

# 第3章

## 项目部署的环境准备

通过前面章节的分析，我们已经明确了将要使用的框架类型和框架版本，本章将根据前面章节所描述的需求分析，搭建一个完整的项目开发环境，即便读者的计算机中已经具备这些环境，也建议浏览一遍本章内容，因为这对后续开发过程中理解代码和命令行很有帮助。

### 3.1 集群规划与服务器配置

在本书中，我们需要在个人计算机上搭建一个具有 3 台节点服务器的微型集群。3 台节点服务器的具体设置如下。

- 节点服务器 1：IP 地址为 192.168.10.102，主机名为 `hadoop102`。
- 节点服务器 2：IP 地址为 192.168.10.103，主机名为 `hadoop103`。
- 节点服务器 3：IP 地址为 192.168.10.104，主机名为 `hadoop104`。

3 台节点服务器的安装部署情况如表 3-1 所示。

表 3-1 3 台节点服务器的安装部署情况

hadoop102	hadoop103	hadoop104
CentOS7.5	CentOS7.5	CentOS7.5
JDK8	JDK8	JDK8
Hadoop3.3.4	Hadoop3.3.4	Hadoop3.3.4

在个人计算机上安装部署 3 台节点服务器的具体流程，读者可以在通过关注“尚硅谷教育”公众号获取的本书附赠课程资料中找到，此处不再赘述。

### 3.2 安装 JDK 与 Hadoop

在准备好集群环境后，我们需要安装 JDK 与 Hadoop。

#### 3.2.1 准备虚拟机环境

在正式安装 JDK 与 Hadoop 前，首先需要在 3 台节点服务器上进行一些配置。

## 1. 创建安装目录

(1) 在/opt目录下创建 module、software 文件夹。

```
[atguigu@hadoop102 opt]$ sudo mkdir module
[atguigu@hadoop102 opt]$ sudo mkdir software
```

(2) 修改 module、software 文件夹的所有者。

```
[atguigu@hadoop102 opt]$ sudo chown atguigu:atguigu module/ software/
[atguigu@hadoop102 opt]$ ll
```

总用量 8

```
drwxr-xr-x. 2 atguigu atguigu 4096 1月 17 14:37 module
drwxr-xr-x. 2 atguigu atguigu 4096 1月 17 14:38 software
```

之后所有的软件安装操作都将在 module 和 software 文件夹中进行。

## 2. 配置3台虚拟机免密登录

为什么需要配置免密登录呢？这与 Hadoop 分布式集群的架构有关。我们搭建的 Hadoop 分布式集群是主从架构，在配置了节点服务器间免密登录之后，就可以方便地通过主节点服务器启动从节点服务器，而不用手动输入用户名和密码。

(1) 免密登录原理如图 3-1 所示。

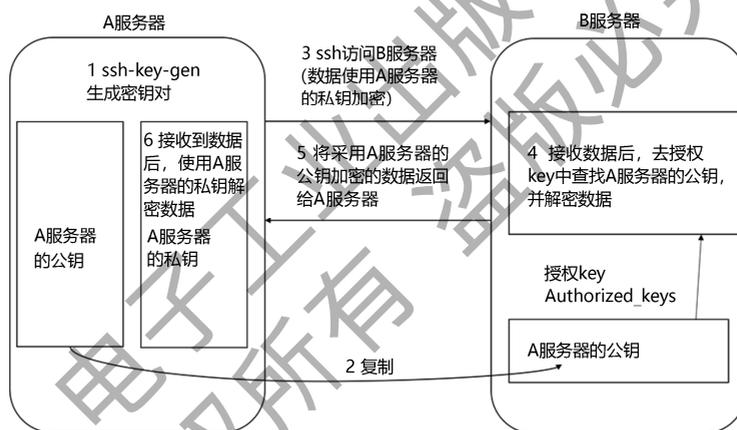


图 3-1 免密登录原理

(2) 生成公钥和私钥。

```
[atguigu@hadoop102 .ssh]$ ssh-keygen -t rsa
```

然后连续按 3 次 Enter 键，就会生成 2 个文件：id\_rsa（私钥）、id\_rsa.pub（公钥）。

(3) 将公钥复制到想要配置免密登录的目标机器上。

```
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop102
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop103
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop104
```

**注意：**需要在 hadoop102 上采用 root 账号，配置免密登录到 hadoop102、hadoop103、hadoop104；还需要在 hadoop103 上采用 atguigu 账号，配置免密登录到 hadoop102、hadoop103、hadoop104。

.ssh 文件夹下的文件功能解释如下。

- known\_hosts: 记录 SSH 访问过计算机的公钥。
- id\_rsa: 生成的私钥。
- id\_rsa.pub: 生成的公钥。
- authorized\_keys: 存放授权过的免密登录服务器公钥。

### 3. 配置时间同步

为什么要配置节点服务器间的时间同步呢？

即将搭建的 Hadoop 分布式集群需要解决 2 个问题：数据的存储和数据的计算。

Hadoop 对大型文件的存储采用分块的方法，将文件切分成多块，以块为单位，分发到各台节点服务器上进行存储。当这个大型文件再次被访问时，需要从 3 台节点服务器上分别拿出数据，再进行计算。由于计算机之间的通信和数据的传输一般是以时间为约定条件的，如果 3 台节点服务器的时间不一致，就会导致在读取块数据时出现时间延迟，进而可能导致访问文件时间过长，甚至失败，所以配置节点服务器间的时间同步非常重要。

第一步：配置时间服务器（必须是 root 用户）。

(1) 检查所有节点服务器 ntp 服务状态和开机自启状态。

```
[root@hadoop102 ~]# systemctl status ntpd
[root@hadoop102 ~]# systemctl is-enabled ntpd
```

(2) 在所有节点服务器关闭 ntp 服务和开机自启动。

```
[root@hadoop102 ~]# systemctl stop ntpd
[root@hadoop102 ~]# systemctl disable ntpd
```

(3) 修改 ntp 配置文件。

```
[root@hadoop102 ~]# vim /etc/ntp.conf
```

修改内容如下。

① 修改 1（设置本地网络上的主机不受限制），将以下配置前的“#”删除，解开此行注释。

```
#restrict 192.168.10.0 mask 255.255.255.0 nomodify notrap
```

② 修改 2（设置为不采用公共的服务器）。

```
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

将上述内容修改为：

```
#server 0.centos.pool.ntp.org iburst
#server 1.centos.pool.ntp.org iburst
#server 2.centos.pool.ntp.org iburst
#server 3.centos.pool.ntp.org iburst
```

③ 修改 3（添加一个默认的内部时钟数据，使用它为局域网用户提供服务）。

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

(4) 修改/etc/sysconfig/ntpd 文件。

```
[root@hadoop102 ~]# vim /etc/sysconfig/ntpd
```

增加如下内容（让硬件时间与系统时间一起同步）。

```
SYNC_HWLOCK=yes
```

重新启动 ntpd 文件。

```
[root@hadoop102 ~]# systemctl status ntpd
ntpd 已停
```

```
[root@hadoop102 ~]# systemctl start ntpd
```

```
正在启动 ntpd: [确定]
```

执行：

```
[root@hadoop102 ~]# systemctl enable ntpd
```

第二步：配置其他服务器（必须是 root 用户）。

配置服务器每 10 分钟与时间服务器同步一次。

```
[root@hadoop103 ~]# crontab -e
```

编写脚本。

```
*/10 * * * * /usr/sbin/ntpdate hadoop102
```

修改 hadoop103 的节点服务器时间，使其与另外 2 台节点服务器时间不同步。

```
[root@hadoop103 hadoop]# date -s "2023-1-11 11:11:11"
```

10 分钟后查看该服务器是否与时间服务器同步。

```
[root@hadoop103 hadoop]# date
```

#### 4. 编写集群分发脚本

集群间数据的复制通用的 2 个命令是 `scp` 和 `rsync`，其中，`rsync` 命令可以只对差异文件进行更新，使用起来非常方便，但是在使用时需要操作者频繁输入各种命令参数，为了能够更方便地使用该命令，我们编写一个集群分发脚本，主要实现目前集群间的数据分发功能。

第一步：脚本需求分析。循环复制文件到所有节点服务器的相同目录下。

(1) 原始复制。

```
rsync -rv /opt/module root@hadoop103:/opt/
```

(2) 期望脚本效果。

```
xsync path/filename #要同步的文件路径或文件名
```

(3) 在 `/home/atguigu/bin` 目录下存放的脚本，`atguigu` 用户可以在系统任何地方直接执行。

第二步：脚本实现。

(1) 在 `/home/atguigu` 目录下创建 `bin` 目录，并在 `bin` 目录下使用 `vim` 命令创建文件 `xsync`，文件内容如下。

```
[atguigu@hadoop102 ~]$ mkdir bin
[atguigu@hadoop102 ~]$ cd bin/
[atguigu@hadoop102 bin]$ touch xsync
[atguigu@hadoop102 bin]$ vim xsync
#!/bin/bash
#获取输入参数个数，如果没有参数，就直接退出
pcount=$#
if((pcount==0)); then
echo no args;
exit;
fi

#获取文件名称
p1=$1
fname=`basename $p1`
echo fname=$fname

#获取上级目录到绝对路径
pdir=`cd -P $(dirname $p1); pwd`
echo pdir=$pdir

#获取当前用户名称
user=`whoami`

#循环
for((host=103; host<105; host++)); do
    echo ----- hadoop$host -----
    rsync -rvl $pdir/$fname $user@hadoop$host:$pdir
done
```

(2) 修改脚本 `xsync`，使其具有执行权限。

```
[atguigu@hadoop102 bin]$ chmod +x xsync
```

(3) 调用脚本的形式：`xsync` 文件名称。

```
[atguigu@hadoop102 bin]$ xsync /home/atguigu/bin
```

## 3.2.2 安装 JDK

JDK 是 Java 的开发工具箱，是整个 Java 的核心，包括 Java 运行环境、Java 工具和 Java 基础类库，JDK 是学习大数据技术的基础。即将搭建的 Hadoop 分布式集群的安装程序就是使用 Java 开发的，所有 Hadoop 分布式集群要想正常运行，必须安装 JDK。

(1) 在 3 台虚拟机上分别卸载现有的 JDK。

```
[atguigu@hadoop102 opt]# sudo rpm -qa | grep -i java | xargs -n1 sudo rpm -e --nodeps
[atguigu@hadoop103 opt]# sudo rpm -qa | grep -i java | xargs -n1 sudo rpm -e --nodeps
[atguigu@hadoop104 opt]# sudo rpm -qa | grep -i java | xargs -n1 sudo rpm -e --nodeps
```

(2) 将 JDK 导入 `opt` 目录下的 `software` 文件夹中。

① 在 Linux 下的 `opt` 目录中查看软件包是否导入成功。

```
[atguigu@hadoop102 opt]$ cd software/
[atguigu@hadoop102 software]$ ls
jdk-8u212-linux-x64.tar.gz
```

② 解压 JDK 到 `opt/module` 目录下，使用 `tar` 命令来解压 `.tar` 或 `.tar.gz` 格式的压缩包，使用 `-z` 选项指定解压 `.tar.gz` 格式的压缩包。使用 `-f` 选项指定解压文件，使用 `-x` 选项指定解包操作，使用 `-v` 选项显示解压过程，使用 `-C` 选项指定解压路径。

```
[atguigu@hadoop102 software]$ tar -zxvf jdk-8u212-linux-x64.tar.gz -C /opt/module/
```

(3) 配置 JDK 环境变量，方便使用 JDK 的程序调用 JDK。

① 先获取 JDK 路径。

```
[atgui@hadoop102 jdk1.8.0_144]$ pwd
/opt/module/jdk1.8.0_212
```

② 新建 `/etc/profile.d/my_env.sh` 文件，需要注意的是，`/etc/profile.d` 路径属于 `root` 用户，需要使用 `sudo vim` 命令才可以对它进行编辑。

```
[atguigu@hadoop102 software]$ sudo vim /etc/profile.d/my_env.sh
```

在 `profile` 文件末尾添加 JDK 路径，添加的内容如下。

```
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk1.8.0_212
export PATH=$PATH:$JAVA_HOME/bin
```

保存后退出。

```
:wq
```

③ 修改环境变量后，需要执行 `source` 命令使修改后的文件生效。

```
[atguigu@hadoop102 jdk1.8.0_212]$ source /etc/profile.d/my_env.sh
```

(4) 通过执行 `java -version` 命令，测试 JDK 是否安装成功。

```
[atguigu@hadoop102 jdk1.8.0_212]# java -version
java version "1.8.0_212"
```

如果执行 `java -version` 命令后无法显示 Java 版本，就执行以下命令重启服务器。

```
[atguigu@hadoop102 jdk1.8.0_212]$ sync
[atguigu@hadoop102 jdk1.8.0_212]$ sudo reboot
```

(5) 分发 JDK 给所有节点服务器。

```
[atguigu@hadoop102 jdk1.8.0_212]$ xsync /opt/module/jdk1.8.0_212
```

(6) 分发环境变量。

```
[atguigu@hadoop102 jdk1.8.0_212]$ xsync /etc/profile.d/my_env.sh
```

(7) 执行 `source` 命令，使环境变量在每台虚拟机上生效。

```
[atguigu@hadoop103 jdk1.8.0_212]$ source /etc/profile.d/my_env.sh
```

```
[atguigu@hadoop104 jdk1.8.0_212]$ source /etc/profile.d/my_env.sh
```

### 3.2.3 安装 Hadoop

在搭建 Hadoop 分布式集群时，每台节点服务器上的 Hadoop 配置基本相同，因此只需要在 `hadoop102` 节点服务器上进行操作，配置完成之后同步到另外 2 台节点服务器上即可。

(1) 将 Hadoop 的安装包 `hadoop-3.3.4.tar.gz` 导入 `opt` 目录下的 `software` 文件夹，该文件夹被指定用来存储各软件的安装包。

① 进入 Hadoop 安装包路径。

```
[atguigu@hadoop102 ~]$ cd /opt/software/
```

② 解压安装包到 `opt/module` 文件中。

```
[atguigu@hadoop102 software]$ tar -zxvf hadoop-3.3.4.tar.gz -C /opt/module/
```

③ 查看是否解压成功。

```
[atguigu@hadoop102 software]$ ls /opt/module/  
hadoop-3.3.4
```

(2) 将 Hadoop 添加到环境变量中，可以直接使用 Hadoop 的相关指令进行操作，而不用指定 Hadoop 的目录。

① 获取 Hadoop 安装路径。

```
[atguigu@ hadoop102 hadoop-3.3.4]$ pwd  
/opt/module/hadoop-3.3.4
```

② 打开 `/etc/profile` 文件。

```
[atguigu@ hadoop102 hadoop-3.3.4]$ sudo vim /etc/profile.d/my_env.sh
```

在 `profile` 文件末尾添加 Hadoop 路径，添加的内容如下。

```
##HADOOP_HOME  
export HADOOP_HOME=/opt/module/hadoop-3.3.4  
export PATH=$PATH:$HADOOP_HOME/bin  
export PATH=$PATH:$HADOOP_HOME/sbin
```

③ 保存后退出。

```
:wq
```

④ 执行 `source` 命令，使修改后的文件生效。

```
[atguigu@ hadoop102 hadoop-3.3.4]$ source /etc/profile.d/my_env.sh
```

(3) 测试是否安装成功。

```
[atguigu@hadoop102 ~]$ hadoop version  
Hadoop 3.3.4
```

(4) 如果执行 `hadoop version` 命令后无法显示 Java 版本，就执行以下命令重启服务器。

```
[atguigu@ hadoop101 hadoop-3.3.4]$ sync
```

```
[atguigu@ hadoop101 hadoop-3.3.4]$ sudo reboot
```

(5) 分发 Hadoop 给所有节点服务器。

```
[atguigu@hadoop100 hadoop-3.3.4]$ xsync /opt/module/hadoop-3.3.4
```

(6) 分发环境变量。

```
[atguigu@hadoop100 hadoop-3.3.4]$ xsync /etc/profile.d/my_env.sh
```

(7) 执行 `source` 命令，使环境变量在每台虚拟机上生效。

```
[atguigu@hadoop103 hadoop-3.3.4]$ source /etc/profile.d/my_env.sh
```

```
[atguigu@hadoop104 hadoop-3.3.4]$ source /etc/profile.d/my_env.sh
```

### 3.2.4 Hadoop 的分布式集群部署

Hadoop 的运行模式包括本地模式、伪分布式模式和完全分布式模式。本次主要搭建实际生产环境中比较常用的完全分布式模式，在搭建完全分布式模式之前，需要对集群部署进行提前规划，不要将过多的服务集中到一台节点服务器上。我们将负责管理工作的 NameNode 和 ResourceManager 分别部署在两台节点服务器上，在另一台节点服务器上部署 SecondaryNameNode，所有节点服务器均承担 DataNode 和 NodeManager 角色，并且 DataNode 和 NodeManager 通常存储在同一台节点服务器上，所有角色尽量做到均衡分配。

(1) 集群部署规划如表 3-2 所示。

表 3-2 集群部署规划

节点服务器	hadoop102	hadoop103	hadoop104
HDFS	NameNode DataNode	DataNode	SecondaryNameNode DataNode
YARN	NodeManager	ResourceManager NodeManager	NodeManager

(2) 对集群角色的分配主要依靠配置文件，配置集群文件的细节如下。

① 核心配置文件为 core-site.xml，该配置文件属于 Hadoop 的全局配置文件，我们主要对分布式文件系统 NameNode 的入口地址和分布式文件系统中数据落地到服务器本地磁盘的位置进行配置，代码如下。

```
[atguigu@hadoop102 hadoop]$ vim core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <!-- 指定 NameNode 的地址 -->
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop102:8020</value>
  </property>
  <!-- 指定 Hadoop 数据的存储目录 -->
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/opt/module/hadoop-3.3.4/data</value>
  </property>
  <!-- 配置 HDFS 网页登录使用的静态用户为 atguigu -->
  <property>
    <name>hadoop.http.staticuser.user</name>
    <value>atguigu</value>
  </property>
  <!-- 配置该 atguigu (superUser) 允许通过代理访问的主机节点 -->
  <property>
    <name>hadoop.proxyuser.atguigu.hosts</name>
    <value>*</value>
  </property>
```

```

<!-- 配置该 atguigu (superUser) 允许通过代理用户所属组 -->
<property>
  <name>hadoop.proxyuser.atguigu.groups</name>
  <value>*</value>
</property>
<!-- 配置该 atguigu (superUser) 允许通过代理的用户-->
<property>
  <name>hadoop.proxyuser.atguigu.users</name>
  <value>*</value>
</property>
</configuration>

```

② HDFS 的配置文件为 `hdfs-site.xml`，在这个配置文件中，我们主要对 HDFS 文件系统的属性进行配置。

```

[atguigu@hadoop102 hadoop]$ vim hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <!-- NameNode Web 端访问地址-->
  <property>
    <name>dfs.namenode.http-address</name>
    <value>hadoop102:9870</value>
  </property>
  <!-- SecondaryNameNode Web 端访问地址-->
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop104:9868</value>
  </property>
  <!-- 测试环境指定 HDFS 副本的数量为 1 -->
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration >

```

③ 关于 YARN 的配置文件 `yarn-site.xml`，主要配置如下两个参数。

```

[atguigu@hadoop102 hadoop]$ vim yarn-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <!-- 为 NodeManager 配置额外的 shuffle 服务 -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <!-- 指定 ResourceManager 的地址-->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop103</value>
  </property>
  <!-- task 继承 NodeManager 环境变量-->
  <property>
    <name>yarn.nodemanager.env-whitelist</name>

```