**Red Hat Enterprise Linux 7.4安装Anaconda2**

[1. 镜像下载 1](#_Toc15893580)

[1) 镜像名称 1](#_Toc15893581)

[2) 获取资源方式 1](#_Toc15893582)

[2. 建虚拟机 1](#_Toc15893583)

[1) Vmware12\_pro虚拟化模拟安装 1](#_Toc15893584)

[2) 创建虚拟机 1](#_Toc15893585)

[3) 加载镜像 2](#_Toc15893586)

[4) 开启虚拟机 2](#_Toc15893587)

[5) 无IP地址问题 2](#_Toc15893588)

[6) 修改hadoop用户权限 3](#_Toc15893589)

[7) 分区查看和赋权 3](#_Toc15893590)

[8) 分区说明 4](#_Toc15893591)

[3. 制作yum离线仓库 4](#_Toc15893592)

[1) 挂载镜像 4](#_Toc15893593)

[2) 卸载和删除镜像文件 4](#_Toc15893594)

[3) 安装yum 4](#_Toc15893595)

[4) 安装yum的离线仓库repo 5](#_Toc15893596)

[4. Anconda2安装 5](#_Toc15893597)

[1) 安装bzip 5](#_Toc15893598)

[2) 安装anaconda2 6](#_Toc15893599)

[5. xgboost 安装 7](#_Toc15893600)

[1) pip安装命令检查 7](#_Toc15893601)

[2) xgboost安装 7](#_Toc15893602)

[6. 安装lightGBm 7](#_Toc15893603)

[7. 其他命令工具安装 8](#_Toc15893604)

[8. JDK安装 8](#_Toc15893605)

[9. Scala安装 9](#_Toc15893606)

[10. sbt安装（开发环境） 9](#_Toc15893607)

[10. Anaconda作用 11](#_Toc15893608)

**说明**

目的: 银行系统离线环境安装Anaconda2-5.0.0-Linux-x86\_64.sh

系统版本: Red Hat Enterprise Linux Server release 7.4 (Maipo)

虚拟化: 采用vmware虚拟化,vmware12\_pro

硬件环境 : 2核4线程;6G内存;单机离线(仅主机模式),windows10\_pro

# 镜像下载

## 镜像名称

rhel-server-7.4-x86\_64-dvd.iso

## 获取资源方式

百度网盘 :

链接：https://pan.baidu.com/s/102GIZeM9hTLMN5uQ15DiEA

提取码：i0yy

第三方网站 :

<http://www.linuxfly.org/post/659/>

官网 :

<https://access.redhat.com/downloads>

# 建虚拟机

若为实体机服务器操作,略过建虚拟机步骤

## Vmware12\_pro虚拟化模拟安装

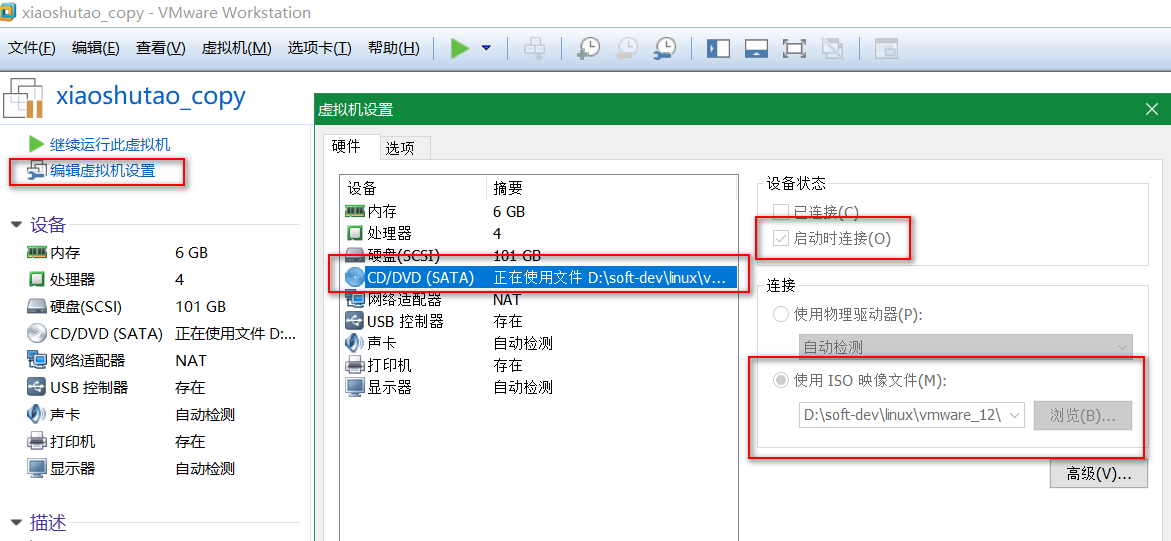
参考博客: <https://blog.csdn.net/yangs_2012/article/details/78784764>

<https://blog.csdn.net/weixin_38187469/article/details/79720107>

## 创建虚拟机

创建新的虚拟机->选择“自定义（高级）->workstation12->稍后安装操作系统->操作系统的类型(linux & redhat enterprise linux7 64位)-> 设置虚拟机名称和保存路径->设置处理器数量和核心数,设置内存,设置网络(仅主机)->IO控制器(LSI Logic控制器)->磁盘类型SCSI->创建新虚拟磁盘->40G磁盘单个文件存储

## 加载镜像

\

## 开启虚拟机



点击“Enter”键开始安装->配置安装使用的语言(中文)->时间设置(亚洲上海24小时制,关闭网络)->键盘选择(汉语)->语言支持(汉语)->安装源选择(默认配置,自动化检测)->软件选择为basic->安装位置(自动化分区)-> KDUMP(省内存建议关闭功能)->网络和主机名(关闭以太网)-> SECURITY POLICY(打开)->开始安装->完成后,设置账户和密码->重启

## 无IP地址问题

1. 查看网卡信息

dmesg | grep -i eth

1. 查看系统网卡设备

ip addr show

1. 修改网卡配置参数

cd /etc/sysconfig/network-scripts/

1. 打开onboot

vi ifcfg-ens33

修改ONBOOT参数为yes

解决没有IP的问题

如果还是没有IP地址可以设置网络模式,仅主机等设置

1. 重启网卡

service network restart

ping 检测下网络有没有通; ping 127.0.0.1 ; ping localhost ;

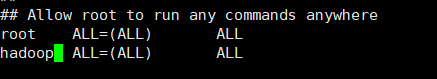
## 修改hadoop用户权限

1. 更改配置文件写权限

root下chomod u+w /etc/sudoers

1. 加入hadoop用户进入sudoers

vim /etc/sudoers



1. 保存并回复文件权限

chmod u-w /etc/sudoers

## 分区查看和赋权

1. 确认/opt目录有剩余空间

df -h查看分区情况

以下操作在hadoop用户下进行

1. /opt目录空间赋给hadoop用户

修改/opt增加写权限并创建目录

sudo chmod o+w /opt

## 分区说明

为hadoop,spark平台寻找合适的数据存储位置, disk0, disk1, disk2为集群的多个硬盘.POC项目提供的硬盘资源可能会比较紧张,只有一个硬盘. 可通过df –h查找最大的硬盘分区,替换以下目录

修改/mnt下所有disk所属用户，例如：

sudo chown hadoop:hadoop /mnt/disk0

sudo chown hadoop:hadoop /mnt/disk1

sudo chown hadoop:hadoop /mnt/disk2

…

# 制作yum离线仓库

目的 : 解决rpm安装缺少依赖问题或者通过缺乏依赖信息去镜像库拷贝, 选择官网下载速度太慢以及依赖包多的时候下载麻烦 ,第三方站提供的可能存在不全或者版本问题

推荐采用安装离线yum, 建立yum本地仓库, 挂载镜像拷贝镜像中的包,从而解决包中现有资源的版本和依赖问题

注意: 整理资源中已包含所需要的所需要的rpm包, 可以跳过yum离线仓库和镜像挂载

## 挂载镜像

1. 采用FTP客户端工具上传镜像iso文件

mkdir /home/hadoop/soft\_package

上传到目录/home/hadoop/soft\_package

1. 创建挂载目录

sudo mkdir /mnt/iso

1. 挂载并拷贝rpm包

mount -o loop /home/hadoop/soft\_package/rhel-server-7.4-x86\_64-dvd.iso /mnt/iso

cd /mnt/iso ; mkdir /home/hadoop/rpm\_dir ;

find . -iname “\*.rpm” | xagrs -i cp {} /home/hadoop/rpm\_dir

## 卸载和删除镜像文件

umount /mnt/iso

rm -rf /home/hadoop/soft\_package/rhel-server-7.4-x86\_64-dvd.iso

## 安装yum

1. 安装yum

cd /home/hadoop/rpm\_dir

find . -iname “\*yum\*.\*”

或者采用直接上传已经整理好的rpm包, 省去挂载镜像

sudo rpm -ivh yum-3.4.3-154.el7.noarch.rpm

1. 检验安装

yum list

## 安装yum的离线仓库repo

1. 安装repo

rpm -ivh deltarpm-3.6-3.el7.x86\_64.rpm python-deltarpm-3.6-3.el7.x86\_64.rpm createrepo-0.9.9-28.el7.noarch.rpm

1. 安装本地存储库

创建一个目录，这里选用/home/hadoop/rpm\_dir(mkdir已操作)作为仓库目录

将RPM包移动到该目录(已经存在rpm包,自定义目录需要移动rpm包到此目录)

1. 修复一些所有权和文件系统权限：

# chown -R root.root /home/hadoop/rpm\_dir

# createrepo /home/hadoop/rpm\_dir

# chmod -R o-w+r /home/hadoop/rpm\_dir

1. 存储库配置文件

/etc/yum.repos.d/myrepo.repo 内容如下

[local]

name=My Awesome Repo

baseurl=file:///home/hadoop/rpm\_dir

enabled=1

gpgcheck=0

检验安装

yum install packagename

会自动安装需要的依赖

# Anconda2安装

## 安装bzip

1. 说明

生产环境是在root环境下进行, 这里在hadoop用户下进行

1. 安装bzip2

安装anaconda2解压需要

bzip2-1.0.6-13.el7.x86\_64.rpm

1. 安装方式  
   1) <https://pypi.org/>官网下载安装包
2. yum离线配置仓库安装包
3. 使用已整理安装包
4. 安装

yum install bzip2-1.0.6-13.el7.x86\_64.rpm

## 安装anaconda2

1. 下载 Anaconda2-5.0.0-Linux-x86\_64.sh 离线包

资源包已整理, 直接上传即可

1. 上传

mkdir /home/hadoop/soft\_package/anaconda2

上传到此目录

1. 安装

bash Anaconda2-5.0.0-Linux-x86\_64.sh

mkdir /home/Hadoop/anacon2(作为安装路径)

依次执行命令：回车 -> yes -> 回车 -> yes

1. 参考链接

<https://blog.csdn.net/hezhiqiang1314/article/details/57077404>

<https://repo.anaconda.com/archive/>

<https://docs.anaconda.com/anaconda/install/hashes/Anaconda2-5.0.0-Linux-x86_64.sh-hash/>

1. 激活配置环境

source ~/.bashrc

根据安装时用户的权限,选择系统环境还是用户环境

1. 检验安装anaconda2

python命令查看验证,显示有ananconda2字符串信息

import scipy不报错

注意: 新xshell窗口等, 需要关闭,重新打开,记住新的source配置路径

1. 缺少gcc安装(不报错则忽略)

查看对应版本

rpm -qa | grep gcc 查看对应版本

<https://access.redhat.com/downloads/content/479/ver=/rhel---8/8.0/x86_64/product-software>

gcc-4.8.5-16.el7.x86\_64.rpm

gcc-c++-4.8.5-16.el7.x86\_64.rpm

rpm -ivh gcc-4.8.5-16.el7.x86\_64.rpm gcc-c++-4.8.5-16.el7.x86\_64.rpm

注意 : 根据提示去官网下载依赖包安装

或者去yum离线仓库yum install 包名

后面遇到类似问题都是类似解决思路, 只演示已提供整理资源安装方式

# xgboost 安装

## pip安装命令检查

1. 注意

pip命令是anaconda2的命令,使用前再检验一遍

1. 查看输入pip命令

which pip查看

1. 解决如果不正确,手动创建

sudo ln -s /home/hadoop/ancon2/bin/pip /usr/bin/pip

1. 检验

输入pip

which pip

## xgboost安装

1. 安装

pip install xgboost-0.72.1-py2.py3-none-manylinux1\_x86\_64.whl

1. 检验

python(conda版本)命令; import xgboost不报错

1. 注意

不推荐采用tar包安装方式(以下是tar包安装方式)

解压tar -zxvf xgboost-0.72.1.tar.gz

cd xgboost-0.72.1/

安装到目标文件夹 ~/anaconda2/bin/python setup.py install(安装失败,采用安装whl文件成功)

进入xgboost/demo/guide-python; python basic\_walkthrough.py运行正常

1. 缺乏依赖

如果缺乏依赖libmpfr.so.4, 则进行安装,

libmpfr.so\*所对应的rpm包是mpfr，安装上mpfr

https://pypi.org/project/xgboost/#files

https://pypi.org/project/xgboost/0.72.1/#files

https://blog.csdn.net/bon\_mot/article/details/72964377

# 安装lightGBm

1. 下载安装包在线下载地址

[lightgbm-2.2.2-py2.py3-none-manylinux1\_x86\_64.whl](https://files.pythonhosted.org/packages/4c/3b/4ae113193b4ee01387ed76d5eea32788aec0589df9ae7378a8b7443eaa8b/lightgbm-2.2.2-py2.py3-none-manylinux1_x86_64.whl)

这里采用FTP客户端直接上传已经整理好资源包

1. 安装

pip install lightgbm-2.2.2-py2.py3-none-manylinux1\_x86\_64.whl

1. 检验

python

>>import lightgbm 不报错

或者conda2 list | grep lightgbm

1. 安装cmake

如果不报错,则无需安装

查看版本cmake –version

采用官网下载安装或者yum离线仓库自动安装, 思路同上

# 其他命令工具安装

1. zip的安装和unzip工具安装

离线仓库yun自动安装

cd /home/hadoop/rpm\_dir

采用find . -iname “\*zip\*.\*” ; find . -iname “\*unzip\*.\*”

yum install 包名

或者采用官网直接下载相应包进行安装

# JDK安装

1. JDK版本

Java SE Development Kit 8u201

1. 下载链接

https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

[jdk-8u201-linux-x64.tar.gz](javascript:%20void(0))

下载的平台类型应该为Linux

这里已经整理好安装包,下载后直接上传即可

1. 查看已安装jdk

rpm -qa|grep gcj

rpm -qa|grep jdk查看是否存在已安装jdk,卸载系统自带

1. 安装

将压缩包解压到指定目录(/opt)

mkdir /opt/jdk

tar -zxvf jdk-8u201-linux-x64.tar.gz -C /opt/

1. 创建软连接方便版本升级。

ln –s /opt/jdk1.8.0\_111 /opt/jdk

1. 配置环境变量

vi /etc/profile.d/java.sh(可以拷贝同目录下的配置文件,修改内容即可)

增加如下内容：

export JAVA\_HOME=/opt/jdk

export JRE\_HOME=$JAVA\_HOME/jre

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

export CLASSPATH=$JAVA\_HOME/lib:$JRE\_HOME/lib:$CLASSPATH

修改完成后，执行source /etc/profile 使配置生效

检验

java -version

也可以采用直接修改/etc/profile文件方式

# Scala安装

1. Scala版本

2.11.4

2. 下载链接

<http://www.scala-lang.org/download/all.html>

注意下载的平台类型应该为Linux

已经整理安装包,直接上传即可

1. 安装

将安装包解压到指定目录(/opt)

mkdir /opt/scala

tar -zxvf scala-2.11.4.tgz -C /opt

1. 创建软连接方便版本升级。

ln –s /opt/scala-2.11.4 /opt/scala

1. 配置环境变量

/etc/profile/scala.sh

增加如下内容：

export SCALA\_HOME=/opt/scala

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

export CLASSPATH=$SCALA\_HOME/lib:$CLASSPATH

配置完成之后source /etc/profile使之生效

检验

scala命令进行待输入状态不报错

# sbt安装（开发环境）

1. 版本

sbt-1.2.8

1. 上传仓库文件

上传服务器或者虚拟机/home 下 .sbt 和 .ivy2文件夹,这里已经整理好直接拷贝

里面包含boot和cache和local及repositories等文件夹或文件

建议:传输文件时,提前将文件夹进行打包,防止传输过程中出现错误或者丢包

1. 解压安装

将安装包解压到指定目录(/opt)

mkdir /opt/sbt

tar -zxvf sbt-1.2.8.tgz -C /opt

创建软连接方便版本升级。

ln –s /opt/sbt-1.2.8 /opt/sbt

1. 配置环境变量

vi /etc/profile.d/sbt.sh

export SBT\_HOME=/opt/sbt

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

配置完成之后source /etc/profile使之生效

1. 修改sbt的repository，使用私服

cd /opt/sbt

vi conf\sbtconfig.txt

　-Dsbt.global.base=/home/hadoop/.sbt

　-Dsbt.boot.directory=/home/hadoop/.sbt/boot/

　　 -Dsbt.ivy.home=/home/hadoop/.ivy2

保存退出

1. vim ~/.sbt/repositories(没有的话，手工创建)

文件内容如下：

[repositories]

local

maven-local

maven-central: http://repo1.maven.org/maven2/

aliyun-mvn: http://maven.aliyun.com/nexus/content/groups/public/

aliyun-ivy: https://maven.aliyun.com/nexus/content/groups/public/, [organization]/[module]/(scala\_[scalaVersion]/)(sbt\_[sbtVersion]/)[revision]/[type]s/[artifact](-[classifier]).[ext]

保存退出

注意:

解压repository.tgz,将repository下的所有目录拷贝到~/.ivy2/local下

采用直接拷贝现有离线仓库(local和cache),因此不需要再操作此步骤

1. 检验

sbt sbtVersion

sbt

编译打包检验

拷贝一个简单的代码项目,进行sbt clean;sbt compile;sbt assembly进行测试

sbt clean; sbt compile; sbt assembly

# Anaconda作用

1. 包管理器:

conda（包管理器）可以很好的帮助你在计算机上安装和管理这些包，包括安装、卸载和更新包

1. 环境管理器
2. 数据分析的标准环境
3. 附带了一大批常用数据科学包，它附带了 conda、Python 和 150 多个科学包及其依赖项
4. 应用层面虚拟化环境,使用anaconda环境能够运行scala等程序

推荐使用pip freeze 或者pip list 查看版本和名称

<https://www.anaconda.com/distribution/>

<https://conda.io/en/latest/>