命令 --help ：查询终端命令的说明信息 （-后表示 选项，--固定选项 可以显示简要的帮助信息 ）

man 命令 ：可以查询 终端命令的 详细使用手册

联网配置 ： https://blog.csdn.net/javaee\_sunny/article/details/80383937

1. linux系统自带openJDK 需要卸载掉 ，装上需要的使用sun公司的 jdk
   1. rpm -qa | grep java 查找那些有哪些跟java有关的包
   2. Rpm -e --nodeps 名称 卸载相应的套件 （注：-e 卸载 ; rpm卸载需要依赖， --nodeps忽略依赖 ）
   3. 解压 tar 包 tar zxvf 名称
   4. 配置 jdk 的环境变量 （配置环境变量地址 ： etc（系统配置文件）/profile）

export JAVA\_HOME=/home/mayadong/JDK/jdk1.8.0\_181

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

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

* 1. Source 路径/文件 重新加载配置文件

2、sudo 用于临时获取 超级用户权限

3、Ifdonfig eth0不显示 IP地址

1. 切换到root
2. Dhclient eth0
3. Ifconfig eth0

4、查看所有进程

ps -ef

5、查看 端口号和进程号

（1）ps aux | grep kafka

（2）ps -ef | grep kafka

1. netstat -anp 查看开放的端口有哪些
2. service iptables stop 关闭防火墙
3. vi /etc/sysconfig/iptables 开放防火墙端口配置文件

注意：在配置开放端口时 ，顺序也是固定的，写在 22端口下 ，不然不起作用

1. tail -f 用户显示文件的 后面的详情 ，-f 实时监听 文件数据的变化

7、当启动mysql 服务时 报([失败]mysql/mysql.pid). 存在的情况

（1）、mysql/data/数据目录mysql用户没有权限

解决 ：chown -R mysql.mysql /opt/mysql/data

（2）mysql 进程可能已经存在 ps -ef | grep mysql

解决：kill 相关进程

（3）etc/my.cnf 配置文件 没有指定 datadir =

或 指定文件过大 应换成 M 级别的

（4）、错误日志目录不存在

    解决方法：使用“chown” “chmod”命令赋予mysql所有者及权限

（5）selinux惹的祸，如果是centos系统，默认会开启selinux

|  |  |
| --- | --- |
| 1  2 | 解决方法：先临时改为警告模式:[root@www php]# setenforce 0  然后打开/etc/sysconfig/selinux，把SELINUX=enforcing改为SELINUX=disabled |

 （ 6）、启动失败注意检查数据目录的权限，要与mysql程序运行用户权限一致。

|  |  |
| --- | --- |
| 1 | [root@db01 mysql]# chown  -R mysql.mysql ../mysql |

 　　最好是整个mysql程序目录都授权予mysql用户。

1. 、服务器 导出 csv 文件

select id , tel,email from hr\_users limit 3 into outfile '/tmp/junjun.csv'character set gbk   
fields terminated by ',' optionally enclosed by '"'   
lines terminated by '\n';

1. 查看一个机器上 是否存在一个 命令

whereis 命令

1. 如何安装包

yum install 包名

1. 赋权限 ： chmod 744 文件 :shell 脚本执行权限

Chmod 755 文件 :shell脚本定时任务执行权限

1. shell定时复制数据主机之间如何建立信任关系
2. 加密算法两种方式 ：对称秘钥加密 、非对称秘钥加密（ssh采用的方法）
   * 1. 对称秘钥加密 ：加密、解密都是使用 同一个钥匙（密码）
        1. 优点：速度快
        2. 缺点：存在不安全因素 ，给别人使用密码时，可能会泄露
     2. 非对称秘钥加密 ：使用时生成两个密钥 ， 公钥放在公共的服务器 ，私钥放在个人服务器 ，在认证时 私钥和公钥结合使用才能认证成功 公钥是可以存在多个的
        1. 两种使用场合 ：
           1. 公钥将文件加密，解密时对应的私钥才能解密 （加密文件）
           2. 私钥将文件加密，解密时对应的公钥才能解密 （发送文件数字签名）

发文件时 +额外数据（私钥加密），收文件之前公钥认证解密 ，这样就不需要手动输入密码了

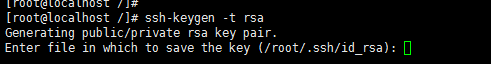
* + - 1. 安全性好
      2. 速度慢

1. 公钥放一台主机 ，私钥放一台主机 每次拷贝文不需要手动认证输入密码

A172.30.3.165 节点从B172.30.3.126节点拉文件 ：

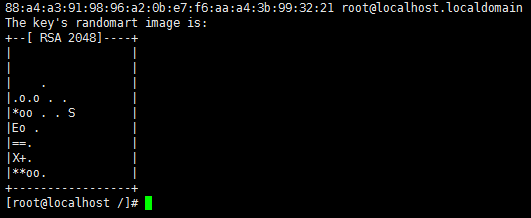
scp [root@172.30.3.126](mailto:root@172.30.3.126):/tmp/文件名字 /tmp/文件名字

* + 1. 创建秘钥对 ssh -keygen -t rsa



提示 ：密钥的保存路径，默认是/root/.ssh/id\_rsa 直接回车

以后的步骤也依次回车



1. 在/root/.ssh/ 生成如下文件



1. .将id\_rsa.pub 公钥拷贝到主机 B /root/ 目录下

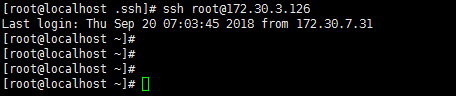
scp id\_rsa.pub [root@172.30.3.126:/root/](mailto:root@172.30.3.126:/root/)

1. 在主机B ，将id\_rsa.pub 生成认证秘钥
   * 1. 如果 /root下无 .ssh文件夹 则创建 mkdir .ssh(创建后文件夹是隐藏的)
     2. 将id\_rsa.pub输出到.ssh目录下 ，创建新的认证文件authorized\_keys

cat id\_rsa.pub >> .ssh/authorized\_keys

* + 1. 到/root/下分别 对文件赋予权限
       1. chmod 700 .ssh
       2. chmod 600 .ssh/authorized\_keys

1. 测试 ：ssh [root@172.30.3.126](mailto:root@172.30.3.126) 远程登录就不需要密码了



1. 、如果只有私钥 ，然后远程节点 配置好了公钥
2. 将私钥 放置 /root/.ssh 目录
3. 开放 root目录、.ssh目录 和 私钥的权限

chmod go-w ~/

chmod 700 ~/.ssh

chmod 600 ~/.ssh/文件名

命令解释

1. **rpm**：**rpm命令**是RPM软件包的管理工具，用来管理Linux各项套件的程序
2. Sudo ：在命令前加上 sudo 可以让普通用户具备root权限
3. SSH 包含3个组件
   1. ssh 远程登录节点 ssh 用户名@IP
      1. 不允许空密码或错误密码认证登录
      2. 不允许root用户登录
      3. 有两个版本 ssh，ssh2安全性更高
   2. sftp 文件共享连接 ， xftp连接就是 sftp实现的
   3. scp 文件拷贝共享
4. scp 两个 节点上的文件的 传输
   * 1. 向目标节点 传文件 ： scp 文件名 root@ ip ：/目录
     2. 向目标节点 拉文件 ： scp root@ip:/文件地址 文件目录

注意 ：在远程拷贝至目标节点时 ，就等于创建了 一个一模一样的文件 ，此时原文件的 文件属性已发生了变化（如文件创建的时间） 如果不期望文件属性发生变化

scp -p -r

-p ： 文件属性不发生变化 （建议加上）

-r ： 如果拷贝的是目录 需要加-r ，递归

-P(大写) ： 指定端口号，如果端口号在ssh 中发生了变化

1. crontab (cron tab): 用于提交或管理当前用户周期性的任务，系统安装完成默 认会安装并开启此服务，进程每分钟检查是否有定时任务如果有则执行  
    /etc/crontab：用户执行系统任务 如：清理日志、缓存、写数据的内存等全局的

at ：用于执行 一次性的定时任务 如 10 分钟之后关机

crontab 使用 ：

* 1. 查询服务状态 ：service crond status
  2. 开启服务 ：service crond start
  3. 重启服务 ：service crond restart
  4. 重新载入配置 ：service crond relaod

1. 编辑并添加 任务 ： crontab -e \*\*\*\*\* connamd

注意 ：不指定用户 如:\*\*\*\*\* root connamd

1. 第一个\* ：分钟 0-59 可以用\* 和 \*/1 表示
   1. \* \* \* \* \* = \*\1 \* \* \* \* ：每分钟执行
2. 第二个\* ：小时 0-23
3. 第三个\* ：天 1-31
4. 第四个\* ：月份 1-12
5. 第五个\* ：星期 0-6 0表示周日
6. 添加任务
7. 查询任务 ： crontab -l 查询当前用户的所有定时任务
8. 删除任务 ： crontab -r 删除当前用户的 所有定时任务
9. (> >>) 文件输出
   1. > 在源文件上覆盖
   2. >> 追加内容
   3. 0 > 标准输入
   4. 1> 标准输出 (1>>)
   5. 2 > 错误输出 (2>>)

./lftp\_to\_ftp\_server.sh var.sh >> ftp.log 2>&1 &

意思： shell的日志输出到ftp.log , 错误输出也会重定向到 1 标准输出 打印到log中

最后一个 & ，表示在后台执行的意思 但是退出账号会终止

配合 nohup 表示在后台执行 且退出账后不会终止shell的执行