器IP）
* monitor.instance：CM

双击运行bin目录下的数据拉取程序DownloadFromSftp.vbe。

# 数据同步服务器上部署数据同步程序

## 安装JDK

需要JDK8及以上版本

[root@CM-SYNC ~]# rpm -ivh jdk-8u131-linux-x64.rpm

[root@CM-SYNC ~]# vi /etc/profile

加入

export JAVA\_HOME=/usr/java/jdk1.8.0\_131

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

[root@CM-SYNC ~]# . /etc/profile

[root@CM-SYNC ~]# java -version

java version "1.8.0\_131"

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

## 创建一个用户用于接收数据仓库导出的可回购债券清单

[root@CM-SYNC ~]# useradd -d /CMSYNC/cmdir/edw\_data cmuserput

[root@CM-SYNC ~]# passwd cmuserput

## 创建一个用户用于部署同步程序

[root@CM-SYNC ~]# useradd cmsync

[root@CM-SYNC ~]# passwd cmsync

注意：本用户必须与数据仓库用于向数据同步服务器传输数据的FTP用户（即上节创建的cmuserput）位于同一组内

[root@CM-SYNC ~]# groupadd cm

[root@CM-SYNC ~]# usermod –g cm cmuserput

[root@CM-SYNC ~]# usermod –g cm cmsync

## 安装PostgreSql客户端

**拥有PostgreSql客户端执行权限的用户需与部署同步程序的用户一致，以下使用cmsync用户**

先安装依赖

[root@CM-SYNC ~]# yum install -y perl-ExtUtils-Embed zlib-devel openssl-devel pam-devel libxml2-devel libxslt-devel tcl-devel python-devel readline-devel gcc dos2unix

下载源码

[root@CM-SYNC ~]# wget https://ftp.postgresql.org/pub/source/v9.4.10/postgresql-9.4.10.tar.bz2

[root@CM-SYNC ~]# tar xf postgresql-9.4.10.tar.bz2

[root@CM-SYNC ~]# cd postgresql-9.4.10/

开始编译

[root@CM-SYNC postgresql-9.4.10]# ./configure --prefix=**/usr/local/pgsql-9.4.10** --with-tcl --with-perl --with-python --with-openssl --with-pam --without-ldap --with-libxml --with-libxslt --enable-thread-safety --with-segsize=10

上面红字加粗位置为安装目录，可以任意替换

[root@CM-SYNC postgresql-9.4.10]# gmake world

[root@CM-SYNC postgresql-9.4.10]# gmake install-world

配置环境变量

[root@CM-SYNC postgresql-9.4.10]# chown cmsync:cmsync -R /usr/local/pgsql-9.4.10

[root@CM-SYNC postgresql-9.4.10]# su - cmsync

[cmsync@CM-SYNC ~]$ vi .bash\_profile

加入

export LANG=en\_US.utf8

export PGHOME=/usr/local/pgsql-9.4.10 #指向安装位置

export LD\_LIBRARY\_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD\_LIBRARY\_PATH

export PATH=$PGHOME/bin:$PATH:.

export MANPATH=$PGHOME/share/man:$MANPATH

[cmsync@CM-SYNC ~]$ . .bash\_profile

## 部署数据同步程序

将程序解压到合适的位置（推荐/CMSYNC/cmapp），保证数据同步服务用户对该目录有读写权限，然后进入程序目录，修改权限和换行符

[cmsync@CM-SYNC MasterDataSync-1.1]$ chmod u+x bin/\*

[cmsync@CM-SYNC MasterDataSync-1.1]$ dos2unix u+x bin/\*

[cmsync@CM-SYNC MasterDataSync-1.1]$ dos2unix u+x conf/\*

修改配置文件conf/Application.properties

* ftp.username：cmuserget（内评系统数据中转服务器下载用户名，需要提前创建）
* ftp.password：内评系统数据中转服务器下载用户密码
* ftp.host：10.129.64.127（内评系统数据中转服务器地址）
* ftp.port：21（内评系统数据中转服务器FTP端口）
* ftp.cscs.remoteDirectory：/CMSWT/cmdir/cscs\_data（内评系统数据中转服务器上暂存中证数据的目录，应和数据拉取程序中配置的ftp.remoteDirectory一致，需要提前创建，并保证ftp上传用户、ftp下载用户对该目录有读写权限）
* ftp.cscs.localDirectory：/CMSYNC/cmdir/cscs\_data（数据同步服务器上暂存中证数据的目录，需要提前创建，并保证数据同步用户对该目录有读写权限）
* ftp.csdc.remoteDirectory：/CMSWT/cmdir/edw\_data（内评系统数据中转服务器上暂存结算EDW导出数据的目录，需要提前创建，并保证数据同步用户对该目录有读写权限）
* ftp.csdc.localDirectory：/CMSYNC/cmdir/edw\_data（数据同步服务器上暂存结算EDW导出数据的目录，需要提前创建，并保证数据同步用户对该目录有读写权限）
* stage.jdbc.url：内评系统stage数据库jdbc连接串，需要修改其中的ip（生产数据库服务器IP：10.128.13.17）和端口（生产数据库服务器端口：3432）部分
* stage.jdbc.user：cs\_master\_stg（内评系统stage数据库用户名）
* stage.jdbc.password：内评系统stage数据库密码
* stage.cscs.baseDir：/CMSYNC/cmdir/cscs\_data（数据同步服务器上暂存中证数据的目录，应和ftp.cscs.localDirectory一致，需要提前创建，并保证数据同步用户对该目录有读写权限）
* stage.cscs.archiveDir：/CMSYNC/cmdir/cscs\_archive（加载完毕的中证数据进行归档时存放的路径，需要提前创建，并保证数据同步用户对该目录有读写权限）
* stage.csdc.baseDir：/CMSYNC/cmdir/edw\_data（数据同步服务器上暂存EDW导出数据的目录，应和ftp.csdc.localDirectory一致，需要提前创建，并保证数据同步用户对该目录有读写权限）
* stage.csdc.archiveDir：/CMSYNC/cmdir/edw\_archive（加载完毕的EDW导出数据进行归档时存放的路径，需要提前创建，并保证数据同步用户对该目录有读写权限）
* target.jdbc.url：内评系统target数据库jdbc连接串，需要修改其中的ip（生产数据库服务器IP：10.128.13.17）和端口（生产数据库服务器端口：3432）部分
* target.jdbc.user：cs\_master\_tgt（内评系统target数据库用户名）
* target.jdbc.password：内评系统target数据库密码
* bond.baseDir：/CMSYNC/cmdir/edw\_data（数据同步服务器上暂存结算导出的债券清单，需要提前创建，并保证数据同步用户对该目录有读写权限）
* bond.archiveDir：/CMSYNC/cmdir/edw\_archive（加载完毕的结算导出债券清单进行归档时存放的路径，需要提前创建，并保证数据同步用户对该目录有读写权限）
* stage.threads、target.threads、bond.threads：4（同步线程数，根据服务器情况合理设置）
* exp.tempDir：/CMSYNC/cmdir/cscs\_export（数据同步服务器上暂存导出给EDW的文件存放路径，需要提前创建，并保证数据同步用户对该目录有读写权限）
* exp.delimiter：|~（导出给EDW的文件的分隔符）
* exp.offset：120（增量导出给EDW的文件时的时间偏移量，单位秒）
* exp.encoding：GBK（导出给EDW的文件编码字符集）
* exp.withHeader：false（导出给EDW的文件是否包含列头）
* exp.ftp.username：cmuserput（EDW接收内评系统导出文件的FTP服务器用户名）
* exp.ftp.password：EDW接收内评系统导出文件的FTP服务器密码
* exp.ftp.host：192.168.12.137 （EDW接收内评系统导出文件的FTP服务器IP）
* exp.ftp.port：21（EDW接收内评系统导出文件的FTP服务器端口）
* exp.ftp.remoteDirectory：/CM（EDW接收内评系统导出文件的FTP服务器上的文件存放路径，需要提前创建，并保证FTP上传用户对该目录有读写权限）
* monitor.url：<http://172.30.0.10:52090/ultraChinaClearService/services/NMSService>（监控服务webservice的调用地址）
* monitor.targetNamespace：<http://ws.accredit.ultrapower.com/>（监控服务的namespace）
* monitor.host：10.128.13.18（数据同步服务器IP）
* monitor.instance：CM

## 调度配置

[cmsync@CM-SYNC MasterDataSync-1.1]$ crontab -e

#每周一至周五每1分钟检查一次是否有新的中证或EDW数据等待导入

\*/1 \* \* \* 1-5 /CMSYNC/cmapp/MasterDataSync-1.1/bin/MasterDataSyncAll.sh &>> /CMSYNC/cmapp/MasterDataSync-1.1/log/mds.log

#每周一至周五每天早上8点检查是否有可回购债券清单加载完成

0 8 \* \* 1-5 /CMSYNC/cmapp/MasterDataSync-1.1/bin/CheckRepobondFile.sh 480

#每年1月31日早上7点30分检查是否有CISP文件加载完成

30 7 1 31 \* /CMSYNC/cmapp/MasterDataSync-1.1/bin/CheckCISPFile.sh 7

#每天下午2:00检查CSCS的文件有没有到达

0 14 \* \* 1-5 /CMSYNC/cmapp/MaserDataSync-1.1/bin/CheckCSCSFile.sh 840

# 应用程序部署

## 环境准备

操作系统: redhat 7.1

Master版本:v2.0

安装包里面一共有如下依赖包,需要先安装完这些依赖包才能开始部署:

1. zlib-1.2.8.tar.gz
2. libxml2-2.9.4.tar.gz
3. pcre-8.39.tar.gz
4. nginx-1.10.1.tar.gz
5. redis-3.2.3.tar.gz
6. jdk-8u131-linux-x64.rpm
7. python-devel-2.7.5-48.el7.x86\_64.rpm

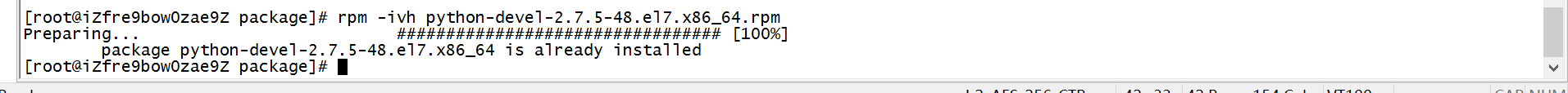
### 安装python-devel

1.如果有网络，可以通过yum去安装pythondevel,yum install python-devel.x86\_64。

2.如果没有网络环境，使用提供的python-devel-2.7.5-48.el7.x86\_64.rpm

yum localinstall python-devel-2.7.5-48.el7.x86\_64.rpm

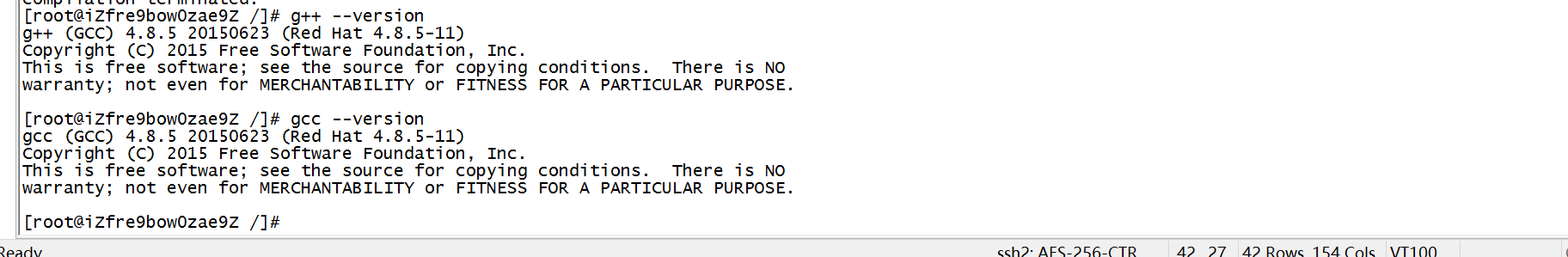
出现如下界面，代表安装成功



### 安装c++编译器

1. 有网络的时候直接运行yum install -y gcc gcc-c++

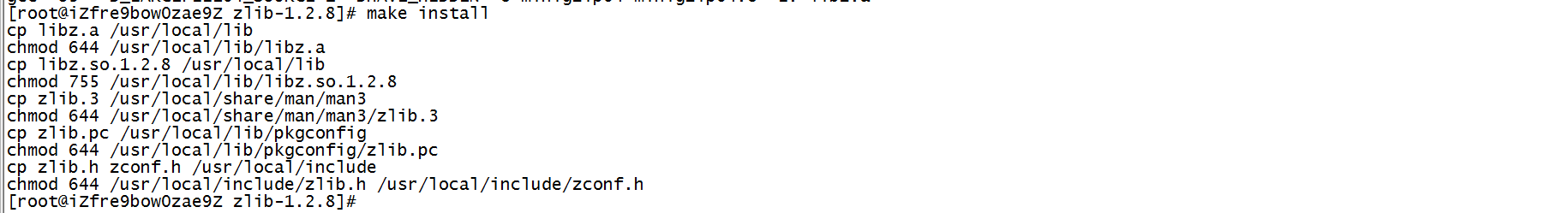
安装完成后可以通过gcc --version 和g++ -- version检查是否安装成功。



### 安装zlib

1. tar zxvf zlib-1.2.8.tar.gz
2. cd zlib-1.2.8
3. ./configure
4. make
5. make install

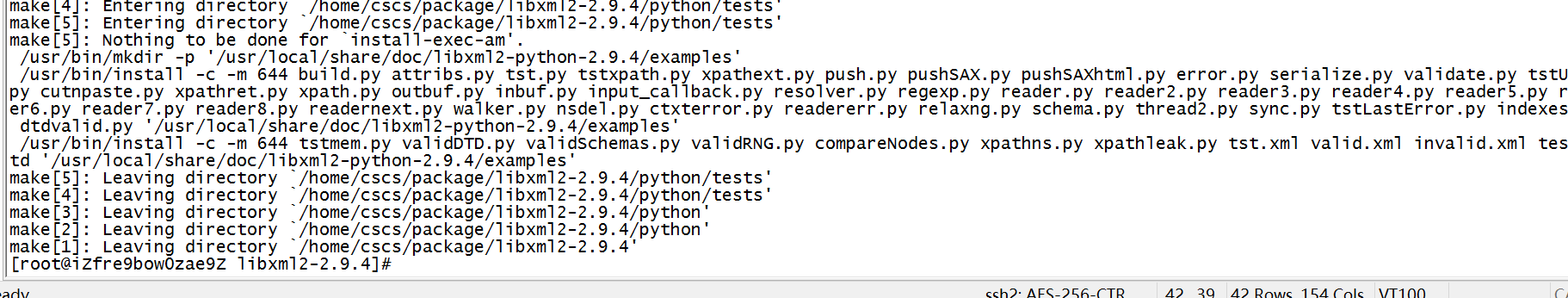
出现如下界面，代表安装成功



### 安装libxml2

1. tar -zxvf libxml2-2.9.4.tar.gz
2. cd libxml2-2.9.4
3. ./configure
4. make
5. make install

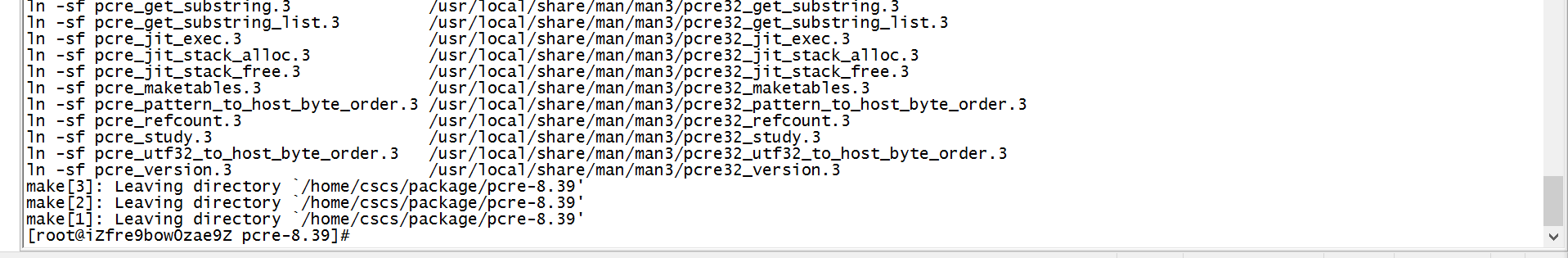
出现如下界面代表安装成功:



### 安装 pcre

1. tar -zxvf pcre-8.39.tar.gz
2. cd pcre-8.39/
3. ./configure
4. make
5. make install

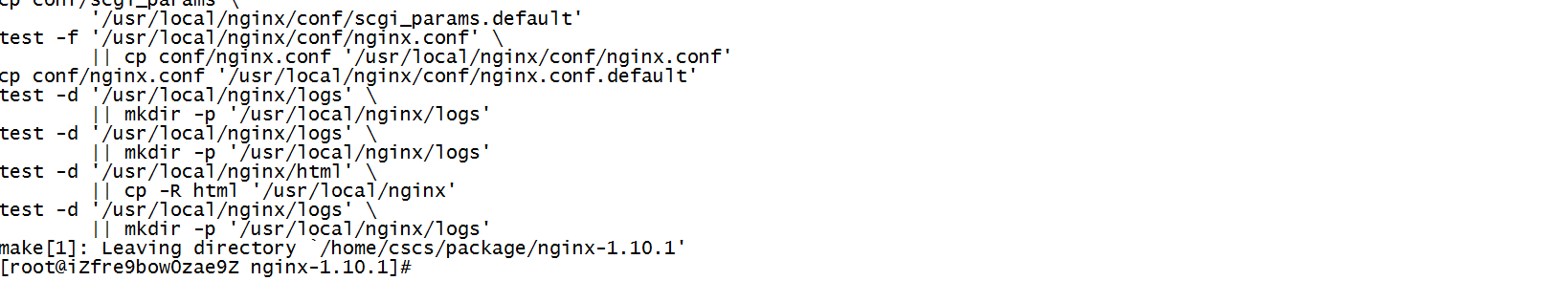
出现如下界面代表安装成功:



### 安装nginx

1. tar -zxvf nginx-1.10.1.tar.gz
2. cd nginx-1.10.1/
3. ./configure
4. make
5. make install

出现如下界面代表安装成功:



### 安装JDK 8

需要JDK8及以上版本。

[root@CM-WEB ~]# rpm -ivh jdk-8u131-linux-x64.rpm

vim /etc/profile

#将下面的内容添加进去：

export JAVA\_HOME=/usr/java/jdk1.8.0\_131

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

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

source /etc/profile

验证是否安装成功: java --version,出现如下信息代表安装成功

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

## 安装和启动redis

安装redis，

1. tar -zxvf redis-3.2.3.tar.gz
2. cd redis-3.2.3
3. 替换redis.conf文件,安装包里有一个已经定义好的redis.conf文件，将它拷贝到解压出来的文件夹里面

cp ../redis.conf ./

1. make
2. make install
3. 启动redis

redis-server /root/redis-3.2.3/redis.conf &

1. 验证是否启动成功

netstat -lnp | grep 6379 有进程代表redis启动成功

## 配置nginx

安装包里面有一个文件nginx.conf，将该文件拷贝到/usr/local/nginx/conf/nginx.conf，并将路径/root/creditmaster修改为/CMWEB/creditmaster，然后启动nginx。

cp nginx.conf /usr/local/nginx/conf/nginx.conf

/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

/usr/local/nginx/sbin/nginx -s reload

验证nginx是否启动成功:

ps -aux |grep nginx

注:可以根据服务器的实际情况调整ngnix的配置

* + 第3行 worker\_processes 8; 8代表CPU的核数目
  + 本地后台服务端口 server 127.0.0.1:7777; 代表后台启动的端口为7777
  + 后台对外服务端口 listen 3030; 代表后台对外服务的端口为3030

## 后台部署

1. 创建后台部署包目录/CMWEB/creditmaster/
2. 将cmaster-0.0.1-SNAPSHOT.jar 拷贝到/CMWEB/creditmaster/
3. 将后台启动命令文件startServer.sh 拷贝到/CMWEB/creditmaster/
4. 修改startServer.sh文件中的数据库链接地址和密码为生产系统实际配置
   1. 地址: --spring.datasource.url=jdbc:postgresql://10.128.13.17:3432/cs\_master\_tgt
   2. 数据库用户: --spring.datasource.username=cs\_master\_tgt
   3. 数据库密码: --spring.datasource.password=abc123
5. 进入目录/CMWEB/creditmaster/ (./startServer.sh ) 启动进程
6. 查看startServer.sh 指定的日志输出文件
7. 通过http://10.128.13.16:1010/v2/api.html 即可查看是否部署成功，出现如下界面，代表后端部署成功。



## 前台部署

1. 准备前台发布的包dist.zip
2. 将dist.zip解压到/CMWEB/creditmaster 如果dist目录已经存在请删除
3. 后端的访问地址为**后台部署第7步中的**http://10.128.13.16:3030/v2/
4. 修改main.\*\*\*\*\*.bundle.js 文件，用后端访问地址替换api\_endpoint:"http://ip:3030 /v2中的对应部分
5. 通过<http://10.128.13.16:3030>查看前台发布

出现如下界面，代表前端部署成功



## 清除缓存

在版本部署成功后，需要手动删除nginx缓存保证更新立即生效.

1. rm -rf /usr/local/nginx/client\_body\_temp
2. rm -rf /usr/local/nginx/fastcgi\_temp
3. rm -rf /usr/local/nginx/proxy\_temp
4. rm -rf /usr/local/nginx/scgi\_temp
5. rm -rf /usr/local/nginx/uwsgi\_temp
6. /usr/local/nginx/sbin/nginx -s reload 重启服务

## 常见错误日志

1. 后台错误日志 (./startSever 指定的日志文件)
2. Nginx 错误日志 /usr/local/nginx/logs

# 备份数据库上部署数据库备份脚本

## 安装JDK

需要JDK8及以上版本

[root@CM-BACK ~]# rpm -ivh jdk-8u131-linux-x64.rpm

[root@CM-BACK ~]# vi /etc/profile

加入

export JAVA\_HOME=/usr/java/jdk1.8.0\_131

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

[root@CM-BACK ~]# . /etc/profile

[root@CM-BACK ~]# java -version

java version "1.8.0\_131"

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

## 安装PostgreSQL客户端

**拥有PostgreSql客户端执行权限的用户需与部署同步程序的用户一致，以下使用pgbak用户**

先安装依赖

[root@CM-BACK ~]# yum install -y perl-ExtUtils-Embed zlib-devel openssl-devel pam-devel libxml2-devel libxslt-devel tcl-devel python-devel readline-devel gcc dos2unix readline-devel python-devel

下载源码

[root@CM-BACK ~]# wget https://ftp.postgresql.org/pub/source/v9.4.10/postgresql-9.4.10.tar.bz2

[root@CM-BACK ~]# tar xf postgresql-9.4.10.tar.bz2

[root@CM-BACK ~]# cd postgresql-9.4.10/

开始编译

[root@CM-BACK postgresql-9.4.10]# ./configure --prefix=**/usr/local/pgsql-9.4.10** --with-tcl --with-perl --with-python --with-openssl --with-pam --without-ldap --with-libxml --with-libxslt --enable-thread-safety --with-segsize=10

上面红字加粗位置为安装目录，可以任意替换

[root@CM-BACK postgresql-9.4.10]# gmake world

[root@CM-BACK postgresql-9.4.10]# gmake install-world

配置环境变量

[root@CM-BACK postgresql-9.4.10]# chown pgbak:pgbak -R /usr/local/pgsql-9.4.10

[root@CM-BACK postgresql-9.4.10]# su - pgbak

[pgbak@CM-BACK ~]$ vi .bash\_profile

加入

export LANG=en\_US.utf8

export PGHOME=/usr/local/pgsql-9.4.10 #指向安装位置

export LD\_LIBRARY\_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD\_LIBRARY\_PATH

export PATH=$PGHOME/bin:$PATH:.

export MANPATH=$PGHOME/share/man:$MANPATH

[pgbak@localhost ~]$ . .bash\_profile

## 部署数据库备份脚本

将程序pg\_bak.zip解压到合适的位置（推荐/CMBKP），进入程序目录，修改权限和换行符

[pgbak@CM-BACK pg\_bak]$ chmod u+x bin/\*

[pgbak@CM-BACK pg\_bak]$ dos2unix bin/\*

[pgbak@CM-BACK pg\_bak]$ dos2unix conf/\*

修改配置文件conf/pg\_bak.conf

* NMS\_HOST：172.30.0.10（监控服务IP）
* NMS\_PORT：51088（监控服务端口）
* LOCAL\_IP：10.128.13.19（数据库备份服务器IP）
* INSTANCE：CM（被监控应用名称）
* JAVA\_HOME：/usr/java/jdk1.8.0\_131（JAVA\_HOME路径）
* PGHOST：10.128.13.17（需要备份的数据库的IP地址）
* PGPORT：3432（需要备份的数据库的端口）
* PGDATABASE：cs\_master\_tgt（需要备份的数据库的数据库名）
* PGUSER：cs\_master\_tgt（需要备份的数据库的登录用户名）
* BAKDIR：/CMBKP/pg\_bak/bak（存放备份文件的路径）
* RETENTION：7（备份文件及备份过程产生的日志文件存放天数）

## 修改数据库服务器的访问权限

修改数据库的访问权限，以使备份服务器能够免密码登录数据库服务器

[postgres@CM-DB ~]$ cd $PGDATA

[postgres@CM-DB cs\_master]$ vi pg\_hba.conf

在**local all postgres trust**和**host all all 0.0.0.0/0 md5**之间加入一行

host all all 10.128.13.19/32 trust

[postgres@CM-DB cs\_master]$ pg\_ctl reload

## 调度配置

[pgbak@CM-BACK ~]$ crontab -e

#每天晚上10点执行备份脚本进行备份

0 22 \* \* \* /CMBKP/pg\_bak/bin/pg\_bak.sh