

---

## 第二阶段面试题

### 一、SHELL

#### 1. 如何获取一个文件每一行的第三个元素?

答案:

```
# awk '{print $3}' 文件名
```

备注: awk 支持按列输出, 通过内置变量\$1,\$2,\$3...可以单独显示任意列, 默认列是以空格或 Tab 缩进为分隔符, 也可以使用-F 选项指定其他分隔符。

#### 2. shell 函数能解决什么实际问题?

答案:

定义函数的格式:

```
function 函数名{
```

```
    代码块
```

```
}
```

```
函数名(){
```

```
    代码块
```

```
}
```

使用函数可以避免代码重复

使用函数可以将大的工程分割为若干小的功能模块, 代码的可读性更强

#### 3. 使用 awk 统计 httpd 访问日志中每个客户端 IP 的出现次数?

答案:

```
# awk '{ip[$1]++}END{for(i in ip){print ip[i],i}}' /var/log/httpd/access_log
```

备注: 定义数组, 数组名称为 ip, 数字的下标为日志文件的第 1 列 (也就是客户端的 IP 地址), ++的目的在于对客户端进行统计计数, 客户端 IP 出现一次计数器就加 1。END 中的指令在读取完文件后执行, 通过循环将所有统计信息输出。

#### 4. 哪些方式可以将标准输出和错误输出重定向到文件?

答案:

```
# 命令 &> 文件名
# 命令 > 文件名 1 2> 文件名 2
# 命令 > 文件名 2>&1
# 命令 &>> 文件名
# 命令 >> 文件名 1 2>> 文件名 2
# 命令 >> 文件名 2>&1
```

#### 5. 正则表达式符号: \*, +, ?, [], [^], \{n\} 分别代表什么含义?

答案:

\*表示匹配前面的字符出现了任意次 (包括 0 次)

+表示匹配前面的字符出现了至少 1 次 (1 次或多次)

? 表示匹配前面的字符出现了 0 次或 1 次

[]表示集合, 匹配集合中的任意单个字符

[^]表示对集合取反

\{n\}表示精确匹配前面的字符出现了 n 次

#### 6. shell 中对变量字符串进行截取的方式有哪些?

答案:

```
# echo ${变量名: 开始位置: 长度}          #注意, 起始位置从 0 开始
# expr substr $变量名 开始位置 长度        #注意, 起始位置从 1 开始
# echo $变量名 | cut -b 开始位置-结束位置  #注意, 起始位置从 1 开始
```

#### 7. 使用 sed 命令如何将文件中所有的大写字母 Q 转换为小写字母 q?

答案:

```
# sed -i 's/Q/q/g' 文件名
```

#### 8. 执行如下命令会对 test.txt 文件产生什么样的影响?

```
# sed -i 's9\945\9\99\98\939g' test.txt
```

答案:

将 test.txt 文件中 94599 替换为 9893

## 9. 工作中你都写过什么脚本？

**答案：**

监控脚本（监控系统、监控服务、监控硬件信息、监控性能、安全监控等）

系统初始化脚本（创建目录，创建账户，安装软件包，设置权限，修改内核参数等）

一键部署（源码安装脚本）

备份脚本（自动备份数据库，备份网站数据，备份日志，备份配置文件等）

日志分析脚本（分析日志数据，汇总并统计相关信息，如 PV、UV 等）

## 9. 编写脚本，用户输入密码，脚本判断密码是否正确，输入正确则提示正确，连续输入错误密码 3 次，则报警？

**答案：**

```
vim test.sh
#!/bin/bash
init=123456
for i in {1..3}
do
    read -p "请输入密码:" pass
    if [ $pass == $init ];then
        echo "密码正确"
        break
    fi
done
echo "警告：密码错误"
```

## 10. 编写脚本，自动生成一个 8 位随机密码？

**答案：**

```
vim test.sh
#!/bin/bash
Str="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
pass=""
for i in {1..8}
do
    num=${RANDOM%${#Str]}
```

---

```
tmp=${Str:num:1}
pass+= $tmp
done
echo $pass
```

## 二、OPERATION

### 1. 源码编译安装 nginx 时，如何启用更多的模块以及禁用不需要的模块？简单

描述下你都使用过哪些模块？

答案：

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

### 2. 如何启动 nginx 的 status 状态页面功能？

答案：

源码安装 nginx 时需要添加--with-http\_stub\_status\_module 模块，安装好后修改 nginx 主配置文件，添加如下内容：

```
location /status {
    stub_status on;
}
```

---

### 3. Nginx 支持哪些类型的虚拟主机，如何配置？

**答案：**

基于端口的虚拟主机

基于域名的虚拟主机

基于 IP 的虚拟主机

这里提供一个模板，根据需要修改对应的内容即可。

```
server {  
    listen IP 地址:端口;  
    server_name 域名;  
    location / {  
        root html;  
        index index.html  
    }  
}
```

### 3. Nginx 支持哪些类型代理服务器，如何配置？

**答案：**

基于 7 层的 HTTP 代理和 MAIL 代理

基于 4 层的 TCP 或 UDP 代理

定义集群服务器模板配置：

```
upstream servers {  
    server 后端真实服务器 IP:端口;  
    server 后端真实服务器 IP:端口;  
}
```

调用集群的配置文件模板：

7 层调度：proxy\_pass <http://servers;>

4 层调度：proxy\_pass servers;

### 4. 如何修改 Linux 内核参数，调整进程可以打开的最大文件数量？

**答案：**

临时规则

# ulimit -Hn 数量

# ulimit -Sn 数量

永久规则

---

```
#vim /etc/security/limit.conf
用户或组 soft  nofile  数量
用户或组 hard  nofile  数量
```

## 5. HTTP 常见状态码有哪些，分别是什么含义？

答案：

状态码	功能描述
200	一切正常
301	永久重定向
302	临时重定向
401	用户名或密码错误
403	禁止访问（客户端 IP 地址被拒绝）
404	文件不存在
414	请求 URI 头部过长
500	服务器内部错误
502	Bad Gateway

## 6. git 的主要功能是什么？你都用她来做什么？

答案：

git 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：

git clone 克隆代码，git add 添加文件，git commit 提交代码，svn log 查看日志，  
git push 推送代码，git pull 更新代码，git branch -v 查看分支，git checkout 切换分支。

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）

## 7. 客户端无法连接服务时，如何查看服务器的服务是否启动，端口是否开放？

答案：

netstat 或 ss 命令可以查看系统中启动的端口信息。

该命令常用选项如下：

-a 显示所有端口的信息

- 
- n 以数字格式显示端口号
  - t 显示 TCP 连接的端口
  - u 显示 UDP 连接的端口
  - l 显示服务正在监听的端口信息，如 httpd 启动后，会一直监听 80 端口
  - p 显示监听端口的服务名称是什么（也就是程序名称）

## 8. Tomcat 如何开启 SSL 加密，SSL 默认的加密端口是多少？

**答案：**

开启一个独立的 Connector，设置端口，并在配置中调用非对称密钥文件，默认的 SSL 端口为 443 端口，Tomcat 配置文件中的默认设置为 8443 端口。

## 9. 工作是否使用过 VPN，用过哪些类型的 VPN？

**答案：**

GRE VPN

PPTP VPN

XL2TP+IPsec VPN

## 10. 如何将一个源码包软件转换称为一个 RPM 软件包？

**答案：**

安装 rpmbuild 工具

首先将源码包放到 SOURCES 目录下，

然后在 SPECS 目录下创建一个 spec 配置文件，并按特定格式要求编辑文件内容，

SPECS 文件可以定义：

- 软件的描述信息，
- 如何编译安装源码软件，
- 对什么文件打包封装为 RPM，
- 安装前和安装后脚本等。

最后使用 rpmbuild -ba spec 文件。

---

## 三、CLUSTER

### 1. 你使用过哪些存储产品？

答案：

NAS 网络附加存储，如 NFS, Samba, FTP

SAN 存储区域网络，如 iSCSI, HP, EMC

分布式存储，如 Ceph

### 2. 部署一个 LVS 集群环境都需要什么组件？

答案：

Director Server：调度服务器，将负载分发到 Real Server 的服务器

该主机需要安装 ipvsadm，创建虚拟集群服务器并添加 real server

Real Server：真实服务器，真正提供应用服务的服务器

该主机需要部署 LNMP 或 LAMP，

做 DR 模式的集群还需要设置 VIP 和防止 ARP 冲突

可以结合 keepalived 软件实现主备 LVS 调度器

VIP：虚拟 IP 地址，公布给用户访问的虚拟 IP 地址

RIP：真实 IP 地址，集群节点上使用的 IP 地址

DIP：调度器连接节点服务器的 IP 地址

### 3. 简述 LVS 三种工作模式，简述他们的区别？

答案：

VS/NAT：通过网络地址转换实现的虚拟服务器，大并发访问时，调度器的性能成为瓶颈；

VS/DR：使用路由技术实现虚拟服务器，节点服务器需要配置 VIP，注意 MAC 地址广播；

VS/TUN：通过隧道方式实现虚拟服务器。

### 4. 列举你知道的 LVS 调度算法？

答案：

轮询 (Round Robin)；

加权轮询 (Weighted Round Robin)；

最少连接 (Least Connections)；

加权最少连接 (Weighted Least Connections )

源地址哈希值 (source hash)。



---

## 6. Ceph 提供了哪些存储方式，客户端通过什么方式访问 Ceph 存储？

**答案：**

块设备的方式：

服务器需要 MON 和 OSD 组件实现块设备共享；

客户端可以使用 Linux 系统直接访问或者通过 KVM 访问共享存储；

Linux 直接访问时，客户端需要安装 ceph-common，并拷贝配置和密钥文件；

通过 KVM 访问时，需要修改 KVM 虚拟机的 XML 文件并定义 Secret 账户。

文件系统的方式

服务器需要额外部署一台 MDS 主机（存储元数据 metadata）

客户端直接通过 mount 或者/etc/fstab 实现挂载访问

客户端挂载时需要提供服务器的 IP 地址、端口、账户和密码

对象存储的方式

服务器需要额外部署一台 RGW 主机（提供 web 服务的接口）

客户端无法直接访问对象存储，必须使用第三方软件

或者开发人员自己编写程序调用 Ceph 的 API

API（是 Ceph 提前准备好的函数，开发人员直接调用这些函数就可以对 Ceph 做读写操作）

## 7. Keepalived 通过什么判断哪台主机为主服务器，通过什么方式配置浮动 IP？

**答案：**

Keepalived 首先做初始化先检查 state 状态，master 为主服务器，backup 为备用服务器。然后再对比所有服务器的 priority，谁的优先级高谁是最终的主服务器。

优先级高的服务器会通过 ip 命令为自己的电脑配置一个提前定义好的浮动 IP 地址。