**运维经典面试题**

**1.不能关机的情况下,如果挂载目录卸载不掉应该怎么办**

umount卸载不掉是因为正在被程序读取或者有数据正在写入

使用 fuser 或者 lsof 查看正在占用的磁盘的进程PID

kill -9 PID强制结束进程

umount卸载：如果kill -9无法杀死进程，可以使用umount -f强制进行卸载，但是这种方法是最不推荐的

**2.如果你向文件系统中写入数据，发现无法写入，怎么处理**

此时应该考虑两种情况：

磁盘空间被占满：此时可以删除一些垃圾数据；或者将一些历史数据导出，将磁盘中的 数据进行删除；如果在安装文件系统时，底层使用了LVM，可以进行在线扩容，增加硬 盘容量

inode号被用完：建议导出一些历史数据，然后将硬盘中的数据删除

**3.客户反映服务器卡顿 ，你会怎么排查**

查看服务器的内存和CPU，确定内存大小和CPU核

查看CPU负载，确认负载高是因为CPU的使用、内存的使用还是io的使用引起的

使用 top 命令进行查看

查看数据库:show full processlist; #查看数据库当前正在使用的线程

通过这个命令可以查看当前登录到数据库的用户、用户的状态和执行的sql语句，通过这些能够分析操作和sql语句是否有误

**4.如果你在执行命令时,发现Linux系统命令执行速度很慢,你会怎么解决**

lscpu：查看CPU的相关信息，看CPU的缓存是否过高

free -mh：查看系统磁盘文件缓存是否过大

主要就是，查看系统中的CPU、负载等情况；检查系统是否在执行定时计划任务等；也 可以选择杀死进程，但是杀死进程前需要根据占用过高的进程做进一步的分析

**5. 了解七层模型吗，说一下；TCP协议和UDP协议的区别是什么；二层和三层的交换机有什么区别，说一下**

七层模型： OSI/ISO七层模型

应用层：给用户一个操作界面，为用户提供服务

表示层：为数据提供加密、压缩、表

会话层：确定数据是否需要进入网络进行传输

传输层：对报文进行分组、组装 ；提供传输协议的选择（TCP协议和UDP协议）差错校验

网络层： IP地址编址

数据链路层： MAC地址编址

物理层：数据实际传输和电气特效定义

TCP协议和UDP协议的区别：

TCP协议：面向连接的、可靠的传输协议，主要用来传输字节流

UDP协议：面向无连接的、不可靠的传输协议，主要用来传输数据报文

安全性较差，但是可靠性高； UDP协议因为是无连接的，实时性好，安全性高，可 靠性低

二、三层交换机的区别：

二层交换机：二层使用的是MAC地址寻址，采用存储转发的方式进行数据交换；

三层交换机：三层交换机具有一定的路由转发功能，可以连接外网；还可以选择路 由转发的线路，提高转发的效率

**6.说一下TCP三次握手和四次挥手**

三次握手：

第一次握手：客户端将标志位SYN置为1，随机产生一个序列号Seq，将这个数据包 发送给服务器，然后客户端进入SYN\_Sent状态。

第二次握手：服务器收到客户端发送的数据包后，会把 SYN 和 ACK （确认序号有效）都置为 1 ，将 Ack （确认号）记为 Seq+1 ，然后再随机产生一个序列号 Seq，将数包返回给客户端，然后进入 SYN\_RCVD 状态。

第三次握手：客户端收到数据包后，确认 Ack 为自己产生的序列号 Seq+1 、 ACK=1 ，

将 SYN 和 ACK 置为 1 ，将 Ack 记为服务器端产生的序列号 Seq+1 ，将数据包发送给服务器，服务器端进行确认，建立连接，此时客户端和服务器端进入 ESTABLISHED 状态。

**9.软链接和硬链接有什么区别，怎么做链接**

我们可以使用 ln 制作硬链接，使用 ln -s 制作软链接

软链接：软链接只是一个路径，可以类似于 Windows 系统中的快捷方式，依赖于源文件存在，源文件如果被删除，那么链接也会失效；软链接可以跨区创建；可以对目录创建硬链接：硬链接和源文件具有同一个 inode 号，是同一个文件，修改源文件硬链接也会被修改，删除源文件硬链接仍然可以使用；硬链接不能对目录创建（会引入循环，破坏目录树结构）；不能跨区创建；

**10.Raid知道吗，说 一 下软raid和硬raid的区别，说 一 下常见的Raid的原理**

Raid：磁盘阵列，提供比单个磁盘更多的存储和冗余存储的技术

软raid：由操作系统模拟出的raid，硬盘损坏，raid就会失效

硬raid：由硬件raid卡制作的raid，硬盘损坏不会导致raid损坏，能够保证数据冗余

Raid原理：

Raid0：是一种条带形式的存储，由两块或两块以上的磁盘组成，数据在写入时会以条带的形式写入所有的磁盘，存储速度最快，但是损坏几率最高，没有磁盘容错功能

Raid1：镜像存储，必须由2的倍数块硬盘组成，数据再写入磁盘时会同时复制到另 一块硬盘中，因此写入速度最慢，磁盘利用率只有50%，具有磁盘容错

Raid10：硬盘先组成raid1再组成raid0，中和Raid1和Raid0的优缺点

Raid01：硬盘先组成raid0再组成raid1

Raid5：至少需要三块硬盘，每次写入数据时都会写入一个校验信息，每个校验信 息都会被写入到不同的硬盘下，具有磁盘容错和冗余功能，磁盘利用率为n-1块盘

**11.监管Linux系统资源使用哪个命令**

dstat，默认未安装，需要安装。用来替换vmstat、iostat、netstat、nfsstat和ifstat这些命令 的工具。

**12.CentOS操作系统的启动过程**

CentOS7：开机，加电自检，加载分区表，加载grub菜单（/etc/grub2 ），确定系统的第一个进程（systemd），确定系统的启动级别（/etc/systemd/system/default.t arget），操作系统初始化（/usr/lib/systemd/system/default.target），根据系统启动级别确定系统启动的服务项（/usr/lib/systemd/system,/etc/systemd/system ），加载环境变量（/etc/bashr c,/etc/profile ）

**13.你在工作中如何修改Apache的主页文件；在工作中会用nginx吗，说一下 nginx和apache的区别**

apache在安装完成后，有一个配置文件目录， yum安装在 /etc/httpd/conf/httpd.con f ，源码包安装会在 /usr/local/apache/conf/http.conf ，在配置文件中有 Directory Index 模块，找到后取消注释，在后面添加需要主页文件类型即可。

区别

Nginx是一种轻量级的web服务器，比Apache占用的资源更少

Nginx的工作模式是异步非阻塞模式，而Apache是同步阻塞型的工作模式，所以 Nginx能抗高并发。Nginx有高度模块化的设计，编写模块比Apache简单

**14.说一下代理和负载均衡的区别**

· 代理分为正向代理和反向代理

正向代理：为客户端处理请求，将客户端要请求的数据从后台服务器进行下载然后 返回给客户端。客户端认为自己访问的是网站，实际上访问的是代理服务器的IP

反向代理：通常和负载均衡搭配使用，接收所有客户端的请求，然后将请求交给后 台的服务器进行处理，降低后台服务器的处理压力，后台一般会有多台服务器，通 常会和负载均衡搭配使用

· 负载均衡：负载均衡只有在后台服务器大于等于两台时才会有意义

**15.lvs和nginx都能做负载均衡，那么二者有什么区别**

· lvs工作在四层，只处理请求，没有网络流量；nginx工作在七层，支持http的操作，因此lvs抗压能力比nginx强

lvs的稳定性比nginx高，因为nginx没有现成的双机热备方案

· nginx的配置难度低，只要修改配置文件打开upstream模块； lvs需要搭建集群，还需要 在iptables内写入路由转发规则

· lvs工作在四层，支持的软件比nginx多

· nginx访问失败会返回状态码； lvs会直接断开

· nginx自带健康检查功能，如果后台服务器宕机，那么nginx不会向这台服务器上分发请 求

· lvs本身是不支持正则匹配的，不能做动静分离

· nginx可以跨平台运行， lvs只能在linux操作系统上有运行

· nginx对网络的依赖性比较低，理论上只要能ping通，就能进行访问； lvs对网络的依赖 性比较高

**16.说一下lvs的工作原理**

lvs主要通过ipvs （钩子函数）实现功能，而ipvs主要工作在防火墙的input链上，用户可以 使用ipvsadm对lvs进行管理

**17.介绍一下lvs常见的工作模式**

·DR（路由）负载均衡服务器和后台的真实服务器位于同一局域网内，客户端通过路由器访问负载均 衡服务器，负载均衡服务器会进行MAC地址转换，因为负载均衡服务器和后台服务器位 于同一网络内，可以直接将请求交给后台服务器进行处理，后台服务器会直接把数据返 回给客户端

NAT负载均衡服务器和真实服务器位于同一局域网内，客户端访问负载均衡服务器，负载均 衡服务器会进行目标地址转换，将访问请求交给后台的服务器；当后台服务器返回数据时，负载均衡服务器会进行源地址转换，将数据交给客户端

·TUN（隧道）隧道模式的处理过程和DR模式相似，负载均衡服务器只会处理入站的报文，会将请求 交给距离你最近的服务器进行处理，然后服务器直接返回数据，提高了访问速度

**19.说一下Apache的三种工作模式**

· prefork：单进程单线程模式，同一时间内只能处理一个请求。优点是运行稳定，不需 要担心线程安全的问题；但是处理请求时单线程独占资源空间，会造成资源的浪费

· worker：单进程多线程模式，在同一时间内能够同时处理多个用户请求

· event：单进程多线程模式， event要比worker模式优秀，因为event工作模式下允许线 程在处理完请求后将资源释放去继续处理另一个请求

**20.说一下Nginx的工作原理（ Nginx为什么消耗资源少，并发能力强）**

异步非阻塞：即当系统内没有资源处理请求时，它会将请求挂起并注册一个事件，然后将 这部分资源释放掉，当系统有多余的线程时再来处理挂起的请求，这样能够实现在资源固 定的情况下处理更大的并发请求

**21.Nginx和Tomcat怎么进行数据传输**

如果是处理静态资源，Nginx直接处理返回给用户

如果处理动态资源， Nginx会将请求交给后台的Tomcat服务器进行处理，然后返回给用 户

**23.介绍一下Nginx的常用模块**

· http\_ssl\_module：实现加密传输的模块

· http\_proxy\_module：实现反向代理

· http\_upstream\_module：负载均衡模块，通常和proxy模块搭配使用

· http\_rewrite\_module：重写模块，实现域名重写

**24.说一下Nginx做反向代理时常用的算法有哪些**

· rr ：轮询算法，将用户的请求平均的分配到后台的服务器上，如果后台服务器宕机，会 自动将宕机服务器清除

· wrr ：加权轮询，根据权重分配请求，权重越大，处理的请求的越多

· ip\_hash：根据请求的hash值进行分配， hash值相同的会被分配到同一台机器上

fair：根据后台服务器的响应时间进行分配，响应时间短的优先分配，必须下载nginx的 upstream\_fair模块才能使用

· least\_conn：哪台服务器的连接数最少，就分配到那台服务器上

**25.说一下你知道的Nginx的特性**

·跨平台：可以在Linux和Windows上运行

·配置简单

·异步非阻塞的工作模式，能抗高并发

网络依赖性低，理论上只要能ping通，就能进行负载均衡，而且能够有效地区分内外网 流量

· 自带健康检查，如果后台有一台服务器宕机， Nginx就不会向这台服务器上分配请求

· 消耗内存少，稳定性高

**26.Nginx常用的参数优化**

· 隐藏版本信息

· 修改上传文件限制

· 控制客户端请求Nginx的速率

· 控制并发的连接数，防止ddos攻击

· 修改Nginx的默认用户

· 防盗链

· 开启ssl模块，使用https进行访问

· gzip压缩

· 动静分离， server端添加location模块

**27.说一下location模块（ 路由功能） 的匹配优先级**

· 首先进行精确匹配 =

· 其次进行前缀匹配 ^~

· 按文件中的顺序进行正则匹配

· 不带任何前缀的正则匹配

· 最后进行通用匹配

· 只要有一条匹配成功，就停止匹配

**28.解释一下数据库的主从复制的原理（ 或者会问怎么做的数据库的数据同步）**

· 从服务器会向主服务器提交主机偏移位

· 从服务器的io线程会主动连接主服务器的io线程，主服务器会通过io线程将bin-log日志推送给从服务器，然后进入休眠状态

· 从服务器的io线程会唤醒sql线程，将bin-log日志转化为存储日志，进行数据同步，然 后线程会进入休眠状态，等待下一次同步

**29.数据库主从不同步的原因**

网络延迟：数据库主从同步是基于binlog日志的异步复制，而binlog日志是通过网络进 行传输的，网络延迟大时，可能会造成主从不同步

·主从数据库负载不一致：主从同步时，主数据库需要一个io线程，而从服务器需要一个 io线程和一个sql线程，当一台服务器负载过大时，会导致提供给线程的资源不足，主 从同步就无法完成

·MySQL异常宕机

·主从数据库设置的 max\_allowed\_packet 不一致，当主库执行一个较大的sql语句，从库无法成功执行，导致数据不同步

·可能在数据同步过程中发生了bug

· MySQL主从数据库的版本不一致，如果主数据库的版本较高、从数据库版本较低，也会导致同步不成功

**30.解释一下数据库的读写分离**

如果被这么问到不仅要解释什么是读写分离，还要说出数据库主从复制的原理 ，切忌只说读写分离，回答一 定要完整

· 读写分离就是，主服务器负责写入数据，后台的一台或多台从服务器提供读操作

· 当主数据库写入数据时，从服务器会将数据同步过去

· 解释主从同步的原理

**31.如果数据库主库宕机了，怎么切换到从库，如果有多个从库，其余的从库如何处理**

· 登录所有的从库，执行show slave status\G；查看Pos信息，选择Pos值最大的从库作为主库，执行stop slave命令

· 修改my.conf配置文件，开启bin-log日志，然后重启数据库，登录数据库执行restart master，修改对应服务器IP信息

· 登录其他从库，执行change master，查看状态

**32.解释一下事务的四个特征，和什么是数据库的回滚**

· 事务的四大特征：

持久性：对数据库的所有操作被保存后将不能进行回滚

原子性：对事务的操作要么全部被执行，要么全部都不执行

一致性：事务的开启前后，数据库的完整性没有被破坏

隔离性：同一时间内，同一个事务只允许一个请求；不同的事务不会受到干扰

· 数据库的回滚：对数据库进行操作后，如果操作过程中发生错误，那么事务将回到修改之前的状态

**33.MySQL数据库的备份方式**

· 增量备份

· 异地备份

· 重要的数据库会在设计架构时，搭建一个一摸一样的数据库做完整备份·通过命令行备份：mysqldump -uroot -p 数据库名 > \*.sql文件

· 通过Rsync+Inotify进行冷备

· 设计架构时，会搭建数据库的集群，进行热备

**34.MySQL的历史了解吗，它和Mariadb有什么关系**

Mariadb是MySQL5.7版本之后的一种叫法，因为原来的MySQL在2009年被Oracle公司收购 后，变成了收费版本， MySQL开发者重新开发了Mariadb作为开源项目继续进行更新和维护

**35.数据库常用的引擎有什么，说一下区别**

MYISAM：性能优异，查询速度快；但是不支持事务、锁、外键约束等高级功能

INNODB：支持事务、锁、外键约束等高级功能

MEMORY：将数据储存在内存中，方便快速读写

**36. 了解非关系型数据库吗，和MySQL有什么区别**

MySQL属于关系型数据库，和非关系型数据库的区别主要体现在以下几个方面：

存储方式：

关系型数据库中的数据都是存储在表格中的，表与表之间能够通过能进行关联协作，在提取数据时很方便；非关系型数据库是通过键值对进行存储的

存储结构：

关系型数据库中，数据存储时的结构都是预定义好的，结构比较稳定，但是修改数据很困难；非关系型数据库使用的是动态结构，很容易适应数据结构和内容的变化

查询方式：

关系型数据库通过SQL语言进行数据查询，使用标准化的语言进行数据查看；非关系型数据库的查询语言是非标准化的，使用的是UNQL

扩展

关系型数据库的数据存储在多个表中，涉及到多表联合查询，需要提升服务器的硬件资源来进行扩展，是一种纵向扩展；非关系型数据库的存储是分布式的，可以向地址池中增加服务器来扩展性能，是一 种横向扩展

**37.解释一下数据库的锁机制**

数据库的锁机制是一种对数据的保护机制，因为数据库支持并发，所以可能存在同一时间 内多个用户对同一个数据进行修改，这可能会破坏数据的一致性；当用户对数据进行修改 时，系统会先对这个数据进行加锁，这就保证了事务对这个数据的控制，当这个数据被修 改完成之前，其他的事务不能对这个数据进行修改，保证了数据的一致性。

**38.说一下高可用负载均衡集群，原理是什么**

如果被问到keepalived如何解决单节点故障 ， 就回答原理

· 使用LVS做负载均衡集群，使用keepalived做高可用集群

· LVS负载均衡集群

LVS主要采用的是基于IP的负载均衡技术，它由两个核心组件构成： ipvs （钩子函 数）、 ipvsadm；其中， ipvs是LVS的核心组件，是一个框架，而ipvsadm是一个管 理命令，用来定义路由转发规则，工作在用户空间中

· keepalived高可用集

keepalived基于VRRP（虚拟路由冗余协议）来实现高可用功能， keepalived可以监 控集群中各个节点的功能是否正常，如果节点异常或者服务故障， keepalived会将 故障节点从集群中清除出去，在故障节点恢复正常后， keepalived会将节点重新拉 入集群中。

VRRP（虚拟路由冗余协议）：是为了解决静态路由状态下出现的单节点路由故障问 题。 VRRP实际上是一个虚拟路由器集群，这个集群工作在实际的物理路由器之上， 这些路由器中有一个master路由器，其余的路由器都是backup级别；只有master路 由器不断的进行vrrp数据包的发送，而backup路由器只负责数据包的接收，同时负 责监控master的运行状态，因此不会发生抢占的现象。如果master路由器出现故障， backup路由器无法接收到来自master服务器的数据包，那么就会认定master服 务器故障，接着剩下的所有backup路由器会进行选举，优先级最高的backup会成为 master，这个过程非常短，能够保证服务的持续性。

**39.什么是DNS，说一下DNS的解析过程，以及使用的端口，解释 一 下智能DNS（分离解析）**

· DNS：域名解析系统

· DNS解析过程：

首先查看本地hosts文件， hosts文件无法解析则查看本地缓存，本地缓存无法解析 则请求DNS服务器进行域名解析，如果是本地解析直接返回解析记录。这是DNS的递归

首先请求根域服务器，返回一级域服务器的IP地址，然后请求一级域服务器返回二级域服务器的IP地址，这是DNS的迭代

请求三级域服务器返回域名地址。 DNS服务器将解析记录在本地记录后，将解析记录发送给客户端，完成访问。这是DNS的递归

递归：直接返回域名地址；迭代：告诉你下一步怎么走，在这里就是获取DNS服务器地址

· 使用TCP/UDP53端口

· 智能DNS：DNS服务器会根据用户发起的查询自动判断用户所属的运营商，将请求交给相应的运营商进行解析，减少访问时间，提高访问速度

**40.怎么添加DNS解析**

vim /etc/resolv.conf

**41.说一下常用服务的端口号**

· DNS：TCP/UDP53端口

· DHCP：UDP67/68端口

· http：80

· https：443

· FTP：20/21

· ssh：22

· telnet远程连接： 23

· SMTP邮件协议： 25

· pop3：110

· IMAP：143

· Tomcat：8080

· php-fpm：9000

· mysql：3306

**42.路由器的两种工作模式， 怎么获取动态路由（DHCP原理）**

· 静态路由static

· 动态路由DHCP（动态主机配置协议）

· DHCP租约过程

首先客户端在局域网内广播发送DHCP\_Discover包，寻找DHCP服务器

DHCP服务器响应，发送DHCP\_Offer包

客户端选择IP，发送DHCP\_Request包

服务器确定租约，发送DHCP\_ACK包

**43.说一下常见的网络状态码**

http状态码主要分为5类：

· 1\*\* ：服务器收到请求，需要请求者继续进行后续操作

· 2\*\* ：操作被成功接收

· 3\*\* ：重定向，需要进一步操作完成请求

· 4\*\* ：客户端错误，主要指语法错误或者无法完成请求

· 5\*\* ：服务器端错误，服务器在处理请求的过程中出现错误

常见的错误状态码：

· 400：服务器不理解请求的语法

· 401：未授权，请求进行身份验证，登录需要授权的网站可能返回此提示 · 403：服务器拒绝请求，可能是服务器或主机拒绝了访问

· 404：服务器找不到页面，访问服务器上不存在的页面会返回这个提示 · 405：客户端请求中的方法被禁用

· 500：服务器内部错误，无法完成请求

· 502：服务器充当网关或者代理，从上游服务器收到了无效响应

· 503：服务器超载或暂停出现的状态码，通常只是暂时的

· 504：服务器充当网关或代理，从上游服务器收到的响应超时

· 505：服务器不支持http协议的请求，无法完成处理

**44.解释一下502是什么意思， 说 一 下你的排错思路**

·502：服务器充当网关或者代理，从上游服务器收到了无效响应。简单的说就是网关错误，web服务器通信失败

· 排错思路

因为是和web服务器通信失败，所以可以先检查网络，看网络是否畅通也可能是因为web服务器未启动，检查web服务器进程是否在运行状态，端口是否开放web服务器处理的请求太多，无法响应这个请求，检查web服务器的日志来定位错 误原因

**45.解释一下http协议，和https协议有什么区别**

· http协议：超文本传输协议，基于的是TCP/IP协议，是一种无连接（限制每次连接只处 理一个请求，处理完成后立即断开）、无状态（服务器端对数据没有记忆能力）的协 议，只能由客户端向服务器端请求数据，服务器端不会主动发送数据

· 区别

http是明文传输，容易被攻击， https是http协议+ssl加密组成，确保数据传输过程是 加密进行的

http是无状态连接， https的连接过程是加密过的

**46.SSH远程无法连接 ，说一下你的检查思路**

主要从四个方面进行分析：网卡，端口， IP，防火墙

· 检查客户端网线连接是否正常，重新插一下网线，重新尝试远程连接

· 去机房检查网络连接、网卡是否正常，检查IP地址

· 检查防火墙是否开启，如果开启防火墙，查看防火墙规则中是否添加了远程连接的端口

· 查看IP安全策略中是否添加了需要远程连接的网段

**47.说一下Linux系统中查看系统资源都能使用哪些命令**

· netstat：能够查看进程和端口

· top 、htop：查看系统负载； htop比top更加人性化，支持鼠标点击

· ps aux：查看所有进程，能够显示PID

· ps ef：查看使用的终端

· free：查看系统内存的使用情况

· w：查看系统负载，登录系统的用户和持续登录时间

· uptime：查看系统负载

**48.列出Linux中常用的文本处理命令**

· cat，less ，more：查看文件内容； less和more支持翻页

· head：使用-n选项可以指定查看文件的前n行

· tail：动态查看文件内容

· grep：行匹配命令

· cut：列提取命令，可以指定分隔符，默认使用空格作为分隔符

· awk：基于行的处理命令

· sed：对文件中的内容进行删除、替换、增加的命令

· sort：排序命令

· uniq：取消重复行

49.Linux系统磁盘的相关命令

· df：查看文件系统的大小和使用情况

· du：查看目录及目录下文件的大小

· fdisk：centos6的磁盘分区命令

· gdisk：centos7的磁盘分区命令

· mount：挂载命令

· umount：卸载命令

· lsblk：列出系统中所有的块设备信息

· blkid：列出设备名称和UUID

· mke2fs：创建文件系统

· fsck：检测分区

**50.说一下df和du的区别**

· df主要是查看文件系统的大小

· du是用来查看文件大小的命令

**51.Linux系统如何进行调优**

· 配置时间服务器，进行时间同步

· 避免直接使用root用户，防止权限溢出

· 配置国内yum源

· 开启ssh服务，使用ssh密钥对登录，修改默认连接端口

· 备份系统中的重要文件，比如： /etc/passwd ，/etc/shadow，/etc/group ，/etc/gshadow

· 配置合理的SELinux上下文规则

· 配置合理的防火墙规则

**52.Linux查看内核版本使用什么命令**

uname -a

**53.Linux中你认为哪些目录比较重要**

· /etc：存放了系统中重要的配置文件

· /bin：存放了系统命令

· /usr/：系统安装时安装的软件默认安装在这个目录下

· /lib：存放系统程序需要调用的函数库

· /dev：设备文件保存位置

· /home：用户家目录

**54.统计文件行数使用哪个命令**

· wc -l：统计文件行数

· wc -c ：统计文件字符数

· wc -w：统计文件字数

**55.解释一下B/S和C/S结构**

· B/S结构：浏览器/服务器架构， 一般建立广域网上，通常面向的是不可知的用户群，对安全的控制能力相对较弱。有逻辑上相互分离的数据层、表示层和业务层构成。表示 层向客户提供数据，业务层实施业务和数据规则， 数据层定义数据访问标准，三层体 系结构中的核心是组件对象模型。

· C/S结构：客户端/服务器架构， 一般建立在局域网上，通过专门的客户端、服务器进 行数据传输和交换，安全性较好。是一个典型的两层架构，将任务合理的分配到客户端 和服务端，降低了系统的通讯对资源的消耗，需要安装客户端才能进行操作。

**56.简单说一下rpm的用法，rpm安装和yum安装的区别是什么**

·rpm -ivh：安装rpm包， -i是安装， -v是显示详细信息， -h是以#号输出安装进度

·rpm -Uvh：升级安装，没有直接安装，有旧版本升级安装

·rpm -Fvh：升级安装，有旧版本升级安装，没有旧版本不安装

·rpm -e ：卸载

rpm安装和yum安装最大的区别在于， yum安装会主动从yum仓库中下载所需要的依赖包；但是rpm安装不会主动解决依赖关系，需要提前将需要的依赖软件安装好

**57.说一下防火墙的四表五链**

**·四表，承载链**

raw：数据报文跟踪

filter：数据报文过滤

mangle：数据报文修改

nat：路由转发，实现dnat （目标地址）和snat （源地址）转换

**·五链，承载规则**

input：数据包入站应用此链中的规则

output：数据包出站应用此链中的规则

forward：处理数据包转发的报文请求

prerouting：数据包做路由前选择，配合DNAT使用

postrouting：数据包做路由后选择，配合SNAT使用

**58.swap分区的作用**

swap分区是一个交换分区，当物理内存不足时， swap分区会释放一部分内存给当前正在运 行的程序使用。这部分空间一般来自那些很长时间没有运行过的程序，系统将这部分空间临时存放在swap空间中，给那些内存占有率比较高的程序使用。

**59.怎么查看系统中的进程和线程**

进程： top，ps -aux，ps -ef

线程： ps -aT （查看系统中的所有线程）

ps -T -p <PID> ：查看某个进程中的线程

top -H -p <PID> ：查看进程中的线程

60.buffer和cache的区别

buffer：缓冲，是为了提高内存和硬盘（或者其他io设备）之间的数据交换速度而设计 的

cache：缓存从CPU的角度来看，是为了提高内存和CPU之间的数据交换速度而设计的

从内存读取和磁盘读取的角度来看，可以理解为系统为了更高的数据读取速率，将 已经读取过的数据存放在内存中。

**61.如何查看系统的资源限制**

ulimit -a

**62.说一下ulimit命令**

ulimit 命令主要用来限制使用系统资源（内存、 CPU等）的范围，是Linux系统的一个 内置的bash命令ulimit 设置项仅在当前shell才会生效，需要永久生效需要修改配置文件 将限制写入 ~/.profile或者~/.bashrc 只对当前的用户生效写入 /etc/security/limits.conf 下可针对性配置，在系统层面永久生效

·调整相关硬限制值 (Hard Limit) 设置一次后,下一次设置的值只能比上一次的值要小

· 如果不加S或者H修饰，默认同时修改硬限制和软限制

**63.怎么查看当前的系统环境变量，怎么查看所有的系统环境变量**

· 查看当前的系统变量： set

· 查看所有的系统环境变量： env

· set：显示当前shell定义的私有变量

· env：显示用户的环境变量

· export：显示当前导出成用户变量的shell变量，并显示变量属性

**64.程序自动启动有哪几种方式**

· 在CentOS7中，使用 systemctl enable 软件 ，设置开机自启

· 在CentOS6中，使用 service 软件 enable ，设置开机自启

· 或者将命令写入 /etc/rc.d/rc.local 下，会在开机自动运行

/etc/rc.local是/etc/rc.d/rc.local的软链接

65.用过抓包命令吗，说一下怎么指定网卡，怎么指定所有网卡 ， -nn和 -vv什么含义

tcpdump可以将网络中传送的数据包的“头”完全截获下来提供分析，它支持针对网络层、协 议、主机、网络或端口的过滤

· 包分析工具： tcpdump

· 指定网卡： -i

· 指定所有网卡： -i any

· -nn：禁止IP和端口的名称解析

· -vv：显示详细信息， v 的个数越多，显示的信息越详细

66.接触过redis集群吗 ， 说 一 下redis集群的三种实现方式 ， 原理是什么

· 主从模式

slave端启动后，会主动向master端发送sync命令， master会拍摄快照保存到后台

（RDB持久化），同时会缓存保存这段时间的命令，然后将快照和命令发送到slave 端， slave端加载快照文件和缓存命令进行数据同步。

如果master节点宕掉，那么redis集群就无法对外提供写服务。

· 哨兵模式

以主从模式作为基础，在集群加入监控服务器，监控服务器可以是一台也可以是一 个集群。

监控服务器以每秒一次的频率向集群中的master节点、 slave节点和其他的监控节点发送 ping 命令。

如果一个节点距离最后一次有效回复 ping 命令的时间超过了配置文件中规定的时 间，那么这个节点就会被标记为主观下线。

如果一个master节点被标记为主观下线，那么监视master节点的所有监控服务器会 以每秒一次的频率确认master节点确实进入了主观下线状态。如果由足够多的监控 服务器将master节点标记为了主观下线状态，那么这个master节点会被标记为客观 下线状态。

使用哨兵模式时，客户端连接的是哨兵的IP和端口，由哨兵来提供Redis集群的相关 服务，如果master节点挂掉，那么哨兵就会从slave节点中重新选择一个作为master 节点

高可用集群：

高可用集群解决了当数据量过大时，单台服务器的存储容量不够的问题。高可用集群可以对数据进行分片，将数据储存在多个服务器中。

在cluster集群中，每个节点都可以作为一个主从模式的集群（一主一从或者一主多 从），其中slave只作为备用，不提供服务。客户端可以直接连接master节点进行读写。

多个节点之间通过网络进行数据共享

cluster集群通过哈希函数进行数据分区，因为Redis数据库是通过 key-value 的方式 进行数据存储，所以在构建cluster集群时，所有的键会根据哈希函数分配到 0-1638 3 的虚拟槽中。

如果master节点被标记为客观下线状态，那么会从他的slave节点中选择一个替换 它，保证集群的高可用。

**67.服务器内存不够怎么处理**

首先查看服务器的内存容量，然后查看服务器的运行程序和进程，分析是由于程序运行过多和线程占用造成的内存不足，还是由于服务器本身的硬件资源不足导致的。也有可能是因为服务器中毒引起的内存不足

**68.CPU资源不足怎么处理**

使用 top 命令查看一下哪个进程占用的CPU资源过高

找到进程中消耗资源最多的线程id，使用的命令是 top -H -p 进程id

将线程id转换为16进制， 使用命令 printf "%0x\n" 线程id 或者 echo 'obase=16,线程id' | bc查看线程的状态信息作进一步分析，使用命令进制 jstack 进程id | grep -A 10 线程id的16

**69.遇到磁盘空间不足的报错，你应该如何处理**

可能一：

使用 df -h 命令查看磁盘空间的使用情况，确定哪个目录占用的磁盘空间过高

确定目录后，使用 du -h 命令进行逐级定位，确认到占用空间最大的大文件

查看文件内容，确认一下是否需要保留，如果需要保留就通过压缩导出，不需要保留可以直接删除，或者对磁盘进行扩容,或者可以直接使用 find 命令查找目录下大于400MB的文件名称，然后进行删除 find 目录 -size +400M

可能二：

使用 df -h 命令并不能发现大文件，那么可能是文件被删除，但是进程仍然在调用 这个文件,此时可以通过 lsof | grep delete ，找到占用的进程，把这个进程kill掉然后重启 服务器即可

**70.如何将新磁盘扩展到已有分区**

首先将磁盘格式化，CentOS6使用 fdisk 命令，CentOS7使用 gdisk 命令

使用 pvcreate 将新分区创建为物理卷

使用 pvdispalay 列出原有的物理卷组

使用 vgextend 把卷加入到卷组中

使用 df -h 查出需要进行扩展的分区名称，使用 lvresize -L 要扩展的容量 分区名 进行 扩容

使用 xfs\_growfs 挂载点 让文件系统进行识别

**71.Linux如何开机自启**

echo '启动命令 ' > /etc/rc.local

chmod +x /etc/rc.local

#服务有自己的启动脚本文件

mv 脚本启动配置文件 /usr/local/bin

mv 脚本启动配置文件 /usr/local/sbin

systemctl enable 服务名称

**72.说一下文件描述符对系统有什么影响**

Linux服务器的内核文件描述符的个数存在限制（对于不同的Linux服务器类型，数值也不 一样），默认限制在1024左右。这个数值对于不太繁忙的文件是足够的，但是对于web服 务器和加压测试的服务器来说，这个数值是不够用的。因为web服务器和加压测试的服务 器会打开大量的文件，导致文件描述符不够用，进而导致无法访问新的文件。

**73.Linux怎么挂载新磁盘**

使用 mount 命令对磁盘进行挂载

将挂载写入 /etc/fstab 配置文件，执行mount -a实现永久挂载

重启系统

**74.找到修改30天以上的文件并删除**

find 目录 -ctime +30

#c：表示属性被修改过

#a：表示文件被访问过

#m：表示内容被修改过

#time：按天查找

#min：按分钟查找

**75.服务器系统日志放在哪里**

/var/log/message ：系统启动后的信息和错误日志

/var/log/security ：与系统安全相关的日志

/var/log/maillog ：与系统邮件相关的日志

/var/log/cron ：与定时计划任务相关的日志

/var/log/boot.log & /var/log/dmesg ：与系统启动相关的日志

**76.如果你的一块磁盘中，只有几个小文件，使用df命令查看，发现磁盘是满的，什么原因有僵尸进程存在，大文件虽然删除了，但是僵尸进程仍然在调用这个文件，所以会显示磁盘被占满怎么解决**

使用 lsof | grep deleted 找到删除的文件对应的进程号，使用 kill -9 关闭进程，重启系统

**77.查看操作系统的版本信息**

#查看操作系统的版本信息: uname -a

#查看CPU版本信息: cat /proc/cpuinfo

#查看内核版本信息: cat /proc/version

#查看发行版本信息: cat /etc/issue

**78.解释一下什么是HAC的脑裂，能说一下解决方案吗**

因为主机异常或宕机时，集群内的主机都认为是对方出现异常，而导致的抢占资源现象。

解决方案：

添加冗余的心跳线，比如同时使用串行电缆和以太网电缆

设置参考IP，当出现线路中断的情况时，让2个节点都去ping参考IP，不通就说明是 本节点故障， 放弃抢占，让正常的节点接管服务

出现脑裂现象时，直接强制关闭一个节点，备节点没有收到心跳隐患，就会通过单 独的线路关闭 主节点电源

**79.Linux进程有哪些状态**

R(TASK\_RUNNING) ：运行状态

S(TASK\_INTERRUPTIBLE) ：可中断的睡眠状态

D(TASK\_UNINTERRUPTIBLE) ：不可中断的睡眠状态

T(TASK\_STOPPED) ：挂起状态，可以被唤醒

Z(TASK\_ZAOMBLE) ：僵尸状态，程序已经中止，但是父程序没有调用 wait 函数获取子 进程的相关信息，那么子进程的状态信息仍然保留在内存中，变成僵尸进程

**80.Linux查看系统资源的命令**

CPU：vmstat，iostat，mpstat，top ，dstat

内存： vmstat，free ，dstat

磁盘io：vmstat，iostat，top ，dsta

**81.Linux查看服务运行端口的命令**

netstat -anpt，lsof -i -P

82./etc/profile和 .bash\_profile有什么区别

· /etc/profile：存放的是系统环境变量

· .bash\_profile：存放的是当前用户的环境变量

**83.数据库的锁是如何制定的**

数据库是一个多用户使用的共享资源，所以会存在多用户同时读取一个数据的情况，如果 不加以控制，可能会破坏数据的一致性

锁机制就是为了加强对数据的并发控制

**84.MySQL为什么会出现锁表的情况，怎么解锁，如何避免这种情况的出现**

· 锁

高并发情况下造成的io拥堵

数据主从同步失败

使用命令进行锁表： lock

· 解锁： unlock

· 增加数据库的缓存可以避免锁表的情况出现

**85.innodb引擎是插入快还是读取快**

写入快，因为写操作不会锁定全表，所以在并发较高时，使用innodb引擎效率更高

**86.Docker和虚拟机的区别**

· Docker启动属于秒级启动，虚拟机启动需要几分钟去进行启动

· Docker属于操作系统级别的虚拟化，通过Docker守护进程直接和内核进行交互，几乎没有性能损耗；虚拟机是硬件级别的虚拟化，需要通过Hypevisor层，性能损耗比较大

· Docker更轻量，占用内存小，在同样的硬件环境下， Docker运行的镜像数量要远多于 虚拟机数量。

· Docker通过Namespaces和Cgroups实现对应用程序的进程之间的隔离，虚拟机从操作系统层面实现隔离，所以虚拟机的隔离性更强、安全性更好

**87.Dockerfile文件**

· FROM：定义基础镜像，必须指定而且要在其他的dockerfile指令之前

· MAINTAINER：定义镜像制作人的基本信息

· RUN：声明要安装或执行的文件，可以运行任何被基础镜像支持的命令

· CMD：容器启动时需要进行的操作，可以是执行系统脚本或者系统命令，只能在文件 中出现一次，如果写入多个默认只有最后一个生效

· ENTERPOINT：容器启动时需要进行的操作，指定容器启动时需要执行的命令，只能在 文件中出现一次，如果写入多个默认只有最后一个生效

· USER：定义容器中的用户

· EXPOSE：将容器的端口信息映射到物理机的端口上

· ENV：在容器中添加一个环境变量

· ADD：把一个路径下的文件复制到另一个路径下，如果这个文件是压缩包，那么复制时 会强制解压

· COPY：把一个路径下的文件复制到另一个路径下

· VOLUME：数据卷挂载，实现容器的数据持久化

· WORKDIR：指定工作目录，容器启动后，会直接切换到工作目录下

· ONBUILD：指定的命令在子镜像中执行

**88.Linux中常用的网络命**

netstat

ifconfig

ping

traceroute

host

**89.NetworkManager**

Centos7以后的网卡图形化管理界面，优先级高于Network

**90.简单介绍 一 下docker定义**

基于linux comtainer的内核技术， go语言开发 http2.0协议 基于namespace的名称空间隔离，基于cgroup做资源限制，基于chroot做伪根，基于网桥技术的网络通信

**91.KVM和docker有什么区别**

· Docker 容器的启动可以在秒级实现，这相比传统的虚拟机方式要快得多。其次， Docker对系统资源的利用率很高，一台主机上可以同时运行数千个 Docker 容器。

· 容器除了运行其中应用外，基本不消耗额外的系统资源，使得应用的性能很高，同时系统的开销尽量小。传统虚拟机方式运行 10 个不同的应用就要起 10 个虚拟机，而

Docker 只需要启动 10 个隔离的应用即可。

· 虚拟化技术依赖物理CPU和内存，是硬件级别的；而docker构建在操作系统上，利用操 作系统的containerization技术，所以docker甚至可以在虚拟机上运行。

· 虚拟化系统一般都是指操作系统镜像，比较复杂，称为“系统”；而docker开源而且轻 量，称为“容器”，单个容器适合部署少量应用，比如部署一个redis 、一个memcached。

· 传统的虚拟化技术使用快照来保存状态；而docker在保存状态上不仅更为轻便和低成 本，而且引入了类似源代码管理机制，将容器的快照历史版本一一记录，切换成本很低。

· 传统的虚拟化技术在构建系统的时候较为复杂，需要大量的人力；而docker可以通过Dockfile来构建整个容器，重启和构建速度很快。更重要的是Dockfile可以手动编写，这 样应用程序开发人员可以通过发布Dockfile来指导系统环境和依赖，这样对于持续交付 十分有利。

· 当然KVM对比于容器也有一个比较大的优势就是可以使用不同的操作系统或内核。所 以，举例说，你可以使用微软Azure，同时运行Windows Server2012的实例和SUSELinux企业级服务器的实例。至于Docker，所有容器都必须使用同样的操作系统和内核。

**92.为什么说docker是轻量级**

· 不依赖于系统硬件

· 创建和销毁简单

· 扩容和缩容简单

· 秒级启动

· 基于linux内核技术

**93.Docker常见的网络模式**

brige：docker创建容器时，默认使用此网络模式，会创建一个docker0网桥，和宿主机 进行通信

· none：创建容器时，关闭容器的网络功能

· host：桥接。使用宿主机的IP和端口

**94.Docker中Namespaces和Cgroups的隔离类型**

· User：用户和用户组

· UTS：主机名和域名

· PID：进程编号

· Mount：挂载（文件系统）

· Network：网络栈，网络设备，端口

IPC：消息队列，共享内存，信号量

**95.Docker如何缩容**

使用kubernetes中deployment控制器对docker进行缩容

具体的操作方式是修改deployment的yaml文件，改变replication的实例数

96.kubernetes中组件和插件的功能

**组件**

apiserver：本质是一个web服务器，通过resetful接口编写，是整个kubernetes集群 的入口

scheduler：调度器，将pod负载到node节点上

controllermanager：维稳集群状态

kubelet：apiserver和CRI之间的接口

etcd：键值对数据库，存储集群状态信息

kube-proxy：通过ipvs 、iptables实现网络分发、负载均衡

**插件**

coredns：kubernetes集群内部的DNS解析服务

flannel：基于CNI的扁平化网络

Ingress nginx：提供7层负载

**97.介绍 一 下kubernetes集群中的控制器**

批处理任务

job

cronjob

守护进程

有状态： statefulset。持久化存储，基于pv 、pvc实现；持久的网络标识，基于无头 服务实现；有序部署回收或启停

无状

RC

RS：集合式的标签选择器

Deployment：官方支持滚动更新和回滚，通过创建RS来引导Pod

DaemonSet：node节点上只能有一个运行结果

**98.pod的分类**

· 自主式pod

· 控制器管理的pod

pod是kubernetes集群中最小的单位。kubernetes集群在创建pod时，首先，kubctl会向kubeapi发送指令， etcd会存储kubeapi调度kubectl的过程，启动一个基础容器pause，如果 这个pod没有运行节点，那么scheduler会将这个pod调度到node节点上

**99.python删除文件**

· os.remove() 或os.unlink(）删除文件

· os.removedirs(path)，删除文件夹，但是文件夹必须为空。

**100.ansible任务执行模式**

· ad-hoc模式(点对点模式)

使用单个模块，支持批量执行单条命令。 ad-hoc 命令是一种可以快速输入的命令，而 且不需要保存起来的命令。 就相当于bash中的一句话shell。

· playbook模式(剧本模式)

是Ansible主要管理方式，也是Ansible功能强大的关键所在。 playbook通过多个task集 合完成一类功能，如Web服务的安装部署、数据库服务器的批量备份等。可以简单地 把playbook理解为通过组合多条ad-hoc操作的配置文件。

**101.如何查看隐藏中的进程**

在 Linux 系统中，进程通常是通过命令 ps、top 或 htop 等工具查看的

ps 命令可以用来列出当前用户或所有用户的进程：ps aux

top 命令提供了一个实时更新的进程列表，显示系统中活动的进程

htop 是 top 的一个增强版本，提供了一个更友好的界面

pgrep 命令可以根据名称或其他属性查找进程。可以帮助你找到特定的进程: pgrep -l <process\_name>

某些进程可能会通过特定手段隐藏自己，如 rootkit 或恶意软件。要检测这些隐藏的进程：pstree -p

**使用 lsof 命令**列出所有打开的文件和对应的进程，有助于发现异常进程