

---

# **devopsBook**

***Release v1***

**maxiaoMax**

**May 12, 2021**



CONTENTS:

<b>1</b>	<b>Linux 基础知识</b>	<b>1</b>
1.1	基础操作 . . . . .	1
1.2	文本文件操作 . . . . .	4
1.3	Vi 基础操作 . . . . .	6
1.4	用户管理 . . . . .	7
1.5	权限管理 . . . . .	8
1.6	磁盘操作 . . . . .	10
1.7	内存操作 . . . . .	11
1.8	网络设置 . . . . .	14
1.9	管道与重定向 . . . . .	14
<b>2</b>	<b>Linux 高级知识</b>	<b>17</b>
2.1	常用正则表达式 . . . . .	17
2.2	常用 find 命令 . . . . .	17
2.3	常用 sed 命令 . . . . .	17
2.4	常用 awk 命令 . . . . .	17
<b>3</b>	<b>aws 学习笔记</b>	<b>19</b>
3.1	计算机 . . . . .	19
3.2	存储 . . . . .	19
3.3	数据库 . . . . .	19
3.4	网络 . . . . .	19
3.5	中间件 . . . . .	19
3.6	权限控制 . . . . .	19
3.7	管理工具 . . . . .	19
3.8	容器 . . . . .	19
<b>4</b>	<b>devops 实践</b>	<b>21</b>
4.1	运行状态机 . . . . .	21
4.2	运维开发标准化建设 . . . . .	21
4.3	jenkins 新环境部署 . . . . .	22

4.4 环境升级 . . . . .	22
5 关于 . . . . .	23

## LINUX 基础知识

### 1.1 基础操作

#### 1.1.1 查看帮助：whatis/man/info

常见参数：

```
-v : 一般是指 verbose  
-h : 一般是 human
```

#### 1.1.2 切换目录：cd

```
cd - : 切换到上一个目录  
cd .. : 切换到上级目录  
cd ~ : 切换到当前用户的 home 目录  
cd ~username : 切换到其它用户的 home 目录
```

#### 1.1.3 查看当前目录：pwd

```
pwd : 查看当前目录  
pwd -P :
```

### 1.1.4 查看目录下文件：ls

```
ls -a : 查看所有文件，包括隐藏文件
ls -l : 显示文件详细信息
ls -ld : 只看目录本身
ls -ltr : 按修改时间顺序查看文件
```

### 1.1.5 创建文件夹：mkdir

```
mkdir dirname : 创建目录，如果父目录不存在会报错
mkdir -p dirname : 创建多层目录，如果父目录不存在会自动创建
mkdir -m=755 dirname : 创建目录，并设置目录权限
```

### 1.1.6 复制文件：cp

```
cp -p source target : 保留原始文件属性
cp -r source target : 复制文件夹
cp -i source target : 目录文件存在则询问
```

### 1.1.7 移动文件：mv

```
mv -u source target : 只有 target 不存在或者旧时才 move
mv -b source target : 备份文件
mv -n source target : 不要覆盖文件
```

### 1.1.8 删除文件：rm

```
rm -rf : 删除文件或文件夹，忽略警告
rm -ir : 删除前先询问
```

### 1.1.9 重命名文件：rename

一般重命名文件使用 mv 命令就足够了，rename 命令主要用来批量重命名文件。

```
rename source target filename : 将文件名中的 source 字符串替换为 target 字符串，支持正则表达式
rename reguler filename : 对文件名执行正则表达式
```

### 1.1.10 转换文件：dd

dd if=source of=target : source 和 target 可以是设备也可以是文件

### 1.1.11 查找文件：which/whereis/locate/find

which : 在 PATH 环境变量制定的位置中, 寻找某个命令的位置, 并返回第一个搜索结果

whereis : 查找代码, 帮助文件和二进制文件。-b 只查找二进制文件, -m 只查找说明文件, -s 只查找源代码

locate : 去/var/lib/mlocate/mlocage.db 中查找文件, 一般一天运行一次 updatedb 更新一次数据库, 可以在 crontab 中设置。

find dir -name "" : 遍历文件, 参数比较复杂

### 1.1.12 输出字符串：echo

echo -e : 开启转义, 默认不开启

### 1.1.13 环境变量：env

env : 显示当前用户全部环境变量

### 1.1.14 一次性计划任务：at

at : 在指定时间执行任务

atq : 查看任务列表

atrm : 删除任务

batch : 与 at 类似, 只有系统不繁忙时才执行。

### 1.1.15 长期计划任务：crontab

crontab 的格式为: 分钟小时日期月份星期几命令, 星期天等于 0 或者 7

\* : 代表所有允许的取值

/n : 代表一定时间间隔

n-m : 代表一定取值范围

x,y,z : 代表离散的取值

% : 代表换行, 可以用\% 转义

## 1.2 文本文件操作

### 1.2.1 查看短文本文件：cat

```
cat -n file : 查看文本文件，显示行号  
cat -n file : 查看文本文件，显示行号，空行不加行号
```

### 1.2.2 查看长文本文件：more

```
more file : 查看文件
```

### 1.2.3 查看文件前 n 行：head

```
head -n 10 file : 显示头 10 行  
head -n -10 file : 不显示后 10 行
```

### 1.2.4 查看文件后 n 行：tail

```
tail -n 10 file : 查看文件后 10 行  
tail -f file : 持续刷新显示文件末尾
```

### 1.2.5 通过分割列来提取字符：cut

类似 python 的 split()

```
cut -d: -f1 -s file : -d 后为分隔符,-f 后为提取第几列,-s 不显示不含分隔符的行
```

### 1.2.6 以不同格式输出：od

```
od -b file : 以 8 进制输出  
od -c file : 以 ascii 码输出
```



### 1.2.7 转换或删除字符：tr

```
[ :alnum:] : 所有的字母和数字
[:alpha:] : 所有的字母
[:blank:] : 所有的水平空格
[:space:] : 所有的水平和垂直空格
[:cntrl:] : 所有的控制字符
[:digit:] : 所有的数字
[:graph:] : 所有可打印的字符，不包括空格
[:lower:] : 所有的小写字母
[:upper:] : 所有的大写字母
[:punct:] : 所有的标点
[=CHAR=] : 所有符合指定的字符
```

### 1.2.8 比较文件差异：diff

```
diff -b -B source target : -b 忽略空格差异, -B 忽略空行差异
diff -q source target : 仅显示是否有差异
```

### 1.2.9 查看文件的时间属性：stat

```
stat file : 查看文件的 inode 内容
atime :
mtime :
ctime :
```

### 1.2.10 修改文件的时间属性：touch

```
touch file : 将文件的 atime,mtime,ctime 修改为系统当前时间，文件不存在则创建
touch -m/-c/-a file : 单独修改一个时间
```

### 1.2.11 识别文件格式：file

```
file filename : 识别文件格式
file -f filename : 从文件中读取文件列表
```

## 1.3 Vi 基础操作

### 1.3.1 删除：dd

dd : 删除光标所在行  
5dd : 删除光标处开始的 5 行

### 1.3.2 复制：yy

5yy : 复制光标处开始的 5 行

### 1.3.3 粘贴：p

p : 将之前删除或复制的数据粘贴到光标后

### 1.3.4 搜索：/和 ?

/ : 从上到下搜索  
? : 从下到上搜索  
n : 跳到搜索到的下一行  
N : 跳到搜索到的上一行

### 1.3.5 撤销：u

撤销之前的操作

### 1.3.6 保存及退出：:q,:w

:w : 保存  
:q : 退出  
:q! : 强制退出，放弃修改  
:wq! : 强制保存退出

### 1.3.7 显示行号: `:set nu`

```
:set nu : 显示行号  
:set nonu : 不显示行号
```

### 1.3.8 跳转: `:n`

```
:n : 跳转到第 n 行  
:1 : 跳转到第一行  
:$ : 跳转到最后一行
```

### 1.3.9 清空文件: `:%!d`

```
:%d : 清空小文件, 大文件会比较慢  
cat /dev/null > filename : 推荐方法
```

### 1.3.10 显示格式: `:set ff`

```
:set ff : 显示文件格式  
:set ff=unix : 将文件格式转换为 unix
```

## 1.4 用户管理

### 1.4.1 创建用户: `useradd`

### 1.4.2 修改密码: `passwd`

### 1.4.3 删除用户: `userdel`

### 1.4.4 修改用户: `usermod`

### 1.4.5 创建组: `groupadd`

### 1.4.6 切换用户身份: `su`

```
su username : 切换用户身份, 但环境变量和工作目录还是原来用户的  
su - username : 切换用户身份和环境变量
```

### 1.4.7 临时使用 root 权限：sudo

可以使用 visudo 命令修改/etc/sudoers

### 1.4.8 查看用户登录记录：last

用户登录信息保存在/var/log/wtmp

```
last user : 显示用户的登录记录
```

### 1.4.9 查看用户登录失败记录：lastb

用户登录失败信息保存在/var/log/btmp

### 1.4.10 查看用户历史命令：history

命令历史储存在 ~/.bash\_history

```
export HISTTIMEFORMAT='%F %T ' : 设置环境变量后的命令才会有时间戳  
ctrl + R : 可以搜索自己使用过的命令  
!n : 执行第 n 条命令
```

### 1.4.11 查看登录用户信息：w/who/whoami

```
w : 查看登录用户信息  
whoami :  
who :
```

### 1.4.12 查看用户组信息：id

## 1.5 权限管理

### 1.5.1 用户身份：UID

```
0 : root 用户  
1-999 : 系统服务用户，默认不可登录  
1000+ : 普通用户
```

## 1.5.2 组身份：GID

组信息保存在/etc/group 中

默认组：与 UID 相同，用户创建时创建  
扩展组：后加入的组

## 1.5.3 文件权限：

读：r 或者 4  
写：w 或者 2  
执行：x 或者 1，目录的 x 权限代表用户可以进入

ls -l 的结果总共 10 位

第一位：文件类型，-普通文件，d 目录，l 链接，b 块设备，c 字符设备，p 管道  
2-4 位：文件所有者权限，顺序 r,w,x，没有该项权限用-代替  
5-7 位：文件所属组权限  
8-10 位：其它用户权限

## 1.5.4 临时属主权限：SUID

## 1.5.5 临时属组权限：SGID

## 1.5.6 只可管理自己的数据不能删除他人文件：SBID

## 1.5.7 设置隐藏权限：chattr

i：仅能修改，不能删除和新建  
a：仅允许追加内容  
S：内部变化后马上同步到硬盘  
s：彻底删除，不可恢复  
A：不再更新文件的最后修改时间  
D：检查压缩文件中的错误  
d：默认将文件或目录进行压缩  
u：删除后保留硬盘中的数据，日后可恢复  
X：可以直接访问压缩文件的内容

### 1.5.8 显示隐藏属性：lsattr

## 1.6 磁盘操作

### 1.6.1 常见目录定义：

```
/boot :  开机所需文件，内核，开机菜单等
/dev :   任何设备与接口都以文件形式保存在此目录
/etc :   配置文件
/home :  用户目录
/opt :   放置第三方软件
/tmp :   任何人都可以使用的共享临时目录
/proc :  虚拟文件系统，包括系统内核，进程
/var :   日志
/lost+found :  文件系统发生错误后保存丢失的文件片段
```

### 1.6.2 常见文件系统：

```
Ext4 :  RHEL6 默认
XFS :   RHEL7 默认
NFS :   网络文件系统，主要用于远程文件共享
PROC :  虚拟的进程文件系统
EFS :   aws 提高的弹性文件系统
```

### 1.6.3 挂载硬件设备：mount

挂载点必须是一个已经存在的目录。

需要 mount 重启后还生效的，需要在/etc/fstab 添加 mount，否则重启后会失效。

/etc/fstab 格式：设备文件挂载目录格式类型权限选项（默认为 default）自检优先级

```
mount -a :  挂载所有在/etc/fstab 中设置的 mount
```

## 1.6.4 取消挂载：umount

## 1.6.5 查看挂载点信息：df

```
df -a : 显示所有的文件系统
df --total : 显示总体使用量
df -h : 易读格式
df -i : 展示 Inode 的信息
df -T : 显示文件系统类型
```

## 1.6.6 查看磁盘使用量：du

```
du -sh dirname : -s 仅显示使用量总和, -h 易读格式
```

## 1.6.7 文件链接：ln

```
ln : 硬链接, 相当于创建副本, 修改一个与其链接的文件也会被修改, 但删除文件后与其链接的文件不会被删除
ln -s : 软链接 (符号链接), 原文件删除就失效, 相当于快捷方式
```

## 1.6.8 逻辑卷：

## 1.6.9 用户磁盘空间限制：quota

## 1.6.10 虚拟文件系统：VFS

# 1.7 内存操作

## 1.7.1 查看内存占用：free

free 命令读取的是/proc/meminfo, 可以在/etc/sysctl.conf 中调整 swappiness 来调整 swap 使用策略。

```
free -h -s 3 : 每 3 秒查询一次内存使用情况, 以易读方式显示
free : 正在未被使用的内存
available : free + cache + buffer
```

## 1.7.2 查看进程: ps

```
ps -ef  
ps aux
```





1.7.3 查看进程占用资源: top

1.7.4 改变进程优先级: nice

1.7.5 结束进程: kill

1.7.6 ulimit

1.7.7 显示进程号: pgrep

1.7.8 后台执行程序: nohup

1.7.9 使程序后台执行: bg

1.7.10 使程序前台执行: fg

1.7.11 查看后台进程: jobs

1.7.12 查看进程间通信状态: ipcs

## 1.8 网络设置

1.8.1 查看网卡状态: ifconfig

1.8.2 查看端口状态: netstat

1.8.3 查看路由设置: route

1.8.4 域名解析: nslookup

1.8.5 测试网络状况: ping

1.8.6 测试远程端口是否开放: telnet

1.8.7 路由跟踪: traceroute

1.8.8 下载文件: wget

1.8.9 curl

1.8.10 监听网卡数据包: tcpdump

---

## 1.9 管道与重定向

1.9.1 管道命令符: |

**1.9.3 通配符：**

**1.9.4 特殊字符转义：**



## LINUX 高级知识

### 2.1 常用正则表达式

### 2.2 常用 find 命令

### 2.3 常用 sed 命令

### 2.4 常用 awk 命令



## AWS 学习笔记

### 3.1 计算机

#### 3.1.1 服务器 ec2

### 3.2 存储

#### 3.2.1 s3:

### 3.3 数据库

### 3.4 网络

### 3.5 中间件

### 3.6 权限控制

### 3.7 管理工具

### 3.8 容器





## 4.1 运行状态机

agent 是一个 workflow 解释器。类似 aws 的 step function。workflow 由扩展过的 yaml 语言描述。由多个 task 组成。

```
- "name": "test task"
  "type": "task"
  "resource": ""
  "fail_action": ""
  "input": ""
  "output": ""
  "when": ""
  "timeout": "30"
```

resource 是去 gitlab 拉取最新代码执行。

每个 task 都会自动保存 log，不用再去写保存日志的代码。

agent 会自动获取环境信息作为上下文 content，可以在 workflow 中直接调用。

外部参数可以通过固定 json 文件的方式输入，在 workflow 中调用。

agent 在运行前会自动更新。

agent 是被动 agent，平时不会运行，等待被唤醒。运行结束后就停止。

## 4.2 运维开发标准化建设

URI 结构：gitlab 仓库地址:gitlab brunch:plugin name:function name

建设好 gitlab 仓库，master 分支要严格保护，经过测试的代码才可以合并进来

plugin 基于 base plugin 开发

单元测试

测试环境

跨环境: aws azure

多版本支持: 11.1 11.2

跨 python 版本: python2 python3

跨 os: windows,linux

## 4.3 jenkins 新环境部署

## 4.4 环境升级

这是我的运维学习笔记总结