



链滴

史上最全! 保姆级 Hadoop 安装教学

作者: [Sakura6868](#)

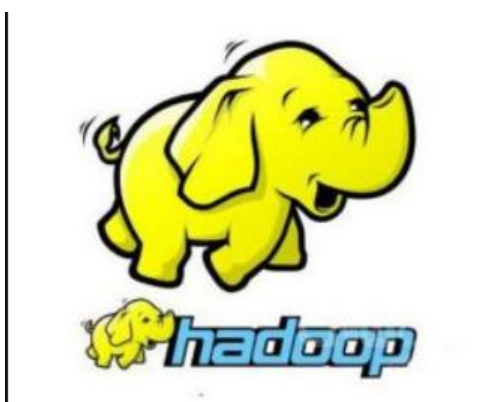
原文链接: <https://ld246.com/article/1628758274710>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



学大数据,不管怎么样始终都绕不开Hadoop这个黄色的小象



而安装Hadoop可以说是进入大数据领域的第一步了,作为学校里大数据专业还在坚持学大数据的同学,过这几年的学习还是积累了些许经验的,来一波保姆级Hadoop安装教学.

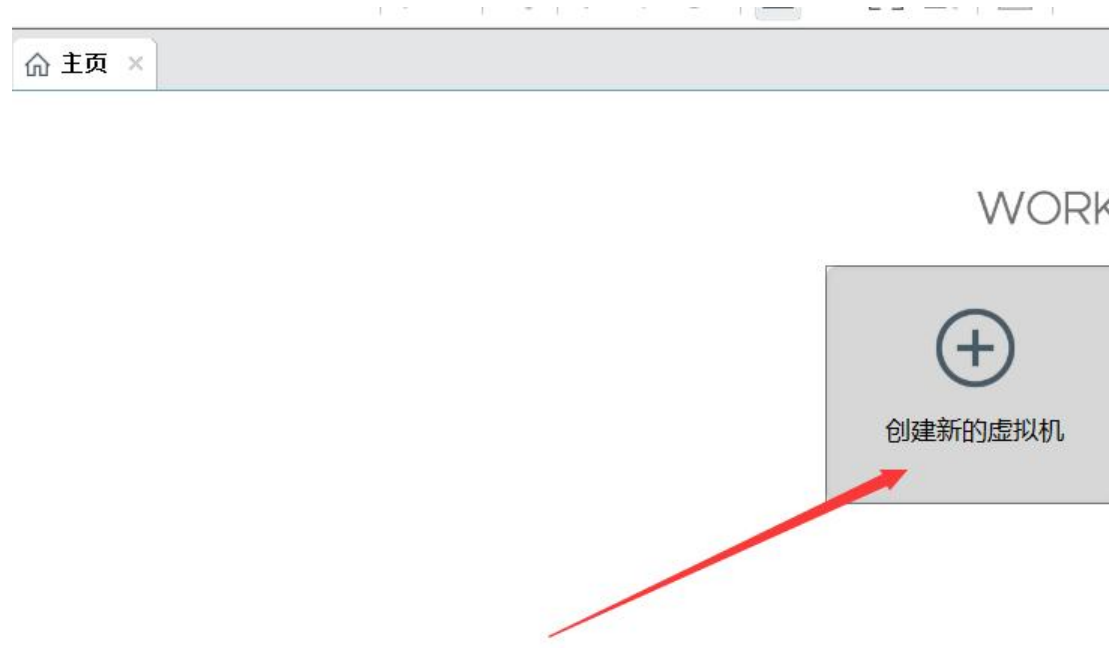
首先默认你有些许Linux的基础,并且电脑上已经安装好vmware workstation 等类似的虚拟机安装软件
当然你有钱买云服务器当我没说)

Linux虚拟机安装

- 下载centos镜像

这里我们下载的是centos7.6的镜像[传送门](#)

- 安装 centos虚拟机
 - 点击创建新的虚拟机



- 选择安装程序光盘这一栏,把我们下好的镜像导入,这里下面会检测出我们要安装的系统



- 狂点下一步,然后再点击完成即可.这是虚拟机会自动打开
- 一路回车,直到出现可视化界面,选择安装语言为中文



- 为了速度我们最好选择最小安装



- 然后就是设置root账户密码和创建我们平时使用时的用户



- 再就是慢慢等待了(大概五六分钟)
 - 最侯点击重启就大功搞成了!

安装Hadoop前的准备工作

- 静态IP和主机名配置
 - 打开ifcfg-ens33文件修改配置

```
vi /etc/sysconfig/network-scripts/ifcfg-ens33
```

```
.....  
BOOTPROTO=static #将dhcp改为static  
ONBOOT=yes #将no改为yes  
IPADDR=192.168.10.200 #添加IPADDR属性和ip地址  
PREFIX=24 #添加NETMASK=255.255.255.0或者PREFIX=24  
GATEWAY=192.168.10.0 #添加网关GATEWAY  
DNS1=114.114.114.114 #添加DNS1和备份DNS  
DNS2=8.8.8.8
```

- 重启网络服务

```
systemctl restart network  
# or  
service network restart
```

- 修改主机名

```
hostnamectl set-hostname master
```

注意：配置完ip和主机名后，最好reboot一下

- 配置/etc/hosts文件

```
vi /etc/hosts  
# 在后面添加上  
192.168.216.114 master
```

- 关闭防火墙

```
systemctl stop firewalld  
systemctl disable firewalld  
#最好也把selinux关闭掉，这是linux系统的一个安全机制，进入文件中将SELINUX设置为disabled  
vi /etc/selinux/config  
SELINUX=disabled
```

- 时间同步

- 输入tzselect 依次选5 9 1 1
- 下载ntp

```
yum install -y ntp
```

- 配置/etc/ntp.conf

```
vim /etc/ntp.conf
```

server 127.127.1.0
fudge 127.127.1.0 stratum 10

- 启动:/bin/systemctl restart ntpd.service
- 设置免密登陆
 - ssh-keygen 后 一路回车
 - ssh-copy-id -i /root/.ssh/id_rsa -p 22 root@master 后输入密码即可

Hadoop单机安装与配置

- JDK安装
 - 检查一下是否已经安装过或者系统内置JDK,如果有内置的, 将其卸载

```
rpm -qa | grep jdk #如果有,请卸载  
rpm -e xxxxxxxx --noeps #将查询到的内置jdk强制卸载
```

- 上传jdk(这里我是上传到/opt/software/目录中)
- 解压jdk到/opt/apps/下

```
cd /opt/software  
tar -zxvf jdk-8u152-linux-x64.tar.gz -C /opt/apps/
```

- 改名为jdk

```
cd /opt/apps  
mv jdk-8u152/ jdk
```

- 配置Jdk的环境变量: /etc/profile

```
vim /etc/profile  
# 后面添加  
#jdk environment  
export JAVA_HOME=/opt/apps/jdk  
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
```

- 使当前窗口生效

```
source /etc/profile
```

- 验证java环境

```
java -version  
javac
```

- Hadoop单机安装
 - 上传hadoop(这里我是上传到/opt/software/目录中)
 - 解压到/opt/apps/下

```
cd /opt/software/  
tar -zxvf hadoop-2.7.6.tar.gz -C /opt/apps/
```

- 改名为hadoop

```
cd /opt/apps
mv hadoop-2.7.6/ hadoop
```

- 配置hadoop的环境变量

```
vi /etc/profile
#hadoop environment
export HADOOP_HOME=/opt/apps/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

- 使当前窗口生效

```
source /etc/profile
```

- 验证hadoop

```
hadoop version
```

- 配置 hadoop-env.sh文件

```
vi $HADOOP_HOME/etc/hadoop/hadoop-env.sh
# 更改如下内容
export JAVA_HOME=/simple/jdk1.8.0_152
```

Hadoop 伪分布式安装与配置

伪分布式模式介绍

首先我们要先了解了解伪分布式有什么特点

1.特点

- 在一台机器上安装，使用的是分布式思想，即分布式文件系统，非本地文件系统。
- Hdfs涉及到的相关守护进程(namenode,datanode,secondarynamenode)都运行在一台机器上，是独立的

java进程。

2. 用途

比Standalone mode 多了代码调试功能，允许检查内存使用情况，HDFS输入输出，以及其他的守护进程交互。

互。

由于我们在前面已经进行了免密登陆 静态ip host映射 的配置也安装了jdk和Hadoop 所以我们接下来直接进入文件的配置

文件配置

- core-site.xml的配置

```
[root@master ~]# cd $HADOOP_HOME/etc/hadoop
[root@master hadoop]# vi core-site.xml
<configuration>
```



```
<!-- 配置分布式文件系统的schema和ip以及port,默认8020-->
```

```
<property>  
<name>fs.defaultFS</name>  
<value>hdfs://localhost/</value>  
</property>  
</configuration>
```

扩展: hadoop1.x的默认端口是9000, hadoop2.x的默认端口是8020, 使用哪一个都可以

- hdfs-site.xml的配置

```
[root@master hadoop]# vi hdfs-site.xml  
<configuration>  
<!-- 配置副本数, 注意, 伪分布模式只能是1。-->  
<property>  
<name>dfs.replication</name>  
<value>1</value>  
</property>  
</configuration>
```

- hadoop-env.sh的配置: 指定jdk的环境 (和单机模式一样,这里不再赘述)

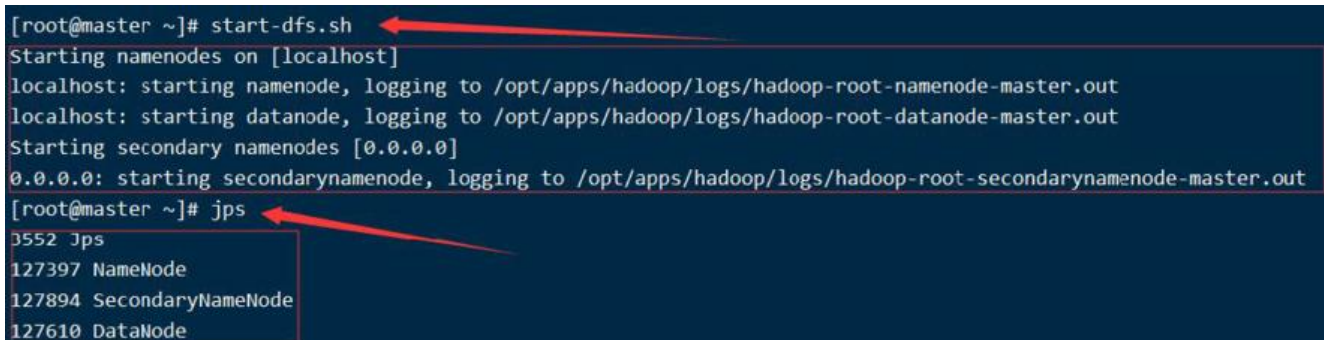
格式化NameNode

```
hdfs namenode -format
```

启动HDFS

```
start-dfs.sh
```

- jps查看进程



```
[root@master ~]# start-dfs.sh  
Starting namenodes on [localhost]  
localhost: starting namenode, logging to /opt/apps/hadoop/logs/hadoop-root-namenode-master.out  
localhost: starting datanode, logging to /opt/apps/hadoop/logs/hadoop-root-datanode-master.out  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: starting secondarynamenode, logging to /opt/apps/hadoop/logs/hadoop-root-secondarynamenode-master.out  
[root@master ~]# jps  
3552 Jps  
127397 NameNode  
127894 SecondaryNameNode  
127610 DataNode
```

WebUI_50070

可以在浏览器上输入: 192.168.10.200:50070 来查看一下伪分布式集群的信息

- 1. 浏览一下页面上提示的ClusterID,BlockPoolID
- 2. 查看一下活跃节点(Live Nodes)的个数, 应该是1个

Overview 'localhost:8020' (active)

Started:	Tue Mar 24 05:26:07 CST 2020
Version:	2.7.6, r085099c66cf28be31604560c376fa282e69282b8
Compiled:	2018-04-18T01:33Z by kshvachk from branch-2.7.6
Cluster ID:	CID-da832497-15bf-4115-bf94-5df92286567b
Block Pool ID:	BP-267391110-192.168.10.131-1584826384251

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 29.57 MB of 46.49 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 38.51 MB of 39.69 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	17.3 GB
DFS Used:	32 KB (0%)
Non DFS Used:	2.13 GB
DFS Remaining:	14.27 GB (82.45%)
Block Pool Used:	32 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)

简单解释:

Compiled:编译 hadoop是由kshvachk工具集成的

Cluster ID:集群id

Block Pool ID:datanode节点的block池的id,每个datanode节点的都要一样

完全分布式集群的安装与配置

单机式和伪分布式不能用于生产环境,只能在平时的调试和学习中用到,我们真正用到的还是完全分布的集群

虚拟机说明

采用虚拟机克隆的方式克隆两台虚拟机使得三台虚拟机配置如下

```
主机名      IP
master 192.168.10.200
slave1 192.168.10.201
slave2 192.168.10.202
```

说明:如果是克隆操作的slave1 slave2就不需要关防火墙了只用在/etc/hosts中把这两台克隆来的映

加上 然后把ip改一下

注意，注意，注意：

1.如果你是从伪分布式过来的，最好先把伪分布式的相关守护进程关闭：stop-all.sh

2.删除原来伪分布式的相关设置

如果原来使用的是默认路径,现在已经没有用了

如果原来使用的跟现在全分布式路径一样,因为这里跟之前的初始化的内容不一样,而且这个文件要让系统自动生成

综上:要删除掉namenode和datanode的目录

守护进程布局

我们搭建hdfs的完全分布式，顺便搭建一下yarn。hdfs和yarn的相关守护进程的布局如下：

master: namenode,datanode,ResourceManager,nodemanager

slave1: datanode,nodemanager,secondarynamenode

slave2: datanode,nodemanager

Hadoop的配置文件的配置

- 配置前说明：

1.我们先在master机器节点上配置hadoop的相关属性。

2.在 `<value>` `</value>` 之间的值不能有空格

3.master配好后直接克隆两台虚拟机,修改ip等配置即可

- 配置core-site.xml文件

```
[root@master ~]# cd $HADOOP_HOME/etc/hadoop/
[root@master hadoop]# vi core-site.xml
<configuration>
<!-- hdfs的地址名称: scheme,ip,port-->
<property>
<name>fs.defaultFS</name>
<value>hdfs://master:8020</value>
</property>
<!-- hdfs的基础路径, 被其他属性所依赖的一个基础路径 -->
<property>
<name>hadoop.tmp.dir</name>
<value>/opt/apps/tmp</value>
</property>
</configuration>
```

- 配置hdfs-site.xml文件

```
[root@master hadoop]# vi core-site.xml
<configuration>
<!-- namenode守护进程管理的元数据文件fsimage存储的位置-->
<property>
<name>dfs.namenode.name.dir</name>
```

```

<value>file://${hadoop.tmp.dir}/dfs/name</value>
</property>
<!-- 确定DFS数据节点应该将其块存储在本地文件系统的何处-->
<property>
<name>dfs.datanode.data.dir</name>
<value>file://${hadoop.tmp.dir}/dfs/data</value>
</property>
<!-- 块的副本数-->
<property>
<name>dfs.replication</name>
<value>3</value>
</property>
<!-- 块的大小(128M),下面的单位是字节-->
<property>
<name>dfs.blocksize</name>
<value>134217728</value>
</property>
<!-- secondarynamenode守护进程的http地址: 主机名和端口号。参考守护进程布局-->
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>slave1:50090</value>
</property>
<!--文件的检测目录-->
<property>
<name>fs.checkpoint.dir</name>
<value>file:///${hadoop.tmp.dir}/checkpoint/dfs/cname</value>
</property>
<!--日志edits的检测目录-->
<property>
<name>fs.checkpoint.edits.dir</name>
<value>file:///${hadoop.tmp.dir}/checkpoint/dfs/cname</value>
</property>
<property>
<name>dfs.http.address</name>
<value>master:50070</value>
</property>
</configuration>

```

- 配置mapred-site.xml文件

如果只是搭建hdfs,只需要配置core-site.xml和hdfs-site.xml文件就可以了, 但是如果要学习MapRed ce

是需要YARN资源管理器的, 因此, 在这里, 提前配置一下相关文件

```

[root@master hadoop]# cp mapred-site.xml.template mapred-site.xml
[root@master hadoop]# vi mapred-site.xml
<configuration>
<!-- 指定mapreduce使用yarn资源管理器-->
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<!-- 配置作业历史服务器的地址-->
<property>

```

```
<name>mapreduce.jobhistory.address</name>
<value>master:10020</value>
</property>
<!-- 配置作业历史服务器的http地址-->
<property>
<name>mapreduce.jobhistory.webapp.address</name>
<value>master:19888</value>
</property>
</configuration>
```

- 配置yarn-site.xml文件

```
[root@master hadoop]# vi yarn-site.xml
<configuration>
<!-- 指定yarn的shuffle技术-->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<!-- 指定resourcemanager的主机名-->
<property>
<name>yarn.resourcemanager.hostname</name>
<value>master</value>
</property>
<!--下面的可选-->
<!--指定shuffle对应的类 -->
<property>
<name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<!--配置resourcemanager的内部通讯地址-->
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8032</value>
</property>
<!--配置resourcemanager的scheduler的内部通讯地址-->
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<!--配置resourcemanager的资源调度的内部通讯地址-->
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8031</value>
</property>
<!--配置resourcemanager的管理员的内部通讯地址-->
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:8033</value>
</property>
<!--配置resourcemanager的web ui 的监控页面-->
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:8088</value>
```

```
</property>
</configuration>
```

- 配置hadoop-env.sh脚本文件(同单机模式不再赘述)
- 配置slaves文件, 此文件用于指定datanode守护进程所在的机器节点主机名

```
[root@master hadoop]# vi slaves
master
slave1
slave2
```

- 配置yarn-env.sh文件, 此文件可以不配置, 不过, 最好还是修改一下yarn的jdk环境比较好

```
# User for YARN daemons
export HADOOP_YARN_USER=${HADOOP_YARN_USER:-yarn}

# resolve links - $0 may be a softlink
export YARN_CONF_DIR="${YARN_CONF_DIR:-$HADOOP_YARN_HOME/conf}"

# some Java parameters
# export JAVA_HOME=/home/y/libexec/jdk1.6.0/  写成:export JAVA_HOME=/opt/apps/jdk
if [ "$JAVA_HOME" != "" ]; then
    #echo "run java in $JAVA_HOME"
    JAVA_HOME=$JAVA_HOME
fi

if [ "$JAVA_HOME" = "" ]; then
    echo "Error: JAVA_HOME is not set."
    exit 1
fi
```

- 另外两台机器配置说明

当把master机器上的hadoop的相关文件配置完毕后, 我们有以下两种方式来选择配置另外几台机器hadoop.

- "scp" 进行同步(本方法适用于多台虚拟机已经提前搭建出来的场景)
- 虚拟机克隆

从头开始再安装两台虚拟机还是很麻烦的这里我们选择克隆

- 打开一个新克隆出来的虚拟机, 修改主机名
- 修改ip地址
- 重启网络服务
- 其他新克隆的虚拟机重复以上1~3步
- 免密登陆的验证

从master机器上, 连接其他的每一个节点, 验证免密是否好使, 同时去掉第一次的询问步骤

- 建议：每台机器在重启网络服务后，最好reboot一下

具体操作步骤上面都有这里不再赘述

格式化NameNode

```
# 在master进行操作  
hdfs namenode -format
```

如果你操作顺利,下面就可以启动Hadoop集群了!

启动脚本和关闭脚本介绍

1. 启动脚本

```
-- start-dfs.sh :用于启动hdfs集群的脚本  
-- start-yarn.sh :用于启动yarn守护进程  
-- start-all.sh :用于启动hdfs和yarn
```

2. 关闭脚本

```
-- stop-dfs.sh :用于关闭hdfs集群的脚本  
-- stop-yarn.sh :用于关闭yarn守护进程  
-- stop-all.sh :用于关闭hdfs和yarn
```

3. 单个守护进程脚本

```
-- hadoop-daemons.sh :用于单独启动或关闭hdfs的某一个守护进程的脚本  
-- hadoop-daemon.sh :用于单独启动或关闭hdfs的某一个守护进程的脚本
```

reg:

```
hadoop-daemon.sh [start|stop] [namenode|datanode|secondarynamenode]
```

```
-- yarn-daemons.sh :用于单独启动或关闭hdfs的某一个守护进程的脚本
```

```
-- yarn-daemon.sh :用于单独启动或关闭hdfs的某一个守护进程的脚本
```

reg:

```
yarn-daemon.sh [start|stop] [resourcemanager|nodemanager]
```

最后每台主机进行 jps进程查看操作 如果启动的进程是按照我们的进程布局来的,那么恭喜你Hadoop群搭建成功!

当然如果有些进程没有启动成功 我们可以对症下药来修改相应进程的配置文件