# Red Hat Enterprise Linux 7.4安装

[Red Hat Enterprise Linux 7.4安装 1](#_Toc15469873)

[1. 镜像下载 2](#_Toc15469874)

[2. 建虚拟机 2](#_Toc15469875)

[3. 制作yum离线仓库 4](#_Toc15469876)

[4. anconda2 安装(生产环境是在root环境下进行) 5](#_Toc15469877)

[5. xgboost 安装 6](#_Toc15469878)

[6. 安装lightGBm 6](#_Toc15469879)

[7. 其他命令工具安装 7](#_Toc15469880)

[8. JDK安装 7](#_Toc15469881)

[9. Scala安装 8](#_Toc15469882)

[10. sbt安装（开发环境） 8](#_Toc15469883)

[11. Anaconda介绍 10](#_Toc15469884)

**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

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

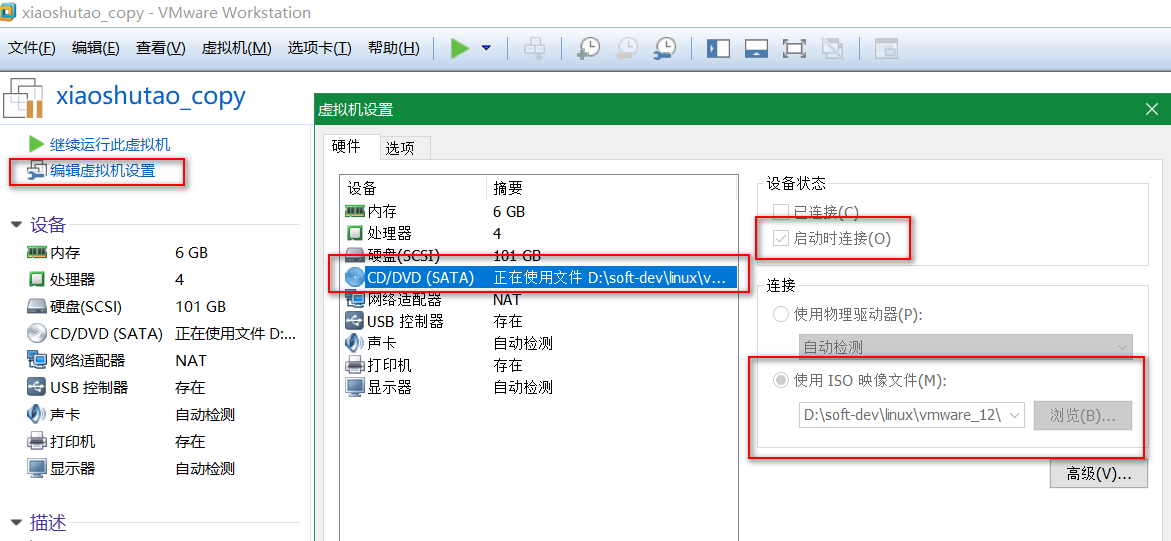
Vmware12\_pro模拟安装配置

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

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

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

查看网卡信息

dmesg | grep -i eth

查看系统网卡设备

ip addr show

修改网卡配置参数

cd /etc/sysconfig/network-scripts/

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

重启网卡

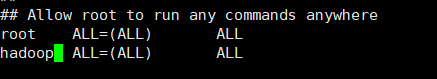
service network restart

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

创建root用户和hadoop用户

在hadoop用户下进行;加入hadoop用户进入sudoers

Root下chomod u+w /etc/sudoers



保存退出后,chmod u-w /etc/sudoers

以下操作在hadoop用户下进行

df -h查看分区情况

把/opt目录空间赋给hadoop用户

确认/opt目录有剩余空间

修改/opt增加写权限并创建目录

sudo chmod o+w /opt

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 ,第三方网站提供的可能存在不全或者版本问题

挂载镜像

mkdir /mnt/iso ;

mount -o loop /home 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

umount /mnt/iso

安装yum和repo

cd /home/hadoop/rpm\_dir

# 从镜像文件获取rpm包或者从相应官网直接搜索对应版本下载

find . -iname “\*yum\*.\*”

sudo rpm -ivh yum-3.4.3-154.el7.noarch.rpm(提示缺乏依赖)

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

检验安装

输入yum list或者输入yum查看是否报错提示

本地存储库

创建一个目录，例如。 /home/hadoop/rpm\_dir(已操作)

将RPM包移动到该目录(已操作)

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

创建存储库配置文件，例如/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

# yum install packagename

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

## anconda2 安装(生产环境是在root环境下进行)

这里在hadoop用户下进行

安装bzip2

安装anaconda2解压需要

bzip2-1.0.6-13.el7.x86\_64.rpm  
<https://pypi.org/>官网下载

yum离线配置仓库自动安装下载

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

安装anaconda2

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

上传到/home/hadoop/package/anaconda2/

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

依次执行命令：回车 -> yes -> 回车 -> yes

参考：

<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/>

激活配置环境

source ~/.bashrc(root或者home目录根据实际选择,这里采用系统变量配置)

python命令查看验证

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

缺少gcc安装(不报错则忽略)

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 \*.rpm

rpm -qa | grep gcc

## xgboost 安装

缺乏依赖libmpfr.so.4

libmpfr.so\*所对应的rpm包是mpfr，安装上mpfr包之后就顺利的安装完成了gcc。

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

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

<https://blog.csdn.net/bon_mot/article/details/72964377>

注意pip命令是anaconda2的命令,使用前再检验一遍

输入pip命令和which pip查看 ; 如果不正确,手动创建

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

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

检验1: python(conda版本)命令; import xgboost不报错

检验2: 进入xgboost/demo/guide-python; python basic\_walkthrough.py运行正常;

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)

## 安装lightGBm

注意 : lightGBm-master 2.2.4在redhat上安装失败,这里采用如下安装

[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)

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

检验:

>>python

>>import lightgbm 不报错

或者conda2 list | grep lightgbm

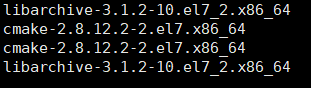
安装cmake

如果不报错,则无需安装

下载cmake https://cmake.org/download/

./bootstrap make -j8 sudo make install

cmake –version



或者采用yum离线仓库一键安装cmake

yum install package\_name

注意:

编译lightGBM：(安装失败)

cd LightGBM mkdir build cd build cmake ../ make -j8

git clone --recursive https://github.com/microsoft/LightGBM ;

cd LightGBM

mkdir build ;

cd build

cmake ..

make -j4

glibc> = 2.14是必需的。

## 其他命令工具安装

Zip的安装和Unzip工具安装

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

yum install \*\*\*

## JDK安装

JDK版本：Java SE Development Kit 8u201

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

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

注意下载的平台类型应该为Linux。

rpm -qa|grep gcj

rpm -qa|grep jdk查看是否存在已安装jdk,卸载系统自带

安装步骤：

1. 将压缩包解压到指定目录(/opt)

如果没有opt目录赋权给hadoop用户,则需要sudo; 或者采用root用户操作

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

2.创建软连接方便版本升级。

ln –s /opt/jdk1.8.0\_111 /opt/jdk

这样以后升级jdk版本时，仅需要重修修改此软连接，环境变量等的配置自动生效

1. 配置环境变量

也可以采用直接修改/etc/profile文件方式

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

## Scala安装

Scala版本：2.11.4

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

安装步骤：

1.解压：

将安装包解压到指定目录(/opt)

mkdir /opt/scala

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

2.创建软连接方便版本升级。

ln –s /opt/scala-2.11.4 /opt/scala

2.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）

安装sbt-1.2.8

拷贝 服务器或者虚拟机/home 下 .sbt and .ivy2

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

下载地址(第二个地址快些)：

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

https://github.com/sbt/sbt/releases

安装步骤：

将安装包解压到指定目录(/opt)

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

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

sudo vi sbt\conf\sbtconfig.txt

　　6

保存退出

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

第一次执行时，会下载一些文件包，然后才能正常使用，要确保联网了

sbt sbtVersion

配置完成后，运行sbt命令，会下载sbt相关的jar包。由于国内被墙等问题，下载会特别慢。可以通过下载Light Activator，这个程序里面有许多已经准备好的repository。

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

下载地址：<http://www.scala-lang.org/download/>

在下载页面选择Lightbend Activator，如下图绿色框出部分;



网站对于库已经不再更新推出, 采用直接拷贝现有离线仓库(local和cache)

检验

sbt sbtVersion

sbt

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

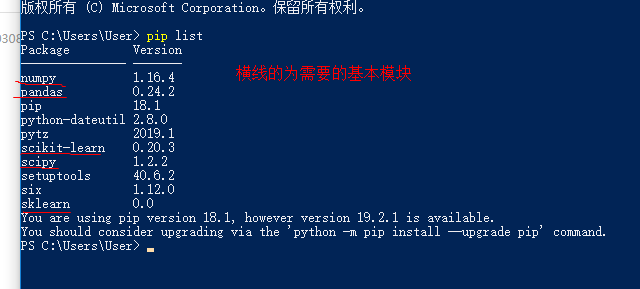
## Anaconda介绍

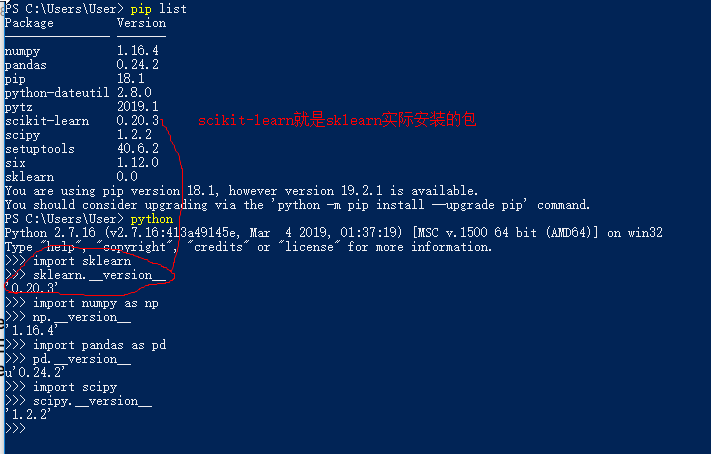
1. 包管理器:

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

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/>

<https://numpy.org/>

开源[Anaconda Distribution](https://docs.anaconda.com/anaconda/)是在Linux，Windows和Mac OS X上执行Python / R数据科学和机器学习的最简单方法。全球有超过1500万用户，它是开发，测试和培训的行业标准。一台机器，使个别数据科学家能够：

* 快速下载1,500多个Python / R数据科学包
* 使用[Conda](https://conda.io/docs/)管理库，依赖项和环境
* 使用[scikit-learn](https://scikit-learn.org/stable/)，[TensorFlow](https://www.tensorflow.org/)和[Theano](https://pypi.org/project/Theano/)开发和训练机器学习和深度学习模型
* 使用[Dask](https://dask.org/)，[NumPy](http://www.numpy.org/)，[pandas](https://pandas.pydata.org/)和[Numba](http://numba.pydata.org/)分析具有可伸缩性和性能的数据
* 使用[Matplotlib](https://matplotlib.org/)，[Bokeh](https://bokeh.pydata.org/en/latest/)，[Datashader](http://datashader.org/)和[Holoviews](http://holoviews.org/)可视化结果

[hadoop@localhost ~]$ pip freeze

alabaster==0.7.10

anaconda-client==1.6.5

anaconda-navigator==1.6.8

anaconda-project==0.8.0

asn1crypto==0.22.0

astroid==1.5.3

astropy==2.0.2

Babel==2.5.0

backports-abc==0.5

backports.functools-lru-cache==1.4

backports.shutil-get-terminal-size==1.0.0

backports.ssl-match-hostname==3.5.0.1

beautifulsoup4==4.6.0

bitarray==0.8.1

bkcharts==0.2

blaze==0.11.3

bleach==2.0.0

bokeh==0.12.7

boto==2.48.0

Bottleneck==1.2.1

cdecimal==2.3

certifi==2017.7.27.1

cffi==1.10.0

chardet==3.0.4

click==6.7

cloudpickle==0.4.0

clyent==1.2.2

colorama==0.3.9

conda==4.3.27

conda-build==3.0.22

conda-verify==2.0.0

configparser==3.5.0

contextlib2==0.5.5

cryptography==2.0.3

cycler==0.10.0

Cython==0.26.1

cytoolz==0.8.2

dask==0.15.2

datashape==0.5.4

decorator==4.1.2

distributed==1.18.3

docutils==0.14

entrypoints==0.2.3

enum34==1.1.6

et-xmlfile==1.0.1

fastcache==1.0.2

filelock==2.0.12

Flask==0.12.2

Flask-Cors==3.0.3

funcsigs==1.0.2

functools32==3.2.3.post2

futures==3.1.1

gevent==1.2.2

glob2==0.5

gmpy2==2.0.8

greenlet==0.4.12

grin==1.2.1

h5py==2.7.0

heapdict==1.0.0

html5lib==0.999999999

idna==2.6

imageio==2.2.0

imagesize==0.7.1

ipaddress==1.0.18

ipykernel==4.6.1

ipython==5.4.1

ipython-genutils==0.2.0

ipywidgets==7.0.0

isort==4.2.15

itsdangerous==0.24

jdcal==1.3

jedi==0.10.2

Jinja2==2.9.6

jsonschema==2.6.0

jupyter-client==5.1.0

jupyter-console==5.2.0

jupyter-core==4.3.0

jupyterlab==0.27.0

jupyterlab-launcher==0.4.0

lazy-object-proxy==1.3.1

lightgbm==2.2.2

llvmlite==0.20.0

locket==0.2.0

lxml==3.8.0

MarkupSafe==1.0

matplotlib==2.0.2

mccabe==0.6.1

mistune==0.7.4

mpmath==0.19

msgpack-python==0.4.8

multipledispatch==0.4.9

navigator-updater==0.1.0

nbconvert==5.3.1

nbformat==4.4.0

networkx==1.11

nltk==3.2.4

nose==1.3.7

notebook==5.0.0

numba==0.35.0+10.g143f70e90.dirty

numexpr==2.6.2

numpy==1.13.1

numpydoc==0.7.0

odo==0.5.1

olefile==0.44

openpyxl==2.4.8

packaging==16.8

pandas==0.20.3

pandocfilters==1.4.2

partd==0.3.8

path.py==10.3.1

pathlib2==2.3.0

patsy==0.4.1

pep8==1.7.0

pexpect==4.2.1

pickleshare==0.7.4

Pillow==4.2.1

pkginfo==1.4.1

ply==3.10

prompt-toolkit==1.0.15

psutil==5.2.2

ptyprocess==0.5.2

py==1.4.34

pycairo==1.13.3

pycodestyle==2.3.1

pycosat==0.6.2

pycparser==2.18

pycrypto==2.6.1

pycurl==7.43.0

pyflakes==1.5.0

Pygments==2.2.0

pylint==1.7.2

pyodbc==4.0.17

pyOpenSSL==17.2.0

pyparsing==2.2.0

PySocks==1.6.7

pytest==3.2.1

python-dateutil==2.6.1

pytz==2017.2

PyWavelets==0.5.2

PyYAML==3.12

pyzmq==16.0.2

QtAwesome==0.4.4

qtconsole==4.3.1

QtPy==1.3.1

requests==2.18.4

rope==0.10.5

ruamel-yaml==0.11.14

scandir==1.5

scikit-image==0.13.0

scikit-learn==0.19.0

scipy==0.19.1

seaborn==0.8

simplegeneric==0.8.1

singledispatch==3.4.0.3

six==1.10.0

snowballstemmer==1.2.1

sortedcollections==0.5.3

sortedcontainers==1.5.7

Sphinx==1.6.3

sphinxcontrib-websupport==1.0.1

spyder==3.2.3

SQLAlchemy==1.1.13

statsmodels==0.8.0

subprocess32==3.2.7

sympy==1.1.1

tables==3.4.2

tblib==1.3.2

terminado==0.6

testpath==0.3.1

toolz==0.8.2

tornado==4.5.2

traitlets==4.3.2

typing==3.6.2

unicodecsv==0.14.1

urllib3==1.22

wcwidth==0.1.7

webencodings==0.5.1

Werkzeug==0.12.2

widgetsnbextension==3.0.2

wrapt==1.10.11

xgboost==0.72.1

xlrd==1.1.0

XlsxWriter==0.9.8

xlwt==1.3.0

zict==0.1.2