* **周一所需：新建一台虚拟机，安装操作系统RHEL8.2**

要求：**必须带GUI图形环境**

**虚拟机内存2G，硬盘大小20G**

**网络类型private1(vmnet1)**

构建Yum仓库(提示8系统光盘提供两个仓库)

操作系统镜像：**rhel-8.2-x86\_64-dvd.iso**

**关闭系统进行拍摄快照**

今日所需tools.tar.gz与users.sql

####################################################

* **关闭所有虚拟机的SELinux**

[root@svr7 ~]# setenforce 0 #修改当前运行模式

[root@svr7 ~]# getenforce #查看当前运行模式

Permissive

[root@svr7 ~]# vim /etc/selinux/config #永久修改

SELINUX=permissive

* **设置所有虚拟机防火墙**

[root@svr7 ~]# systemctl stop firewalld

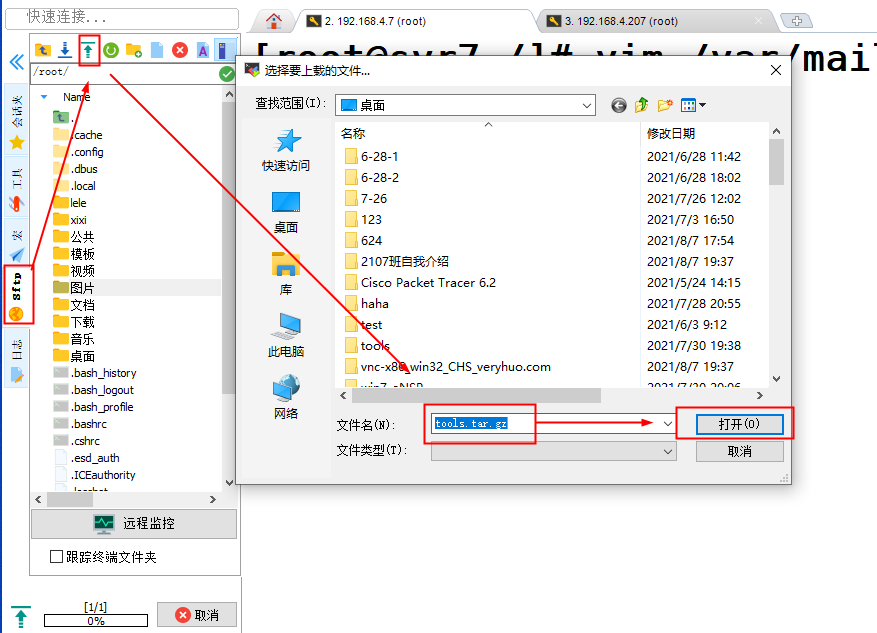
[root@svr7 ~]# systemctl disable firewalld

**一、源码编译安装**

**真机将tools.tar.gz上传到虚拟机A**

]# scp /linux-soft/1/tools.tar.gz root@192.168.4.7:/root

**真机为Windows：**



**虚拟机A**

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C /

[root@svr7 ~]# ls /tools/

inotify-tools-3.13.tar.gz other

[root@svr7 ~]#

RPM软件包：yum -y install rpm -ivh

源码包---编译(gcc与make)---》可以执行二进制的程序--》运行安装

•主要优点

–获得软件的最新版，及时修复bug

–软件功能可按需选择/定制，有更多软件可供选择

–源码包适用各种平台

–……

步骤1：安装开发工具gcc与make，释放源代码至指定目录

步骤2：tar解包，释放源代码至指定目录

步骤3：./configure 配置，指定安装目录/功能模块等选项

步骤4：make 编译，生成可执行的二进制程序文件

步骤5：make install 安装，将编译好的文件复制到安装目录

**1.安装开发工具**

[root@svr7 ~]# **yum -y install gcc make**

[root@svr7 ~]# rpm -q gcc

gcc-4.8.5-28.el7.x86\_64

[root@svr7 ~]# rpm -q make

make-3.82-23.el7.x86\_64

[root@svr7 ~]#

**2.进行解压缩**

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C /

[root@svr7 ~]# ls /tools/

inotify-tools-3.13.tar.gz other

**3.进行tar解包**

[root@svr7 ~]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /usr/local/

[root@svr7 ~]# ls /usr/local/

[root@svr7 ~]# cd /usr/local/inotify-tools-3.13/

[root@svr7 inotify-tools-3.13]# ls

**4.运行configure脚本**

作用1：检测当前系统是否安装gcc

作用2：指定安装位置与功能

]# **cd /usr/local/inotify-tools-3.13/**

]# ./configure --help

]# **./configure --prefix=/opt/myrpm** #指定安装位置，此步骤不产生相应的目录

常见的报错信息：gcc开发工具没有安装

checking for gcc... no

checking for cc... no

checking for cl.exe... no

configure: error: no acceptable C compiler found in $PATH

See `config.log' for more details.

**5.进行make编译，变成可以执行的程序（放在内存中）**

[root@svr7 ~]# cd /usr/local/inotify-tools-3.13/

[root@svr7 inotify-tools-3.13]# **make**

**6.进行make install安装**

[root@svr7 ~]# cd /usr/local/inotify-tools-3.13/

[root@svr7 inotify-tools-3.13]# **make install**

[root@svr7 inotify-tools-3.13]# ls /opt/

[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/

bin include lib share

[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/bin/

inotifywait inotifywatch

[root@svr7 inotify-tools-3.13]#

1. 数据同步

* 命令用法
  1. rsync [选项...] 源目录 目标目录
* 同步与复制的差异
  1. 复制：完全拷贝源到目标
  2. 同步：增量拷贝，只传输变化过的数据
* rsync操作选项
  1. -n：测试同步过程，不做实际修改
  2. --delete：删除目标文件夹内多余的文档
  3. -a：归档模式，相当于-rlptgoD
  4. -v：显示详细操作信息
  5. -z：传输过程中启用压缩/解压
* **本地同步**

[root@svr7 ~]# mkdir /mydir /todir

[root@svr7 ~]# cp /etc/passwd /mydir

[root@svr7 ~]# touch /mydir/1.txt

[root@svr7 ~]# ls /mydir

[root@svr7 ~]# rsync -av /mydir /todir #同步目录本身

[root@svr7 ~]# ls /todir

[root@svr7 ~]# rsync -av /mydir**/** /todir #同步目录内容

[root@svr7 ~]# ls /todir

[root@svr7 ~]# touch /mydir/2.txt

[root@svr7 ~]# rsync -av /mydir**/** /todir #同步目录内容

[root@svr7 ~]# ls /todir

[root@svr7 ~]# echo 123 > /mydir/1.txt

[root@svr7 ~]# rsync -av /mydir**/** /todir #同步目录内容

[root@svr7 ~]# ls /todir

[root@svr7 ~]# rsync -av --delete /mydir**/** /todir/

[root@svr7 ~]# ls /mydir/

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# touch /todir/a.txt

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# rsync -av --delete /mydir**/** /todir/

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# ls /mydir/

* **远程同步**
* 与远程的 SSH目录保持同步
  1. 下行：rsync [...] user@host:远程目录 本地目录
  2. 上行：rsync [...] 本地目录 user@host:远程目录

**虚拟机A的/mydir目录的内容与虚拟机B的/opt进行同步**

虚拟机A：

]# rsync -av --delete /mydir**/** root@192.168.4.207:/opt

……..connecting (yes/no)? **yes**

root@192.168.4.207's password: **#输入密码**

虚拟机B：

[root@pc207 ~]# ls /opt/

[root@pc207 ~]# vim /etc/ssh/sshd\_config

**UseDNS no** #禁用DNS解析

[root@pc207 ~]# systemctl restart sshd

1. **实时数据同步**

虚拟机A的/mydir/目录的内容与虚拟机B的/opt进行同步

* **实现ssh无密码验证(公钥与私钥)**

**虚拟机A**

1.生成公钥与私钥

[root@svr7 ~]# ssh-keygen #一路回车

[root@svr7 ~]# ls /root/.ssh/

id\_rsa(私钥) id\_rsa.pub(公钥) known\_hosts(记录曾经远程管理过的机器)

2．将虚拟机A的公钥传递给虚拟机B

]# ssh-copy-id root@192.168.4.207

]# rsync -av --delete /mydir/ root@192.168.4.207:/opt

* 基本用法
  1. inotifywait [选项] 目标文件夹
* 常用命令选项
  1. -m，持续监控（捕获一个事件后不退出）
  2. -r，递归监控、包括子目录及文件
  3. -q，减少屏幕输出信息
  4. -e，指定监视的 modify、move、create、delete、attrib 等事件类别

inotifywait

rsync -av --delete /mydir/ root@192.168.4.207:/opt

* **书写shell脚本(了解)**

脚本：可以运行一个文件，实现某种功能

中文:新建用户zhangsan shell： useradd zhangsan

[root@svr7 /]# **vim /root/hello.sh**

**echo hello world**

**hostname**

**ifconfig | head -2**

[root@svr7 /]# **chmod +x /root/hello.sh #赋予执行权限**

[root@svr7 /]# **/root/hello.sh #绝对路径运行脚本**

shell中的循环：

for适合有一定次数的事件

while无限次数事件

重复性：循环解决

死循环：while循环

while 条件

do

重复执行的事情

done

[root@svr7 /]# **vim /etc/rsync.sh**

while /opt/myrpm/bin/inotifywait -rqq /mydir/

do

rsync -a --delete /mydir**/** root@192.168.4.207:/opt

done

[root@svr7 /]# **chmod +x /etc/rsync.sh**  #赋予执行权限

[root@svr7 /]# **/etc/rsync.sh &** #放入后台运行脚本程序

[root@svr7 /]# **jobs -l**  #查看后台运行的进程

[1]+ 17707 运行中 /etc/rsync.sh &

[root@svr7 /]# kill 17707 #停止脚本

1. **数据库服务基础（数据库系统）**

数据库：存放数据的仓库

在**数据库系统**中，有很多的**数据库**，在每一个库中有很多的**表格**

**真机将users.sql上传到虚拟机A**

]# **ls /linux-soft/1/users.sql**

]# **scp /linux-soft/1/users.sql root@192.168.4.7:/root**

* 常见的关系型 数据库管理系统
  1. 微软的 SQL Server
  2. IBM的 DB2
  3. 甲骨文的 Oracle、MySQL
  4. 社区开源版 MariaDB
  5. ……
* **部署MariaDB 数据库系统**

[root@svr7 /]# **yum -y install mariadb-server**

[root@svr7 /]# **systemctl restart mariadb**

* **MariaDB基本使用**

1. Linux系统的管理指令不能使用
2. 所有的数据库系统指令都必须以 ; 结尾
3. 数据库系统的指令大部分不支持tab补全

[root@svr7 /]# **mysql**  #进入数据库系统

> **create database nsd01;** #创建nsd01数据库

> **show databases;**  #查看所有数据库

> **drop database nsd01;**  #删除数据库nsd01

> **show databases;**  #查看所有数据库

> **exit**  #退出数据库系统

[root@svr7 /]# **mysql**  #进入数据库系统

> **use mysql;**  #进入到mysql数据库

> **show tables;**  #查看当前库中所有表格

> **show databases;** #查看所有数据库

> **use test;**  #切换到test数据库

> **exit;**  #退出数据库系统

* **为数据库系统管理员设置密码**
  1. mysqladmin [-u用户名] [-p[旧密码]] password '新密码'

**数据库系统管理员**:对于数据库系统有最高权限，名字为root，能够登陆数据系统的用户信息有mysql库中user表进行储存

**Linux系统管理员**: 对于Linux系统有最高权限，名字为root，能够登陆Linux系统的用户信息/etc/passwd进行储存

[root@svr7 /]# mysqladmin -u root password '456'

[root@svr7 /]# mysql -u root -p #交互式进行登录

Enter password:

[root@svr7 /]# mysql -u root -p456 #非交互式进行登录

已知旧密码修改新密码

]# mysqladmin -u root -p456 password '123'

]# mysql -u root -p123 #非交互式的方式

* **数据库系统恢复出厂设置（密码登录不进去的同学）**

[root@svr7 ~]# rm -rf /var/lib/mysql/ #删除数据库所有数据

[root@svr7 ~]# systemctl stop mariadb

[root@svr7 ~]# yum -y reinstall mariadb-server #重装数据库

[root@svr7 ~]# systemctl restart mariadb

[root@svr7 ~]# mysql

* **恢复数据到数据库中**

1.建立新的数据库

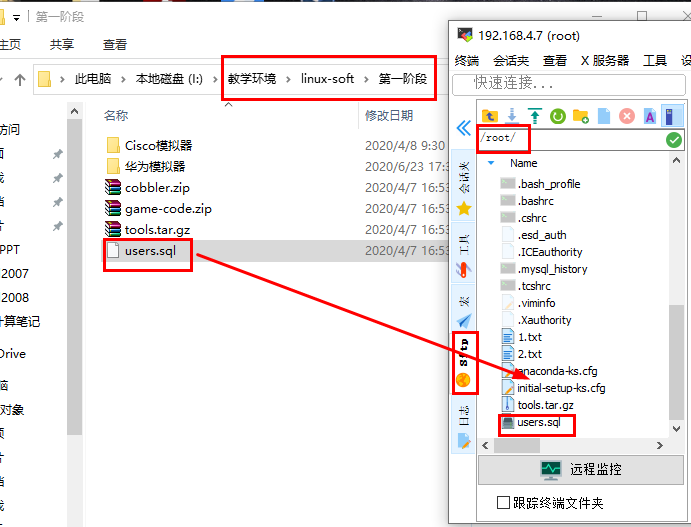
[root@svr7 /]# mysql -u root -p123

MariaDB [(none)]> create database nsd2107;

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit;

2.传递备份好的数据文件users.sql到虚拟机A中



[root@svr7 /]# ls /root

abc02 **users.sql** 图片 桌面

[root@svr7 /]#

3.恢复数据到数据库

]# mysql -u root -p123 nsd2107 < /root/users.sql

]# mysql -u root -p123

MariaDB [(none)]> use nsd2107; #切换到数据库nsd2107

MariaDB [nsd2107]> show tables; #查看当前库有哪些表格

+-------------------+

| Tables\_in\_nsd2107 |

+-------------------+

| base |

| location |

+-------------------+

* **表格操作：**
* **增(insert) 删（delete） 改（update） 查(select)**
* **表字段、表记录**

|  |  |  |
| --- | --- | --- |
| 编号 | 姓名 | 住址 |
| 1 | Dc | 东村 |
| 2 | Tc | 西村 |

* **查(select)**

格式: select 表字段，表字段，…… from 库名.表名;

[root@svr7 /]# mysql -u root -p123

> use nsd2107;

> select \* from base; #查看base所有表字段内容

> select \* from location; #查看location所有表字段内容

> use test;

> select \* from nsd2107.base;

> use nsd2107;

> select id,name from base;

[root@svr7 /]# mysql

> use nsd2107;

查询密码为456的记录

> select \* from base where password='456';

查询id编号为4的记录

> select \* from base where id='4';

查询id编号为4并且密码必须为123的记录

> select \* from base where id='4' and password='123';

查询id编号为4或者密码为123的记录

> select \* from base where id='4' or password='123';

* **增(insert)**

格式：insert 表名 values (‘值’,‘值’,‘值’);

MariaDB [nsd2107]> insert base values('10','dc','789');

MariaDB [nsd2107]> insert base values('11','tcc','369');

MariaDB [nsd2107]> select \* from base ;

* **改（update）**

格式：

update 表名 set 表字段='新值' where 表字段='值'；

> select \* from base ;

> update base set password='8888' where id='1';

> select \* from base ;

> update base set password='9999' where id='2';

> select \* from base ;

* **删（delete）**

> use nsd2107;

> delete from base where id='4' ;

> select \* from base ;

> delete from base where id='3' ;

> select \* from base ;

#########################################

命令补充

* **获取命令帮助**

方式一：命令 --help

[root@localhost ~]# cat --help

方式二：man 命令

[root@localhost ~]# man cat #按q退出

[root@localhost ~]# man passwd #显示passwd命令帮助

[root@localhost ~]# man 5 passwd

数字5表示帮助的类型，表示配置文件类型

* **历史命令**

管理/调用曾经执行过的命令

* + history：查看历史命令列表
  + history -c：清空历史命令
  + !n：执行命令历史中的第n条命令
  + !str：执行最近一次以str开头的历史命令

[root@svr7 ~]# vim /etc/profile

HISTSIZE=1000 #默认记录1000条

[root@localhost ~]# history #显示历史命令列表

[root@localhost ~]# history -c #清空历史命令

[root@localhost ~]# history

[root@localhost ~]# cat /etc/redhat-release

[root@localhost ~]# ls /root

[root@localhost ~]# history

[root@localhost ~]# !cat #指定最近一条以cat开头的历史命令

[root@localhost ~]# !ls #指定最近一条以ls开头的历史命令

* **du，统计文件的占用空间**
  + du [选项]... [目录或文件]...
  + -s：只统计每个参数所占用的总空间大小
  + -h：提供易读容量单位（K、M等）

[root@localhost ~]# du -sh /root

[root@localhost ~]# du -sh /etc

[root@localhost ~]# du -sh /boot

[root@localhost ~]# du -sh /

* **date，查看/调整系统日期时间**
  + date +%F、date +%R
  + date +"%Y-%m-%d %H:%M:%S"
  + date -s "yyyy-mm-dd HH:MM:SS"

]# date

]# date -s "2008-9-6 11:11:11" #修改系统时间

]# date

]# date -s "2020-9-5 15:37:11"

]# date

[root@localhost ~]# date +%Y #显示年

[root@localhost ~]# date +%m #显示月

[root@localhost ~]# date +%d #显示日期

[root@localhost ~]# date +%H #显示时

[root@localhost ~]# date +%M #显示分

[root@localhost ~]# date +%S #显示秒

[root@localhost ~]# date +%F #显示年-月-日

[root@localhost ~]# date +%R #显示时:分

* **制作链接(链接)文件（制作快捷方式）**

格式：ln -s /路径/源数据 /路径/快捷方式的名称 #软链接

]# ln -s /etc/sysconfig/network-scripts/ /ns

]# ls /

]# ls -l /ns #查看快捷方式的信息

]# touch /ns/haha.txt

]# touch /ns/maohehaozi.txt

]# touch /ns/shukehebeita.txt

]# ls /etc/sysconfig/network-scripts/

软链接优势：可以针对目录与文件制作快捷方式，支持跨分区

软链接缺点：源数据消失，快捷方式失效

格式：ln /路径/源数据 /路径/快捷方式的名称 #硬链接

硬链接优势：源数据消失，快捷方式仍然有效

硬链接缺点：只能针对文件制作快捷方式，不支持支持跨分区

[root@localhost ~]# rm -rf /opt/\*

[root@localhost ~]# echo 123 > /opt/A.txt

[root@localhost ~]# ln -s /opt/A.txt /opt/B.txt #软链接

[root@localhost ~]# ls /opt/

[root@localhost ~]# ln /opt/A.txt /opt/C.txt #硬链接

[root@localhost ~]# ls /opt/

[root@localhost ~]# cat /opt/B.txt

[root@localhost ~]# cat /opt/C.txt

[root@localhost ~]# rm -rf /opt/A.txt

[root@localhost ~]# ls /opt/

[root@localhost ~]# cat /opt/B.txt #软链接失效

cat: /opt/B.txt: 没有那个文件或目录

[root@localhost ~]# cat /opt/C.txt #硬链接仍然有效

* **zip归档工具，跨平台**
* 归档+压缩操作: zip [-r] 备份文件.zip 被归档的文档...

[-r]:被归档的数据有目录，必须加上此选项

]# zip -r /opt/abc.zip /etc/passwd /home

]# ls /opt/

* 释放归档+解压操作: unzip 备份文件.zip [-d 目标文件夹]

]# mkdir /nsd20

]# unzip /opt/abc.zip -d /nsd20

]# ls /nsd20

]# ls /nsd20/etc/

]# ls /nsd20/home/