尚硅谷视频-云计算-shell编程

1. Shell脚本主要是简化工作流程，面向过程的脚本语言。如果是高级的应用，需要python脚本，面向对象的脚本语言。学会编程思想，需要多练才有自己的编程思想。Shell是命令解释器。
2. Linux是bash，unix是csh。两者基本操作是一致的，部分命令不一致。
3. Echo命令

echo –e “” ；-e 对特殊字符进行转义；-n 在脚本能够使用

若使用echo –e ‘’中的单引号有特殊用法

4 History命令

history -w ； 将当前用户所记录到内存中的命令存入当前用户的home下，home/.bash\_history文件中；或者注销当前用户可存入到home/.bash\_history文件中。History –c 清除内存和home/.bash\_history文件中的命令。

5 ！使用方法

！！；执行上一条命令 ； ！23 ；执行history命令中的23行

6 alias命令

查看命令的别名，比如：alias ll='ls -l --color=auto'；将别名写入到home/.bashrc文件中，并source .bashrc，即.bashrc文件在不用重启系统前提下进行生效。

7 优先级的命令

1）执行绝对或相对路径的命令；2）执行别名；3）执行Bash的内部命令；4）执行$PATH环境变量的命令

8. Bash快捷键

Ctrl+C;终止当前运行 Ctrl+Z 暂停并放入后台 Ctrl+u 清除命令行

9. Bash的标准输入和输出

|  |  |  |  |
| --- | --- | --- | --- |
| 设备 | 设备文件名 | 文件描述符 | 类型 |
| 键盘 | /dev/stdin | 0 | 标准输入 |
| 显示器 | /dev/stdout | 1 | 标准输出 |
| 显示器 | /dev/stderr | 2 | 标准错误输出 |

输出重定向：正确和错误的都会输出到一个文件中

命令 >> 文件 2>&1 ； 命令 >> 文件1 2>>文件2；正确在文件1，错误在文件2

若不按照以上命令设置，会导致错误的命令结果不会输出到文件中

输入重定向：wc < xxx.txt；打补丁或者提取关键词的时候会用该命令

10. 多命令执行

； && ||

命令1 && 命令2 || 命令3 ； 可用于if else语句

11. find和grep的区别，find主要用于查找文件； grep 主要用于查找文件中关键词

12. 通配符用于匹配文件名

13. Bash特殊符号1

‘’ “” `` $()等特殊符号的作用

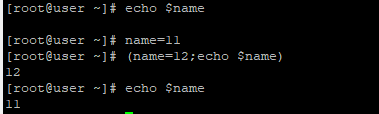
14. Bash特殊符号2

在命令行中写入(命令1)，表示命令1在子bash中执行的命令；

在命令行中写入{命令2}，表示命令2在当前bash中执行的命令；

以上就是小括号和大括号的区别；

Pstree查看进程数



15. 练习题

无需使用service命令或者服务脚本的绝对路径去执行程序。

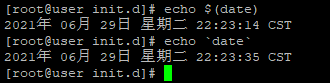
分析:service启动的程序是通过执行/etc/rc.d/init.d文件下的脚本来启动的或者绝对路径启动的程序。因此通过配置$PATH路径来启动程序。结合7的知识点，命令优先级，配置home/.bash\_profile文件中的PATH变量，将/etc/rc.d/init.d/路径配置到PATH变量中。最后source home/.bash\_profile即可生效。

16.Bash的变量和运算符

Shell的变量的默认变量是字符串型，如果是数值计算，则指定变量类型是数值型。

赋值中的变量名=值，等号左右两侧不能有空格。双引号括起来的内容中有特殊字符的就有特殊含义，包括$,!,空格等。单引号括起来的就是普通字符。

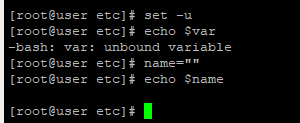
$(命令)或`命令`，把命令的结果值进行赋值。其中里面的命令是系统命令



17.变量分类：

Set 命令可以查询环境变量和自定义变量；env是查询环境变量的

在shell中：



如果使用set –u ，可以识别出空值和未定义值的区别，未定义会报出错误，空值就是空值。

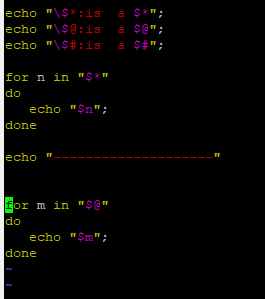
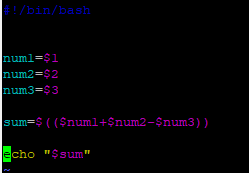
Unset 表示删除自定义变量

用户自定义变量：

环境变量：用户自定义环境变量；export定义的变量是自定义环境变量。

系统自定义环境变量;PATH,若修改PATH的值，修改home/.bash\_profile文件或者PATH=”$PATH”:XXXX修改PATH的值。环境变量在子bash中也可以使用，若只是自定义普通变量只可以在本bash中使用。其中PS1变量中值是’[\u@\h \W]\$’，若用双引号表示值则为” [\u@\h \W]\\$ “，注意\\$，因为双引号中若只是\$，则\表示是转义字符符号，因此需要将这个转义字符变成普通符号，则需要在\前面加上\符号。LANG变量的含义用处，在不同操作系统中值是不一样的。

位置参数变量：用于命令执行的形参。。如果两值相加就是$(())表达式。包括$#，$@等符号的使用。



预定义变量：

$?，$$，$!变量的使用，$?显示上一个命令结束的正确与否

18.键盘输入命令

Read –e –s –n –p

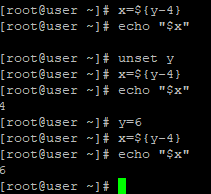
19.declare命令

Declare 用于声明变量可以是整数型或者数组型；-x 声明变量是环境变量；

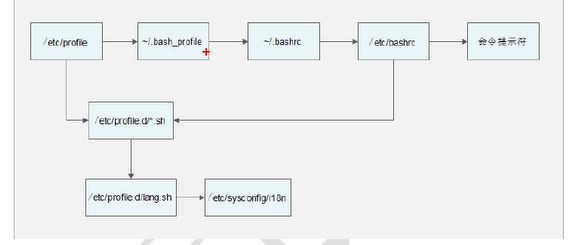
20.shell运算符

$(())运算符是常用的运算符

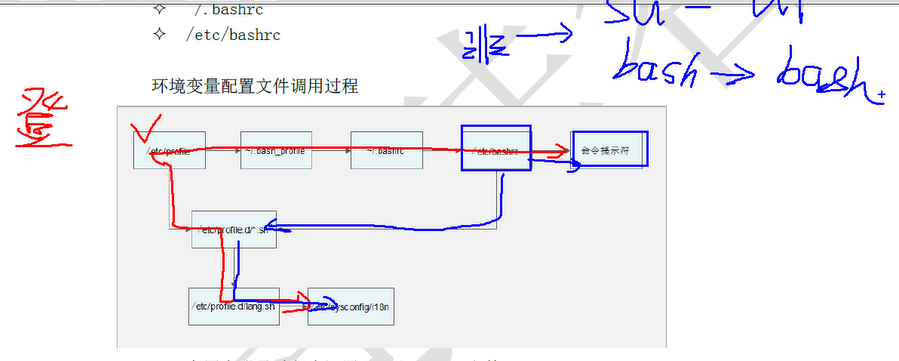
21.变量的测试使用方法，记住x=$(y-new)用于检测y值是否为空或者不存在的方法。



22.环境变量配置文件，配置文件生效命令是source 配置文件; . 配置文件的两种方式，若不是配置文件，使用该命令会出现报错，则需要重启，是其非配置文件生效



首次登录执行的是/etc/profile文件，接着分成两条路（红色路线）执行生效各个文件；如果是非登录的用户，比如su，bash等方式登录则执行蓝色路线的配置文件。配置文件的加载顺序如下图所示



23.其他配置文件。

修改ssh方式登录后的欢迎信息和tty方式登录后的欢迎信息的配置文件。tty的是/etc/issue文件和ssh的/etc/issue.net以及/etc/ssh/sshd\_config文件。如果需要填写登录之后的提示信息，则修改/etc/motd文件，其中有的文件生效需要重启系统即可。注销用户可以生效的文件~./bash\_logout文件。

24.正则表达式用来匹配文本的字符串；通配符用来匹配文本名；

25.grep，sed，awk命令可以使用正则表达式。正则有基础正则和扩展正则，基础正则和扩展正则的命令是有区别的。

26.正则实例：

匹配邮箱，学习使用正则表达式，在使用grep命令检索的时候，使用grep –E去检索；正则主要用于网页端使用，shell端使用量少。能够使用使用正则的命令有grep ,awk,sed。

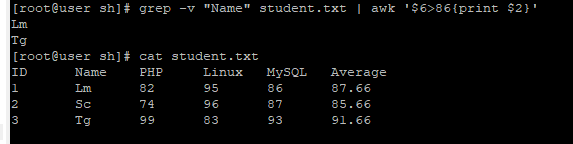
27.cut和grep

grep检索出来的文本是行，cut检索出来的是列，默认分隔符是“TAB”,若分割的是空格，则无法使用cut，则使用awk。Cut的常用命令是cut –f –d 命令，-f是列数，-d 是分隔符。

28.printf

grep | printf这样单独组合失效，只有printf 和awk组合才可生效

29.awk



Awk的原理：虽然展现是列的形式，但实际判断是一行行去判断，每行的数据放入到$0中，每列数据放入到$1，$2…中，命令判断后再输出到相关的$x

awk '$2 ~ /Sc/ {print $5}' student.txt 其中//中填写的是正则表达式，而~表示匹配的含义，$2中是否包含Sc字符串。

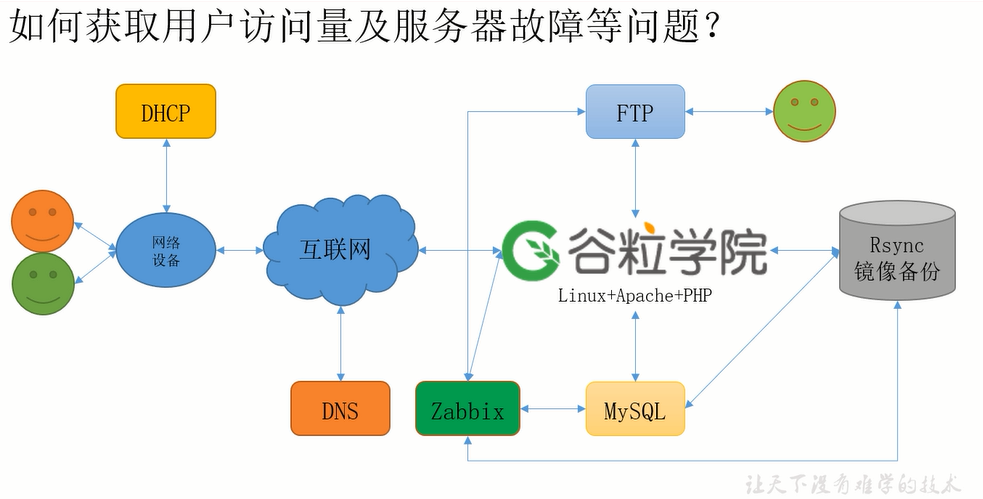
Awk多使用，多练,注意begin的用法，第一个命令出现问题的需要理解awk的原理



尚硅谷视频-云计算-网络服务编程

网络服务的标准：通过各种协议（http/ftp/telnet/smtp）进行网络访问的软件模块

运维后台使用的所有技术



DHCP给局域网内的主机分配IP

DNS解析域名

Zabbix作为监控，查看服务器的运行状况，网站访问量

MYSQL在生产环境配置通常是主从配置

为了保障服务器不被任何人可以访问任何文件和目录，搭建ftp服务，限制非运维人员的上传，删除，下载等任务

如何学习网络服务

每个实验至少搭建2遍，并且写出自己的搭建手册，报错原因记录下来

1. Centos6和Centos7的对比

在centos7中大部分情况也可以使用centos6的命令

Centos7中配置文件可以使用即可使用配置命令进行永久生效，也可修改配置文件进行永久生效；Centos6中只可以通过配置文件进行永久生效

1）CENTOS的主机名：

Centos 6 配置文件是 /etc/sysconfig/network(永久配置)

Centos 7 配置文件是 /etc/hostname(永久配置)

文件系统

Centos6是EXT4;Centos7是XFS

EXT4的单个文件系统(单个分区)容量达到1EB，XFS可以达到8EB

网络设置：

Centos6 网卡名：eth0 Centos7 网卡名：ens33

网络配置命令：

Centos6 ifconfig/setup Centos7 ip/nmtui

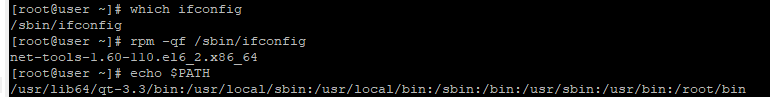
Ifconfig配置临时网络；setup配置永久网络

网络服务：

Centos6 默认使用network服务

Centos6 的which命令用于查看可执行文件的位置。Rpm –qf 查看可执行文件所属于哪个安装包。

Centos7 默认使用NetworkManager服务



修改网络配置的文件：

/etc/sysconfig/network-scripts/eth0文件，记住里面常用的参数配置

Etho HWaddr指的是MAC地址

内核的不同：

从2升级到了3

默认数据库不同:

6是Mysql;7是MariaDB

防火墙不同：

6是iptables；7是firewalld，生产环境通常是firewalld停掉，安装iptables。

时区同步，若配置服务器集群，则需要和时间服务器进行时间同步，若服务器没有连接网络，而时间服务器可以自己搭建去进行时间同步

6是ntpq –p ；修改时区 /etc/sysconfig/clock 修改语言 /etc/sysconfig/il8n

7是chronyc sources ；修改时区 timedatectl set-timezone Asia/Shanghai

服务操作命令

6是service xxx status 查看服务状态 service --status-all 查看所有服务状态

Chkconfig xxx on 设置服务自启动 chkconfig --list 查看所有服务的自启动状态

7是 systemctl start xxxx 启动服务 systemctl disable 服务 设置服务不自启动

以上服务操作命令用于rpm包，但源码包的启动，服务自启动都根据本身情况去使用

2)TCP/IP五层常见协议

应用层协议：FTP/HTTP/SMTP/Telnet/DNS/SSH等

传输层协议：TCP(可靠传输协议)/UDP等

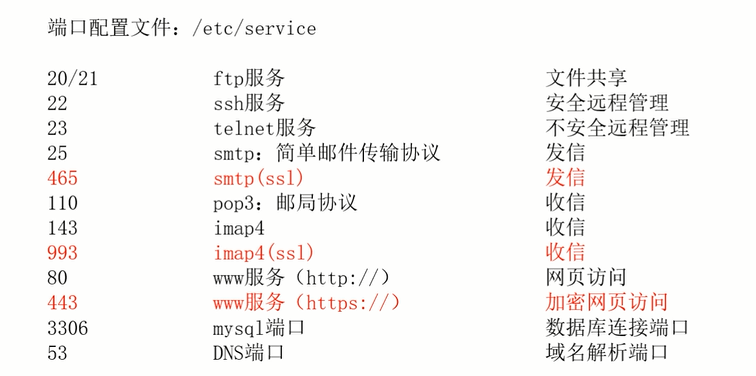
网络层协议：IP/ARP(地址解析协议，IP解析成MAC )/ICMP等

数据链路层：PPP协议,MAC地址

物理层：不常用

其中IP（网络地址）表示的是逻辑网络，在网络层；MAC（物理地址）代表的是物理的硬件地址，在链路层

端口配置文件：/etc/services



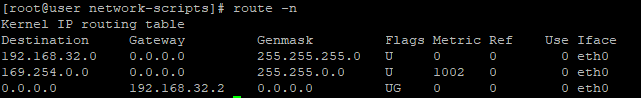
3）交换机和路由器的区别

在局域网内或者同一个网段（192.168.1.1和192.168.1.2）中，可以使用交换机,解析MAC地址找到对方电脑。路由可以跨越不同的网段（192.168.1.2和192.168.2.2），连接不同的网段。

路由：不同的网段，路由选择（网段之间连接的路径选择，网段的走向由路由决定），

网关：不同的网段，路由选择以及默认路由（网段的走向已经手动配置，固定的路径），NAT转化（私有IP通过路由转化成公网IP）。

网关是特殊的路由，网关中有路由表，路由表中是私有地址和网关映射关系，通过网关可以将私有地址和公网连接。比如私有地址是192.168.1.1和192.168.1.2，都和网关地址192.168.1.0做映射，然后192.168.1.0去连接公网地址。查看路由表的命令是route –n



4)DNS的配置  
DNS用于解析服务域名，配置DNS服务地址有114.114.114.114.和8.8.8.8。

配置完成以上的DNS和路由后，然后使用service network service，最后在/etc/sysconfig/network-scripts/eth0文件中查看是否已生效

5）基础网络知识

VM虚拟机使用NAT模式，因主机可以访问公网，因此虚拟机也可以访问公网。

Nslookup 命令是解析域名对应的IP地址，一般情况是通过DNS去解析域名访问网站，该命令可以测试DNS配置是否成功。

若手动配置域名和IP之间的映射配置，需要在/etc/hosts文件中配置。

Netstat 查看系统的网络连接状态，路由信息，接口

-a 查看所有活动连接 –n 以数字形式显示 –t 查看tcp协议相关信息 –p 显示pid -l 监听

Traceroute命令可以查看A到B之间经过的路由信息，但NAT模式不可以使用该命令，桥接模式可以。

Ping 检测网络的互通性

Arp ：将IP解析成MAC地址

Nmap：网络探测命令

2.ssh服务

1）ssh用户登录

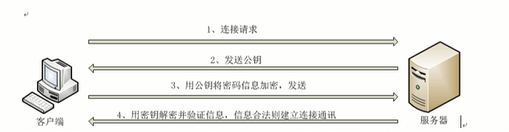
Ssh是通过密文发送信息，而telnet是通过明文发送信息，因此ssh替代了telnet。

因此使用ssh服务建立连接。

2)SSH的账号密码和密钥两种登陆验证方式

账号密码验证：

1. 第一次发送的连接请求，没有发送用户名和密码查看服务器是否在线
2. 服务器会生成密钥对，其中公钥发送给了客户端，私钥自身留下，密钥对是互相解锁的
3. 客户端发送密码，通过公钥进行加密
4. 服务端解锁，建立连接



登陆方式：

Linux系统：Ssh 用户名@ip或者ssh ip

Windows系统：使用xshell工具

密钥对验证：



1. 客户端发送自身的公钥(C-公钥)给服务端；
2. 若C-公钥和S-公钥匹配，则服务器发送S-公钥给客户端，另外用C-公钥加密的challenge发送到客户端
3. C-公钥加密的challenge被C-私钥解锁，再通过S-公钥加密的challenge给服务端。
4. 服务端验证两端的challenge后，则建立连接。

密钥对的登陆验证方式比账号密码登陆的优先级高

3）SSH环境准备

关闭selinux（/etc/selinux/config中将其配置关闭）和清空防火墙规则(iptables -F)并设置防火墙开机不启动（永久不开启防火墙，chkconfig iptables off）

最终重启linux服务器。

4）ssh服务可以设置有端口限制，root禁止登陆，限制ssh监听的IP，禁止密码登陆方式

，以上的配置在/etc/ssh/sshd\_config文件中配置。

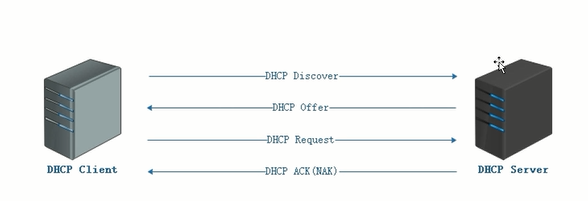
5）ssh衍生的sftp和scp服务

3.DHCP服务

DHCP服务应用于应用层的局域网，属于应用层协议，DHCP报文会在传输层中会封装成UDP数据包进行UPD不可靠协议工作，到了网络层会封装成IP数据包，最终封装成数据链路层的帧进行发送，主要作用于分配网络资源，集中管理。

DHCP工作原理

DHCP客户端因为没有IP，需要发出广播到服务端，服务端发出DHCP offer，允许客户端连接，于是客户端发出请求，最终服务端发出ACK回复或者NAK。



DHCP具体的4部曲：

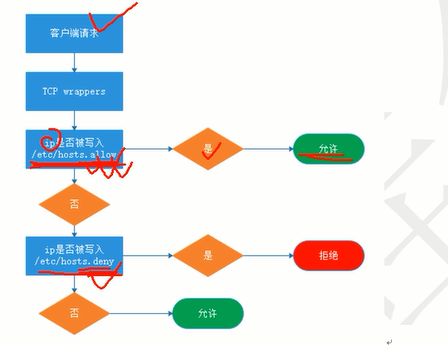
1. DHCP客户端本身没有IP，以IP 0.0.0.0为源地址，255.255.255.255.为目的地址向局域网中开启监听端口为67的DHCP服务端发送广播，发送DISCOVER信息，信息中包含有客户端的MAC地址和主机名。
2. DHCP服务端发现有空闲IP，发送Offer信息，该信息中包含服务端的IP和MAC地址（硬件地址，在局域网中使用MAC地址可找到相应机器）。另外包含客户端需要的ip信息：子网掩码，默认网关，租约期限。因为客户端没有IP，因此不能使用OSI的7层协议去传送数据，所以服务端将自己的IP作为源地址，使用255. 255. 255.255为目的地址，

来广播DHCP OFFER信息来传送到客户端已打开的UDP端口68。在实际生产环境中，客户端收到第一个OFFER信息，就会做出回应。

4．TCP Wrappers（简单防火墙）

TCP Wrappers工作在传输层的安全工具，对TCP的特定服务有访问控制权限，鉴定方式是调用libwrap.so库文件的程序会收到TCP Wrappers的安全控制，常见程序有rpcbind，sshd，telnet。

TCP Wrappers过滤外来访问的工作原理



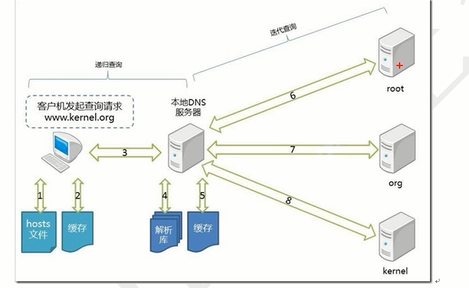
TCP Wrappers的使用主要靠两个配置文件/etc/hosts.allow和/etc/hosts.deny文件对访问进行控制

1. DNS

域名系统服务。

正向域名解析：域名->IP

反向域名解析: IP->域名



尚硅谷视频-数据库

1.关系型和非关系型数据库

2.Mysql的常见数据类型

字符串类型（Char和Varchar类型，Char是固定的长度，Varchar是任意的长度）；

数值类型（int和float类型）

日期和时间类型（Date和Time类型）

3.数据库的约束

主键约束：唯一约束加非空约束

唯一约束：值不唯一，没有重复，但可存在多个null

非空约束：not null

外键约束：两个或多个表之间字段的参考关系

4.mysql的索引

表和索引是同一级别的，索引的出现加快了表的查询速度，原理是将原表中的部分字段信息取出，对部分字段信息进行查询，然后通过部分字段信息中的一个信息找到原表中的相应信息。索引类似于目录。

5.数据库的锁

因为并发的存在，使用事务和锁的搭配，来解决并发问题

6.数据库的存储引擎

Oracle和SqlServer数据库只有一种存储引擎，mysql是插件式存储引擎：

INNODB:支持事务，外键，锁等功能

MYISAM:默认引擎，查询和插入速度快，但不支持事务，锁等功能

MEMORY:数据暂存到内存中，保存数据快

…

7.mysql的安装

需要安装服务端和客户端，安装完成后，mysql的主配置文件是/etc/my.cnf文件。

Mysql的pid在/var/run/mysqld中；数据目录在/var/lib/mysql；套接字文件在/var/lib/mysql/mysql.sock。

8.mysql的备份

1）mysqldump命令：

导出数据库：mysqldump -uroot -p123456 atguigu > ./testguigu.sql；

导入数据库：mysql -uroot -p123456 atguigu < testguigu.sql

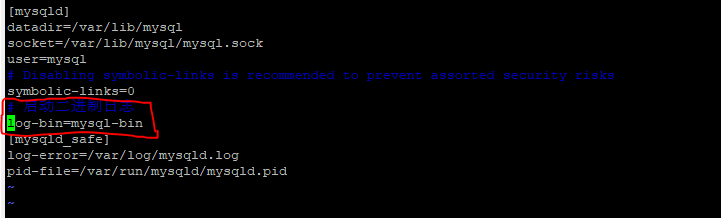
1. mysqlhotcopy命令：

原理是将数据目录的数据库移除导出，再重新导入即可。

1. mysql-bin日志：

启动该日志在/etc/my.cnf中修改：

该日志只可记录增删改信息；



查看mysql-bin日志，mysqlbinlog mysql-bin.000001。

mysqlbinlog --start-datetime '2021-07-24 14:29:00' --stop-datetime '2021-07-24 14:30:30' mysql-bin.000001|mysql -uroot –p

以上通过时间差的方法可以恢复原先丢失的信息

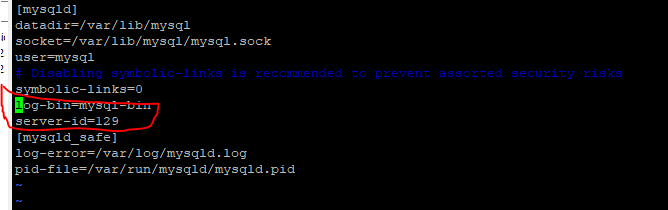
1. mysql基本操作

set password for root@localhost = password('123456');

9.mysql主从备份（master-slave模式）

1）首先原理是mysql-bin日志实现主从服务器mysql的信息同步

2）主从开启mysql-bin日志，在/etc/my.cnf文件中添加如下信息，主的server-id是128



1. 在主服务器上授权，从服务器保存授权的信息
2. 在主服务器上执行

grant replication slave on \*.\* to slave\_user@'192.168.32.129' identified by '123456'

其中\*.\*代表任意库，任意表，slave\_user是从服务器的用户名。

1. show master status，查看主服务器的授权信息
2. 将主服务器的授权信息配置到从服务器，从服务器接受授权，以下命令在从服务器中执行

change master to master\_user='slave\_user', master\_password='123456', master\_host='192.168.32.128', master\_log\_file='mysql-bin.000003', master\_log\_pos=265;

1. 在从服务器中启动授权start slave；
2. 查看授权的相关线程是否打开show slave status\G
3. 从服务器的授权信息在/var/lib/mysql/master.info中可以查看

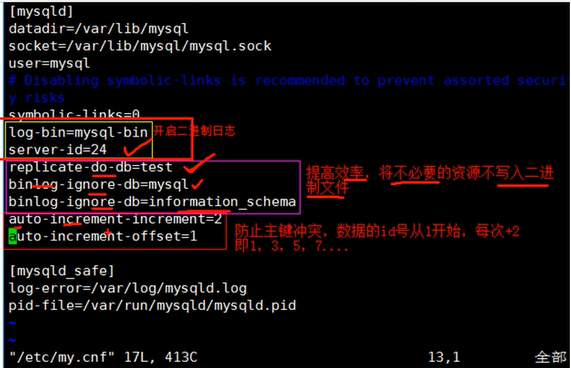
10.mysql的主主备份

主从服务器即实现了备份功能，也实现了在查询数据时候的负载均衡功能。

主主服务器是加快了写入效率，均摊写入的压力。

主从配置的方案是主服务器配置主的，从服务器配置从的；而主主配置的方案是原先的从服务器配置成主的，原先主的配置成从的，这样两个服务器都是主，也是从。

备注：/etc/my.cnf可以优化：



11.mysql的一主多从

一主多从的配置，只不过需要对从服务器的配置多备份几个即可

12.mysql的多主一从

从服务器需要开启多线程服务，并且从服务器的mysql开启配置项需要留意。

不方便实验。

13.mysql的读写分离

Mysql的中间件amaeba可实现。

14.redis

Nosql出现是解决大数据的问题

Redis属于key-value格式的非关系型数据库，主要用于缓存数据使用，比如京东购买商品，为何可以快速查询某物品，是因为部分数据从mysql查询后缓存到了redis数据库，然后这些数据从redis取出，增加了查询效率。

15.集群

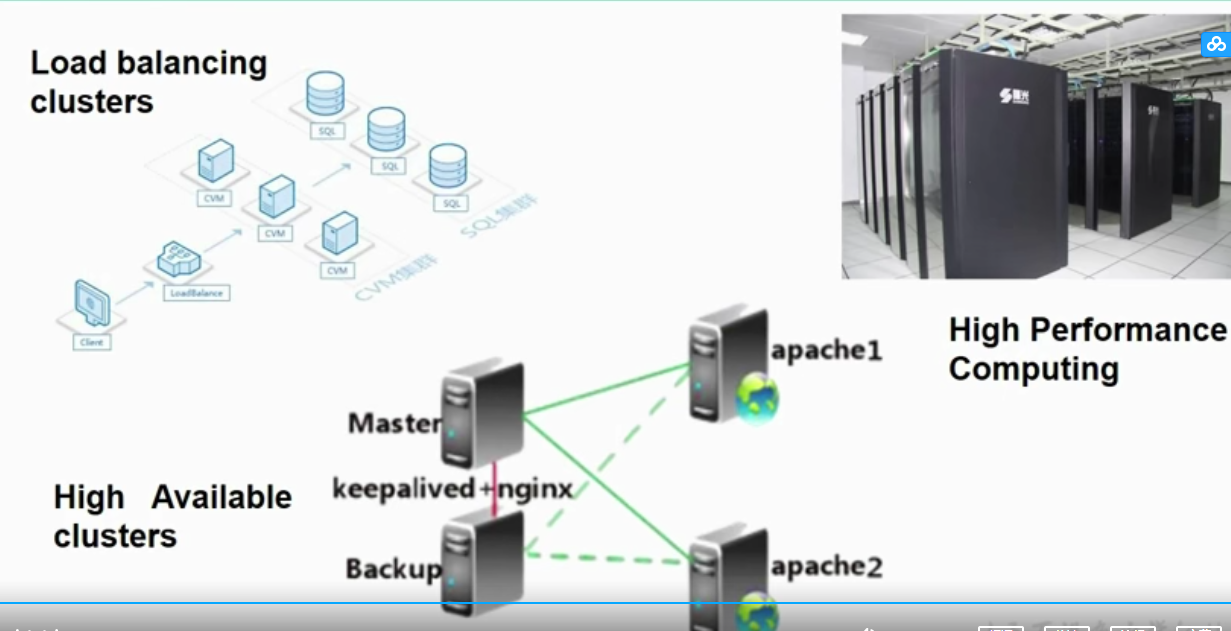
集群分类：高可用和负载均衡以及超级计算机的3大应用技术

负载均衡集群LBC: 分担服务的总体压力（每个子节点任务是一致）

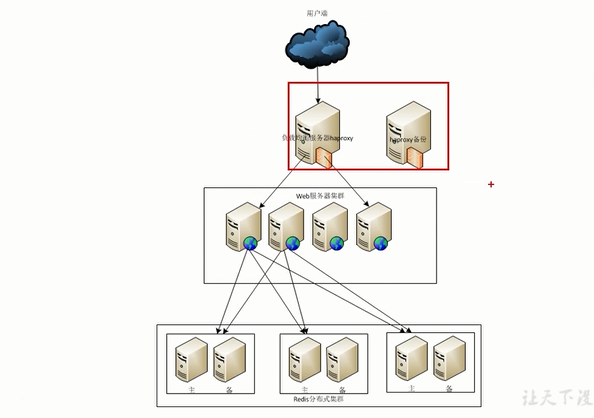
高可用集群HAC：尽可能保障服务状态的可用性

高性能运算集群HPC:提供单台服务器提供不了的计算能力（每个子节点任务不一致）

生产环境的配置是LBC+HAC;HPC+HAC。



16.负载均衡知识点

其中红框中服务器配置是主备份，属于高可用。前两部分的属于负载均衡集群（LVS）的。最后一部分属于存储集群。

16.1分类成软/硬件设备

软件：LVS,NGINX,HAPROXY

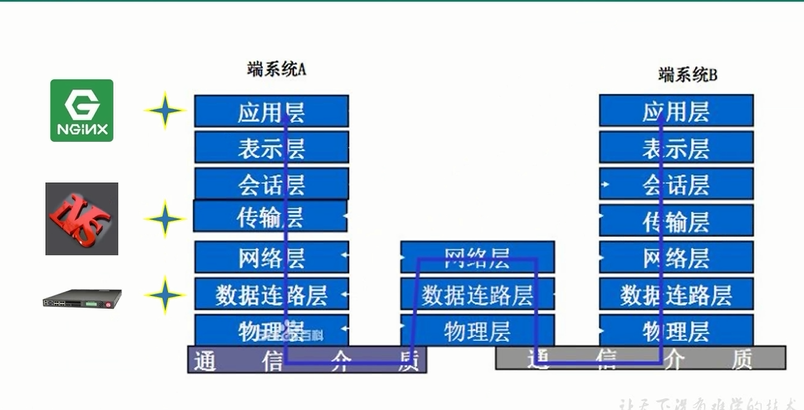
硬件：f5

应用层/表示层/会话层：主机名，应用层http协议->传输层：UDP/TCP，端口，传输协议；->网络层：IP地址，路由选择；->数据链路层：mac地址，交换机负责区域->物理层：bit流信息

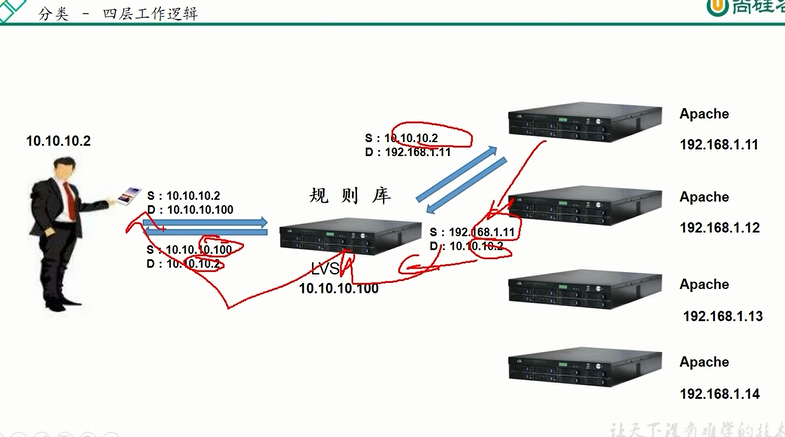
数据链路层：数据报文通过该层发出，F5设备在该层应用。

传输层：LVS和F5可以应用。

应用层：nginx可以应用。

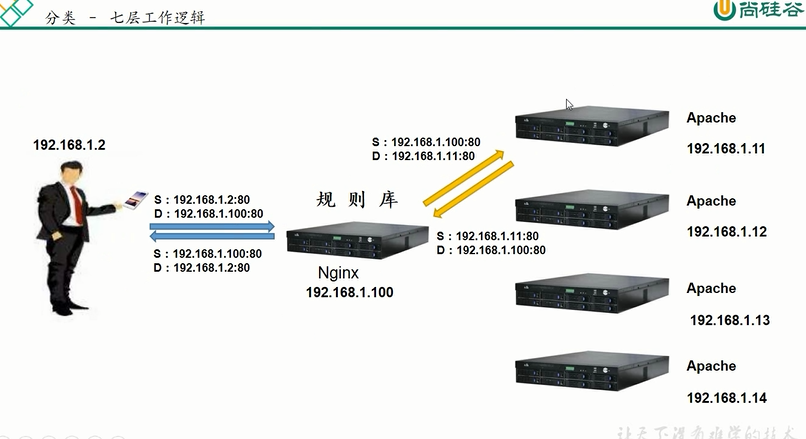


16.2 四层工作逻辑



LVS只是起到转发作用

16.3.七层工作逻辑



Nginx不仅起到转发作用，而且起到了“保存信息的作用”，导致nginx相比于LVS压力更大。

由于nginx应用在应用层，可以看到下层更多的信息，在转发的时候更精确，只可应用于应用层的http协议，B/S架构。Nginx\_plugins却不限于B/S架构。而LVS优势在于负载均衡压力小，可以应用于B/S和C/S架构。



因此负载均衡使用LVS还是nginx，根据实际场景来判断。

17.LVS\_NAT工作模式

