云计算-阶段 2-必备知识

Shell 编程 ——

##
##########

Day01

01. 编写脚本时，如何将错误的信息重定向导出，生成一份日志文件？

根据情况，看看是否需要覆盖或追加，选择 2>或者 2>>可以将错误信息重定向导出，从而生成一份日志文件。

重定向符号的使用详情如下表：

重定向符号	功能描述
>	标准正确重定向（覆盖）
2>	标准错误重定向（覆盖）
>>	标准正确重定向（追加）
2>>	标准错误重定向（追加）
&>	标准正确和错误重定向（覆盖）
<	重定向导入

02. 简单描述执行脚本的不同方式及其区别（假设脚本名称为 test.sh）？

对于有执行权限的脚本可以使用相对路径或绝对路径执行

对于没有执行权限的脚本可以使用 bash 或 source 执行脚本

使用 source 执行脚本时不会开启子进程，其他的方式会开启子进程

执行效果如下：

```
# ./test.sh                    //需要脚本有可执行权限，执行脚本会开启子进程
# bash test.sh                 //不需要脚本有可执行权限，执行脚本会开启子进程
# source test.sh           //执行脚本，不开启子进程
```

03. 编写 shell 脚本时，定义变量名有哪些规则要求？

变量名称可以使用数字、字母、下划线(_)

变量名称不能以数字开始，变量名称不可以使用特殊符号

04. 编写 shell 时，如何让脚本支持位置参数？

使用位置变量可以让脚本能够读取位置参数，如$1,$2,$3,$4 等等。

05. Shell 如何判断上一个命令的执行结果是否成功？

使用$?可以判断上一个命令的执行结果是否成功，通常 0 代表成功，非 0 代表失败

Day02

01. 请写出下面运算指令的执行结果分别是多少？

```
# i=5
# expr  3 + $i
# echo $[i*2]
# let i++; echo $i
# echo "scale=2;12/$i" |bc
```

结果依次为：
8
10
6
2.00

02. 判断 Linux 当前登录用户的数量，登录数量大于 3，则发送邮件报警,邮件内容在 mail.txt 文件中

```
# num=$(who |wc –l)
# [  $num  -gt  3  ] && mail -s  Warning  root  < mail.txt
```

其他 Shell 支持的判断条件如下表：

判断符号	功能描述
==	测试字符串是否相同
!=	测试字符串是否不同
-z	测试字符串是否为空
-eq	测试数字是否相等
-ne	测试数字是否不相等
-gt	测试数字是否大于
-ge	测试数字是否大于等于
-lt	测试数字是否小于
-le	测试数字是否小于等于
-e	测试文件或目录是否存在
-f	测试是否存在且为文件
-d	测试是否存在且为目录

03. 编写脚本，判断当前用户是否为 root，如果是则安装 httpd 软件，否则提示“当前用户不是管理员”。

```
# cat test.sh
#!/bin/bash
if [  $USER == root  ];then
      yum –y install httpd
```

```
else
        echo "当前用户不是管理员"
fi
```

Day03

01. 编写脚本，循环读取用户输入的用户名和密码，判断用户名是否为 tom，密码是否为 123456，匹配则脚本提示登录成功，否则提示重试，失败 3 次则脚本退出？

for 循环的版本：

```
# cat test.sh
for   i   in   {1..3}
do
            read   -p   "请输入用户名:"     user
            read   -p   "请输入密码":        pass
            if   [   $user   ==   "root" ] && [ $pass –eq 123456 ];then
                        echo "登录成功"
                        exit
            else
                        echo "用户名或密码错误，请重新输入:"
            fi
done
```

while 循环的版本：

```
# cat test.sh
i=1
while [ $i –le 3 ]
do
            read   -p   "请输入用户名:"     user
            read   -p   "请输入密码":        pass
            if   [   $user   ==   "root" ] && [ $pass –eq 123456 ];then
                        echo "登录成功"
                        exit
            else
                        echo "用户名或密码错误，请重新输入:"
            fi
```

```
        let i++
done
```

02.在编写 shell 脚本时 break 和 continue 命令有什么区别？

break 和 continue 都是用来中断循环语句的命令。break 会结束整个循环体；continue 仅结束当前循环，跳入下一次循环中。

Day04

01. 编写一个生成 6 位随机密码的脚本？

```
# cat   test.sh
#!/bin/bash

str="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
pass=""
for i   in   {1..6}
do
      num=$[RANDOM%62]
      tmp=${str:$num:1}
      pass=$pass$tmp
done
echo   $pass
```

02. 请写出正则表达式的符号及其功能描述。

基本正则列表

正则符号	功能描述
^	匹配行首
$	匹配行尾
[]	集合，匹配集合中的任意单个字符
[^]	对集合取反
.	匹配任意单个字符
*	匹配前一个字符任意次数
\{n,m\}	匹配前一个字符 n 到 m 次
\{n\}	匹配前一个字符 n 次
\{n,\}	匹配前一个字符 n 次以上

\(\)	保留

扩展正则列表

正则符号	描述
+	最少匹配一次
？	最多匹配一次
{n,m}	匹配 n 到 m 次
()	组合为整体，保留
|	或者
\b	单词边界

Day05

01.使用 sed 修改 httpd.conf 配置文件中的 DocumentRoot，将默认网站根路径从 /var/www/html 修改为/data/web?

```
# sed   -i    '/DocumentRoot/s#/var/www/html#/data/web#'   /etc/httpd/conf/httpd.conf
```

其他 sed 常用指令列表如下：

指令	描述
a	追加新内容到文件
I	插入新内容到文件
c	修改内容（行修改）
s	替换内容(关键词替换)
d	删除（行删除）
p	打印

Day6

01.使用 awk 分析 Apache 日志，统计每个客户端 IP 访问服务器的次数？

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

02.如何使用 awk 提取 Linux 主机的参数信息，如 CPU 负载，内存剩余容量，根分区剩余容量，本机 IP 地址，本机能登录的用户个数等信息？

```
# df |awk '/\//{print $4}'                                  //根分区剩余容量

# free | awk   '/Mem/{print $4}'                           //内存剩余容量

# uptime | awk   '{print $10}'                                    //CPU 负载信息

# ifconfig enp3s0   | awk   '/pack/{print $5}'     //本机 IP 地址

# awk –F: '/bash$/{x++}   END{print "本机能登录的用户量:",x}'   /etc/passwd
```

服务器运维技术 ——

##

Day01

01. Nginx 如何实现网站的用户认证？

Nginx 默认就支持用户认证功能 ,在配置文件中添加 auth_basic 和 auth_basic_user_file 即可，其中 auth_basic_user_file 定义了存储用户名和密码的文件名称，最后通过 htpasswd 命令创建对应的账户和密码即可。

```
# cat /usr/local/nginx/conf/nginx.conf
… …
server {
            … …
      auth_basic   "Prompt:";
      auth_basic_user_file "/usr/local/nginx/pass";                    //指定密码文件
}
```

```
# htpasswd –cm   /usr/local/nginx/pass     tom                    //创建密码文件及账户
```

02. 制作 HTTPS 加密网站时，如何生成私钥和证书？

```
# openssl genrsa -out cert.key                                    //生成私钥

# openssl req-new -x509 -key cert.key-out cert.pem                //生成证书
```

03.Nginx 实现 HTTPS 加密网站，编译安装时需要启用什么模块？

编译时使用./configure --with-http_ssl_module 命令启动 ngx_http_ssl_module 模块。

Day02

01. 创建 LNMP 环境所需要安装的软件包列表有哪些，她们的作用分别是什么？

Linux：操作系统
Nginx：Web 服务器，接受用户的 HTTP 请求
mariadb-server：数据库服务器
mariadb：数据库客户端软件
mariadb-devel：依赖包
php：PHP 解释器
php-fpm：PHP 服务
php-mysql：PHP 连接 mysql 的扩展包。

02. Nginx 如何实现页面的动静分离？

Nginx 可以使用 location 匹配用户的请求，根据正则表达式判断用户访问的是静态页面还是动态页面。
样例配置如下：

```
# cat   /usr/local/nginx/conf/nginx.conf
… …
      server{
            listen 80;
            server_name localhost;
            location / {                                          //识别静态页面（除了 PHP 外的其他所有页面）
                  root html;
                  index index.html;
            }
            location ~* \.php$ {                                  //识别动态页面
                  root html;
                  fastcgi_pass 127.0.0.1:9000;
                  include   fastcgi.conf;
            }
      }
```

03. Nginx 如何识别不同的浏览器返回不同的页面，如 UC、IE 浏览器？

Nginx 可以使用 http_user_agent 识别用户的浏览器，另外可以使用 rewrite 设置不同浏览器返回不同的页面。
样例配置如下：

```
rewrite    ^/       http://www.tmooc.cn;                              //访问任何页面跳转网页到 www.tmooc.cn
if ($http_user_agent ~* msic){
            rewrite ^/(.*)    /msie/$1;                          //如果用户的浏览器是 MSIE，则跳转页面
}
```

04. Nginx 如何将所有对 a.html 页面的请求跳转到 b.html 页面？

```
rewrite    a\.html    b.html;                                    //访问 a.html 跳转到 b.html
```

Day03

01. 配置 nginx 反向代理，实现 Web 集群调度器，应该使用什么语句？

Nginx 可以使用 upstream 定义集群，默认算法为轮询调度，可以修改为 ip_hash 和最少连接算法。
具体配置文件模板如下：

```
# cat /usr/local/nginx/conf/nginx.conf
… …
http{
    upstream servers {
        #ip_hash;
        server 192.168.2.100 weight=2 max_fails=2 fail_timeout=30;
        server 192.168.2.200;

    server {
        listen 80;
        server_name localhost;
        location / {
            proxy_pass http://servers;
        }
    }
}
```

02. Nginx 如果需要实现 TCP/UDP 的代理服务器功能，需要安装什么模块？

Nginx 采用模块化设计，使用--with-stream 模块可以实现 TCP/UDP 代理功能。

03. 说说你是如何优化 nginx？

我们可以从以下几个方面着手优化 nginx：

1. 优化并发数量（修改 nginx 配置并且修改 Linux 系统配置）

2. 增加浏览器缓存机制
3. 启用页面压缩功能
4. 设计自定义报错页面

代码示例如下：

```
# ulimit –Hn 100000                                    //max open files
# ulimit –Sn 100000                                    //max open files
# cat /usr/local/nginx/conf/nginx.conf
worker_processes 2;                                    //与 CPU 核心数量一致
worker_connections 60000;                                    //每个 worker 最大并发连接数
location ~* \.(jpg|jpeg|gif|png|css|js|ico|xml)$ {
      expires 30d;                                           //静态数据缓存 30 天
}
error_page    404     /404.html;                       //自定义报错页面
```

Day04

01.简单描述 memcached 的功能？

Memcached 是一款：缓存数据库、KV 数据库、noSQL 数据库
她可以缓存 SQL 数据库的数据，用来加速 SQL 数据库的访问速度，加速整体网站的访问速度
使用 memcached 缓存可以共享网站的 Session 会话信息
PHP 连接 memcached 数据库是需要安装 php-pecl-memcache

02. Memcached 写数据和查数据的指令分别是什么？怎么删除数据？

Memcached 中写使用 set 指令、查数据使用 get 指令、删除数据使用 delete 指令。

03. PHP 默认会把 Session 会话信息保存在哪里？

默认 php-fpm 配置文件中定义的 Session 会话信息保存在/var/lib/php/session 目录。

04. 怎么修改 php-fpm 的配置文件，可以实现使用 memcached 实现 session 共享？

```
[root@web1 ~]# vim    /etc/php-fpm.d/www.conf                  //修改该配置文件的两个参数
//文件的最后 2 行
修改前效果如下:
php_value[session.save_handler] = files
php_value[session.save_path] = /var/lib/php/session
//原始文件，默认定义 Sessoin 会话信息本地计算机（默认在/var/lib/php/session）
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
修改后效果如下:
php_value[session.save_handler] = memcache
php_value[session.save_path] = "tcp://192.168.2.5:11211"
```

```
//定义 Session 信息存储在公共的 memcached 服务器上，主机参数中为 memcache（没有 d）

//通过 path 参数定义公共的 memcached 服务器在哪（服务器的 IP 和端口）
```

Day05

01. Tomcat 默认启动的端口号是多少？

8080 端口、8009 端口、8005 端口。

02. JDK 与 JRE 的区别是什么？

JDK 是 java 开发工具组包，里面包含了 java 开发需要的编译工具、排错等工具，同时还包含有运行 java 代码所需要的解释器等工具；
而 JRE 仅仅包含运行 java 代码所需要的工具，JRE 是 JDK 的子集。

03. 简单说明 tomcat 架构

Connector 负责监听端口接受用户的请求，并将请求转发给 Engine，Engine 判断用户访问的目标网站，将请求调度给具体的虚拟主机，<Host>定义虚拟主机名称、定义网站的根路径，定义网站相关属性配置>

04. 简单描述 varnish 的功能？

Varnish 可以使用内存或文件做缓存，性能和速度更高，使用 Varnish 对网站的页面内容（如图片，视频、音频等资料缓存），使用代理将数据缓存在不同地区，结合 DNS 分离解析，实现不同客户访问离自己最近的代理服务器，加速 Web 的访问速度。

Day06

01.常用 subversion 命令：

```
# svnadmincreate /var/svn/project1                              //创建仓库
# svnimport ./ file:///var/svn/project1/init.d -m "Init"        //导入代码到仓库
# svn co svn://10.47.214.131/ code                              //将代码从服务器下载到本地副本
# svn ci –m "test"                                                  //提交代码到服务器
# svn update                                                            //更新本地副本上的代码
# svn log    svn://10.47.214.131                                    //查看版本库的日志
# svn list svn://10.47.214.131                                  //列出服务器文件列表
# svn add test.sh                                                   //新建文件
# svn mkdir    test                                                     //创建目录
# svn cat svn://10.47.214.131/test.sh                               //查看服务器仓库中的代码文件内容
```

```
# svn diff                                                    //对比本地副本与服务器仓库的差异
# svn revert test.sh                                          //回滚数据（还原数据）
# svn merge -r 10:5 test.sh                                   //还原数据到第 5 个版本
# svn co –r2 svn://10.47.214.131    code2                     //下载特定版本的代码到本地
```

02.将源码包转换为 RPM 需要什么工具：

```
# yum –y install rpm-build
```

04. rpm-build 将源码包制作成 RPM 时，需要写什么配置文件？
源码制作 RPM 需要写 spec 文件。

Day07

01.创建 VPN 的常用方式有哪些？
GRE VPN、PPTP VPN、XL2TPD+IpSec VPN、SSL VPN。

02.Linux 系统使用什么命令可以查看、动态加载内核模块？
lsmod 可以查看已经加载的内核模块
modprobe 可以动态加载新的内核模块

03.Linux 系统使用什么命令可以创建 GRE VPN？

```
# ip tunnel add tun0    mode gre \
>    remote 201.1.2.5 local 201.1.2.10
```

04.Linux 系统如何开启路由转发功能？

```
[root@proxy ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

05.NTP 服务采用分层设计，Stratum 层数最多不能超过多数？
最多不能超过 15 层

06.RHEL7 系统中默认使用的时间服务器软件是什么？
是 chrony

07.使用 pssh 可以实现什么功能？
1）可以使用密码批量、多并发远程其他主机
2）使用密钥批量、多并发远程其他主机
3）批量、多并发拷贝数据到其他主机
4）批量、多并发从其他主机下载数据到本机
5）批量、多并发杀死其他主机的进程

安全与监控 ——

##

Day01

01.在 Linux 系统中设置账户有效期，以及临时锁定账户的命令是什么？

```
# chage -E 时间   账户                    #使用该命令可以设置账户有效期

# passwd -l   账户                        #使用该命令可以临时锁定一个账户
```

02. 在 Linux 系统中设置特殊属性锁定文件的命令是什么？

```
# lsattr   文件名                         #使用该命令可以查看文件的特殊属性

# chattr   +i   文件名                    #使用该命令可以锁定一个文件
```

03. 在 Linux 系统中 sudo 提权需要修改哪个配置文件？

修改/etc/sudoers 文件可以实现给普通用户提权的功能。

04. 如何通过 sudo 授权 tom 执行 useradd 命令？

```
[root@localhost ~]# vim   /etc/sudoers
… …
tom     ALL=(ALL)     /usr/sbin/useradd
```

05. 如何设置 SSH 属性，禁止 root 远程登陆，如何设置 ssh 黑白名单？

```
[root@localhost ~]# vim   /etc/ssh/sshd_config
… …
PermitRootLogin no                                          #禁止 root 远程登陆

AllowUsers zhangsan tom useradm@192.168.4.0/24              #定义账户白名单

##DenyUsers   USER1   USER2                                  #定义账户黑名单
```

06. 如何关闭 SELinux？

```
[root@proxy ~]# setenforce 0                    #临时关闭
    [root@proxy ~]# vim /etc/selinux/config
    SELINUX=disable                             #永久禁用 SELinux
```

Day02

01. 对称密钥、非对称密钥常用的算法有哪些？信息摘要的算法有哪些？

常用对称算法：AES、DES

常用非对称算法：RSA、DSA

常用信息摘要算法：MD5、SHA512

02. 简单描述 AIDE 软件的功能？

aide 对文件数据进行权限、账户、哈希值等校验，并将校验结果写入数据库，在系统被人攻击后，使用 aide 再次对文件数据进行校验，对比两次校验的差异，从中发现被人入侵的文件。

03. nmap 常用的扫描选项有哪些？

-sT, TCP 连接扫描（全开）
-sS, TCP SYN 扫描（半开）
-sU，UDP 扫描
-sP，ICMP 扫描
-A, 目标系统全面分析

04. 使用 tcpdump 抓包时，有哪些常用选项和过滤条件？

//监控选项如下：
// -i，指定监控的网络接口（默认监听第一个网卡）
// -A，转换为 ACSII 码，以方便阅读
// -w，将数据包信息保存到指定文件
// -r，从指定文件读取数据包信息
//tcpdump 的过滤条件：
// 类型：host、net、port、portrange
// 方向：src、dst
// 协议：tcp、udp、ip、wlan、arp、……
// 多个条件组合：and、or、not

Day03

01. 审计的目的是什么，系统自带的审计软件是什么？

审计的目的是基于事先配置的规则生成日志，记录可能发生在系统上的事件（正常或非正常行为的事件），审计不会为系统提供额外的安全保护，但她会发现并记录违反安全策略的人及其对应的行为。

系统自带的审计软件为 audit。

02. Nginx 如何添加和删除模块？

重新编译安装时，可以指定需要或删除的模块，使用--with-模块，可以添加需要的模块；使用--without-模块，可以删除不需要的模块。

03. Nginx 如何限制每个用户的并发连接数？

使用 limit_req_module 模块，可以实现该功能。

具体操作如下：

```
[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
… …
http{
… …
limit_req_zone $binary_remote_addr zone=one:10m rate=1r/s;
 server {
        listen 80;
        server_name localhost;
        limit_req zone=one burst=5;
            }
}

//备注说明：
//limit_req_zone 语法格式如下：
//limit_req_zone key zone=name:size rate=rate;
//上面案例中是将客户端 IP 信息存储名称为 one 的共享内存，内存空间为 10M
//1M 可以存储 8 千个 IP 信息，10M 可以存储 8 万个主机连接的状态，容量可以根据需要任意调整
//每秒中仅接受 1 个请求，多余的放入漏斗
//漏斗超过 5 个则报错
[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

04. 常见的 HTTP 请求方法有哪些？

请求方法	功能描述

GET	请求指定的页面信息，并返回实体主体
HEAD	类似于 get 请求，只不过返回的响应中没有具体的内容，用于获取报头
POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）
DELETE	请求服务器删除指定的页面
PUT	向服务器特定位置上传资料
…	其他

05. Bash 历史命令和 mysql 历史命令的记录文件分别是什么?

bash 历史命令记录文件为~/.bash_history

mysql 历史命令记录文件为~/.mysql_history

06. 如何降级使用普通账户启动 tomcat 服务？

```
[root@web1 ~]# useradd tomcat
[root@web1 ~]# chown -R tomcat:tomcat /usr/local/tomcat/
    //修改 tomcat 目录的权限，让 tomcat 账户对该目录有操作权限
[root@web1 ~]# su -c /usr/local/tomcat/bin/startup.sh tomcat
```

07. 使用 diff 命令生成补丁文件的常用选项有哪些？

diff 命令常用选项：

-u　输出统一内容的头部信息（打补丁使用），计算机知道是哪个文件需要修改

-r　递归对比目录中的所有资源（可以对比目录）

-a　所有文件视为文本（包括二进制程序）

-N　无文件视为空文件（补丁文件中说明空文件怎么变成第二个文件）

08. Linux 系统中使用什么命令可以对程序打补丁，如何使用？

patch 命令；patch -p1 < ../xx.patch

Day04

01. iptables 默认的 4 表 5 链分别是什么，并简单描述功能？

iptables 默认的 4 个分别表是 filter、nat、mangle、raw

filter 表是过滤表，用来过滤数据包的功能；

nat 表是地址转换表，用来实现网络地址转换功能，如将内网地址转换为外网地址等；

mangle 表用来修改特定数据包的包头信息；

raw 表是跟踪表，用来跟踪数据包的状态。

iptables 默认的 5 个链分别是：INPUT，OUTPUT，FORWARD，POSTROUTING，PREROUTING

INPUT 链用来存放入站数据包的规则

OUTPUT 链用来存放出站数据的规则

FORWARD 链用来存放转发数据的规则

PREROUTING 链用来存放路由前数据包的规则

POSTROUTING 链用来存放路由后数据包的规则

02. iptables 的基本用法格式与常用选项？

基本语法格式如下：

iptables [-t 表名] 选项 [链名] [条件] [-j 目标操作]

常用选项如下：

类别	选项	用途
添加规则	-A	在链的末尾追加一条规则
	-I	在链的开头（或指定序号）插入一条规则
查看规则	-L	列出所有的规则条目
	-n	以数字形式显示地址、端口等信息
	--line-numbers	查看规则时，显示规则的序号
删除规则	-D	删除链内指定序号（或内容）的一条规则
	-F	清空所有的规则
默认策略	-P	为指定的链设置默认规则

常用目标操作指令：

ACCEPT：允许通过/放行；DROP：直接丢弃，不给出任何回应；

REJECT：拒绝通过，必要时会给出提示；LOG：记录日志，然后传给下一条规则。

03. iptables 命令的注意事项与规律？

a) 可以不指定表，默认为 filter 表
b) 可以不指定链，默认为对应表的所有链
c) 如果没有匹配的规则，则使用防火墙默认规则
d) 选项/链名/目标操作用大写字母，其余都小写

04. 常用 iptables 案例？

[root@svr7 ~]# iptables -t filter -A INPUT -p tcp -j ACCEPT

[root@svr7 ~]# iptables -I INPUT -p udp -j ACCEPT
[root@svr7 ~]# iptables -I INPUT 2 -p icmp -j ACCEPT
[root@svr7 ~]# iptables -L INPUT --line-numbers
[root@svr7 ~]# iptables -D INPUT 3
[root@svr5 ~]# iptables -t filter -P INPUT DROP

05. iptables 常用匹配规则有哪些？

类别	选项	用法
通用匹配	协议匹配	-p 协议名
	地址匹配	-s 源地址、-d 目标地址
	接口匹配	-i 收数据的网卡、-o 发数据的网卡
隐含匹配	端口匹配	--sport 源端口、--dport 目标端口
	ICMP 类型匹配	--icmp-type ICMP 类型

06. 写若干 iptables 扩展规则？

基于 mac 地址的规则案例：

[root@svr1 ~]# iptables -A INPUT -m mac --mac-source 00:0C:29:74:BE:21 -j DROP
丢弃 00:0C:29:74:BE:21 这台主机入站的所有数据包

基于多端口的规则案例：

[root@svr1 ~]# iptables -A INPUT -p tcp -m multiport \
--dports 20:22,25,80,110,143,16501:16800 -j ACCEPT
允许任何主机使用 TCP 协议访问本机的 20 到 22 端口，25、80、110、143 端口，16501 到 16800 端口

基于 IP 范围的规则案例：

[root@svr1 ~]# iptables -A INPUT -p tcp --dport 22 -m iprange \
--src-range 192.168.4.10-192.168.4.20 -j ACCEPT
允许 192.168.4.10 到 192.168.4.20 范围的主机，使用 TCP 协议访问本机的 22 端口

07. 让内部 192.168.4.0/24 网络的所有主机都可以通过网关 172.40.3.100 访问外部网络？

iptables -t nat -I POSTROUTING -s 192.168.4.0/24 -j SNAT --to-source 172.40.3.100

Day05

01. 监控资源一般分为共有数据资源和私有数据资源，列出几个常见的共有和私有资源？

a) 公开数据

Web、FTP、SSH、数据库等应用服务

TCP 或 UDP 端口

b) 私有数据

CPU、内存、磁盘、网卡流量等使用信息

用户、进程等运行信息

02. 常用的查看系统信息的命令有哪些？

a) ifconfig
b) netstat 或 ss
c) ping
d) traceroute
e) iostat
f) ps
g) uptime
h) free
i) swapon -s
j) df -h

03. 常见的监控软件有哪些？

a) Cacti

基于 SNMP 协议的监控软件，强大的绘图能力

b) Nagios

基于 Agent 监控，强大的状态检查与报警机制

插件极多，自己写监控脚本潜入到 Nagios 非常方便

c) Zabbix

基于多种监控机制，支持分布式监控

04. 部署 zabbix 监控时，监控服务器和被监控端需要安装哪些软件？

监控服务器：zabbix_server、zabbix_agentd、zabbix_proxy、zabbix_get、zabbix_sender

被监控服务器：zabbix_agentd、zabbix_get、zabbix_sender

05. 部署 zabbix 的流程？

搭建 LNMP 环境（动静分离、优化 buffer 缓存）；

源码安装 zabbix 软件（监控服务器和被监控端主机安装的软件略有差异）；

导入初始化数据库数据（初始化数据就在 zabbix 源码包中）；

将 zabbix 页面上线到 LNMP 平台（zabbix 页面代码在源码包中）；

访问 Web 初始化 zabbix（根据提示优化 php 配置，安装缺少的依赖包）；

登陆 zabbix web 控制台！

06. 监控一台 Linux 服务器的流程？

通过 Web 控制台添加主机，绑定监控模板到主机，或者自定义监控项与主机绑定，查看监控数据或监控图。

Day06

01. 配置 zabbix 报警的流程？

创建触发器，设置触发器表达式，设置发邮件的邮件服务器，设置邮件的收件人，创建 Action（设置动作的条件与动作的操作行为），测试监控报警结果。

02. 简单说明自动发现的功能？

a) 自动发现（Discovery）

当 Zabbix 需要监控的设备越来越多，手动添加监控设备越来越有挑战，此时，可以考虑使用自动发现功能；需要批量一次性添加一组监控主机，也可以使用自动发现功能。

b) 自动发现可以实现：

自动发现、添加主机，自动添加主机到组

自动连接模板到主机，自动创建监控项目与图形等

03. 简单描述 zabbix 的主动与被动监控？

主动和被动都是对被监控端主机而言的，默认 zabbix 采用的是被动监控。

被动监控：Server 向 Agent 发起连接，发送监控 key，Agent 接受请求，响应监控数据

主动监控：Agent 向 Server 发起连接，Agent 请求需要检测的监控项目列表，
Server 响应 Agent 发送一个 items 列表，Agent 确认收到监控列表，
TCP 连接完成，会话关闭，Agent 开始周期性地收集数据

区别：Server 不用每次需要数据都连接 Agent，Agent 会自己收集数据并处理数据，
Server 仅需要保存数据即可

04. Zabbix 监控 nginx 状态的简单流程？

```
[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf
UnsafeUserParameters=1                                  //是否允许自定义 key
Include=/usr/local/etc/zabbix_agentd.conf.d/        //加载自定义监控配置目录


[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/nginx.status     //创建自定义监控
UserParameter=nginx.status[*],/usr/local/bin/nginx_status.sh $1
```

```
[root@web1 ~]# killall zabbix_agentd
[root@web1 ~]# zabbix_agentd
```

写自定义监控脚本（仅供参考，非完整脚本）

```
[root@web1 ~]# vim /usr/local/bin/nginx_status.sh
#!/bin/bash
case $1 in
active)
    curl -s http://127.0.0.1/status |awk '/Active/{print $NF}';;
waiting)
    curl -s http://127.0.0.1/status |awk '/Waiting/{print $NF}';
accepts)
    curl -s http://127.0.0.1/status |awk 'NR==3{print $2 }';;
esac
[root@web1 ~]# chmod +x    /usr/local/bin/nginx_status.sh
```

测试效果：

```
[root@web1 ~]# zabbix_get    -s 127.0.0.1 -k 'nginx.status[accepts]'
```

05. 常见的 TCP 连接的状态有哪些？

熟悉 TCP 三次握手，参考图-1。

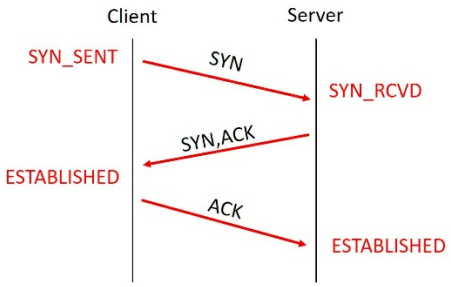

图-1

熟悉 TCP 连接的四次断开，参考图-2。

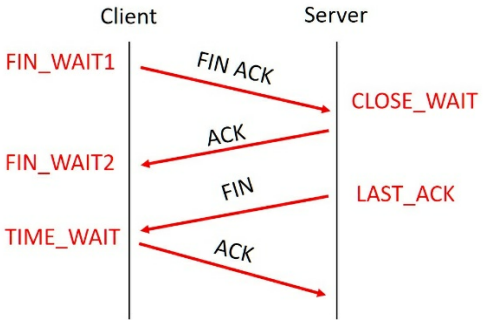

图-2