正式课程笔记：码云

https://gitee.com/niubenlinux/nsd2010

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

1. 环境的准备

* **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

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：

]# ls /opt/

1.txt haha passwd

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

* **监控目录内容变化工具**

**将真机的tools.tar.gz传递数据到虚拟机A**



[root@svr7 ~]# ls /root

**tools.tar.gz**  下载

公共 音乐

[root@svr7 ~]#

源码编译安装步骤：

步骤一:安装开发工具

]# yum -y install make

]# yum -y install gcc

]# rpm -q gcc

]# rpm -q make

步骤二:进行tar解包

]# tar -xf /root/tools.tar.gz -C /usr/local/

]# ls /usr/local/

]# ls /usr/local/tools/

]# tar -xf /usr/local/tools/inotify-tools-3.13.tar.gz -C /usr/local/

]# ls /usr/local/

步骤三：运行configure脚本进行配置

作用1：检测系统是否安装gcc

作用2：可以指定安装位置及功能

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

]# ./configure --prefix=/opt/myrpm **#指定安装位置**

常见错误：没有安装gcc

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.

步骤四：make进行编译，产生可以执行的程序

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

]# **make**

步骤五：make install进行安装

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

]# **make install**

]# ls /opt/

]# ls /opt/myrpm/

]# ls /opt/myrpm/bin/

* 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

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 a+x /etc/rsync.sh #赋予执行权限

[root@svr7 /]# /etc/rsync.sh & #运行脚本程序

[root@svr7 /]# jobs -l

[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

* 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

> use mysql; #切换到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;

* **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

* **恢复数据到数据库中**

1.建立新的数据库

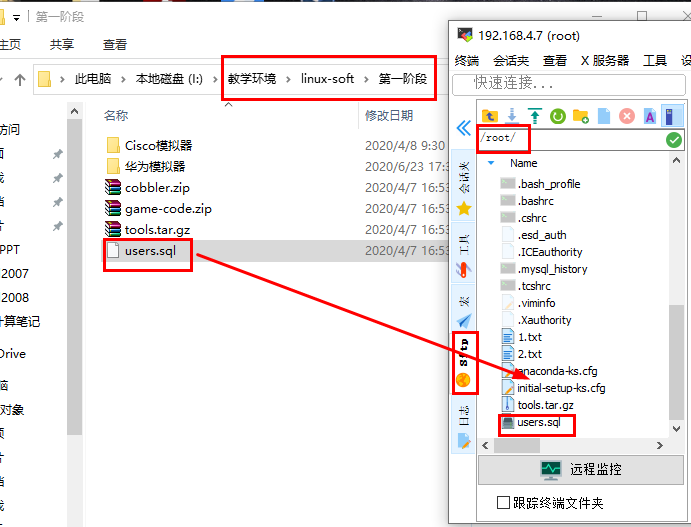
[root@svr7 /]# mysql -u root -p123

MariaDB [(none)]> create database nsd2010;

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 nsd2010 < /root/users.sql

]# mysql -u root -p123

MariaDB [(none)]> use nsd2010; #切换到数据库nsd2010

MariaDB [nsd2010]> show tables; #查看当前库有哪些表格

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

| Tables\_in\_nsd2010 |

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

| 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 nsd2010;

> 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 nsd2010.base;

> use nsd2010;

> select id,name from base;

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

> use nsd2010;

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

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

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

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

* **增(insert)**

格式：insert 表名 values (‘值’,‘值’,‘值’);

MariaDB [nsd2010]> insert base values('10','dc','789');

MariaDB [nsd2010]> insert base values('11','tcc','369');

MariaDB [nsd2010]> 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 nsd2010;

> delete from base where id='4' ;

> select \* from base ;

> delete from base where id='3' ;

> select \* from base ;

* **数据库授权操作**

MariaDB [(none)]> 交互指令

* 1. GRANT 权限列表 ON 数据库名.表名 TO 用户名@客户机地址 IDENTIFIED BY '密码';

> grant select on nsd2010.\* to dcc@localhost identified by '123';

> exit；

当dcc从localhost本地登录输入密码123，将会获得nsd2010库中所有表格的查询权限

[root@svr7 /]# mysql -u dcc -p123 #测试

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

补充内容（了解）

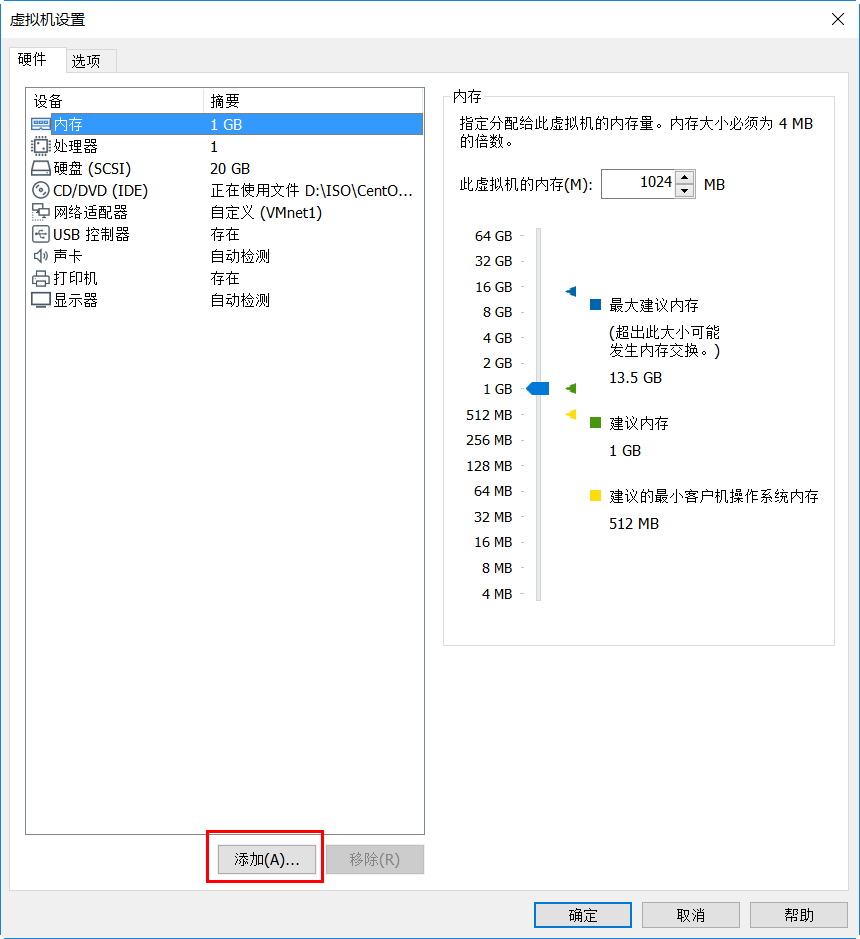
**一、链路聚合（网卡team、网卡组队、聚合链路）**

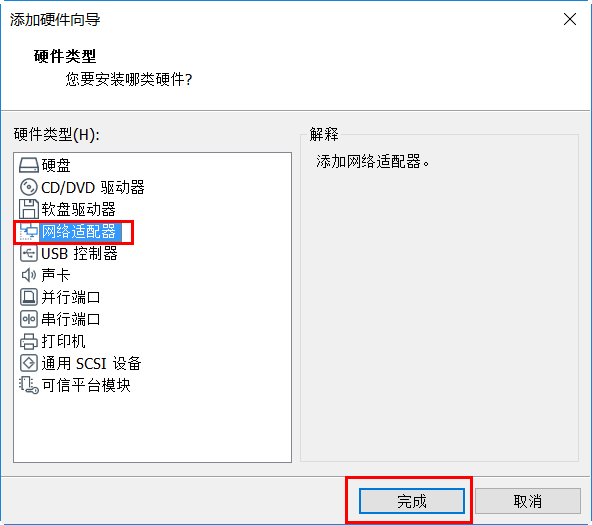
作用：备份网卡设备，提高可靠性

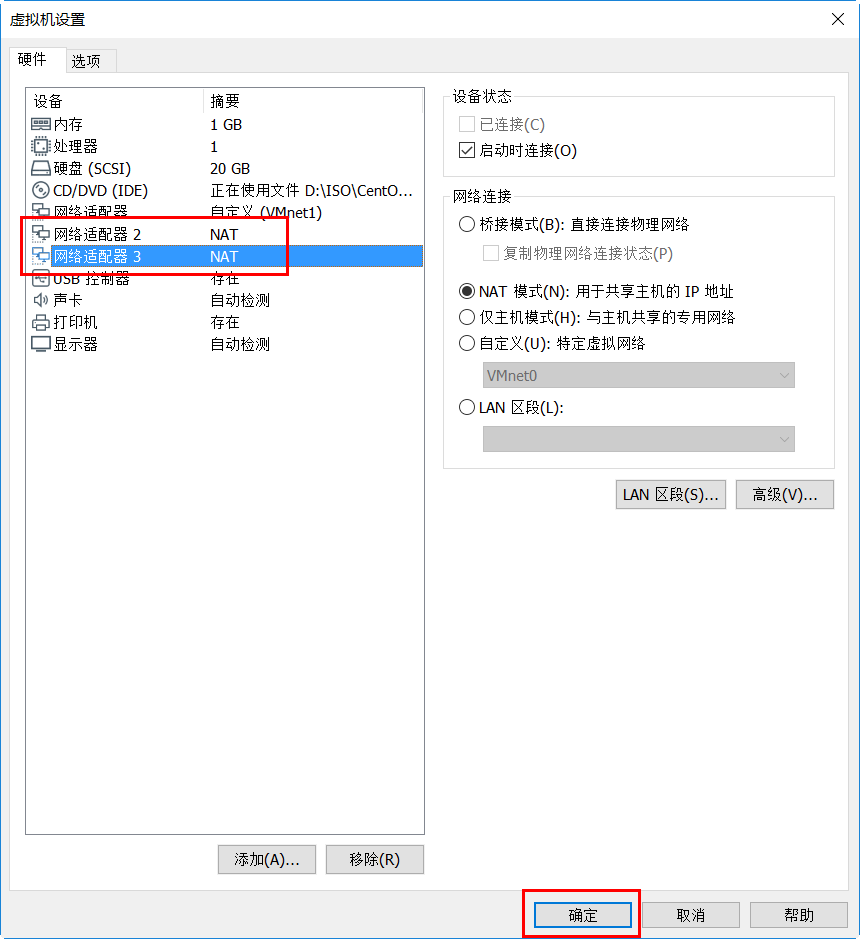
虚拟网卡：team0

1.关闭虚拟机添加两块网卡









[root@A ~]# ifconfig | less

2.建立虚拟网卡设备，参考 man teamd.conf 进行全文查找 /example

[root@A ~]# nmcli connection add type team ifname team0 con-name team0 autoconnect yes config '{"runner": {"name": "activebackup"}}'

[root@A ~]# nmcli 连接 添加 类型 组队 网卡名 team0命名为team0

每次开机自启动 工作方式 活跃备份

[root@A ~]# ifconfig

3.添加成员

[root@A ~]# nmcli connection add type team-slave autoconnect yes ifname eth1 con-name team0-1 master team0

[root@A ~]# nmcli connection add type team-slave autoconnect yes ifname eth2 con-name team0-2 master team0

[root@A ~]# nmcli 连接 添加 类型 组队-成员 每次开机自动启用 网卡名 eth2 命名为 team0-2 主设备为 team0

4.激活，以配置文件的名进行激活

successfully（成功）

[root@A ~]# nmcli connection up team0

[root@A ~]# nmcli connection up team0-1

[root@A ~]# nmcli connection up team0-2

5.为team0配置IP地址

[root@A ~]# nmcli connection modify team0 ipv4.method manual ipv4.addresses 192.168.1.1/24 connection.autoconnect yes

[root@A ~]# nmcli connection up team0

[root@A ~]# teamdctl team0 state #专门查看team0状态的命令

[root@A ~]# ifconfig eth1 down

[root@A ~]# teamdctl team0 state

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

**二、VDO 了解内容**

•Virtual Data Optimizer（虚拟数据优化器）

–一个内核模块，目的是通过**重删**减少磁盘的空间占用，以及减少复制带宽

–VDO是基于块设备层之上的，也就是在原设备基础上映射出mapper虚拟设备，然后直接使用即可

•重复数据删除

–输入的数据会判断是不是冗余数据

–判断为重复数据的部分不会被写入，然后对源数据进行更新，直接指向原始已经存储的数据块即可

•压缩

–对每个单独的数据块进行处理

[root@svr7 ~]# yum  -y  install  vdo  kmod-kvdo    #所需软件包

•制作VDO卷

•vdo基本操作：参考man vdo 全文查找/example

–vdo  create  --name=VDO卷名称  --device=设备路径 --vdoLogicalSize=逻辑大小

–vdo  list

–vdo  status  -n  VDO卷名称

–vdo  remove  -n  VDO卷名称

–vdostatus  [--human-readable] [/dev/mapper/VDO卷名称]

•VDO卷的格式化加速（跳过去重分析）：

–mkfs.xfs  –K   /dev/mapper/VDO卷名称

–mkfs.ext4  -E  nodiscard  /dev/mapper/VDO卷名称

前提制作VDO需要2G以上的内存

[root@nb ~]# vdo create --name=vdo0 --device=/dev/sdc --vdoLogicalSize=200G

[root@nb ~]# mkfs.xfs -K /dev/mapper/vdo0

[root@nb ~]# mkdir /nsd01

[root@nb ~]# mount /dev/mapper/vdo0 /nsd01

[root@nb ~]# df -h

[root@nb ~]# vdostats --hum /dev/mapper/vdo0 #查看vdo设备详细信息

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

/dev/mapper/vdo0  /nsd01  xfs  defaults,\_netdev 0 0