

# NSD Study

2018-03

达内 IT 培训集团

案例练习,准备:

```
rht-vmctl reset classroom
rht-vmctl reset server
rht-vmctl reset desktop
```

#####

案例 1:为虚拟机 server0 配置以下静态地址参数, 实现永久设置

- 主机名:server0.example.com

- IP 地址:172.25.0.11
- 子网掩码:255.255.255.0
- 默认网关:172.25.0.254
- DNS 服务器:172.25.254.254

案例 2:为虚拟机 desktop0 配置以下静态地址参数, 实现永久设置

- 主机名:desktop0.example.com
- IP 地址:172.25.0.10
- 子网掩码:255.255.255.0
- 默认网关:172.25.0.254
- DNS 服务器:172.25.254.254

案例 3:指定 yum 软件源

为 server0 指定可用的 yum 软件源

- YUM 软件库的地址为  
http://classroom.example.com/content/rhel7.0/x86\_64/dvd
- 将此配置为虚拟机 server0 的默认软件仓库
- 确认可用的仓库列表
- 利用 yum 仓库安装 httpd 与 vsftpd

案例 4:指定 yum 软件源

为 desktop0 指定可用的 yum 软件源

- YUM 软件库的地址为  
http://classroom.example.com/content/rhel7.0/x86\_64/dvd
- 将此配置为虚拟机 server0 的默认软件仓库
- 确认可用的仓库列表
- 利用 yum 仓库安装 httpd 与 vsftpd

案例 5: 虚拟机 server0 上操作, 复制、粘贴、移动

以 root 用户新建/nsddir/目录, 在此目录下新建 readme.txt 文件, 并进一步完成下列操作

```
[root@server0 ~]# mkdir /nsddir
[root@server0 ~]# touch /nsddir/readme.txt
[root@server0 ~]# ls /nsddir/
readme.txt
[root@server0 ~]#
```

### 1) 将 “I love Linux” 写入到文件 readme.txt

```
[root@server0 ~]# echo 'I love Linux' >> /nsddir/readme.txt #也可以用 vim 写入
[root@server0 ~]# cat /nsddir/readme.txt
I love Linux
[root@server0 ~]#
```

### 2) 将 readme.txt 重命名为 mylove.txt

```
[root@server0 ~]# mv /nsddir/readme.txt /nsddir/mylove.txt
[root@server0 ~]# ls /nsddir/
mylove.txt
[root@server0 ~]#
```

### 3) 将/etc/passwd、/boot、/etc/group 同时拷贝到/nsddir 目录下

```
[root@server0 ~]# cp -r /etc/passwd /boot/ /etc/group /nsddir/
[root@server0 ~]# ls /nsddir/
boot group mylove.txt passwd
[root@server0 ~]#
```

### 4) 将 ifconfig 命令的前两行内容, 追加写入 mylove.txt

```
[root@server0 ~]# ifconfig | head -2
[root@server0 ~]# ifconfig | head -2 >> /nsddir/mylove.txt
[root@server0 ~]# cat /nsddir/mylove.txt
```

### 5) 将主机名永久配置文件, 拷贝到/nsddir 目录下

```
[root@server0 ~]# cp /etc/hostname /nsddir/
[root@server0 ~]# ls /nsddir/
boot group hostname mylove.txt passwd
[root@server0 ~]#
```

### 6) 将 DNS 永久配置文件, 拷贝到/nsddir 目录下

```
[root@server0 ~]# cp /etc/resolv.conf /nsddir/
[root@server0 ~]# ls /nsddir/
boot group hostname mylove.txt passwd resolv.conf
[root@server0 ~]#
```

### 7) 新建目录结构/nsd/test

```
[root@server0 ~]# mkdir -p /nsd/test
```

### 8) 在目录/nsd/test 创建文件 nsd.txt 并写入内容 NSD Student

```
[root@server0 ~]# echo NSD Student > /nsd/test/nsd.txt #也可以用 vim 写入
```

```
[root@server0 ~]# cat /nsd/test/nsd.txt
NSD Student
[root@server0 ~]#
```

**9) 将/nsd/test/nsd.txt 文件复制到/root 目录下, 同时 改名为 tedu.txt**

```
[root@server0 ~]# cp /nsd/test/nsd.txt /root/tedu.txt
[root@server0 ~]# ls /root/
```

**10) 利用 vim 修改文件/etc/hostname 将其原有内容全部删除, 写入新的内容为 server0.example.com**

```
[root@server0 ~]# vim /etc/hostname
server0.example.com
```

**11) 将 /etc/passwd 、 /etc/resolv.conf 、 /etc/hostname 同时拷贝到 /nsd/test/目录下**

```
[root@server0 ~]# cp /etc/passwd /etc/resolv.conf /etc/hostname /nsd/test/
[root@server0 ~]# ls /nsd/test/
hostname nsd.txt passwd resolv.conf
[root@server0 ~]#
```

**12) 将文件 /nsd/test/hostname 重改名为 hn.txt**

```
[root@server0 ~]# mv /nsd/test/hostname /nsd/test/hn.txt
[root@server0 ~]# ls /nsd/test/
hn.txt nsd.txt passwd resolv.conf
[root@server0 ~]#
```

**13) 创建目录结构/nsd/test/kernel**

```
[root@server0 ~]# mkdir -p /nsd/test/kernel
```

**案例 6:虚拟机 server0 上操作, 查找并处理文件**

**- 创建目录 /root/findfiles/**

```
[root@server0 ~]# mkdir /root/findfiles
```

**- 利用 find 查找所有用户 student 拥有的文件,把它们拷贝到 /root/findfiles/ 文件夹中**

```
[root@server0 ~]# find / -user student -type f
[root@server0 ~]# find / -user student -type f -exec cp -r {} /root/findfiles/ \;
```

**- 利用 find 查找/boot 目录下大于 10M 并且必须是文件, 拷贝到/opt**

```
[root@server0 ~]# find /boot -size +10M -type f
[root@server0 ~]# find /boot -size +10M -type f -exec cp -r {} /opt \;
```

– 利用 find 查找/boot/ 目录下以 vm 开头且必须是文件，拷贝到/opt

```
[root@server0 ~]# find /boot -name "vm*" -type f
[root@server0 ~]# find /boot -name "vm*" -type f -exec cp -r {} /opt \;
```

– 利用 find 查找/etc 目录下，以 tab 作为结尾的，并统计其个数

```
[root@server0 ~]# find /etc -name "*.tab"
[root@server0 ~]# find /etc -name "*.tab" | wc -l
```

– 利用 find 查找/etc 目录下，以 .conf 作为结尾的，并统计其个数

```
[root@server0 ~]# find /etc -name "*.conf"
[root@server0 ~]# find /etc -name "*.conf" | wc -l
```

**案例 7:虚拟机 server0 上操作,查找并提取文件内容**

1.在文件 /usr/share/dict/words 中查找到所有包含字符串 seismic 的行,将输出信息,写入到/opt/nsd.txt

```
[root@server0 ~]# grep "seismic" /usr/share/dict/words
[root@server0 ~]# grep "seismic" /usr/share/dict/words > /opt/nsd.txt
```

2.将/etc/login.defs 文件有效内容 写入到/root/login.txt

```
[root@server0 ~]# grep -v "^#" /etc/login.defs | grep -v "^$"
[root@server0 ~]# grep -v "^#" /etc/login.defs | grep -v "^$" > /root/login.txt
```

3.提取/etc/passwd 以 bash 结尾的行，将其信息写入/opt/system.txt

```
[root@server0 ~]# grep "bash$" /etc/passwd > /opt/system.txt
```

**案例 8:虚拟机 server0 上操作,tar 制作/释放归档压缩包 (zcf、ztf、zxf、jcf、jtf、jxf、cf、tf)**

1) 备份/boot、/home 这两个文件夹，保存为 boothome.tar.gz 文件

```
[root@server0 ~]# tar -zcf boothome.tar.gz /boot /home
```

2) 查看 boothome.tar.gz 文件内包含哪些内容

```
[root@server0 ~]# tar -tf boothome.tar.gz
```

**3) 将 boothome.tar.gz 释放到文件夹/root/boothome/下**

```
[root@server0 ~]# mkdir /root/boothome  
[root@server0 ~]# tar -xf boothome.tar.gz -C /root/boothome
```

**4) 创建一个名为 /root/backup.tar.bz2 的归档文件，其中包含 /usr/local 目录中的内容**

```
[root@server0 ~]# tar -jcf /root/backup.tar.bz2 /usr/local
```

**案例 9:虚拟机 server0 上操作**

**新建用户 tom，其用户 ID 为 1234，密码是 abcdef**

```
[root@server0 ~]# useradd -u 1234 tom  
[root@server0 ~]# echo abcdef | passwd --stdin tom
```

**创建下列用户、组以及组的成员关系：**

**– 一个名为 stugrp 的组**

```
[root@server0 ~]# groupadd stugrp
```

**– 一个名为 natasha 的用户，其属于 stugrp 组，这个组是该用户的从属组**

```
[root@server0 ~]# useradd -G stugrp natasha
```

**– 一个名为 harry 的用户，其属于 stugrp 组，这个组是该用户的从属组**

```
[root@server0 ~]# useradd -G stugrp harry
```

**– 一个名为 sarah 的用户，其在系统中没有可交互的 Shell (/sbin/nologin)，并且不是 adminuser 组的成员**

```
[root@server0 ~]# useradd -s /sbin/nologin sarah
```

**– natasha、harry、sarah 的密码都要设置为 pass123**

```
[root@server0 ~]# echo pass123 | passwd --stdin natasha  
[root@server0 ~]# echo pass123 | passwd --stdin harry  
[root@server0 ~]# echo pass123 | passwd --stdin sarah
```

**案例 10:虚拟机 server0 上操作**

**为用户 natasha 配置一个定时任务**

**– 每天在本地时间 14:23 执行**

- 需要完成的任务操作为 `/bin/echo hiya`

```
[root@server0 ~]# crontab -e -u natasha
23 14 * * * /bin/echo hiya
```

### 案例 11:虚拟机 server0 上操作

将文件 `/etc/fstab` 拷贝为 `/var/tmp/fstab`, 并调整文件 `/var/tmp/fstab` 权限

```
[root@server0 ~]# cp /etc/fstab /var/tmp/
```

满足以下要求:

- 此文件的拥有者是 `root`
- 此文件属于 `root` 组
- 此文件对任何人都不可执行
- 用户 `natasha` 能够对此文件执行读和写操作

```
[root@server0 ~]# setfacl -m u:natasha:rw /var/tmp/fstab
```

- 用户 `harry` 对此文件既不能读, 也不能写

```
[root@server0 ~]# setfacl -m u:harry:--- /var/tmp/fstab
```

- 所有其他用户 (当前的和将来的) 能够对此文件进行 读操作

### 案例 12:虚拟机 server0 上操作

创建一个共用目录 `/home/admins`, 要求如下:

```
[root@server0 ~]# mkdir /home/admins
```

- 此目录的组所有权是 `adminuser`

```
[root@server0 ~]# groupadd adminuser
[root@server0 ~]# chown :adminuser /home/admins
[root@server0 ~]# ls -ld /home/admins
```

- `adminuser` 组的成员对此目录有读写和执行的权限, 除此以外的其他所有用户没有任何权限

```
[root@server0 ~]# chmod g+w,o=--- /home/admins
[root@server0 ~]# ls -ld /home/admins
```

- root 用户能够访问系统中的所有文件和目录
- 在此目录中创建的文件，其组的所有权会自动设置为 属于 adminuser 组

```
[root@server0 ~]# chmod g+s /home/admins  
  
[root@server0 ~]# mkdir /home/admins/test  
[root@server0 ~]# ls -ld /home/admins/test
```

### 案例 13：虚拟机 server0 上操作,绑定到 LDAP 验证服务

- 使用系统 classroom.example.com 提供的 LDAP 服务
- 验证服务的基本 DN 是：dc=example,dc=com
- 账户信息和验证信息都是由 LDAP 提供的
- 连接要使用证书加密，证书可以在下面的链接下载：  
<http://classroom.example.com/pub/example-ca.crt>
- 确认本地可以识别 ldapuser10 用户

#### 1. 安装一个客户端 sssd 软件,与 LDAP 服务端沟通

```
[root@server0 /]# yum -y install sssd
```

#### 2. 安装 authconfig-gtk 图形的工具 配置 sssd 软件

```
[root@server0 /]# yum -y install authconfig-gtk  
[root@server0 /]# rpm -q authconfig-gtk
```

#### 3. 运行 authconfig-gtk 图形的工具

```
[root@server0 /]# authconfig-gtk  
  
选择 LDAP  
dc=example,dc=com          #指定服务端域名  
classroom.example.com      #指定服务端主机名  
  
勾选 TLS 加密  
使用证书加密: http://classroom.example.com/pub/example-ca.crt  
  
选择 LDAP 密码
```

#### 4. 重起客户端服务 sssd 服务，设置开机自启动

```
[root@server0 /]# systemctl restart sssd    #重起 sssd 服务  
[root@server0 /]# systemctl enable sssd     #设置开机自启动
```

#### 5. 验证

```
[root@server0 ~]# grep 'ldapuser10' /etc/passwd  
[root@server0 ~]# id ldapuser10
```



#### 案例 14: 虚拟机 server0 上操作, 访问 NFS 共享

- 查看 classroom.example.com 的 NFS 共享
- 将 classroom.example.com 的 NFS 共享目录, 挂载到本地/home/guests

##### 1. 查看共享 classroom.example.com

```
[root@server0 ~]# showmount -e classroom.example.com
Export list for classroom.example.com:
/home/guests 172.25.0.0/255.255.0.0
```

##### 2. 访问共享内容, 将服务端的共享文件夹数据, 挂载到本地/home/guests 以本地的/home/guests 作为访问点

```
[root@server0 ~]# mkdir /home/guests

[root@server0 ~]# mount classroom.example.com:/home/guests/ /home/guests/

[root@server0 ~]# ls /home/guests/

[root@server0 ~]# su - ldapuser0
```

#### 案例 15: 虚拟机 server0 上操作, (GPT 分区模式) 规划分区

关闭虚拟机, 图形添加一块 60G 的硬盘并规划分区:

划分 4 个 10G 的主分区;

1 个 12G 的主分区

```
[root@server0 ~]# lsblk

[root@server0 ~]# parted /dev/sdb
.....
(parted) mktable gpt    #设置为 GPT 分区模式
(parted) print          #显示分区表信息

(parted) mkpart
分区名称?  []? nsd
文件系统类型?  [ext2]? ext4
起始点?  0
结束点?  10G
警告: The resulting partition is not properly aligned for best performance.
忽略/Ignore/放弃/Cancel? i          #选择忽略
```

```
(parted) print
.....
Number  Start   End     Size    File system  Name  标志
1       17.4kB  10.0GB  10000MB                nsd

(parted)
(parted) mkpart nsd ext4  10G 20G
(parted) mkpart nsd ext4  20G 30G
(parted) mkpart nsd ext4  30G 40G
(parted) mkpart nsd ext4  40G 52G
(parted) print
.....

Number  Start   End     Size    File system  Name  标志
1       17.4kB  10.0GB  10000MB                nsd
2       10.0GB  20.0GB  9999MB                nsd
3       20.0GB  30.0GB  10.0GB                nsd
4       30.0GB  40.0GB  10.0GB                nsd
5       40.0GB  52.0GB  12.0GB                nsd
```

## 案例 16: 在 server 上操作, 分区使用

### 1、案例 15 中新添加 60G 硬盘的第一个主分区

- 格式化 xfs 分区, 实现该分区开机自动挂载/mnt/xpart

```
[root@server0 ~]# mkfs.xfs /dev/vdb1
[root@server0 ~]# mkdir /mnt/xpart
[root@server0 ~]# vim /etc/fstab
/dev/vdb1 /mnt/xpart xfs defaults 0 0
[root@server0 ~]# mount -a
[root@server0 ~]# df -h
```

### 2、案例 15 中新添加 60G 硬盘的第二个主分区

- 完成开机自动挂载, 挂载点/mnt/mypart, 文件系统为 ext4

```
[root@server0 ~]# mkfs.ext4 /dev/vdb2
[root@server0 ~]# mkdir /mnt/mypart
[root@server0 ~]# vim /etc/fstab
/dev/vdb2 /mnt/mypart ext4 defaults 0 0
[root@server0 ~]# mount -a
[root@server0 ~]# df -h
```

## 案例 17: 虚拟机 server0 操作, 构建 LVM 存储

- 新建一个名为 systemvg 的卷组

```
[root@server0 ~]# vgcreate systemvg /dev/vdb[3-4]
[root@server0 ~]# vgs
```

- 在此卷组中创建一个名为 **vo** 的逻辑卷，大小为 8G

```
[root@server0 ~]# lvcreate -L 8G -n vo systemvg
[root@server0 ~]# lvs
```

- 将逻辑卷 **vo** 格式化为 **xfs** 文件系统

```
[root@server0 ~]# mkfs.xfs /dev/systemvg/vo
[root@server0 ~]# blkid /dev/systemvg/vo
```

- 将逻辑卷 **vo** 挂载到 **/vo** 目录，并在此目录下建立一个测试文件 **votest.txt**，内容为 **"I AM KING"**

```
[root@server0 ~]# mkdir /vo
[root@server0 ~]# mount /dev/systemvg/vo /vo
[root@server0 ~]# df -h
[root@server0 ~]# echo I AM KING > /vo/votest.txt
```

- 实现逻辑卷 **vo** 开机自动挂载到 **/vo**

```
[root@server0 ~]# vim /etc/fstab
/dev/systemvg/vo /vo xfs defaults 0 0
[root@server0 ~]# umount /vo
[root@server0 ~]# mount -a
[root@server0 ~]# df -h
```

#### 案例 18:虚拟机 server0 操作，构建 LVM 存储(修改 PE 大小)

- 在 **datastore** 卷组中的所有逻辑卷，其 PE 的大小为 1M
- 新的逻辑卷命名为 **database**，其大小为 50 个 PE 的大小，属于 **datastore** 卷组
- 使用 **EXT3** 文件系统对逻辑卷 **database** 格式化，此逻辑卷应该在开机时自动挂载到 **/mnt/database** 目录

```
[root@server0 ~]# vgcreate -s 1M datastore /dev/vdb5
[root@server0 ~]# vgs
[root@server0 ~]# vgdisplay datastore #查看 PE 大小

[root@server0 ~]# lvcreate -l 50 -n database datastore
[root@server0 ~]# lvs

[root@server0 ~]# mkfs.ext4 /dev/datastore/database

[root@server0 ~]# mkdir /mnt/database
[root@server0 ~]# vim /etc/fstab
/dev/datastore/database /mnt/database ext3 defaults 0 0
[root@server0 ~]# mount -a

[root@server0 ~]# df -h
```

#### 案例 19:在在 server0、desktop0 上操作

– 将防火墙默认区域设置为 trusted

```
[root@server0 ~]# firewall-cmd --set-default-zone=trusted  
[root@desktop0 ~]# firewall-cmd --set-default-zone=trusted
```

**案例 20: 在 server 上操作, 验证端口转发策略**

- 从 desktop0 上访问 server0 的 5423 端口, 与访问 server0 的 80 端口效果一样

虚拟机 Server0

```
[root@server0 ~]# yum -y install httpd  
[root@server0 ~]# systemctl restart httpd  
[root@server0 ~]# firewall-cmd --permanent --zone=trusted  
--add-forward-port=port=5423:proto=tcp:toport=80  
  
[root@server0 ~]# firewall-cmd --reload  
[root@server0 ~]# firewall-cmd --zone=trusted --list-all
```

**案例 21: 在 server 上操作, 搭建 mariadb 数据库系统 (安装软件, 启动服务即可)**

1. 在 server0 上安装 MariaDB 数据库系统

1) 安装 mariadb-server 软件包

2) 启动 mariadb 服务

```
[root@server0 ~]# yum -y install mariadb-server  
[root@server0 ~]# systemctl restart mariadb  
[root@server0 ~]# systemctl enable mariadb
```

**案例 22: 配置聚合连接**

在两个虚拟机之间配置一个链路, 要求如下:

- 此链路使用接口 eth1 和 eth2
- 此链路在其中一个接口失效时仍然能工作
- 此链路在 server0 上使用下面的地址 172.16.3.20/255.255.255.0
- 此链路在 desktop0 上使用下面的地址 172.16.3.25/255.255.255.0
- 此链路在系统重启之后依然保持正常状态

1. 制作虚拟网卡 team0, 参考 man teamd.conf 全文查找/example 按 n 跳转匹配项

```
# nmcli connection add type team  
autoconnect yes con-name team0 ifname team0  
config '{"runner": {"name": "activebackup"}}'  
  
# nmcli connection 添加 类型为 team 的设备  
每次开机自动启用      配置文件命名为 team0 网卡显示的名字为 team0
```

team0 网卡内部成员工作模式为 '{"runner": {"name": "activebackup"}}' (热备)

```
# ifconfig
```

## 2. 为 team0 添加成员

```
# nmcli connection add type team-slave con-name team0-1 ifname eth1 master team0
# nmcli connection add type team-slave con-name team0-2 ifname eth2 master team0

# 添加 类型为 team-slave 的设备 配置文件命名为 team0-1 网卡为 eth1
主设备为 team0
```

## 3. 配置 team0 的 ip 地址与激活

```
# nmcli connection modify team0 ipv4.method manual
  ipv4.addresses 172.16.3.20/24 connection.autoconnect yes

# nmcli connection up team0      #激活 team0 网卡
# nmcli connection up team0-1    #激活 team0-1 成员
# nmcli connection up team0-2    #激活 team0-2 成员
```

**案例 23:配置 Samba 文件夹共享, 防火墙默认区域设置为 trusted, 修改 SELinux 状态为 Enforcing**

**在 server0 上发布共享文件夹**

- 1) 此服务器必须是 STAFF 工作组的一个成员
- 2) 发布目录 /devops, 共享名为 devops
- 5) 用户 harry 对共享 devops 可读可写, 密码是 123
- 6) 用户 kenji 对共享 devops 可读, 密码是 123

### 1. 安装软件包:samba

```
[root@server0 ~]# yum -y install samba
```

### 2. 建立 Samba 共享帐号

```
[root@server0 ~]# useradd harry
[root@server0 ~]# useradd kenji

[root@server0 ~]# pdbedit -a harry      #将本地用户 harry 设置为 Samba 共享帐号
[root@server0 ~]# pdbedit -a kenji
[root@server0 ~]# pdbedit -L           #显示本地都有那些 Samba 共享帐号
harry:1001:
```

### 3. 修改 Smb 服务配置

```
[root@server0 ~]# mkdir /devops
[root@server0 ~]# echo 123 > /devops/1.txt
[root@server0 ~]# ls /devops

[root@server0 ~]# vim /etc/samba/smb.conf

workgroup = STAFF                #设置工作组

在命令模式下, 按 大写的 G 到全文最后

[devops]                          #设置共享名
path = /devops                   #设置共享的实际路径
write list = harry               #允许 harry 可写
```

### 4. 重起 smb 服务, 设置开机自起

```
[root@server0 ~]# systemctl restart smb    #重起服务
[root@server0 ~]# systemctl enable smb     #设置开机自起
```

### 5. SELinux 设置布尔值 (功能的开关)

- 需要加 -P 选项才能实现永久设置

```
[root@server0 ~]# getsebool -a | grep samba    #查看所有布尔值

[root@server0 ~]# setsebool samba_export_all_ro on #修改布尔值
[root@server0 ~]# setsebool samba_export_all_rw on #修改布尔值
[root@server0 ~]# getsebool -a | grep samba
```

### 6. 修改本地权限

```
[root@server0 ~]# setfacl -m u:harry:rwX /devops
```

### 在 desktop0 上访问共享文件夹

- 1) 访问 server0 上发布共享文件夹, 实现开机自动挂载到 /mnt/smb
- 2) 实现对 /mnt/smb 目录可以读和写

#### 1 创建挂载点

```
[root@desktop0 ~]# mkdir /mnt/smb
```

## 2 所需软件包:cifs-utils

```
[root@desktop0 ~]# yum -y install cifs-utils
[root@desktop0 ~]# vim /etc/fstab

//172.25.0.11/devops /mnt/smb cifs
defaults,user=harry,pass=123,_netdev 0 0

[root@desktop0 ~]# mount -a
[root@desktop0 ~]# df -h
[root@desktop0 ~]# touch /mnt/smb/test.txt
```

### 案例 24: 多用户 Samba 挂载 (了解)

在 desktop0 上访问 server0 提供的共享 devops

- 1) 将此共享永久挂载在 /mnt/dev 目录
- 2) 挂载时以用户 kenji 作为认证
- 3) 必要的时候, 任何普通用户都可以通过用户 harry 来临时获取写的权

```
[root@desktop0 ~]# mkdir /mnt/dev
[root@desktop0 ~]# vim /etc/fstab

//172.25.0.11/devops /mnt/dev cifs
defaults,user=kenji,pass=123,_netdev,multiuser,sec=ntlmssp 0 0 #修改挂载
点, 修改挂载参数

[root@desktop0 ~]# umount /mnt/smb

[root@desktop0 ~]# mount -a
[root@desktop0 ~]# df -h
[root@desktop0 ~]# su - student
[student@desktop0 ~]$ cifscreds add -u harry 172.25.0.11 #提交新的身份
Password:

[student@desktop0 ~]$ ls /mnt/dev/
[student@desktop0 ~]$ touch /mnt/dev/haha.txt
[student@desktop0 ~]$ exit
```

### 案例 25: 普通 NFS 共享的实现

- 在 server0 上配置 NFS 服务

- 1) 只读的方式共享目录 /public, 只能被 172.25.0.0/24 网段中的客户机系统访问

- 在 desktop0 上访问 NFS 共享目录

- 1) 将 server0 的 /public 挂到本地 /mnt/nfsmount
- 2) 这些文件系统在系统启动时自动挂

## 虚拟机 Server0

```
[root@server0 ~]# vim /etc/exports
/public 172.25.0.0/24(ro)
[root@server0 ~]# mkdir /public
[root@server0 ~]# cp /etc/passwd /public/pass.txt

[root@server0 ~]# systemctl restart nfs-server
[root@server0 ~]# systemctl enable nfs-server
```

## 虚拟机 Desktop0

```
[root@desktop0 ~]# mkdir /mnt/nfsmount
[root@desktop0 ~]# vim /etc/fstab

172.25.0.11:/public /mnt/nfsmount nfs defaults,_netdev 0 0
```

### 案例 26:在在 server0、desktop0 上操作

#### - 将防火墙默认区域设置为 trusted

```
[root@server0 ~]# firewall-cmd --set-default-zone=trusted
[root@desktop0 ~]# firewall-cmd --set-default-zone=trusted
```

### 案例 27:为虚拟机 server 配置 Web 服务, 实现网站的部署

#### - 实现客户端访问 172.25.0.11 网页内容为 环太平洋之雷霆再起

```
[root@server0 ~]# echo 环太平洋之雷霆再起 > /var/www/html/index.html
[root@server0 ~]# systemctl restart httpd
[root@server0 ~]# firefox 172.25.0.11
```