第 3 章 平台基础环境配置

学习目标

- 掌握集群网络连接与配置的方法
- 掌握 SSH 无密钥登录配置的方法
- 掌握 Hadoop 全分布式结构
- 掌握 JDK 安装与配置的方法

平台基础环境配置直接关系平台及相关组件的安装是否成功。基础环境配置错误会导致后期安装失败。本章主要介绍基础环境配置涉及的网络连接配置、主机地址映射、无密钥登录、JDK 的安装与配置等内容。

电子工业出版社版权所有 盗版必究

3.1 集群网络连接

3.1.1 实验环境下的大数据 Hadoop 平台集群网络

大数据 Hadoop 平台集群采用 Master/Slave 架构。集群由单独的 Master 节点和多个 Slave 节点服务器组成，实验环境下一般为 3 个节点，分别是 master、slave1 和 slave2，如图 3-1 所示。

master 节点主要承载平台中 HDFS 的 NameNode 节点，负责管理文件系统的名字空间（namespace）及客户端对文件的访问，如打开、关闭、重命名文件或目录。它也负责确定数据块到具体 DataNode 节点的映射。slave 节点主要承载平台中 HDFS 的 DataNode 节点，负责管理在此节点上的存储数据和负责处理文件系统客户端的读写请求。

实验环境下 Hadoop 集群网络需考虑地址规划和连通性。由于实验环境下数据负载较小、可靠性要求不高，链路一般采用单链路连接。IP 地址规划在同一网络中，一般设定地址为 192.168.1.0/24 网段。具体 IP 地址在 CentOS 7 中配置。集群网络配置中会详细介绍 IP 地址配置。

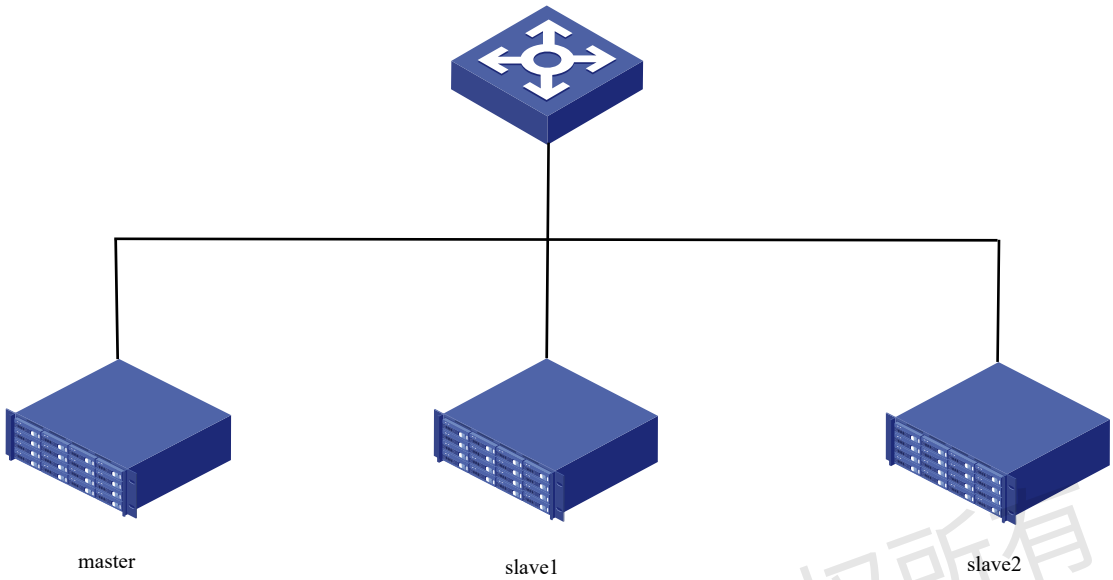

图 3-1　实验环境下的大数据平台网络拓扑结构

3.1.2　生产环境下的大数据 Hadoop 平台集群网络

大数据 Hadoop 平台集群在生产环境下，由于承载生产中大量数据的存储计算，考虑整体容错性，会采用多 master 架构。本实例采用两台 master 服务器和多台 slave 服务器，如图 3-2 所示。

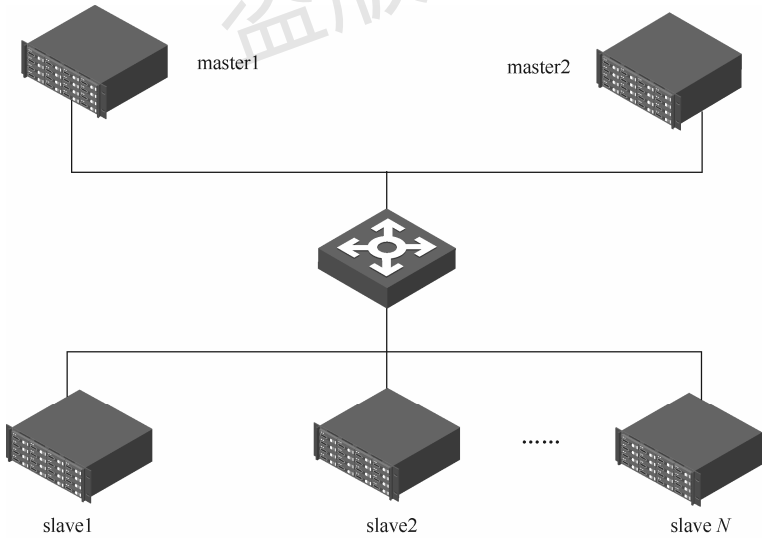

图 3-2　生产环境下的大数据平台网络拓扑结构

电子工业出版社版权所有 盗版必究

生产环境下大数据 Hadoop 平台集群网络需考虑其可用性和弹性。一个高冗余性和可扩展的网络，不但可以提高集群可用时间，还能满足 Hadoop 集群的增长。部署网络需要每个节点双链路连接，防止单链路故障导致集群网络中断。相应地，交换机段需要配置链路绑定，防止产生环路。Hadoop 平台在使用中需检测网络运行状态，防止出现网络延迟和数据过载。

3.2　SSH 无密码登录

3.2.1　SSH 简介

Secure Shell（SSH）是由 IETF（The Internet Engineering Task Force，国际互联网工程任务组）制定的建立在应用层基础上的安全网络协议。它是专为远程登录会话（甚至可以用 Windows 远程登录 Linux 服务器进行文件互传）和其他网络服务提供安全性的协议，可有效弥补网络中的漏洞。通过 SSH，可以把所有传输的数据进行加密，也能够防止 DNS 欺骗和 IP 欺骗。还有一个额外优势就是传输的数据是经过压缩的，所以可以加快传输速度。目前 SSH 已经成为 Linux 操作系统的标准配置。

SSH 只是一种协议，存在多种实现，既有商业实现，也有开源实现。本书主要介绍 OpenSSH 免费开源实现。如果要在 Windows 操作系统中使用 SSH，则需要使用另外的 SSH 客户端软件，如 Putty、SecureCRT、XShell 等。

3.2.2　SSH 特点介绍

（1）SSH 是传输层和应用层上的安全协议，它只能通过加密连接双方会话的方式来保证连接的安全性。当使用 SSH 连接成功后，将建立客户端和服务端之间的会话，该会话在被加密之后进行会话传输。

（2）SSH 服务的守护进程为 sshd，默认监听在 22 端口上。

（3）所有 SSH 客户端工具，包括 ssh、scp、sftp、ssh-copy-id 等命令都是借助于 ssh 连接来完成任务的。也就是说，它们都连接服务端的 22 端口，只不过连接之后将待执行的相关命令转换传送到远程主机，由远程主机执行。

（4）SSH 客户端命令（ssh、scp、sftp 等）读取两个配置文件：全局配置文件/etc/ssh/ssh_config 和用户配置文件~/.ssh/config。实际命令行上也可以传递配置选项。它们生效的优先级是：“命令行配置选项”优先于“用户配置文件~/.ssh/config”，“用户配置文件~/.ssh/config”优先于“全局配置文件/etc/ssh/ssh_config”。

（5）SSH 涉及两个验证：主机验证和用户身份验证。通过主机验证，再通过该主机上的用户验证，就能唯一确定该用户的身份。一个主机上可以有很多用户，所以每台主机的验证只需一次，但主机上每个用户都需要单独进行用户验证。

（6）SSH 支持多种身份验证，最常用的是密码验证机制和公钥认证机制，其中公钥认证机制在某些场景实现双机互信时几乎是必需的。

（7）SSH 客户端具有多个强大的功能，如端口转发（隧道模式）、代理认证、连接共享（连接复用）等。

（8）SSH 服务端配置文件为/etc/ssh/sshd_config，客户端的全局配置文件为/etc/ssh/ssh_config。

电子工业出版社版权所有 盗版必究

3.2.3 非对称加密机制

对称加密：加密和解密使用一样的算法，只要解密时提供与加密时一致的密码就可以完成解密。例如登录密码、银行卡密码，只要保证密码正确就可以。

非对称加密：通过公钥（public key）和私钥（private key）来加密、解密。公钥加密的内容可以使用私钥解密，私钥加密的内容可以使用公钥解密。一般使用公钥加密，私钥解密，但并非绝对如此。在接下来介绍的 SSH 服务中，建议使用分发公钥的方法，但也可以分发私钥。

举例说明：如果 A 生成了（私钥 A，公钥 A），B 生成了（私钥 B，公钥 B），那么 A 和 B 之间的非对称加密会话如下，如图 3-3 所示。

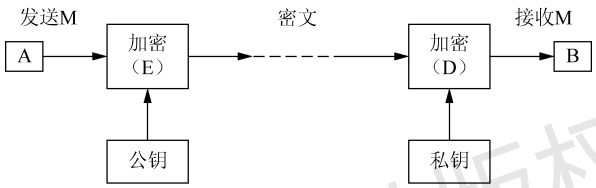

图 3-3　非对称加密、解密示意图

（1）A 将自己的公钥 A 分发给 B，B 拿着公钥 A 将数据进行加密，并将加密的数据发送给 A，A 将使用自己的私钥 A 解密数据。

（2）A 将自己的公钥 A 分发给 B，并使用自己的私钥 A 加密数据，然后 B 使用公钥 A 解密数据。

（3）B 将自己的公钥 B 分发给 A，A 拿着公钥 B 将数据进行加密，并将加密的数据发送给 B，B 使用自己的私钥 B 解密数据。

（4）B 将自己的公钥 B 分发给 A，并使用自己的私钥 B 加密数据，然后 A 使用公钥 B 解密数据。

虽然理论上非对称加密支持以上 4 种场景，但在 SSH 的身份验证阶段，SSH 只支持服务器端保留公钥，客户端保留私钥的方式，所以 SSH 的实现方式只有两种：客户端生成密钥对，将公钥分发给服务器端；服务器端生成密钥对，将私钥分发给客户端。只不过出于安全性和便利性，一般是客户端生成密钥对，并分发公钥。

3.2.4 SSH 安全机制

SSH 之所以能够保证安全，原因在于它采用了非对称加密技术（RSA）。传统的网络服务程序，如 FTP、Pop 和 Telnet 本质上都是不安全的；因为它们在网络上用明文传送数据、用户账号和用户口令，很容易受到中间人（man-in-the-middle）攻击方式的攻击。

但并不是说 SSH 就是绝对安全的，因为它本身提供两种级别的验证方法。

第一种级别（基于口令的安全验证）：只要知道自己的账号和口令，就可以登录到远程主机。所有传输的数据都会被加密，但是不能保证正在连接的服务器就是用户想连接的服务器，可能会有别的服务器在冒充真正的服务器，也就是受到中间人攻击。

第二种级别（基于密钥的安全验证）：用户必须为自己创建一对密钥，并把公钥放在需要访问的服务器上。如果要连接到 SSH 服务器，客户端软件就会向服务器发出请求，请求

电子工业出版社版权所有 侵权必究

使用用户的密钥进行安全验证。服务器收到请求之后，先在该服务器用户的主目录下寻找公钥，然后把它和客户端发送过来的公钥进行比较。如果两个密钥一致，服务器就用公钥加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用私钥在本地解密，再把它发送给服务器完成登录。与第一种级别相比，第二种级别不仅加密所有传输的数据，而且不需要在网络上传送口令，因此安全性更高，可以有效防止中间人攻击。

3.2.5　SSH 基于口令的安全验证

第一次登录对方主机，系统会出现下面的提示：

```
$ ssh user@host
The authenticity of host 'host (192.168.1.6)' can't be established.
RSA key fingerprint is 98:2e:d7:e0:de:9f:ac:67:28:c2:42:2d:37:16:58:4d.
Are you sure you want to continue connecting (yes/no)?
```

该提示信息表示系统无法确认 host 主机的真实性，只知道它的公钥指纹，询问用户是否还想继续连接。所谓公钥指纹，是指公钥长度较长，很难比对，所以对其进行 MD5 计算，将它变成数字指纹，如 98:2e:d7:e0:de:9f:ac:67:28:c2:42:2d:37:16:58:4d。远程主机必须在自己的网站上贴出公钥指纹，以便用户自行核对。

假定经过风险衡量以后，用户决定接收这个远程主机的公钥。

```
Are you sure you want to continue connecting (yes/no)? yes
```

系统会出现一句提示，表示 host 主机已经得到认可。

```
Warning: Permanently added 'host,192.168.1.6' (RSA) to the list of known
hosts
```

然后，系统会要求输入密码。

```
Password: (enter password)
```

如果密码正确，用户即可登录。

当远程主机的公钥被接收以后，它就会被保存在文件$HOME/.ssh/known_hosts 中。下次再连接这台主机时，系统就会认出它的公钥已经保存在本地，从而跳过警告部分，直接提示输入密码。

每个 SSH 用户都有自己的 known_hosts 文件，此外，系统也有一个这样的文件，通常是/etc/ssh/ssh_known_hosts，保存一些对所有用户都可信赖的远程主机的公钥。

3.2.6　基于密钥的安全验证

使用密码登录，每次都必须输入密码，操作过程比较麻烦。SSH 还提供公钥登录的方式，可以省去输入密码的步骤。

公钥登录就是用户将自己的公钥储存在远程主机上。登录时，远程主机会向用户发送一段随机字符串，用户用自己的私钥加密后再返回。远程主机用事先储存的公钥进行解密，如果成功，就证明用户是可信的，直接允许登录 shell，不再要求密码。

这种方法要求用户必须提供自己的公钥。如果没有公钥，可以直接用 ssh-keygen 命令生成：

```
$ ssh-keygen
```

运行上面的命令以后，系统会出现一系列提示问题，可以一直按 Enter 键确认，使用

电子工业出版社版权所有 盗版必究

默认的配置。其中有一个提示问题是，要不要对私钥设置口令（passphrase），如果担心私钥的安全，这里可以设置口令。

运行结束以后，在$HOME/.ssh/目录下会生成两个新文件：id_rsa.pub 和 id_rsa。前者是当前用户的公钥，后者是私钥。

这时再输入下面的命令，将公钥传送到远程主机 host 上面：

```
$ ssh-copy-id user@host
```

以后再登录远程主机，就不需要输入密码，可以直接登录。

3.3 集群网络配置

3.3.1 实验环境下的集群网络配置

根据实验环境下集群网络 IP 地址规划，为主机 master 设置的 IP 地址是 192.168.1.6，掩码是 255.255.255.0；slave1 设置的 IP 地址是 192.168.1.7，掩码是 255.255.255.0；slave2 设置的 IP 地址是 192.168.1.8，掩码是 255.255.255.0。

根据我们为 Hadoop 设置的主机名为“master、slave1、slave2”，映射地址是“192.168.1.6、192.168.1.7、192.168.1.8”，分别修改主机配置文件“/etc/hosts”，在命令终端输入如下命令。

```
#vi /etc/hosts
```

“/etc/hosts”文件给出主机到 IP 的映射关系，修改“/etc/hosts”文件，在文件末尾添加以下配置。

```
192.168.1.6 master
192.168.1.7 slave1
192.168.1.8 slave2
```

配置完毕，执行“reboot”命令，重新启动系统。

3.3.2 生产环境下的集群网络配置

生产环境下集群网络采用双网卡绑定链路连接，所有链路处于负载均衡状态，这种模式的特点增加了带宽，同时支持容错能力。

nmcli 命令是 CentOS 7 下的常用网络管理工具，通过 nmcli 命令可以完成网络的查看、配置和管理工作。以下是集群双网络绑定链路配置步骤。

1．查看主机信息

命令如下。

```
[root@localhost ~]# nmcli dev
DEVICE    TYPE      STATE        CONNECTION
em33      ethernet  connected      --
em37      ethernet  connected      --
em39      ethernet  unavailable    --
```

nmcli dev 命令用来查询网卡设备信息。从上面的命令结果可以看到当前主机配置了 em33、em37、em39 三块网卡，其中 em33 和 em37 两块网卡已连接，em39 未连接。

2．查看网络连接信息

命令如下。

电子工业出版社版权所有 盗版必究

```
[root@localhost ~]# nmcli con sh
NAME        UUID                                  TYPE        DEVICE
em33  4c8a8aee-327b-4f29-9ec4-ea2b2551420b  802-3-ethernet    --
em37  62bde3ef-9f40-42aa-9b3a-4fe8b1aabf42  802-3-ethernet    --
em39  56ef9d05-3a5d-4cd6-af5b-3d39adce619f  802-3-ethernet    --
```

nmcli con sh 命令是 nmcli connnection show 命令的简写版，两个命令等价，用来查询系统当前已配置的连接。在 CentOS 7 系统中，connection 连接是逻辑上的概念，device 设备是网卡。连接上可以配置 IP 地址等信息，而且连接要与网卡关联后才能生效。以上查询结果中 DEVICE 一列全部为“--”，表示 3 个连接都没有与网卡关联。

3．创建绑定虚拟网卡

命令如下。

```
[root@localhost ~]# nmcli con add type team con-name team0 ifname team0 config
'{"runner":{"name": "roundrobin"}}'
[root@localhost ~]# nmcli con sh
NAME              UUID                                  TYPE          DEVICE
team0    5a1976d7-2c91-456e-8dda-0b12c6ef0c04       team          team0
em33     4c8a8aee-327b-4f29-9ec4-ea2b2551420b    802-3-ethernet    --
em37     62bde3ef-9f40-42aa-9b3a-4fe8b1aabf42    802-3-ethernet    --
em39     56ef9d05-3a5d-4cd6-af5b-3d39adce619f    802-3-ethernet    --
```

nmcli con add 命令用来添加名称为 team0 的绑定连接及 team0 的配置文件。

4．配置 IP 及网关

命令如下。

```
[root@localhost ~]# nmcli con modify team0 ipv4.address '10.128.105.157/24'
ipv4.gateway '10.128.105.254'
[root@localhost ~]# nmcli con modify team0 ipv4.method manual
```

nmcli con modify 命令用来修改连接信息，以上命令将 team0 连接的 IP 地址设定为 10.128.105.157，子网掩码为 24 位，网关 IP 地址为 10.128.105.254。

5．启动网卡服务

命令如下。

```
[root@localhost ~]# ifup team0
[root@localhost ~]# /etc/init.d/network restart
Restarting network (via systemctl):                        [  OK  ]
[root@localhost ~]# ip a |grep team0
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master team0
state UP qlen 1000
10: team0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP qlen 1000 inet 10.128.105.157/24 brd 10.128.105.255 scope global team0
```

ifup 命令用来启动连接 team0，/etc/init.d/network restart 命令用来重启主机网络服务，ip a |grep team0 命令用来查询 team0 连接的 IP 地址等相关信息。

6．添加端口到网卡

命令如下。

```
[root@localhost ~]# nmcli con add type team-slave con-name team0-port1 ifname
em33 master team0
```

电子工业出版社版权所有 盗版必究

```
   Connection 'team0-port1' (e01c37f5-b3a1-4e6d-96ab-a6e676ad3c32) successfully
added.
   [root@localhost ~]# nmcli con add type team-slave con-name team0-port2 ifname
em37 master team0
   Connection 'team0-port2' (8b13a9de-d64c-44fc-a86d-ea45d2f511cf) successfully
added.
   [root@localhost ~]# nmcli con sh
   NAME              UUID                                  TYPE           DEVICE
   team0        64b123d5-189d-43b0-a850-77d47c0da86c  team           team0
   team0-port1  24181898-bffe-4ae1-90e8-f47da64b5512  802-3-ethernet  em33
   team0-port2  8b13a9de-d64c-44fc-a86d-ea45d2f511cf  802-3-ethernet  em37
   em33          4c8a8aee-327b-4f29-9ec4-ea2b2551420b  802-3-ethernet  --
   em37          62bde3ef-9f40-42aa-9b3a-4fe8b1aabf42  802-3-ethernet  --
   em39          761f7809-d10c-4e3b-bc06-c0906ca55d1c  802-3-ethernet  --
```

添加 team0-port1 和 team0-port2 两个端口，分别绑定 em33 和 em37 两个网卡。

7．查看状态模式

命令如下。

```
   [root@localhost ~]# teamdctl team0 st
   setup:
     runner: roundrobin
   ports:
     em33
       link watches:
         link summary: up
         instance[link_watch_0]:
           name: ethtool
           link: up
           down count: 0
     em37
       link watches:
         link summary: up
         instance[link_watch_0]:
           name: ethtool
           link: up
           down count: 0
```

8．测试连通性

将 em33 网卡禁用，命令如下。

```
   [root@localhost etc]# ifdown em33
   Device 'em33' successfully disconnected.
```

用 ping 命令测试主机连通性，命令如下。

```
   [root@localhost etc]ping  192.168.1.6
   64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
   64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
   64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
   64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
```

电子工业出版社版权所有 盗版必究

```
64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
64 bytes from 192.168.1.6 : icmp_seq=57 ttl=59 time=0.519 ms
```

查看网络连接状态，命令如下。

```
[root@localhost etc]# nmcli con sh
NAME        UUID                                  TYPE           DEVICE
team0       64b123d5-189d-43b0-a850-77d47c0da86c  team           team0
team0-port2 8b13a9de-d64c-44fc-a86d-ea45d2f511cf  802-3-ethernet em33
```

3.4　SSH 无密码验证配置

3.4.1　生成 SSH 密钥

Hadoop 运行过程中需要管理远端 Hadoop 守护进程，在 Hadoop 启动以后，NameNode 是通过 SSH（Secure Shell）来启动和停止各个 DataNode 上的各种守护进程的。这就要求在节点之间执行指令的时候是不需要输入密码的形式，故需要配置 SSH 运用无密码公钥认证的形式，这样 NameNode 使用 SSH 无密码登录并启动 DataName 进程。同样的原理，DataNode 上也能使用 SSH 无密码登录到 NameNode。

1．安装和启动 SSH 协议

实现 SSH 登录需要 openssh 和 rsync 两个服务，一般情况下默认已经安装，可以通过下面的命令查看结果。

```
[root@master ~]# rpm -qa | grep openssh
openssh-server-7.4p1-11.el7.x86_64
openssh-7.4p1-11.el7.x86_64
openssh-clients-7.4p1-11.el7.x86_64
```

```
[root@master ~]# rpm -qa | grep rsync
rsync-3.1.2-6.el7_6.1.x86_64
```

如果没有安装 openssh 和 rsync，可以通过下面的命令进行安装。

```
[root@master ~]# yum install openssh* -y
[root@master ~]# yum install rsync -y
```

2．切换到 hadoop 用户

```
[root@master ~]# su -hadoop
```

3．每个节点生成秘钥对

```
[hadoop@master ~]$ ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:UxFuDHHLiMNQsZ11o3F7NApvDPfKMBqaxE+rlccKdk4 hadoop@master
The key's randomart image is:
+---[RSA 2048]----+
|    ..o.o.O.= o  |
```

电子工业出版社版权所有 盗版必究

```
|     + + X & * . |
|      B * @ * o  |
|     . * O = o   |
|      = E o o    |
|     . B +       |
|      . o        |
|                 |
|                 |
+----[SHA256]-----+
```

4．查看“.ssh”文件夹及无密码密钥对

查看“/home/hadoop/”下是否有“.ssh”文件夹，且“.ssh”文件下是否有两个刚生成的无密码密钥对。

```
[hadoop@master .ssh]$ cd ~/.ssh/
[hadoop@master .ssh]$ ls
id_rsa  id_rsa.pub
```

5．将 id_rsa.pub 追加到授权 key 文件中

```
[hadoop@master .ssh]$ cat  ~/.ssh/id_rsa.pub >>  ~/.ssh/authorized_keys
[hadoop@master .ssh]$ ls ~/.ssh/
authorized_keys  id_rsa  id_rsa.pub
```

6．修改文件“authorized_keys”权限

通过命令 ll 查看，可以看到修改后 authorized_keys 文件的权限为“-rw-------”，表示所有者可读写，其他用户没有访问权限。如果该文件权限太大，SSH 服务会拒绝工作，出现无法通过密钥文件进行登录认证的情况。

```
[hadoop@master .ssh]$ chmod  600  ~/.ssh/authorized_keys
[hadoop@master .ssh]$ ll ~/.ssh/
总用量 12
-rw-------. 1 hadoop hadoop  395 4月   9 15:34 authorized_keys
-rw-------. 1 hadoop hadoop 1679 4月   9 15:26 id_rsa
-rw-r--r--. 1 hadoop hadoop  395 4月   9 15:26 id_rsa.pub
```

7．配置 SSH 服务

使用 root 用户登录，修改 SSH 配置文件“/etc/ssh/sshd_config”的下列内容，需要将该配置字段前面的#号删除，启用公钥私钥配对认证方式。

```
[root@master ~]# vi /etc/ssh/sshd_config
```

```
PubkeyAuthentication yes
```

8．重启 SSH 服务

设置完后需要重启 SSH 服务，这样才能使配置生效。

```
[root@master ~]# systemctl restart sshd
```

9．切换到 hadoop 用户

```
[root@master ~]# su -hadoop
上一次登录：四 4月  9 15:25:35 CST 2020pts/0 上
```

电子工业出版社版权所有 盗版必究

10．验证 SSH 登录本机

在 hadoop 用户下验证能否嵌套登录本机。若可以不输入密码登录，则说明本机通过密钥登录认证成功。

```
[hadoop@master ~]$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:mCBXMeGA6BsP/ aYJH3Ie5723JAWRSOzBr7FReICWLtQ.

ECDSA key fingerprint is MD5:b2:88:99:ee:00:30:24:61:75:7e:7f:8a:f5:
d0:98:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Last login: Thu Apr  9 16:02:14 2020
```

首次登录主机时会提示系统无法确认 host 主机的真实性，只知道它的公钥指纹，询问用户是否还想继续连接。这时需要输入“yes”，表示希望继续连接。第二次再登录同一个主机时，则不会再出现该提示，可以直接进行登录。

读者需要关注在登录过程中是否需要输入密码，不需要输入密码才表示通过密钥登录认证成功。

3.4.2　交换 SSH 密钥

在 master 和 slave1、slave2 节点之间交换密钥，实现 master 和两个 slave 节点之间能够 SSH 无密码登录。

1．将 master 节点的公钥 id_rsa.pub 复制到每个 slave 节点

hadoop 用户登录，通过 scp 命令实现密钥复制。

```
[hadoop@master ~]$ scp ~/.ssh/id_rsa.pub hadoop@slave1:~/
The authenticity of host 'slave1 (192.168.10.12)' can't be established.
ECDSA key fingerprint is SHA256:VQHM02+x2s0xCE7Qw+EbjdPfgfZV3W7B+gDiozC80c4.
ECDSA key fingerprint is MD5:3e:a1:47:e9:fe:1b:55:7e:cf:a9:90:58:9b:a2:
0d:26.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'slave1,192.168.10.12' (ECDSA) to the list of
known hosts.
hadoop@slave1's password:
id_rsa.pub
100%  395   259.2KB/s   00:00
```

首次远程连接时系统会询问用户是否要继续连接。这时需要输入“yes”，表示希望继续连接。因为目前尚未完成密钥认证的配置，所以使用 scp 命令复制文件需要输入 slave1 节点 hadoop 用户的密码。

2．在每个 slave 节点把 master 节点复制的公钥复制到 authorized_keys 文件

hadoop 用户登录 slave1 和 slave2 节点，执行命令如下。

```
[hadoop@slave1 ~]$ cat ~/id_rsa.pub >>~/.ssh/authorized_keys
```

3．在每个 slave 节点删除 id_rsa.pub 文件

```
[hadoop@slave1 ~]$ rm -f ~/id_rsa.pub
```

电子工业出版社版权所有 盗版必究

4．将每个 slave 节点的公钥保存到 master

以下步骤（1）～（3）为一组，在 slave1 节点上执行完成一组命令后，再在 slave2 节点执行命令。

（1）将 slave 节点的公钥复制到 master。

```
[hadoop@slave1 ~]$ scp ~/.ssh/id_rsa.pub hadoop@master:~/
The authenticity of host 'master (192.168.10.11)' can't be established.
ECDSA key fingerprint is SHA256:mCBXMeGA6BsP/aYJH3Ie5723JAWRSOzBr7FReICWLtQ.
ECDSA key fingerprint is MD5:b2:88:99:ee:00:30:24:61:75:7e:7f:8a:f5:d0:
98:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master,192.168.10.11' (ECDSA) to the list of
known hosts.
hadoop@master's password:
id_rsa.pub
100%  395    51.5KB/s   00:00
```

（2）在 master 节点把从 slave 节点复制的公钥复制到 authorized_keys 文件。

```
[hadoop@master ~]$ cat ~/id_rsa.pub >>~/.ssh/authorized_keys
```

（3）在 master 节点删除 id_rsa.pub 文件。

```
[hadoop@master ~]$ rm -f ~/id_rsa.pub
```

3.4.3 验证 SSH 无密码登录

1．查看 master 节点 authorized_keys 文件

```
[hadoop@master .ssh]$ cat ~/.ssh/authorized_keys
```

可以看到 master 节点 authorized_keys 文件中包括 master、slave1、slave2 共 3 个节点的公钥。

2．查看 slave 节点 authorized_keys 文件

```
[hadoop@slave1.ssh]$ cat ~/.ssh/authorized_keys
```

可以看到 slave 节点 authorized_keys 文件中包括 master 和当前 slave 两个节点的公钥。

3．验证 master 到每个 slave 节点无密码登录

hadoop 用户登录 master 节点，执行 ssh 命令登录 slave1 和 slave2 节点，可以观察到不需要输入密码即可实现 SSH 登录。

```
[hadoop@master ~]$ ssh slave1
Last login: Thu Apr  9 19:24:36 2020
[hadoop@slave1 ~]$
```

```
[hadoop@master ~]$ ssh slave2
Last login: Thu Apr  9 19:32:09 2020
[hadoop@slave2 ~]$
```

4．验证两个 slave 节点到 master 节点无密码登录

```
[hadoop@slave1 ~]$ ssh master
Last login: Thu Apr  9 19:19:07 2020
[hadoop@master ~]$
```

电子工业出版社版权所有 盗版必究

JDK 的安装参考“2.3.2 安装 Java 环境”。安装成功 JDK 后，即可开始安装 Hadoop 分布式环境。

3.5 本章小结

本章主要介绍了 Hadoop 平台搭建的相关集群网络配置、SSH 无密钥登录、JDK 等相关配置，以及相关的动手实操实验详细配置内容。

电子工业出版社版权所有
盗版必究

电子工业出版社版权所有
盗版必究