

云计算面试题



第 二 阶 段

2019 版

第二阶段面试题

选择题

1. 用(**AC**)命令来显示当前目录
 - A. echo \$(pwd)
 - B. echo pwd
 - C. Pwd
 - D. pwd | echo
2. (多选) bash 中, 需要将脚本 demo.sh 的标准输出和标准错误输出重定向至文件 demo.log, 以下哪些用法是正确的(**AC**)
 - A. bash demo.sh &>demo.log
 - B. bash demo.sh>&demo.log
 - C. bash demo.sh >demo.log 2>&1
 - D. bash demo.sh 2>demo.log 1>demo.log
3. (多选) 有命令:
awk' BEGIN { Arr[2,79]=78; _____ }'
为了让执行的结果是 78, 应该填(**BCD**)语句
 - A. print Arr["2/03479"]
 - B. print Arr[2,79]
 - C. idx=2 SUBSEP 79;print Arr[idx]
 - D. print Arr[2, 79]
4. 在 shell 中变量的赋值有四种方法, 其中, 采用 name=12 的方法称(**A**)
 - A. 直接赋值
 - B. 使用 read 命令
 - C. 使用命令行参数
 - D. 使用命令的输出
5. 关于代理服务器的论述, 正确的是(**A**)
 - A. 使用 internet 上已有的公开代理服务器, 只需配置客户端
 - B. 代理服务器只能代理客户端 http 的请求
 - C. 设置好的代理服务器可以被网络上任何主机使用
 - D. 使用代理服务器的客户端没有自己的 IP 地址
6. 一个 bash shell 脚本的第一行是(**D**)
 - A. #/bin/csh

- B. #/bin/bash
- C. /bin/bash
- D. #!/bin/bash

7. 命令 `ifconfig enol6777736 | awk 'NR==4 {print $2}'` 得到结果是 (C)

网卡 enol6777736 信息如下:

```
enol6777736: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
inet 192.168.30.212 netmask 255.255.255.0 broadcast 192.168.30.255
inet6 fe80::20c:29ff:fe3e:2ef9 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:3e:2e:f9 txqueuelen 1000 (Ethernet)
RX packets 866254 bytes 54347914 (51.8 MiB)
RX errors 0 dropped 10391 overruns 0 frame 0 TX packets 5002 bytes 433839
(423.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- A. 1
- B. 192.168.30.212
- C. 00:0c:29:3e:2e:f9
- D. 2

8. 命令 `echo a777helloworld | sed 's/.*\([0-9]\{8\}\).*/\1/'` , 得到的结果是 (B)

- A. 20140501
- B. a777helloworld
- C. txt
- D. 20140501.txt

9. 在 shell 编程中关于\$2 的描述正确的是 (C)

- A. 程序后携带了两个位置参数
- B. 宏替换
- C. 程序后面携带的第二个位置参数
- D. 携带位置参数的个数

10. 在 VPN 技术中, 属于三层隧道协议的是 (C)

- A. L2TP
- B. PPTP
- C. Ipsec
- D. L2F

11. 不是进程和程序的区别 (B)

- A. 程序是一组有序的静态指令, 进程是一次程序的执行过程
- B. 程序只能在前台运行, 而进程可以在前台或后台运行
- C. 程序可以长期保存, 进程是暂时的
- D. 程序没有状态, 而进程是有状态的

12. 不是 shell 具有的功能和特点的是 (D)
- A. 管道
 - B. 输入输出重定向
 - C. 执行后台进程
 - D. 处理程序命令
13. 下列对 shell 变量 FRUIT 操作, 正确的是 (C)
- A. 为变量赋值: \$FRUIT=apple
 - B. 显示变量的值: fruit=apple
 - C. 显示变量的值: echo \$FRUIT
 - D. 判断变量是否有值: [-f "\$FRUIT"]
14. 在 DNS 配置文件中, 用于表示某主机别名的是 (B)
- A. NS
 - B. CNAME
 - C. NAME
 - D. CN
15. 下列变量名中有效的 shell 变量名是 (C)
- A. -2-time
 - B. _2\$3
 - C. trust_no_1
 - D. D、2004file
16. 在 Shell 脚本中, 用来读取文件内各个域的内容并将其赋值给 Shell 变量的命令是 (D)
- A. fold
 - B. Join
 - C. tr
 - D. read
17. 退出交互模式的 shell, 应键入 (B)
- A. <Esc>
 - B. exit
 - C. ^q
 - D. quit
18. (多选) 下面 (BC) 命令可以查看 file1 文件的第 300-500 行的内容
- A. cat file1 | tail -n +300 | head -n 200
 - B. cat file1 | head -n 500 | tail -n +300
 - C. sed -n '300,500p' file1
 - D. sed -n '300;500p' file1

19. 下面哪个命令是用来定义 shell 的全局变量 (D)

- A. exportfs
- B. alias
- C. exports
- D. export

简答题:

1. 请问 subversion 的主要功能是什么? 它的常用命令有哪些?

SVN 是一个版本控制软件, 主要功能为版本控制, 支持日志、数据恢复等功能。

可以实现下列功能:

svn co 下载代码, svn ci 提交代码, svn log 查看日志,

svn update 更新代码, svn diff 查看代码差异, svn rm 删除, svn merge 合并。

工作中的主要应用:

代码管理 (通过版本库上线代码到服务器)

文档管理 (技术指导手册的版本控制)

2. 编写 shell 程序: 实现自动删除 50 个账号的功能。账号名为 stud1 至 stud50。

```
#!/bin/bash
for i in {1..50}
do
    userdel -r stud$i
done
```

3. 正则表达式中各个元字符 \n 的含义是 (换行符),

\s 的含义是 (不可见字符, 如空格, tab 等),

\f 的含义是 (换页符), 预定义的 [:graph:] 的含义是 (出空白外的字符)

4. 脚本实现: 查找目录 /var/log/ 下大于 1M 的文件移动至 /tmp/log 下。

```
#find / -size +1G -exec mv {} /tmp \;
```

5. 描述一次在使用 nginx 中遇见的问题, 及解决过程?

配置文件修改错误, 导致无法启动

根据提示检查配置, 修改为正确的内容

6. 写一个脚本, 实现判断 192.168.1.0/24 网络里, 当前在线的 IP 有哪些, 能 ping 通则认为在线。

```
#!/bin/bash
for i in {1..254}
do
    ping -c2 -i0.2 -W1 192.168.1.$i &>/dev/null
    [ $? -eq 0 ] && echo "192.168.1.$i is up"
```

done

7. 写出 LVS 负载与 Nginx 相比优点和缺点

LVS 为 4 层调度，不能实现 7 层调度，优点是集成在内核，可以实现内核级别的调度，调度算法非常丰富

Nginx 可以做 4 层或 7 层调度，对 7 层的设置如正则，httpd 头部分分析和处理的功能强大，但内置算法相对比较少

8. 进程和程序的区别是

程序是代码的集合，程序保存在硬盘

进程是程序执行的结果，进程运行在内存中

9. 请写出你所使用过或知道的 Nginx 模块名称

执行 ./configure 脚本时，通过添加 --with-模块名的方式启用更多模块，通过添加 --without-模块名的方式禁用不需要的模块。

| | |
|---------------------------------|--------------------|
| ngx_http_auth_basic_module 模块: | 实现网站的用户认证功能 |
| ngx_http_charset_module 模块: | 可以自定义网页编码（如 UTF-8） |
| ngx_http_fastcgi_module 模块: | 转发请求给 PHP 服务 |
| ngx_http_gzip_module 模块: | 实现网站数据压缩功能 |
| ngx_http_log_module 模块: | 记录用户请求的日志 |
| ngx_http_rewrite_module 模块: | 地址重写模块 |
| ngx_http_ssl_module 模块: | 实现 HTTPS 加密网站 |
| ngx_http_stub_status_module 模块: | 显示 Nginx 服务状态 |
| ngx_http_upstream_module 模块: | 定义集群服务器组模块 |
| ngx_stream_core_module 模块: | 定义 4 层调度（集群服务器组）模块 |

10. 请写出你所使用过或知道的高可用架构组件名称，并标注其用途

Keepalived 可以通过浮动 IP 实现服务器的高可用架构，在结合 LVS 调度器，可以将用户请求调度给后端若干台服务器，实现整体集群的高可用架构

11. 请写出 sed -n 1p testfile 这个命令的作用

显示文件的第一行

12. 请写出 LVS 的工作模式、至少写出两种

NAT 模式，DR 模式，Tun 隧道模式

13. Linux 下访问一个 WEB 页面可以用什么命令？

curl, elinks

14. arp 攻击和 DOS 攻击是什么？

ARP 攻击就是通过伪造 IP 地址和 MAC 地址实现 ARP 欺骗，能够在网络中产生大量的 ARP 通信量使网络阻塞，攻击者只要持续不断的发出伪造的 ARP 响应包就能更改目标主机 ARP 缓存中的 IP-MAC 条目，造成网络中断或中间人攻击。

DOS 攻击是制造大流量无用数据，造成通往被攻击主机的网络拥塞，使被攻击主机无法正

15. awk 打印文件逗号分隔的最后一列内容

`awk -F, ' {print $NF}' 文件名`

16. sed 替换文件内容 3 至 10 行 abc 替换为 ABC

`sed '3,10s/abc/ABC' 文件名`

17. 文件 a 内容 aaa\nbbb\nccc , 文件 b 内容 aaaa\nbbb\ddd , 使用任意程序语言写一个小程序实现打印出两个文件内容的交集行内容。

`sort a.txt b.txt | uniq -d`

18. 从 cache.log 中提取包含“WARNING”或“FATAL”但不包含“IGNOR”字段的行，并提取以“:”分隔的每行最后一个字段。

`egrep "(WARNING|FATAL)" cache.log | egrep -v LGNOR | awk -F: ' {print $NF}'`

19. 用一行命令实现：查找当前目录下（含子目录），文件内容中含有 sina 且文件名以“.config_”结尾的文件。

`find ./ -name "*.config_" -exec grep sina {} \;`

20. 脚本编程 1. 请看下面一段日志文件：

```
[huanw@CCN-BJ-5-5IQ lancome]$ more log_wang_2215
220.170.212.209 [07/Dec/2009:22:16:00 200 2098
http://www.mavbellinechina.com/whats new/doublestretch.aspx
113.128.140.233 [07/Dec/2009:22:16:00 200 3065
http://www.mavbellinechina.com/whats
new/2009mineralcreamv.aspx?aid=39624&cid=20996
221.10.8.223 [07/Dec/2009:22:16:00 200 259341
http://www.maybellinechina.com/whats__new/doublestretch%2fmedias%2fmain.swf
220.187.49.199 [07/Dec/2009:22:16:00 304 218
http://www.mavbellinechina.com/whats new/doublestretch.aspx
```

第 1 列：IP 第 2 列：时间 第 3 列：响应代码 第 4 列：字节大小 第 5 列：url

提示：同一个 IP 不止访问一次，同一个 url 不止被访问一次

注：仅需写出脚本代码

1) 请将日志中的 url 单独输出到 url.log 文件

`awk ' {url[$5]++}END{for(i in url){print i}}' log_wang_2215`

2) 请列出一共有多少个 IP 来访问，并统计出访问次数最多的前 5 个 IP

`awk ' {ip[$1]++}END{for(i in ip){print i,ip[i]}}' a |sort -rk 2 |head -5
log_wang_2215`

3) 请算出该日志文件中所有 url 访问流量总和

`awk ' {x+=$4}END{print x}' log_wang_2215`

- 4) 请算出该日志文件中每个 url 的访问流量, 并根据流量从大到小依次排出访问流量最大的前 10 个 url

```
awk ' {ip[$1]+=$4} END {for(i in ip){print i,ip[i] } }' a |sort -rk 2 | head -10
```

21. httpd 这个服务启动命令为/etc/init.d/httpd start, 请写一个脚本, 当服务宕掉, 脚本自动拉起 httpd 服务。(提示 5s 查检一次)

```
#!/bin/bash
while :
do
    netstat -anptu |grep -q httpd
    [ $? -eq 0 ] || /etc/init.d/httpd start
    sleep 5
done
```

22. 设计一个 shell 程序, 在 2012 年 12 月 23 日凌晨 3 点备份并压缩前一天 /svn 目录的所有内容, 存放在/root/bak 目录里, 且文件名为如下形式 svn. 2008. 05. 06. tar. gz, 试写脚本。

```
#!/bin/bash
t=`date +%Y`
[ $t -eq 2012 ] || exit
echo "1 3 23 12 * tar -czf /root/bak/svn.`date +%Y.%m.%d`.tar.gz /svn"
" >> /var/spool/cron/root
```

23. sed 命令配置关闭 selinx (一条命令)

```
sed -i '/^SELINUX=/s/enforcing/permissive/'
```

24. 机器非常卡时 (top 一次命令执行返回非常慢) 要查找出使用率高的进程 (一条命令)

```
ps aux|head -1;ps aux|grep -v PID|sort -rn -k +3|head -1
```

25. 编写 shell 程序, 实现自动删除 50 个账号的功能。账号名为 stud1 至 stud50

```
#!/bin/bash
for i in {1..50}
do
    userdel stud$i
done
使用循环, 求 1+3+5. ...+1001 的和是? (所有奇数和)
#!/bin/bash
for i in {1..1001}
do
    let n=i%2
    [ $n -eq 0 ]&&continue
    x+=i
done
```



```
echo $x
```

26. 写一个 Shell 程序，在/tmp/userdata 目录下建立 10 个目录，即 user1~user10，并设置每个目录的权限，其中其他用户的权限为：读；文件所有者的权限为：读、写、执行；文件所有者所在组的权限为：读、执行

```
#!/bin/bash
for i in {1..10}
do
mkdir /tmp/userdata/user$i
chmod 754 /tmp/userdata/user$i
done
```

27. 写一个 shell 程序，备份并压缩/etc 目录的所有内容文件，存放在/tmp/bak 目录里，且文件名为如下形式 yymmdd_etc，yy 为年，mm 为月，dd 为日

```
tar -czf /tmp/bak/`date +%Y%m%d`_etc /etc
```

31. 用你最熟悉的编程语言，使用循环打印一下如下图形.

```
*
**
***
#!/bin/bash
for i in {1..3}
do
    for a in `seq $i`
    do
        echo -n "*"
    done
echo
done
```

28. 用你最熟悉的编程语言，检查如下文本中括号是否闭合？

```
{ {name=TTY},
{sex=M}
}
```

29. 运用 shell 命令输出文本文件 log 第三列的内容，并进行去重和排序，log 内容显示如下：

1. 20 applea
2. 225 banana
3. 314 apple
4. 33 pear

```
awk '{a[$3]++}END{for(i in a){print i a[i]}}' |sort -n log
```

30. 利用 shell 命令将文本 user.txt 中所有 apple 替换成 banana

```
sed 's/apple/banana/' user.txt
```

31. 编写 crontab, 每周六、日凌晨 3 点的 0, 15, 30, 45 分各执行一次由 shell 编写的脚本
/var/crontab.sh

```
0,15,30,45 3 * * 6,7 /var/crontab.sh
```

32. 写一个脚本, 实现判断 192.168.16.0/22 网络里, 显示当前在线 IP 有哪些, 能 ping 通则认为在线

```
#!/bin/bash
for i in {1..254}
do
    ping -c 2 -i 0.2 -W 1 192.168.16.$i &> /dev/null
    [ $? -eq 0 ]&&echo "192.168.16.$i 通了"
done
```

33. 写一个脚本查找最后创建时间是 3 天前, 后缀是*.log 的文件并删除

```
find / -mtime +3 -name *.log -exec rm -rf {} \;
```

34. 写一个脚本将某目录下大于 100k 的文件移动至/tmp 下

```
find / -size +100k -exec mv {} /tmp \;
```

35. 写一个脚本将数据库备份并打包至远程服务器 192.168.1.1 /backup 目录下

```
#!/bin/bash
Data=" tar -czf `mysqldump -uroot -p 密码 -all-databases` > alldb.sql`
alldb.sql.tar.gz"
scp -r $Data 192.168.1.1:/backup
```

36. 写一个防火墙配置脚本, 只允许远程主机访问本机的 80 端口

```
firewall-cmd --permanent --zone=public --remove-service=dhcpv6-client
firewall-cmd --permanent --zone=public --remove-service=ssh
firewall-cmd --permanent --zone=public --add-service=http
```

37. 写一个脚本进行 nginx 日志统计, 得到访问 ip 最多的前 10 个(nginx 日志路径:
/home/logs/nginx/default/access.log

```
awk ' {ip[$1]++} END{for(i in ip){print i, ip[i]}} '
/home/logs/nginx/default/access.log | sort -k 2 -r | head -10
```

38. 写一个脚本把指定文件里的/usr/local 替换为别的目录

```
sed 's#/usr/local#XXX#' XXX
```

39. 写一个脚本，实现判断 192.168.1.0/24 网络里，当前在线的 P 有哪些，能 ping 通则认为在线【使用 shell/python 实现】

```
#!/bin/bash
for I in {1..254}
do
    ping -c 2 -i 0.2 -W 1 192.168.1.$i &> /dev/null
    [ $? -eq 0 ]&&echo "$i 通了"
done
```

40. 查看/var/log 目录下的文件数, 将文件内容内的“hello”, 替换为“hi”

```
sed 's/hello/hi/' /var/log
```

41. 如何查看 nginx 进程所打开的所有文件

```
rpm -ql nginx
```

42. 请解这条语句的含义: echo "1" > /proc/sys/net/ipv4/ip_forward

将 echo 输出结果覆盖重定向到 ip_forward 文件中

43. shell 脚本一般都用完成什么功能? 请选择一个简单的功能, 用 shell 脚本来完成该功能

自动化辅助运维管理服务器, 比如自动部署服务并开启

```
#!/bin/bash
yum -y install httpd
systemctl start httpd
systemctl enable httpd
```

44. 51. 通过命令行参数给定两个文件名, 如果文件不存在则结束脚本执行: 都存在时返回各个文件的行数, 并说明其中较多行数的文件

```
#!/bin/bash
[ $1 -e ]&&[ $2 -e ]||exit
w1=`cat $1 |wc -l`
w2=`cat $2 |wc -l`
if [ w1 -eq w2 ];then
echo “一样多”
elif [ w1 -gt w2 ];then
echo “w1 多”
else
echo “w2 多”
fi
```

45. 请找出下列内容中 https 证书文件及密钥文件位置, 反向代理名称, 以及启用了那些功能 (包括不限于)

```
events{
worker_connections 1024;
}

http{
include mime.types;

default_type application/octet-stream;

Fog format main Sremote add. Sremote user (time local"Srequest"
# '$status $body_bytes_sent "$http_referer" '
# "$http_user_agent" "$http_x_forwarded_for" ;

#access_log logs/access.log main;

sendfile on;
#tcp_nopush on;

#keepalive_timeout 0;
keepalive_timeout 65;

gzip on;
upstream appl{
    server192.168.0.191:443 weight=5;
    server192.168.0.185:443 weight=3;
}
server {
listen 80;
server_name 192.168.0.187;
rewrite^(.*)$ https://$host$1permanent;

server{
listen 443ssl;
listen 80;
server_name localhost www.sinoinfosec.com;

ssl_certificate /usr/local/nginx/key/www.sinoinfosec.com.crt;
sslcertificatekey /usr/local/nginx/key/www.sinoinfosec.com.key;
```

```
ssl_session_cache shared:SSL:1m;  
ssl_session_timeout 5m;
```

```
ssl_ciphers HIGH:LaNULL:IMD5;  
ssl_prefer_server_ciphers on;  
location / {  
    proxy_pass https://app1;  
    #root html;  
    #index index.html index.htm;  
}
```

证书文件/usr/local/nginx/key/www.sinoinfosec.com.crt

私钥文件/usr/local/nginx/key/www.sinoinfosec.com.key

代理集群名称 app1

功能：虚拟主机，地址重写，并发连接数，代理服务器，加密网站，保持连接

46. 假设我们有一文件名为 ab，请列“出删除第一行”、“删除最后一行”、“显示第一行”、“显示第一行到第二行”

```
sed '1d' ab  
sed '$d' ab  
sed -n '1p' ab  
sed -n '1,2p' ab
```

47. https 访问但是 ssl 证书已过期，此传输过程是否还加密？

加密

48. /data 目录下存在 n 多 *.lck 文件，如何快速删除 2 小时外的 *.lck 文件

```
find /data/*.lck -mmin +120 -exec rm -rf {} \;
```

49. awk 打印文件逗号分隔的第一列大于 2 的行

```
awk -F, '$1>2'
```

50. sed 替换文件内容 3 至 10 行 abc 替换为 ABC

```
sed '3,10s/abc/ABC/'
```

51. 文件 a 内容 aaa\nbbb\nccc, 文件 b 内容 aaaa\nddd\nbbb, 使用任意程序语言写一个小程序实现打印出仅在文件 a 中出现的行。注：文件内容\n 为回车符

```
for a in `cat a`  
do  
    for b in `cat b`  
    do  
        [ $a == $b ]&&echo $a
```

```
done  
done
```

52. 请举例说明 if、for、 while 语句的用法

```
if [ a == a ];then  
echo a  
fi  
for i in {1..10}  
do  
echo $i  
done  
while :  
do  
echo ok  
done
```

53. 说对一下脚本进行说明:

```
#!/bin/bash  
#chkconfig: 2345 55 25          # 运行级别、启动优先级、关闭优先级  
#processname: Extraauthserver   #进程名  
#description: Extraauthserver   #服务描述  
prog=/usr/local/apache/htdocs/Extrabaiduserver/Authserver/Authserver  
lock= ps -ef | grep usr/10cal/apache/htdocs/Extrabaiduserver/Authserver |  
grep -v grep | awk '{print $2}'  
start () {  
    sprog > /dev/null 2> /dev/null &  
    echo "正在启动服务..."  
}  
stop () {  
    if [[ $lock != '' ]];then  
        kill -9 $slock > /dev/null 2>/dev/null &  
        echo "正在停止服务"  
    else  
        Echo "$0 服务未运行"  
    fi  
}  
restart () {  
    stop  
    Start
```

```
}  
case "$1" in  
    "start" )  
        start  
        ;;  
    "stop" )  
        stop  
        ;;  
    "restart")  
        restart  
        ;;  
    "status" )  
        status  
        ;;  
    *)  
        echo "用法: $0 start|status|restart"  
        ;;  
esac
```

首先定义了 start 和 stop 函数，作为服务的开启与关闭，然后使用 case 分支定义了用户的命令输入，如果输入 start 则开启服务，如果输入 stop 则关闭服务，如果输入 restart 则重启服务，如果输入 status 则查看状态