1. 基础概念

块级别：提供块设备（磁盘）

文件级别：提供共享文件夹

DAS：直连式存储。本地磁盘

NAS：网络附加存储。本质上就是共享文件夹

SAN：存储区域网络。

FC SAN：基于光纤的存储区域网络

1. 配置iSCSI
2. 准备三台虚机

vh01.tedu.cn 192.168.4.1 存储服务器

Vh02.tedu.cn 192.168.4.2 mysql1

Vh03.tedu.cn 192.168.4.3 mysql2

[root@room8pc16 nsd1802]# clone-auto7

1. 初始化三台虚拟机：ip地址、主机名、selinux、防火墙、yum

配置IP地址的方式可以使用nmtui

TUI：Text User Interface 文本用户界面

GUI：Graphical User Interface 图形用户界面

CLI：Command Line Interface 命令行界面

1. 为客户端提供存储，需要给存储提供个名字。这个名字推荐使用iqn名称

iqn = iqn.时间.域名的反写.字符串:子字符串

如iqn.2018-06.cn.tedu.nsd1802:teacher

如果为全国200多同学都提供存储，就需要有200多不一样的名字。

1. 在vh01上安装存储端软件包

[root@vh01 ~]# yum install -y targetcli.noarch

1. 为存储端准备硬盘。虚拟机可以在不关机的情况下，直接加硬盘。
2. 创建vdb1作为共享存储的远程磁盘

[root@vh01 ~]# parted /dev/vdb

(parted) mklabel gpt

创建一个主分区，起始位置是1M，结束位置是整个硬盘的10％容量部分

(parted) mkpart primary 1M 10%

(parted) mkpart primary 10% 20%

(parted) print 查看

(parted) quit

[root@vh01 ~]# lsblk

1. 配置target存储

[root@vh01 ~]# targetcli

创建后端存储

/> /backstores/block create formysql /dev/vdb1

创建IQN对象

/> /iscsi create iqn.2018-06.cn.tedu.nsd1802

向IQN对象中添加存储vdb1

/> ls

/> /iscsi/iqn.2018-06.cn.tedu.nsd1802/tpg1/luns create /backstores/block/formysql

授权客户端访问，如客户端iqn名字是iqn.2018-06.cn.tedu.vh02

/> /iscsi/iqn.2018-06.cn.tedu.nsd1802/tpg1/acls create iqn.2018-06.cn.tedu.vh02

绑定地址（可以省略，因为默认已配置好）

/> /iscsi/iqn.2018-06.cn.tedu.nsd1802/tpg1/portals/ create 0.0.0.0

存盘

/> saveconfig

/> exit

8、启动服务

[root@vh01 ~]# systemctl start target

[root@vh01 ~]# systemctl enable target

1. 配置iSCSI客户端
2. 安装软件包

[root@vh02 ~]# yum install iscsi-initiator-utils

1. 修改客户端的iqn名称

[root@vh02 ~]# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-06.cn.tedu.vh02

1. 发现服务端

[root@vh02 ~]# man iscsiadm 查看帮助

[root@vh02 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.4.1 --discover

此时，可以发现存储的iqn名称，但是执行lsblk是看不到新磁盘的。

1. 登陆target。可以通过man帮助中看到的命令执行，也可以执行以下操作

[root@vh02 ~]# systemctl restart iscsi

[root@vh02 ~]# lsblk 出现新硬盘

1. 使用iSCSI磁盘
2. 分区、格式化

[root@vh02 ~]# parted /dev/sda

(parted) mklabel gpt

(parted) mkpart primary 1M 100%

(parted) quit

[root@vh02 ~]# mkfs.ext4 /dev/sda1

1. 安装mariadb

[root@vh02 ~]# yum install -y mariadb-server

1. 挂载再启服务

[root@vh02 ~]# mount /dev/sda1 /var/lib/mysql/

[root@vh02 ~]# chown mysql.mysql /var/lib/mysql/

[root@vh02 ~]# systemctl start mariadb

1. 测试
2. 创建数据库，并写入数据

[root@vh02 ~]# mysql -uroot

MariaDB [(none)]> CREATE DATABASE nsd1802 DEFAULT CHAR SET utf8;

MariaDB [(none)]> USE nsd1802;

MariaDB [nsd1802]> CREATE TABLE students(name varchar(20));

MariaDB [nsd1802]> INSERT INTO students VALUES ("李洋");

MariaDB [nsd1802]> SELECT \* FROM students;

1. 停止mariadb服务，卸载sda1（模拟vh02宕机了）

一定要将共享存储卸载，不能vh02和vh03同时挂载共享存储。因ext4/xfs是单节点文件系统，如果多个节点同时挂载，就会损坏文件系统，使得数据丢失。

1. 在vh03上挂载共享存储，安装数据库，让它为客户机提供服务。

[root@vh02 ~]# shutdown -h now

[root@vh03 ~]# yum install -y iscsi-initiator-utils

[root@vh01 ~]# targetcli

/> /iscsi/iqn.2018-06.cn.tedu.nsd1802/tpg1/acls/ create iqn.2018-06.cn.tedu.vh03

/> saveconfig

[root@vh03 ~]# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-06.cn.tedu.vh03

[root@vh03 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.4.1 --discover

[root@vh03 ~]# systemctl restart iscsi

[root@vh03 ~]# yum install -y mariadb-server

[root@vh03 ~]# mount /dev/sda1 /var/lib/mysql/

[root@vh03 ~]# systemctl start mariadb

[root@vh03 ~]# mysql

MariaDB [(none)]> use nsd1802;

MariaDB [nsd1802]> select \* from students;

Udev：动态管理设备文件的方法

1. Linux中一切皆文件。
2. Udev管理设备文件。如，主机连接U盘出现新的文件（sdb），移除U盘，磁盘文件消失。
3. 硬件文件的信息写入到了/sys目录下，这个目录与proc类似。
4. 我们用udev可以为设备改名，或执行自定义操作。
5. 在/etc/udev/rules.d目录下创建规则文件
6. 接入新设备，如果满足规则，就会按管理员的规则应用
7. 规则文件名：数字-名字.rules
8. 规则文件中，管理员指定硬件满足或不满足哪些条件，就执行什么操作
9. 实例：U盘接入主机，起名为udisk
10. 找到U盘在/sys目录中的说明路径

[root@room8pc16 ~]# udevadm info --query=path --name=/dev/sdb1

/devices/pci0000:00/0000:00:14.0/usb1/1-5/1-5:1.0/host5/target5:0:0/5:0:0:0/block/sdb/sdb1

1. 根据上一步的路径，查询U盘的硬件信息

[root@room8pc16 ~]# udevadm info --query=all --attribute-walk --path=/devices/pci0000:00/0000:00:14.0/usb1/1-5/1-5:1.0/host5/target5:0:0/5:0:0:0/block/sdb/sdb1

出现的内容非常多，只需要找到跟U盘相关的内容即可，如DRIVERS=="usb-storage"

1. 编写规则文件

[root@room8pc16 ~]# vim /etc/udev/rules.d/90-udisk.rules

ACTION=="add", KERNEL=="sd[a-z]\*", DRIVERS=="usb-storage",

SYMLINK+="udisk%n"

将设备连到主机，ACTION（行为）就是add，KERNEL的意思是没有人为干预内核给它起的名。SYMLINIK是创建快捷方式，+=是额外增加。%n是分区号

（4）重新接入U盘即可

NFS：网络文件系统，用于Linux间共享。

1. 实现只读共享
2. 配置vh01为服务器
3. 装包

[root@vh01 ~]# yum install -y nfs-utils

1. 创建共享目录

[root@vh01 ~]# mkdir -pv /nfsroot/nfsro

[root@vh01 ~]# cp /etc/hosts /nfsroot/nfsro/

1. 在配置文件中声明共享目录

[root@vh01 ~]# vim /etc/exports

/nfsroot/nfsro \*(ro)

1. 启动服务

[root@vh01 ~]# systemctl restart nfs

1. 客户端使用共享
2. 查看

[root@vh03 ~]# showmount -e 192.168.4.1

1. 挂载

[root@vh03 ~]# mount 192.168.4.1:/nfsroot/nfsro /mnt/sharero

[root@vh03 ~]# cat /mnt/sharero/hosts

1. 创建读写共享
2. 配置vh01为服务器

（1）服务器上创建共享目录

[root@vh01 ~]# mkdir -m 777 /nfsroot/nfsrw

不管是NFS还是SAMBA，还是FTP，只要是读写，必须设置本地权限和配置文件的权限

1. 声明读写目录

/nfsroot/nfsrw 192.168.4.\*(rw,sync) ->sync表示同步读写

1. 重新输出共享目录

[root@vh01 ~]# exportfs -rv

1. 配置vh03为客户端

[root@vh03 ~]# mount 192.168.4.1:/nfsroot/nfsrw /mnt/sharerw

[root@vh03 ~]# echo 'hello world' > /mnt/sharerw/hi.txt

[root@vh03 ~]# cat /mnt/sharerw/hi.txt

多路径multipath

1. 用途

如果客户端到服务器存储只有一条线路，那么该线路出现故障，存储就不可用了。可以再加一条线路。这个时候，客户端就会从两条线路发现同一存储设备，会给这个存储起两个名字，如sda和sdb。

客户端无论使用哪个存储都不合适。我们可以创建一个虚拟磁盘，如mpatha，客户端挂载mpatha，只要底层不要两条路径全部出现故障，mpatha一直可用。原理参考HSRP。

1. 配置多路径
2. 在vh01和vh03之间加上192.168.2.0/24网络
3. 在vh03上通过192.168.2.0网络也发现一次共享

[root@vh03 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.2.1 --discover

[root@vh03 ~]# systemctl restart iscsi

[root@vh03 ~]# lsblk 此时本地有了sda和sdb

1. 安装多路径软件包

[root@vh03 ~]# yum install -y device-mapper-multipath

4、sda和sdb是同一设备，他们的wwid就是相同的

获得磁盘的wwid：

[root@vh03 ~]# /lib/udev/scsi\_id --whitelisted --device=/dev/sda

3600140589282e00db104fc29d84334a8

1. 生成多路径配置文件

[root@vh03 ~]# mpathconf --user\_friendly\_names n

1. 在配置文件里声明获得wwid的方法，并且声明WWID是3600140589282e00db104fc29d84334a8的设备就是mpatha

[root@vh03 ~]# vim /etc/multipath.conf

defaults {

user\_friendly\_names no

find\_multipaths yes

}

multipaths {

multipath {

wwid "3600140589282e00db104fc29d84334a8"

alias "mpatha"

}

}

1. 启动多路径服务，就可以在/dev/mapper下看到mpatha设备了

[root@vh03 ~]# systemctl start multipathd

[root@vh03 ~]# systemctl enable multipathd