

学神 IT-Linux 运维面试题库

1、先自我介绍

一、个人情况

简单说一下自己的基本情况和学历背景就好

例如：

我是 XX，来自 XX 大学，专业是 XX。大学时参与了 XX 社团的 XX，有些 XX 能力提升，拿过 XX 奖项

二、与岗位匹配的经历

这里主要介绍你有过什么和岗位相关的工作/实习/学校经历。通过分析岗位需求，用具体事例向面试官证明你具备匹配该职位的能力，切忌用形容词来堆砌，主要是通过案例让对方得知你具有的技能 and 经验。

例如：

我之前在 XX 公司的市 XX 岗位，期间我负责 XXX,XXX,XXX 的工作，工作期间参与 XXX 解决方案时，我通过 xxxx 分析与优化并提出了实施有效的解决方案，该方案最后实施并获得 xxxx 效果。

三、与岗位相关的爱好经历

最好是一个与众不同，又可以和岗位匹配的爱好经历。当然实在没有，这点也可以省略。

例如：

爱好可以是刷微博(热点)、技术论坛(可以在稍有著名论坛的某技术论坛，发表 xx 篇文章等)，运动类的爱好：高尔夫、游泳

这样会让你看起来是一个经验丰富、积极向上、不会枯燥乏味的人，做技术相关是很需要这样一个饱满的年轻人的。

四、表达对该工作的兴趣

在准备面试之前，最好是先去了解公司，公司的业务如何，有什么文化特点等等，然后在自我介绍最后说明自己了解了公司的情况，对这方面工作感兴趣，有加入的意愿。

了解公司往往会让面试官觉得眼前一亮，感受到你的真诚，甚至还会直接和你详细聊聊公司的愿景以及下一步的计划。

总结一下，自我介绍要做到：了解公司+了解自己+摆正心态+勤加练习+个人信息+匹配能力+举例/数据论证+强烈加入的意愿 这几点，那么 offer 就离你不远了

案例：

我是王涛，来自北京市朝阳区，毕业于北京大学，专业是网络信息。在大学时参与了网络安全社团，有对于网络安全的能力提升，期中拿过网络安全奖项。

来贵司面试之前，是在北京市某网络科技有限公司工作，任职网络安全工程师岗位，期间我负责维护

及优化公司网络安全的工作，在此工作期间参与 Ddos 攻击及网络安全优化解决方案时，我通过模拟攻击及抓包分析，并提出了实施有效的解决方案，最后该方案实施并获得提高网络安全性及网络稳定性的效果。通过此次工作与研究令我对网络安全的知识与技能得到了进一步的提升。

当然这一些收获与我的爱好很有关系，而我的爱好刷网络技术论坛，多方面了解其他的网络安全事件及解决方案，从中可以有助于我的技能掌握，并能够了解一些新的技能手段，我有爱好除了刷论坛以外，也会刷刷微博，了解一下新闻事件，平时由于上班缺少运动，也会抽空运动打篮球、跑步、游泳，这样可以提活力。

在贵司面试前，我已经对贵司的情况有过一定的了解，经了解后才知道贵司的实力和 IT 资源比较丰富，因此技术岗位和人员比较多，我不仅为来贵司效力，同时也想不断的向贵司元老学习并提升技能，同时也了解到贵司是以创新、专注、分享的企业精神，目标与时俱进，贵司在 IT 行业也取得突出的成绩，在 IT 领域中也做了不少的做贡献，所以对贵司的技术工作特别感兴趣，非常有加入的意愿。

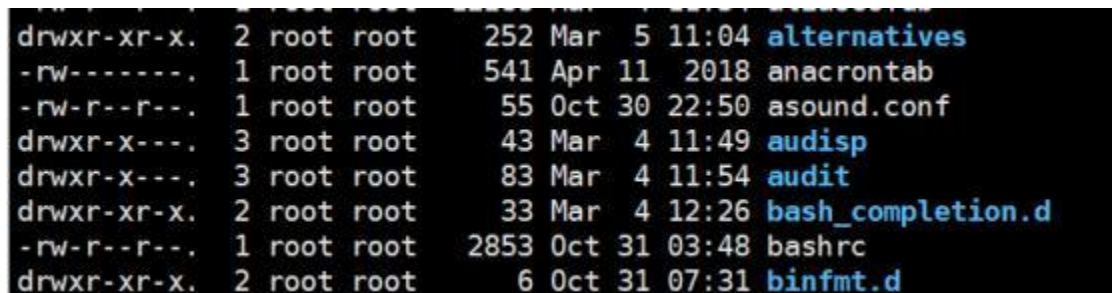
2、如何生成 8 位随机密码

```
date +%s%N | Md5sum | head -c 8
echo $RANDOM | Md5sum | head -c 8
openssl rand -base64 8 | md5sum | cut -c1-8    #八位字母和数字的组合
openssl rand -base64 8 | cksum | cut -c1-8 #八位数字
cat /dev/urandom | head -n 10 | md5sum | head -c 8
cat /dev/urandom | strings -n 8 | head -n 1
cat /dev/urandom | sed -e 's/[^a-zA-Z0-9]//g' | strings -n 8 | head -n 1 这个方法效率低下
cat /proc/sys/kernel/random/uuid | md5sum | cut -c1-8
```

3 取出同时与服务器建立链接最多的 ip 地址

```
netstat -antup |grep ESTABLISHED |awk '{print $5}' |awk -F: '{print $1}' |sort |uniq -c
|sort -nr |head -n 1
```

4 Linux 文件权限一共 10 位长度，分成四段，第三段表示的内容是？



```
drwxr-xr-x. 2 root root 252 Mar 5 11:04 alternatives
-rw----- 1 root root 541 Apr 11 2018 anacrontab
-rw-r--r-- 1 root root 55 Oct 30 22:50 asound.conf
drwxr-x--- 3 root root 43 Mar 4 11:49 audisp
drwxr-x--- 3 root root 83 Mar 4 11:54 audit
drwxr-xr-x. 2 root root 33 Mar 4 12:26 bash_completion.d
-rw-r--r-- 1 root root 2853 Oct 31 03:48 bashrc
drwxr-xr-x. 2 root root 6 Oct 31 07:31 binfmt.d
```

文件所有者所在组的权限

5 禁止 192.168.0.0/24 网段访问服务器

```
[root@xuegod63 ~]# iptables -I INPUT -s 192.168.0.0/24 -j DROP
```

6 简述 raid0 raid1 raid5 三种工作模式的工作原理及特点。

RAID 0: 连续以位或字节为单位分割数据, 并行读/写于多个磁盘上, 因此具有很高的数据传输率, 但它没有数据冗余, 因此并不能算是真正的 RAID 结构。RAID 0 只是单纯地提高性能, 并没有为数据的可靠性提供保证, 而且其中的一个磁盘失效将影响到所有数据。因此, RAID 0 不能应用于数据安全性要求高的场合。

RAID 1: 它是通过磁盘数据镜像实现数据冗余, 在成对的独立磁盘上产生互为备份的数据。当原始数据繁忙时, 可直接从镜像拷贝中读取数据, 因此 RAID 1 可以提高读取性能。RAID 1 是磁盘阵列中单位成本最高的, 但提供了很高的数据安全性和可用性。当一个磁盘失效时, 系统可以自动切换到镜像磁盘上读写, 而不需要重组失效的数据。简单来说就是: 镜像结构, 类似于备份模式, 一个数据被复制到两块硬盘上。

RAID 10: 高可靠性与高效磁盘结构一个带区结构加一个镜像结构, 因为两种结构各有优缺点, 因此可以相互补充。主要用于容量不大, 但要求速度和差错控制的数据库中。

RAID 5: 分布式奇偶校验的独立磁盘结构, 它的奇偶校验码存在于所有磁盘上, 任何一个硬盘损坏, 都可以根据其它硬盘上的校验位来重建损坏的数据。支持一块盘掉线后仍然正常运行

7 通过 apache 访问日志 access.log 统计 IP 和每个地址访问的次数

按访问量列出前 10 名。日志格式样例如下

```
192.168.1.247 - - [02/Jul/2010:23:44:59 +0800] -GET / HTTP/1.1" 200 19
```

```
:
```

```
[root@xuegod63 ~]# service httpd restart
```

```
[root@xuegod63 ~]# curl http://192.168.1.63 #自己产生一些日志
```

```
[root@xuegod63 ~]# cat /var/log/httpd/access_log | awk '{ print $1 }' | uniq -c | sort -rn | head -10
```

//这个别的方法也能统计,但有些命令是必要的 awk, sort, uniq, 主要看是否这些命令都使用了。

8 你认为在 linux 系统调优方面都包括哪些工作,

以 linux 为例, 请简明阐述, 并举一些参数为例。

系统调优包括内核参数优化和应用优化 2 个方面,

内核参数调优如:

进程最多打开文件数

IPV4 等网络参数调优

应用优化如:

CPU 亲和力

进程数

Nginx 配置缓存 压缩 高效传输 超时时间 等。

9 当用户在浏览器当中输入一个网 g 站，说说计算机对 dns 解释经过那些流程？

注： 本机跟本地 dns 还没有缓存。

- a.用户输入网址到浏览器
- b.浏览器发出 DNS 请求信息
- c.计算机首先查询本机 HOST 文件， 看是否存在， 存在直接返回结果，不存在，继续下一步
- d.计算机按照本地 DNS 的顺序，向合法 dns 服务器查询 IP 结果，
- e.合法 dns 返回 dns 结果给本地 dns，本地 dns 并缓存本结果，直到 TTL 过期，才再次查询此结果
- f.返回 IP 结果给浏览器
- g.浏览器根据 IP 信息， 获取页面

10 apache 有几种工作模式， 分别介绍下其特点，并说明什么情况下采用不同的工作模式？

apache 主要有三种工作模式： prefork、worker、event

prefork 的特点是：（预派生）

- 1.这种模式可以不必在请求到来时再产生新的进程， 从而减小了系统开销
- 2.可以防止意外的内存泄漏
- 3.在服务器负载下降的时候会自动减少子进程数

worker 的特点是：

支持混合的多线程多进程的多路处理模块如果对于一个高流量的 HTTP 服务器 worker MPM 是一个比较好的选择， 因为 workerMPM 占用的内存要比 prefork 要小。

Event MPM：

event 模式是在 2.4 版本中才可以稳定运行。这是 Apache 最新的工作模式，它和 worker 模式很像，不同的是在于它解决了 keep-alive 长连接的时候占用线程资源被浪费的问题，在 event 工作模式中，会有一些专门的线程用来管理这些 keep-alive 类型的线程，当有真实请求过来的时候，将请求传递给服务器的线程，执行完毕后，又允许它释放。这增强了在高并发场景下的请求处理。

11.简述 Linux 启动过程

Bios 引导--》启动引导工具 grub--》核心初始化--》载入初始程序 init--》init 初始化--》从 inittab 中读取数据， 决定启动级别--》系统运行

12.简述 Tcp 三次握手的过程

第一次握手， 建立连接， 客户端发送 SYN 包到服务器， 并进入 SYN_SEND 状态， 等待服务器确认；

第二次握手， 服务器收到 SYN， 同时自己也发送一个 SYN 包和一个 ACK 包来确认客户端的 SYN， 并进入 SYN_RECV；

第三次握手， 客户端收到服务器发来的 SYN+ACK 后， 回复服务器端一个 ACK 确认， 发送完毕后， 双方进入 ESTABLISHED 状态。

三次握手成功后， 开始传输数据。

13、 Mysql 主从同步的原理？

第一步： master 记录二进制日志。在每个事务更新数据完成之前， master 在二进制日志记录这些改变。MySQL 将事务写入二进制日志，即使事务中的语句都是交叉执行的。在事件写入二进制日志完成后，master 通知存储引擎提交事务。

第二步： slave 将 master 的 binary log 拷贝到它自己的中继日志。首先，slave 开始一个工作线程——I/O 线程。I/O 线程在 master 上打开一个普通的连接， 然后开始 binlog dump process。Binlog dump process 从 master 的二进制日志中读取事件， 如果已经执行完 master 产生的所有文件， 它会睡眠并等待 master 产生新的事件。I/O 线程将这些事件写入中继日志。

第三步： SQL slave thread (SQL 从线程) 处理该过程的最后一步。SQL 线程从中继日志读取事件， 并重新执行其中的事件而更新 slave 的数据， 使其与 master 中的数据一致。

14、 查看服务器运行状态相关的命令

1 找出系统中使用 CPU 最多的进程？

方法 1：使用 top 命令

运行 top ， 找出使用 CPU 最多的进程 ， 按大写的 P， 可以按 CPU 使用率来排序显示

2 找出系统中使用内存最多的进程？

方法 1：使用 top 命令

运行 top ， 然后按下大写的 M 可以按内存使用率来排序显示

方法 2:

按照实际使用内存， 从大到小排序显示所有进程列表

[root@xuegod63 ~]# ps -aux --sort -rss | more 内存降序排序(去掉减号就是升序)

3 找出系统中对磁盘读写最多的进程?

iostat 命令, 查看哪个进程使用磁盘读写最多

4 找出系统中使用网络最多的进程?

使用 nethogs 找出使用带宽最多的进程

15、你对现在运维工程师的理解和以及对其工作的认识

运维工程师在公司当中责任重大, 需要保证时刻为公司及客户提供最高、最快、最稳定、最安全的服务

运维工程师的一个小小的失误, 很有可能会对公司及客户造成重大损失
因此运维工程师的工作需要严谨及富有创新精神

16、nginx 反向代理的几种模式?

nginx 反向代理一共以下 5 种模式, 前 3 种使用的比较多:

- 1) 轮询: 每个请求按时间顺序逐一分配到不同的后端服务器, 如果后端服务器 down 掉, 能自动剔除。
- 2) 权重: 指定轮询几率, weight 和访问比率成正比, 用于后端服务器性能不均的情况。
- 3) ip_hash: 每个请求按访问 ip 的 hash 结果分配, 这样每个访客固定访问一个后端服务器, 可以解决 session 的问题。
- 4) fair (第三方): 按后端服务器响应时间来分配, 时间短的优先分配。
- 5) url_hash (第三方): 按访问 url 的 hash 结果来分配请求, 使同样的 url 定向到同一个后端服务器, 后端服务器为缓存时比较有效。

17、apache 虚拟主机模式

基于端口, 基于域名, 基于 ip

18、Lvs nginx haproxy 优缺点对比

LVS:

- 1、抗负载能力强。抗负载能力强、性能高, 能达到 F5 硬件的 60%; 对内存和 cpu 资源消耗比较低
- 2、工作在网络 4 层, 通过 vrrp 协议转发(仅作分发之用), 具体的流量由 linux 内核处理, 因此没有流量的产生。
- 2、稳定性、可靠性好, 自身有完美的热备方案; (如: LVS+Keepalived)
- 3、应用范围比较广, 可以对所有应用做负载均衡;
- 4、不支持正则处理, 不能做动静分离。
- 5、支持负载均衡算法: rr (轮循)、wrr (带权轮循)、lc (最小连接)、wlc (权重最小连接)
- 6、配置复杂, 对网络依赖比较大, 稳定性很高。

Nginx:

- 1、工作在网络的 7 层之上, 可以针对 http 应用做一些分流的策略, 比如针对域名、目录结构;
- 2、Nginx 对网络的依赖比较小, 理论上能 ping 通就能进行负载均衡;
- 3、Nginx 安装和配置比较简单, 测试起来比较方便;
- 4、也可以承担高的负载压力且稳定, 一般能支撑超过 1 万次的并发;

- 5、对后端服务器的健康检查，只支持通过端口来检测，不支持通过 url 来检测。
- 6、Nginx 对请求的异步处理可以帮助节点服务器减轻负载；
- 7、Nginx 仅能支持 http、https 和 Email 协议，这样就在适用范围较小。
- 8、不支持 Session 的直接保持，但能通过 ip_hash 来解决。、对 Big request header 的支持不是很好，
- 9、支持负载均衡算法：Round-robin (轮循)、Weight-round-robin (带权轮循)、Ip-hash (Ip 哈希)
- 10、Nginx 还能做 Web 服务器即 Cache 功能。

HAProxy 的特点是：

- 1、支持两种代理模式：TCP (四层)和 HTTP (七层)，支持虚拟主机；
- 2、能够补充 Nginx 的一些缺点比如 Session 的保持，Cookie 的引导等工作
- 3、支持 url 检测后端的服务器出问题的检测会有很好的帮助。
- 4、更多的负载均衡策略比如：动态加权轮循(Dynamic Round Robin)，加权源地址哈希(Weighted Source Hash)，加权 URL 哈希和加权参数哈希(Weighted Parameter Hash)已经实现
- 5、单纯从效率上来讲 HAProxy 更会比 Nginx 有更出色的负载均衡速度。
- 6、HAProxy 可以对 Mysql 进行负载均衡，对后端的 DB 节点进行检测和负载均衡。
- 9、支持负载均衡算法：Round-robin (轮循)、Weight-round-robin (带权轮循)、source (原地址保持)、RI (请求 URL)、rdp-cookie (根据 cookie)
- 10、不能做 Web 服务器即 Cache。

19 三大主流软件负载均衡器适用业务场景

1、网站建设初期，可以选用 Nginx/HAproxy 作为反向代理负载均衡(或者流量不大都可以不选用负载均衡)，因为其配置简单，性能也能满足一般的业务场景。如果考虑到负载均衡器是有单点问题，可以采用 Nginx+Keepalived/HAproxy+Keepalived 避免负载均衡器自身的单点问题。

2、网站并发达到一定程度之后，为了提高稳定性和转发效率，可以使用 LVS、毕竟 LVS 比 Nginx/HAproxy 要更稳定，转发效率也更高。不过维护 LVS 对维护人员的要求也会更高，投入成本也更大。

注：Nginx 与 HAproxy 比较：Nginx 支持七层、用户量最大，稳定性比较可靠。HAproxy 支持四层和七层，支持更多的负载均衡算法，支持 session 保存等。具体选型看使用场景，目前来说 HAproxy 由于弥补了一些 Nginx 的缺点用户量也不断在提升。

20、nginx 调优设置

1：系统优化

Nginx 进程数

cpu 亲和力

nginx 最多可以打开文件数

内核参数优化

2 配置优化

Nginx 实践处理模型 epoll 模式

开启高效传输模式

配置链接超时时间

Fastcgi 调优

Gzip 调优

Expires 缓存调优

防盗链

21、你解决过什么漏洞或故障

MYSQL 主从同步异常

Centos 无法创建新文件

服务器被黑上不了网

木马排查

日志文件删除后磁盘空间不释放

22、公司的系统架构说说

公司使用 apache 作为 web 服务器，使用NFS 共享存储，后端为mysql MHA 集群并使用redis 数据库对数据进行缓存。前端使用 LVS 作为负载均衡服务器， vanish 作为缓存服务器，大大提高了业务的稳定性和访问速度。

23、你对哪些服务器硬件型号比较熟悉， 你们公司使用了哪些型号。

F5 负载均衡服务器， dell 系列服务器如 R730 2U ， IBM 服务器 X3850x6 4U

R730 配置：

CPU: E5-2609V4*2

内存: 256G(16G X16)

硬盘: 4T SAS*3 热插拔

Raid 卡: H730-1G 缓存

光驱: DVD

电源: 750W 双电

机箱规格: 2U 机架式

IBM X3850X6 4U 配置：

CPU: E7-4809V4*4

内存: 768 G(16G X 48) 上限可拓展至 6TB DDR4-2400 频率

硬盘: 4T SAS*8 热插拔

Raid 卡: 730-8i 1G 缓存

电源: 750W 四电

网卡: 双口 HBA 光纤卡

机箱规格: 4U 机架式

编程与应用题

1.添加一个新组为 class01,然后， 添加属于这个组的 30 个用户，

用户名的形式为 stdXX,其中, XX 从 01 到 30

```
#!/bin/bash
groupadd class1
for ((i=1;i<=30;i++))
do
if [ $i -lt 10 ];then
username="std0"$i
else
username="std"$i
fi
useradd -G class1 $username
done#!/bin/bash
groupadd class1
for ((i=1;i<=30;i++))
do
if [ $i -lt 10 ];then
username="std0"$i
else
username="std"$i
fi
useradd -G class1 $username
done
```

2、写一个自动备份 mysql 的脚本

需求: Mysqldump 需要备份的库 时间 频率 除旧

```
mkdir /bak
mkdir /bak/mysql
mkdir /bak/backup
vim mysql.sh
#!/bin/bash
# backup
cc=`date +%F`
mysqldump -uroot -p -p' passwd' -B HA >/bak/mysql/mysql.${cc}.sql >/dev/null
2>&1
cd /bak/mysql/
tar czf /bak/backup/mysql.${cc}.tar.gz *.sql
rm -rf /bak/mysql/*.sql
find /bak/backup/ -mtime 7 -name 'mysql*.tar.gz' -exec rm -rf {} \;
```

3.编写 shell 脚本获取本机的网络地址。

比如: 本机的 ip 地址是:

192.168.100.2/255.255.255.0 , 那么它的网络地址是

192.168.100.1/255.255.255.0

方法一:

```
1. #!/bin/bash
2. #This script print ip and network
3. file= "/etc/sysconfig/network-scripts/ifcfg-eth0"
4. if [ -f $file ];then
5. IP=`grep "IPADDR" $file|awk -F=" '{ print $2 }`
6. MASK=`grep "NETMASK" $file|awk -F=" '{ print $2 }`
7. echo "$IP/$MASK"
8. exit 1
```

9. fi

方法二:

```
1. #!/bin/bash
2. #This programm will printf ip/network
3. #
4. IP=`ifconfig eth0 |grep 'inet ' |sed 's/^.*addr://g'|sed 's/ Bcast.*$//g`
5. NETMASK=`ifconfig eth0 |grep 'inet '|sed 's/^.*Mask://g`
6. echo "$IP/$NETMASK"
7. exit
```

4、写一个脚本,实现判断 192.168.1.0/24 网络中当前在线的 IP

有哪些,能 ping 通则认为在线?

```
[root@Bob10 ~]# vim /root/ping.sh
#!/bin/bash
i=0
while [ $i -le 255 ]
do
    ping -c 3 192.168.1.$i &> /dev/null
    if [ $? == 0 ]
    then
        echo "192.168.1.$i"
    fi
    ((i++))
done
```