

### Linux三剑客实战

### 环境

- windows git bash
- mac iterm term
- 用自己的帐号登录 ssh 手机号码后8位@shell.testing-studio.com
- 没有账号的可以临时用 ssh hogwarts2019@shell.testing-studio.com
- /tmp/nginx.log 保存了一份一天的nginx访问log

### Linux三剑客介绍





# Linux三剑客介绍

- \* grep
  - \* global search regular expression(RE) and print out the line
  - \* 基于正则表达式查找满足条件的行
- \* awk
  - \* 名字来源于三个作者的名字简称
  - \* 根据定位到的数据行处理其中的分段
- \* sed
  - stream editor
  - \* 根据定位到的数据行修改数据







# Linux三剑客的价值

- \* 三剑客
  - \* grep 数据查找定位
  - \* awk 数据切片
  - \* sed 数据修改
- \* 类比SQL
  - \* grep=select \* from table
  - \* awk=select field from table
  - \* sed=update table set field=new where field=old

testerhome.con

### Grep





## grep

- grep pattern file
- \* grep -i pattern file 忽略大小写
- \* grep -v pattern file 不显示匹配的行
- \* grep -o pattern file 把每个匹配的内容用独立的行显示
- \* grep -E pattern file 使用扩展正则表达式
- \* grep -A -B -C pattern file 打印命中数据的上下文
- \* grep pattern -r dir/ 递归搜索



▲ 霍格沃兹测试学院



# pattern正则表达式

- \*基本表达式 (BRE)
  - \* ^ 开头 \$结尾
  - \* [a-z] [0-9] 区间
  - \* \*0个或多个
- \*基本正则(BRE)与扩展正则的区别(ERE)
  - \* ? 非贪婪匹配
  - \* + 一个或者多个
  - \*()分组
  - \* {} 范围约束
  - \* | 匹配多个表达式的任何一个

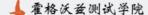


ps -ef | grep bashecho "ABC" | grep -i abcps -ef | grep bash | grep -v grepe

#### 测试题

- 找出nginx.log中所有404和503报错的log数据,取出前3条数据,把命令贴到回复里
- 找出testerhome首页的所有http和https的链接

### awk





# awk pattern语法

- \* awk理论上可以代替grep
- \* awk 'pattern{action}'
- \* awk 'BEGIN{}END{}' 开始和结束
- \* awk'/Running/' 正则匹配
- \* awk '/aa/,/bb/' 区间选择
- \* awk '\$2~/xxx/' 字段匹配
- \* awk 'NR==2' 取第二行
- \* awk 'NR>1' 去掉第一行

```
localhost:~ seveniruby$ echo "
1
2
3
" | awk '$0>2'
3
localhost:~ seveniruby$
```

▲ 霍格沃兹测试学院



# awk的字段数据处理

- \* -F 参数指定字段分隔符
- \* BEGIN{FS="\_"} 也可以表示分隔符
- \* \$0代表原来的行
- \* \$1代表第一个字段
- \* \$N 代表第N个字段
- \* \$NF 代表最后一个字段

testerhome.com





# awk行处理

- \* 把单行分拆为多行
  - \* echo \$PATH | awk 'BEGIN{RS=":"}{print \$0}'
  - \* echo \$PATH | awk 'BEGIN{RS=":"}{print NR,\$0}'
  - \* echo \$PATH | awk 'BEGIN{RS=":"}END{print NR}'
- \* 多行组合为单行
  - \* echo \$PATH | awk 'BEGIN{RS=":"}{print \$0}' | awk 'BEGIN{FS="\n";ORS=":"}{print \$0}'

testerhome.com

ps | awk 'BEGIN{print "start"}{print \$0}END{print "end"}'awk '/ 404 | 500 /' /

#### 测试题

- 找出404和500的数据,只打印状态码这一列,然后排序去重。把命令贴到回复里
- 去testerhome首页找到所有的http的连接,然后打印不带http的纯域名部分

#### sed





### sed

- \* sed [addr]X[options]
- \* -e 表达式
- \* sed -n '2p' 打印第二行
- \* sed 's#hello#world#' 修改
- \* -i 直接修改源文件
- \* -E 扩展表达式
- \* --debug 调试

### testerhome.com

#### pattern表达式

- 20 30.35 行数与行数范围
- /pattern/正则匹配
- //,// 正则匹配的区间

#### action

- d删除
- p 打印, 通畅结合-n参数
- s/REGEXP/REPLACEMENT/[FLAGS]
- 替换时引用 \1\2 匹配的字段

ps | sed -n 1,3pps | sed 's/CMD/command/'ps | sed -n '/ps/p'echo '12345' | sed

#### 测试题

• 对所有404 500 的数据,统计出现这种状态码的url,需要对url汇总(汇总相似的url,把相同的资源但是变化的id去掉)去重,打印前5个出问题的路径,把命令贴到回复里

### 演练



长按识别二维码, 关注 Testerhome 公众号

☑☑☑阅读原文可查看相关链接,并与作者交流

linux, shell