

# 01.Shell脚本概述

- 01.Shell脚本概述
  - 1.Shell脚本的功能
  - 2.Shell脚本是什么
  - 3.Shell脚本的特性

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

- 1.shell基本特性
- 2.shell变量定义
- 3.shell条件测试
- 4.shell数值运算
- 5.shell流程控制 if case
- 6.shell循环语句 for while
- 7.shell数组函数 array function
- 8.shell内置命令 break continue exit
- 9.shell轻量项目

## 1.Shell脚本的功能

Shell 脚本做什么: 基于标准化之上的 -> 工具化

Shell 脚本的作用: 简化操作步骤, 提高效率, 减少人为干预, 减少系统故障

- 1.自动化的完成基础配置: (系统初始化操作、系统更新、内核调整、网络、时区、SSH优化)
- 2.自动化安装程序: 自动化安装 LNMP、LAMP、MySQL、Nginx、Redis
- 3.自动化调整配置文件 Nginx Conf、MySQL Conf
- 4.自动化部署业务: 部署 php、java 秒级回退
- 5.自动化定期备份恢复程序: (MySQL 全备+增量+ binlog + crond + Shell 脚本)
- 6.自动化信息的采集 Zabbix + Shell: 硬件、系统、服务、网络、等等
- 7.自动化日志分析: 取值->排序->去重->统计->分析
- 8.自动化扩容/缩容 zabbix + shell
  - 扩容: 监控服务器cpu, 如cpu负载持续80% + 触发动作(脚本)

- 脚本: 调用api开通云主机->初始化环境->加入集群->对外提供
- 缩容: 监控服务器cpu使用率20%->判断有多少web节点->判断是否超过预设->缩减到对应的预设状态->变更负载的配置
- 9. Shell 什么都能做, 但要符合实际情况以及实际业务需求

## 2.Shell脚本是什么

- 1.系统命令的堆积
- 2.特定的格式 + 特定的语法 + 系统的命令 = 文件
- 3.不会手动搭建或者执行某些程序, 那么一定学不会Shell

### 1.脚本的基本格式

```
shell> vim /tmp/test
#!/bin/bash
echo "Hello world"
```

运行脚本

```
# chmod 755 /tmp/test
# /tmp/test
```

### 2.脚本的执行方式

#### 1.直接使用bash解析器执行脚本

```
# bash /tmp/test
```

#### 2.给脚本添加执行权限, 然后使用./运行

```
# chmod 755 /tmp/test
# ./tmp/test
```

没有执行权限会提示权限不足

## 3.Shell脚本的特性

bash shell 特性

- 1.命令补全和文件路径补全, 如果写错无法补全     **table**
- 2.命令历史记忆功能history
- 3.别名功能alias、unalias
- 4.常用快捷键ctrl+u,k,a,e,l,c,z,d,w,r,y
- 5.前后台作业控制bg,fg,jobs,screen

6.输入输出重定向 >, >>, 1>, 2>>, &>, cat <

7.管道 | 将前者命令的标准输出交给后者命令的输入 tee

8.命令排序

； 没有逻辑关系,无论分号前面的命令执行是否成功都执行后者命令

&& 前面执行成功,则执行后者

|| 前面执行不成功,则执行后者

9.shell 通配符

\* 匹配任意多个字符

? 匹配任意一个字符

[] 匹配括号中任意一个字符a-z,0-9,A-Z,a-Z

() 在子 shell 中执行(cd /boot;ls) (umask 077; touch file1000)

{ } 集合 touch file{1..9}

\ 转义符

10.echo输出颜色、printf 格式化输出文本

```
[root@Shell ~]# echo -e "\033[30m 黑色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[31m 红色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[32m 绿色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[33m 黄色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[34m 蓝色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[35m 紫色字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[36m 天蓝字 \033[0m"
```

```
[root@Shell ~]# echo -e "\033[37m 白色字 \033[0m"
```