

linux

启动关闭和登陆

重启

shutdown-r now

- 立刻重启

shutdown-r 5

- 五分钟之后重启

reboot

- 立刻重启（需要管理员身份）

init 6

- 重新启动

关闭linux

shutdown-h now

- 立刻关机

shutdown-h 5

- 五分钟之后关机

poweroff

- 立刻关机（需要管理员身份）

halt

init 0

本地登陆远程登陆

本地登陆

普通用户登陆提示符

- \$

root登陆提示符

- #

远程登陆

- 使用ssh协议远程登陆linux系统，要远程登陆要保证sshd服务开启

▼ 格式

- #ssh远程主机上的用户名@远程主机的IP地址或FQDN
- FQDN：权限的域名，同时带有主机名和域名的名称

▼ 基础操作命令

▼ 目录

▼ 目录切换

- cd
- ▼ 切换到根目录
 - cd/
- ▼ 切换到根目录下的usr
 - cd/usr
- ▼ 切换到上一级目录
 - cd..
- ▼ 切换到home目录
 - cd~
- ▼ 切换到上次访问目录
 - cd-

▼ 常用目录

- ▼ bin
 - 存放在最经常使用的命令
- ▼ etc
 - 存放的是系统管理所需的配置文件和子目录
- ▼ usr
 - 用户的很多应用程序文件在这个目录下
- ▼ mnt
 - 临时挂载的文件系统
- ▼ tmp
 - 存放一些临时文件
- ▼ boot
 - 存放启动linux时使用的核心文件

- ▼ dev
 - 存放的是linux的外部设备
- ▼ home
 - 该目录的目录名是以用户的账号命名的
- ▼ lib
 - 存放着系统的最基本的动态连接共享库
- ▼ root
 - 系统管理员，也称超级权限者的用户主目录
- ▼ sys
 - 系统sysfs
- ▼ 路径操作
 - ▼ pwd
 - 打印当前所在目录
 - ▼ cd.
 - 指的是当前目录
 - ▼ cd..
 - 指的是当前目录的上一级目录
- ▼ 目录查看
 - ▼ ls
 - 查看当前目录下所有文件和目录
 - ▼ ls -a
 - 查看当前目录下所有目录和文件包括隐藏文件
 - ▼ ls -l或ls -ll
 - 列表查看当前目录下所有目录和文件
 - ▼ ls / 目录名
 - 查看指定目录下所有目录和文件
 - ▼ ls 结果重新定向
 - ▼ >
 - 输出重定向到一个文件或设备 覆盖原来的文件
 - ▼ >!
 - 输出重定向到一个文件或设备 不覆盖原来的文件

- 输出重定向到一个文件或设备 强制覆盖原来的文件
- ▼ >>
 - 输出重定向到一个文件或设备 追加原来的文件
- ▼ <
 - 输入重定向到一个程序
- ▼ 目录操作
 - ▼ 创建目录
 - mkdir
 - ▼ mkdir -p dirname
 - -p确保目录存在，不存在就建一个
 - ▼ mkdir myfiles
 - 在当前目录下创建一个myfiles目录
 - ▼ mkdir /usr/myfiles
 - 在指定目录下创建一个名为myfiles的目录
 - ▼ 删除目录
 - rm -r myfiles 递归删除当前目录下的myfiles目录
 - rm -rf myfiles 递归删除当前目录下的myfiles目录（不询问）
 - ▼ rm -rf*
 - 将当前目录下所有目录和文件删除
 - ▼ rm -rf/*
 - 将根目录下所有目录和文件删除
 - ▼ 修改目录
 - ▼ 重命名目录
 - mv 当前目录 新目录
 - ▼ 剪切目录
 - mv 目录名称 目录新的位置
 - ▼ 拷贝目录
 - ▼ cp -r 目录名称 目录拷贝的目标位置
 - ▼ cp 选项 源文件 目标文件
 - ▼ 选项

- ▼ -a
 - 通常在复制目录时使用，保留连接，文件属性，并复制目录下所有内容
- ▼ -d
 - 复制时保留连接
- ▼ -f
 - 覆盖已经存在的目标文件而不给出提示
- ▼ -i
 - 覆盖已经存在的目标文件前给出提示询问
- ▼ -p
 - 除复制文件之外还将修改时间和访问权限也复制到新文件中
- ▼ -r
 - 若给出的源文件是个目录文件，复制该目录下的所有子目录和文件
- ▼ -l
 - 不复制文件，只生产连接文件

▼ 搜索目录

▼ find 目录 参数 文件名称

▼ 参数

▼ -mount , -xdev

- 只检查和指定目录在一个文件系统下的文件，避免列出文件系统下的文件

▼ -amin n

- 过去n分钟内被读取过

▼ linux的基础命令

▼ 帮助命令

- --help命令
- man命令
- man在前help在后

▼ 查找命令

- ▼ grep
 - ▼ 强大的文本搜索命令
 - ps -ef | grep sshd (查找指定的ssh服务进程)
- ▼ find
 - 在目录机构中搜索文件，并对搜索到的文件执行指定操作
- ▼ locate
 - ▼ 让使用者快速的找到某个路径
 - locate /etc/sh (搜索etc目录下sh开头的所有文件)
- ▼ whereis
 - ▼ 定位可执行文件，源代码文件，帮助文件在文件系统中的位置
 - whereis ls (将和ls有关的文件都查找出来)
- ▼ which
 - ▼ 在PATH变量指定的路径中，搜索某个系统指令的位置，并且返回第一个搜索结果
 - which ls (查找ls命令所在的路径)
- ▼ su、sudo
 - ▼ su
 - 用于用户之间的切换但是切换前的用户依旧保持登陆状态。root用户向普通或虚拟用户切换不需要密码，普通用户切换到任意用户都要密码
 - ▼ su不足
 - 某个用户需要使用root权限，必须吧root密码告诉此用户
 - ▼ su test
 - 切换到test用户，但是路径还是/root
 - ▼ su -test
 - 切换到test用户，路径变成/home/test
 - ▼ su
 - 切换到root用户但是路径还是原来的路径
 - ▼ su-
 - 切换到root用户，并且路径是/root
- ▼ sudo

- 让普通用户临时具有root权限
- ▼ 进程相关命令
 - ▼ ps-ef
 - 查看正在运行的所有进程
 - ▼ kill pid或kill-9 pid (强制杀死进程)
 - pid (进程号)
- ▼ 常用文件处理命令
 - ▼ 新增文件
 - touch
 - ▼ 参数说明
 - ▼ a
 - 改变档案的读取时间记录
 - ▼ m
 - 改变档案的修改时间记录
 - ▼ c
 - 假如档案不存在，不会建立新的档案
 - ▼ f
 - 不使用是为了和其他unix系统的相容性而保留
 - ▼ r
 - 使用参考档的时间记录
 - ▼ d
 - 设定时间和日期，可以使用各种不同格式
 - ▼ t
 - 设定档案的时间记录
 - ▼ 删除文件
 - ▼ rm文件
 - 删除当前目录下的文件
 - ▼ rm -f文件
 - 删除当前目录的文件 (不询问)
 - ▼ rm-rf

- 文件1文件2批量删除
- rm -rf /*删除当前目录下所有隐藏文件
- ▼ r
 - r递归删除
- ▼ f
 - 强制删除
- ▼ 修改文件
 - ▼ vi或vim
 - ▼ 命令模式
 - ▼ 控制光标移动
 - 控制屏幕光标的移动，字符，字或行的删除，查找，移动复制删除，进入insert mode（插入模式）或者到last line mode（底行模式）
 - 上，下，左，右，j（下一行）
 - ▼ 删除当前行
 - dd
 - ▼ 查找
 - /字符
 - ▼ 进入编辑模式
 - i, o, a, insert
 - ▼ 进入底层模式
 - : (shift+;)
 - ▼ 插入模式
 - 只有在插入模式下才能进行文字输入，按【ESC】回到命令模式
 - ▼ 底行模式
 - 将文件保存或退出vi编辑器，亦可以设置编辑环境，如寻找字符串，列出行号
 - ▼ 底行模式常用命令
 - ▼ : q
 - 退出编辑

- ▼ : q!
 - 强制退出
- ▼ : wq
 - 保存并退出
- ▼ 打开文件
 - vi文件名
- ▼ 编辑文件
 - ▼ 使用vi编辑器打开文件后点击：i, a或者o或者insert即可进入编辑模式
 - i在光标所在字符前开始插入
 - a在光标所在字符后开始插入
 - o在光标所在行的下面另起一行插入
 - insert在光标所在字符前开始插入
 - ▼ 保存文件
 - ▼ 第一步
 - ESC进入命令模式
 - ▼ 第二步
 - : (shift+;) 进入底行模式模式
 - ▼ 第三步
 - wq保存并退出
- ▼ 取消编辑
 - ▼ 第一步
 - ESC进入命令模式
 - ▼ 第二步
 - : (shift+;) 进入底行模式
 - ▼ 第三步
 - q! 撤销本次修改并退出
- ▼ 文件查看
 - ▼ cat
 - 看最后一屏
 - ▼ more

- ▼ 百分百显示
 - 可以显示百分比，回车可以下一行，空格可以下一页，q可以结束查看
- ▼ less
 - ▼ 翻页查看
 - 使用键盘上的pgup, pgdn向上和向下翻页，q退出
- ▼ tail
 - ▼ 指定行数或动态查看
 - tail -10 查看文件的最后10行，ctl+c结束
- ▼ 权限修改
 - ▼ rwx
 - ▼ r
 - 可读
 - ▼ w
 - 可修改
 - ▼ x
 - 可执行文件
 - ▼ X
 - 表示只有当该文件是个子目录或者该文件已经被设定过为可执行文件
 - ▼ 第一位
 - -代表文件
 - d代表文件夹
 - ▼ 第一段 (三位)
 - 代表拥有者的权限
 - ▼ 第二段 (三位)
 - 代表拥有者所在的组组员的权限
 - ▼ 第三段 (三位)
 - 代表其他用户的权限
- ▼ 8421法
 - ▼ r
 - 4

- ▼ W
 - 2
- ▼ r
 - 1

rwx: r代表可读, w代表可写, x代表该文件是一个可执行文件, 如果rwx任意位置变为-则代表不可读或不可写或不可执行文件。

示例: 给a.c文件权限改为可执行文件权限, a.c文件的权限是-rw-r--r--

第一位: -就代表是文件, d代表是文件夹

第一段 (3位) : 代表拥有者的权限

第二段 (3位) : 代表拥有者所在的组, 组员的权限

第三段 (最后3位) : 代表的是其他用户的权限

```
a.c
[root@192 myfiles]# ls -l
total 4
-rw-r--r--. 1 root root 137 Apr 24 19:52 a.c
[root@192 myfiles]# _
```

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="3">文件原权限</th></tr> <tr> <td></td><th>①段 421</th><th>②段 rw-</th><th>③段 r--</th></tr> <tr> <td>-</td><td>6</td><td>4</td><td>4</td></tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="3">文件目标权限</th></tr> <tr> <td></td><th>①段 421</th><th>②段 rwx</th><th>③段 r-x</th></tr> <tr> <td>-</td><td>7</td><td>5</td><td>5</td></tr> </table>	文件原权限				①段 421	②段 rw-	③段 r--	-	6	4	4	文件目标权限				①段 421	②段 rwx	③段 r-x	-	7	5	5	权限字母: chmod +x a.c	<pre>[root@192 myfiles]# chmod +x a.c [root@192 myfiles]# ls -l total 4 -rwxr--r--. 1 root root 137 Apr 24 19:52 a.c [root@192 myfiles]# _</pre>
文件原权限																								
	①段 421	②段 rw-	③段 r--																					
-	6	4	4																					
文件目标权限																								
	①段 421	②段 rwx	③段 r-x																					
-	7	5	5																					
	8421法: chmod 755 a.c	<pre>[root@192 myfiles]# ls -l total 4 -rw-r--r--. 1 root root 137 Apr 24 19:52 a.c [root@192 myfiles]# chmod 755 a.c [root@192 myfiles]# ls -l total 4 -rwxr--r--. 1 root root 137 Apr 24 19:52 a.c [root@192 myfiles]# _</pre>																						

▼ 权限

- u表示文件拥有者
- g表示文件拥有者所在组的组员
- o表示其他人
- a表示三者皆是
- +表示增加权限
- -表示取消权限
- =表示设置唯一权限

▼ 文件压缩操作

▼ 打包文件

- .tar

▼ 压缩文件

- .gz

▼ 压缩并打包文件

- .tar.gz

▼ 命令: tar-zcvf打包压缩后的文件名 要打包的文件名

- ▼ Z

- 调用gzip压缩
- ▼ c
 - 打包文件
- ▼ v
 - 显示打包过程
- ▼ f
 - 指定文件名
- ▼ 文件解压
 - ▼ 命令：tar-zxvf
 - ▼ x
 - 代表解压
 - ▼ tar-xvf a.tar
 - 将当前目录下a.tar解压到当前目录下
 - ▼ tar-xvf a.tar-c..../myfiles1
 - 将当前目录下a.tar解压到上一级目录下的myfiles1中
 - c代表指定解压位置
- ▼ 文件或目录的属性字段
 - ▼ 第一个字符代表文件的类型
 - ▼ -
 - 普通文件
 - ▼ d
 - 目录
 - ▼ l
 - 连接文件
 - ▼ b
 - 设备文件
 - ▼ c
 - 字符设备文件
 - ▼ p
 - 命令管道文件，与shell编程有关

- ▼ S
 - sock文件，与shell编程有关
- ▼ 前三个
 - 文件拥有者权限
- ▼ 中间三个
 - 文件拥有者所在组组员的权限
- ▼ 最后三个
 - 其他用户的权限

▼ 网络管理和软件包管理

- ▼ 网络管理
 - ▼ 网卡配置文件、DNS配置文件、主机名配置文件
 - ▼ 网卡配置文件
 - ▼ 文件路径
 - linux系统下网卡有专门的配置文件存储
 - ▼ ls /etc/sysconfig/network-scripts/
 - 一般在这个文件下第一个文件
 - ifcfg-ens33
 - cat /etc/sysconfig/network-scripts/ifcfg-en33

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="1a5eda7a-8e5c-4720-b059-1567cd025837"
DEVICE="ens33"
ONBOOT="yes"
[zhangzhihu@192 ~] $ █
```

- ▼ 初始配置状态

```

TYPE="Ethernet" #类型为以太网
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp" #static是静态IP地址, dhcp, 动态获取IP地址, none
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes" #IPV6是否启用
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33" #网卡名称
UUID="1a5eda7a-8e5c-4720-b059-1567cd025837" #唯一识别码
DEVICE="ens33" #设备名称
ONBOOT="yes" #yes:开启自动启用网络连接,no:不自动启用网络连接

```

注意：网卡配置文件内，选项要大写，小写不报错，但是不生效，参数可以小写

▼ 详细配置内容

```

TYPE="Ethernet" //网络类型: Ethernet 以太网
BOOTPROTO="static " //引导协议: dhcp 自动获取、static 静态、none 不指定
NM_CONTROLLED //是否受 network 程序管理
USERCTL //普通用户是否可以控制网卡
DEFROUTE="yes" //是否启用默认路由
IPV4_FAILURE_FATAL="no" //是否启用 IPV4 错误检测功能
IPV6INIT="no" //是否 IPV6 协议
IPV6_AUTOCONF="no " //是否自动配置 IPV6 地址
IPV6_DEFROUTE="no" //是否启用 IPV6 默认路由
IPV6_PEERDNS="no " //是否通过 DHCP 服务器获取 DNS 地址(IPV6)
IPV6_PEERROUTES="no" //是否通过 DHCP 服务器获取网关的路由信息(IPV6)
IPV6_FAILURE_FATAL="no " //是否启用 IPV6 错误检测功能
NAME="ens33" // 网卡设备的别名
UUID=xxxxxxxxxxxxxxxxxxxxxx //网卡设备的 UUID 唯一标识号
DEVICE="ens33" //网卡的设备名称 (不推荐改)
ONBOOT="yes" //是否开机自启动
IPADDR="192.168.1.10" //网卡的 IP 地址
NETMASK="255.255.255.0" //IP 地址的子网掩码
GATEWAY="192.168.1.1" //默认网关 IP 地址
DNS1="114.114.114.114" //DNS 域名解析服务器的 IP 地址
DNS2="8.8.8.8" //DNS2 的 IP 地址

```

▼ 获取ip、网关等配置信息

获取IP、网关等配置范围信息：



▼ 配置静态ip信息

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO= static          #修改为静态IP模式
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"             #IPV6是否启用
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens3"
UUID="1a5eda7a-8e5c-4720-b059-1567cd02583"
DEVICE="ens3"
ONBOOT="yes"
IPADDR="192.168.177.129"  #需要添加的信息
NETMASK="255.255.255.0"
GATEWAY="192.168.177.2"
DNS1="8.8.8.8"             #网卡名称
DNS2="8.8.8.8"             #唯一识别码
                           #设备名称
                           #yes:开启自动启用网络连接,no:自动启用网络连接
                           #IP地址
                           #子网掩码
                           #默认网关
                           #主DNS
                           #备DNS
```

▼ 配置过程

(1) 使用su切换到root使用vi或vim进行编辑ifcfg-ens33网卡配置文件

```
[zhangzhihu@192 ~]$ su
[zhangzhihu@192 ~]$ su
密码:
[root@192 zhangzhihu]# vi /etc/sysconfig/network-scripts/ifcfg-ens33
```

(2) 将ip获取修改为静态（static），并将IP、子网掩码、网关等信息写入或修改。

(3) 保存退出

```
:wq
```

(4) 配置完成之后需要重启网络服务
service network restart

```
[root@192 zhangzhihu]# service network restart
Restarting network (via systemctl): [确定]
[root@192 zhangzhihu]#
```

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"          #需要添加的信息
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"               #网卡名称
IPV6_AUTOCONF="yes"           #唯一识别码
IPV6_DEFROUTE="yes"           #设备名称
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens3"
UUID="1a5eda7a-8e5c-4720-b059-1567cd02583"
DEVICE="ens3"
ONBOOT="yes"
IPADDR="192.168.177.129"     #IP地址
NETMASK="255.255.255.0"       #子网掩码
GATEWAY="192.168.177.2"       #默认网关
DNS1="8.8.8.8"                 #主DNS
DNS2="8.8.8.8"                 #备DNS
```

▼ 测试网络状态

(1) ping命令进行测试至网关网络状态

```
[root@192 zhangzhihu]# ping 192.168.177.2
PING 192.168.177.2 (192.168.177.2) 56(84) bytes of data.
64 bytes from 192.168.177.2: icmp_seq=1 ttl=128 time=0.281 ms
64 bytes from 192.168.177.2: icmp_seq=2 ttl=128 time=0.524 ms
64 bytes from 192.168.177.2: icmp_seq=3 ttl=128 time=0.450 ms
64 bytes from 192.168.177.2: icmp_seq=4 ttl=128 time=0.447 ms
64 bytes from 192.168.177.2: icmp_seq=5 ttl=128 time=0.445 ms
^C
-- 192.168.177.2 ping statistics --
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.281/0.429/0.524/0.082 ms
[root@192 zhangzhihu]#
```

网络正常状态

```
[root@192 zhangzhihu]# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
^C
-- 192.168.2.1 ping statistics --
12 packets transmitted, 0 received, 100% packet loss, time 11014ms
[root@192 zhangzhihu]#
```

与某些网络不通状态

(2) ping 命令进行测试至Internet网络状态

```
[root@192 zhangzhihu]# ping www.baidu.com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
64 bytes from 104.193.88.77 (104.193.88.77): icmp_seq=1 ttl=128 time=155 ms
64 bytes from 104.193.88.77 (104.193.88.77): icmp_seq=2 ttl=128 time=249 ms
64 bytes from 104.193.88.77 (104.193.88.77): icmp_seq=3 ttl=128 time=168 ms
^C
-- www.wshifen.com ping statistics --
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 155.187/191.097/249.388/41.587 ms
[root@192 zhangzhihu]#
```

与互联网正常状态

注意：访问外网，需要接入宿主网络能上Internet

▼ DNS配置文件

▼ 文件与路径配置

- DNS客户机配置文件，用于设置DNS服务器的IP地址及DNS域名，还包含了主机的域名搜索顺序。该文件是由域名解析器使用的配置文件
- /etc/resolve.conf配置的是本地DNS服务器，也就是本机要访问某个域名时，就会向/etc/resolve.conf中配置的DNS服务器发起DNS解析请求。

▼ nameserve

- 定义DNS服务器的IP地址

▼ domain

- 定义本地域名

▼ search

- 定义域名的搜索列表

▼ sortlist

- 对返回的域名进行排序

DNS配置文件，是用于配置DNS IP，用于访问互联网时所需要。

路径：

cat **/etc/resolv.conf**

```
[root@192 zhangzhihu]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
[root@192 zhangzhihu]#
```

配置DNS：

使用**vi /etc/resolv.conf** 进入后修改

```
# Generated by NetworkManager
nameserver 8.8.8.8
nameserver 8.8.4.4
~
~
~
~
- INSERT -
```

```
[root@192 zhangzhihu]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
nameserver 8.8.4.4
[root@192 zhangzhihu]#
```

▼ 主机名配置文件—文件路径与配置

- **/etc/hostname** 文件是主机配置名
- **/etc/hosts** 文件是linux系统中负责IP地址与域名的快速解析的文件
-

路径：

cat /etc/hostname

cat /etc/hosts

```
[root@localhost zhangzhihu]# cat /etc/hostname
localhost.localdomain
[root@localhost zhangzhihu]# cat /etc/hosts
127.0.0.1 localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@localhost zhangzhihu]#
```

修改主机名为zhangzhihu：

使用**vi /etc/hostname**进入后修改localhost.localdomain为zhangzhihu

使用**vi /etc/hosts**进入后修改localhost.localdomain为zhangzhihu

```
[root@localhost zhangzhihu]# cat /etc/hostname
zhangzhihu
[root@localhost zhangzhihu]# cat /etc/hosts
127.0.0.1 localhost zhangzhihu localhost4 zhangzhihu4
::1 localhost zhangzhihu localhost6 zhangzhihu6
[root@localhost zhangzhihu]#
```

▼ 主机名规则

- 只能出现0-9, a-z, A-Z, -, .
- 开头和结尾不能是.
- 主机名长度必须是1-63个字符长度

▼ hostname-s

- 显示简名，即第一个.之前全部名称

▼ hostname

- 显示全部名称

▼ 网络配置工具NetworkManager

- NetworkManager 是 2004 年 RedHat 启动的项目，旨在能够让 Linux 用户更容易轻松的处理现代网络需求，尤其是无线网络，能够自动发现网卡并配置 IP 地址

▼ NetworkManager 主要管理 2 个对象

- Connection (网卡连接配置)
- Device (网卡设备)
- 他们之间是多对一的关系，但是同一时刻只能有一个 Connection 对于 Device 才生效

▼ NetworkManager 的优点

- 简化网络连接的工作
- 让桌面本身和其他应用程序能感知网络

▼ 使用两种方式

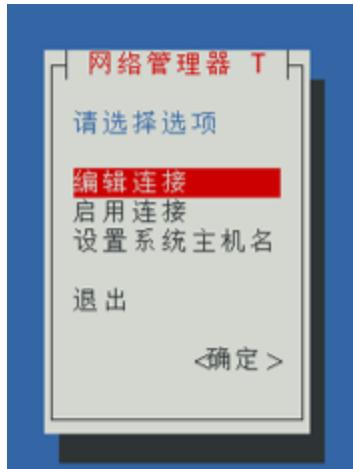
- nmtui 界面操作
- nmcli 命令行操作

▼ nmtui 界面操作—启动 nmtui 界面

▼ 命令

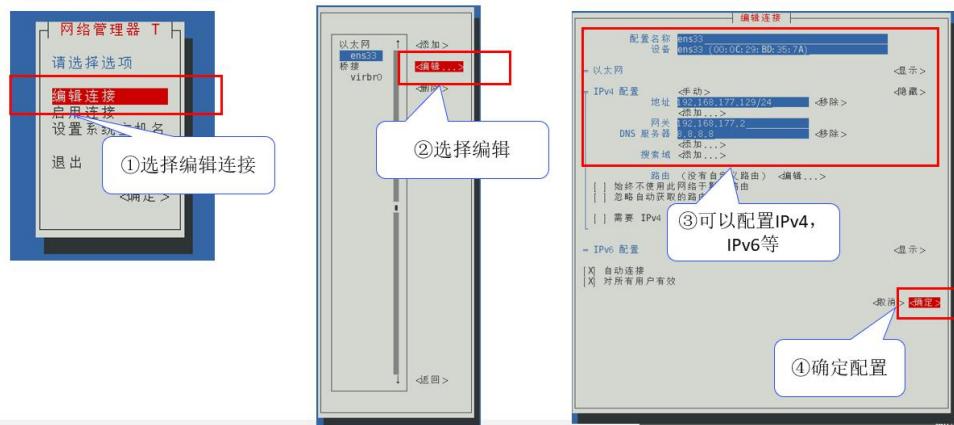
- nmtui

▼ 界面



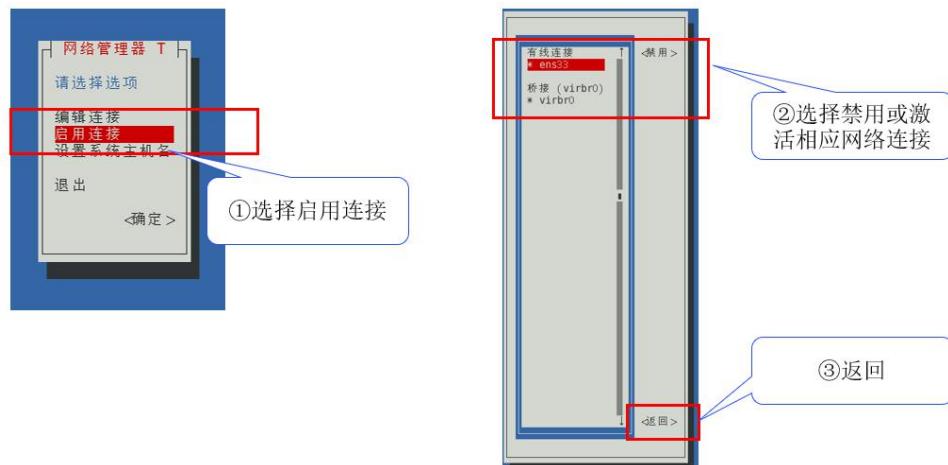
▼ 编辑连接

编辑连接: 设备IP、网关、DNS等信息。使用tab键或方向键, Enter键进行选择



▼ 启用链接

启用连接: 可以禁用网络或切换连接网络方式。使用tab键或方向键, Enter键进行选择



▼ 设置系统主机名

设置主机名: 填写主机名。使用tab键或方向键, Enter键进行选择



▼ 查看连接和设备信息

▼ 查看连接

- nmcli connection

- nmcli con show

▼ 查看设备

- nmcli device
- nmcli device status
- nmcli dev show

查看连接: **nmcli connection
nmcli con show**

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection
名称 UUID          类型    设备
ens33  1a5eda7a-8e5c-4720-b059-1567cd025837  802-3-ethernet  ens33
virbr0  7367fae7-d84c-4de9-a6b4-852e44183768  bridge        virbr0
[zhangzhihu@zhangzhihu ~]$
```

查看设备: **nmcli device
nmcli device status
nmcli dev show**

```
[zhangzhihu@zhangzhihu ~]$ nmcli device
设备  类型 状态 连接
virbr0  bridge  连接的 virbr0
ens33  ethernet  连接的 ens33
lo    loopback  未托管 -
virbr0-nic tun   未托管 -
[zhangzhihu@zhangzhihu ~]$
```

nmcli device

nmcli device status

```
[zhangzhihu@zhangzhihu ~]$ nmcli dev show
GENERAL设备: virbr0
GENERAL类型: bridge
GENERAL硬件: 52:54:00:BF:7E:8A
GENERAL状态: 1500
GENERAL状态: 100 (活动的)
GENERAL连接: virbr0
GENERAL连接路径: /org/freedesktop/NetworkManager/ActiveConnection/2
IP4地址[1]:
  IP4地址: 192.168.122.1/24
  IP4网关: 
  IP4子网掩码: 255.255.255.0
  IP4广播: 
  IP4路由器: 
  IP4地址[1]: 192.168.177.129/24
  IP4网关: 192.168.177.2
  IP4子网掩码: 255.255.255.0
  IP4广播: 192.168.177.255
  IP4地址[1]: fe80::4aa4:f06e:82de:1d4b/64
  IP4网关: 

GENERAL设备: ens33
GENERAL类型: ethernet
GENERAL硬件: 00:0C:29:BD:35:7A
GENERALMTU: 1500
GENERAL状态: 100 (连接的)
GENERAL连接: ens33
GENERAL连接路径: /org/freedesktop/NetworkManager/ActiveConnection/13
WIRED-PROPERTIES 装置: 开
IP4地址[1]:
  IP4地址: 192.168.177.129/24
  IP4网关: 192.168.177.2
  IP4子网掩码: 255.255.255.0
  IP4广播: 192.168.177.255
  IP4地址[1]: 8.8.8.8
  IP4地址[1]: fe80::4aa4:f06e:82de:1d4b/64
  IP4网关: 

GENERAL设备: lo
GENERAL类型: loopback
GENERAL硬件: 00:00:00:00:00:00
GENERALMTU: 65536
GENERAL状态: 10 (未托管)
GENERAL连接: -
GENERAL连接路径: -
IP4地址[1]:
  IP4地址: 127.0.0.1/8
  IP4网关: 
  IP4子网掩码: 255.0.0.0
  IP4广播: 
  IP4地址[1]: ::1/128
  IP4网关: 

GENERAL设备: virbr0-nic
GENERAL类型: tun
GENERAL硬件: 52:54:00:BF:7E:8A
GENERALMTU: 1500
GENERAL状态: 10 (未托管)
GENERAL连接: -
GENERAL连接路径: -
[zhangzhihu@zhangzhihu ~]$
```

nmcli

查看ip:
nmcli

```
[zhangzhihu@zhangzhihu ~]$ nmcli
virbr0: 连接的 to virbr0
  "virbr0"
  bridge, 52:54:00:BF:7E:8A, sw, mtu 1500
  inet4 192.168.122.1/24

ens33: 连接的 to ens33
  "Intel 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000 MT Single Port Adapter)"
  ethernet (e1000), 00:0C:29:BD:35:7A, hw, mtu 1500
  ip4 default
  inet4 192.168.177.129/24
  inet6 fe80::4aa4:f06e:82de:1d4b/64

lo: 未托管
  "lo"
  loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

virbr0-nic: 未托管
  "virbr0-nic"
  tun, 52:54:00:BF:7E:8A, sw, mtu 1500

DNS configuration:
  servers: 8.8.8.8
  interface: ens33

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(5) manual pages for complete usage details.
[zhangzhihu@zhangzhihu ~]$
```

IP配置信息

DNS配置信息

▼ nmcli 命令行操作— dhcp

使用“**dhcp**”创建新的连接

nmcli connection add con-name "dhcp" type ethernet ifname ens33

connection add – 添加新的连接

con-name – 连接名

type – 设备类型

ifname – 接口名

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection add con-name "dhcp" type
ethernet ifname ens33
连接 "dhcp" (cf606c51-49f0-4bd7-9f16-398e4e869078) 已成功添加。
[zhangzhihu@zhangzhihu ~]$
```

更新连接:

nmcli connection up dhcp

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection up dhcp
连接已成功激活 (D-Bus 活动路径:
/org/freedesktop/NetworkManager/ActiveConnection/18)
```

动态获取的IP

```
[zhangzhihu@zhangzhihu ~]$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::/128 scope host
    valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 00:0c:29:bd:35:7a brd ff:ff:ff:ff:ff:ff
inet 192.168.177.132/24 brd 192.168.177.255 scope global dynamic ens33
    valid_lft 1739sec preferred_lft 1739sec
    inet6 fe80::81e5:440a:46c7:3b16/64 scope link
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
link/ether 52:54:00:0b:7e:8a brd ff:ff:ff:ff:ff:ff
inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
    valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
link/ether 52:54:00:0b:7e:8a brd ff:ff:ff:ff:ff:ff
[zhangzhihu@zhangzhihu ~]$
```

▼ nmcli 命令行操作— static

不通过**dhcp**分配IP，使用“**static**”添加地址

nmcli connection add con-name "static" ifname ens33 autoconnect no type ethernet ip4 192.168.177.130 gw4 192.168.177.2

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection add con-name "static"
ifname ens33 autoconnect no type ethernet ip4 192.168.177.130 gw4
192.168.177.2
连接 "static" (16f69b6e-9e28-44e9-a482-fc6714ab1938) 已成功添加。
```

更新连接:

nmcli connection up static

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection up static
连接已成功激活 (D-Bus 活动路径:
/org/freedesktop/NetworkManager/ActiveConnection/17)
```

静态ip

```
[zhangzhihu@zhangzhihu ~]$ ip addr show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 00:0c:29:bd:35:7a brd ff:ff:ff:ff:ff:ff
inet 192.168.177.130/32 brd 192.168.177.130 scope global dynamic ens33
    valid_lft forever preferred_lft forever
    inet6 fe80::ae2e:46a0:16d1:ba80/64 scope link
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
link/ether 52:54:00:0b:7e:8a brd ff:ff:ff:ff:ff:ff
inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
    valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
link/ether 52:54:00:0b:7e:8a brd ff:ff:ff:ff:ff:ff
[zhangzhihu@zhangzhihu ~]$
```

▼ nmcli 命令行操作— dns(静态连接中)

添加**DNS**设置到静态连接中

nmcli connection modify "static" ipv4.dns 8.8.8.8

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection add con-name "static"
ifname ens33 autoconnect no type ethernet ip4 192.168.177.130 gw4
192.168.177.2
连接 "static" (16f69b6e-9e28-44e9-a482-fc6714ab1938) 已成功添加。
```

重启静态连接

添加更多的**DNS**:

nmcli connection modify "static" +ipv4.dns 8.8.4.4

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection modify "static" +ipv4.dns
8.8.4.4
[zhangzhihu@zhangzhihu ~]$
```

```
[zhangzhihu@zhangzhihu ~]$ nmcli connection modify "static" ipv4.dns 8.8.8.8
[zhangzhihu@zhangzhihu ~]$ ping www.baidu.com
ping: www.baidu.com: 未知的名称或服务
[zhangzhihu@zhangzhihu ~]$ nmcli connection up static
连接 Ping 测试互联网通信状态
[org.freedesktop.NetworkManager.ActiveConnection/20]
[zhangzhihu@zhangzhihu ~]$ ping www.baidu.com
PING www.wshifen.com [103.235.46.39] 56(84) bytes of data.
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=1 ttl=128 time=205 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=2 ttl=128 time=199 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=3 ttl=128 time=203 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=4 ttl=128 time=196 ms
^C
-- www.wshifen.com ping statistics --
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 196.513/201.257/205.592/3.544 ms
[zhangzhihu@zhangzhihu ~]$
```

- ▼ 软件包管理
 - ▼ 设置软件仓库
 - ▼ 软件仓库
 - 是一个预备好的目录，或是一系列存放软件的服务器，或者是一个网站，包含了软件包和索引文件
 - ▼ 软件管理工具
 - 例如yum（软件管理工具）可以在仓库中自动的定位并获取正确的RPM（开放软件管理系统）软件包
 - ▼ RPM基本知识
 - RPM是一个开放的软件包管理系统，最初的全称是Red Hat Package Manager
 - ▼ 五大功能
 - ▼ 安装
 - 将软件从包中解出来并安装到硬盘
 - ▼ 卸载
 - 将软件从硬盘清除
 - ▼ 升级
 - 替换软件的旧版本
 - ▼ 查询
 - 查询软件包的信息
 - ▼ 验证
 - 验证系统中的软件与软件包中的软件的区别
 - ▼ RPM包的名称格式
 - name-version.type.rpm
 - ▼ 例如
 - ▼ linuxqq-v1.2.2-beta.i386.rpm
 - name: 软件的名称。linuxqq
 - version: 软件版本号。v1.0.2-beta1
 - type: 包的类型。i386
 - rpm: 文件拓展名
 - ▼ RPM常用命令

- ▼ rpm-l<.pm filename>
 - 安装指定的.pm文件
- ▼ rpm-u<rpm file name >
 - 升级同名包
- ▼ rpm-v<package-name>
 - 校验指定的软件包
- ▼ rpm-e<package-name >
 - 删除指定的软件包
- ▼ YUM基本知识
 - YUM 是"Yellow dog Updater, Modified"的简称，最早是由 Yellow dog 发行版的开发者 Terra Soft 研发，用 Python 写成，那时叫做 YUP (Yellow dog Updater) ，后经杜克大学的 Linux@Duke开发团队进行改进，遂有此名。
 - YUM 的宗旨是自动化地升级，安装/移除 RPM 包，收集 RPM 包的相关信息，检查依赖性并自动提示用户解决。
- ▼ 特点
 - 自动解决包的依赖性问题，能更方便的添加/删除/更新RPM包
 - 便于管理大量系统的更新问题
 - 可以同时配置多个仓库
- ▼ 简洁的配置文件
 - /etc/yum.conf
 - 保持与RPM数据库的一致性
 - 有较详细的日志，可以查看何时安装升级了什么软件包等
- ▼ YUM包含的组件
 - ▼ yum命令
 - yum命令使用 提供的众多功能
 - ▼ yum插件
 - 用于扩展yum的功能
 - ▼ yum仓库
 - 软件仓库

- ▼ yum缓存
 - 下载缓存在/var/cache/yum目录中
- ▼ YUM命令语法
 - YUM 【全局参数】命令 【命令参数】
- ▼ 配置软件仓库—仓库配置文件
 - (/etc/yum.repos.d/*.repo)

YUM 使用仓库配置文件（文件名以repo 结尾文件）配置仓库的镜像站点地址等配置信息。默认情况下，CentOS 7 在/etc/yum.repos.d/目录下包含 7 个配置文件。主要的仓库配置文件为 CentOS-Base. repo，其余配置文件中的仓库默认均未启用。采用分段形式，每一段配置一个软件仓库，配置语法如下。

```
[root@zhangzhihu ~]# ll /etc/yum.repos.d/
总用量 28
-rw-r--r-- 1 root root 1664 8月 30 2017 CentOS-Base.repo
-rw-r--r-- 1 root root 1309 8月 30 2017 CentOS-CR.repo
-rw-r--r-- 1 root root 649 8月 30 2017 CentOS-Debuginfo.repo
-rw-r--r-- 1 root root 314 8月 30 2017 CentOS-fasttrack.repo
-rw-r--r-- 1 root root 630 8月 30 2017 CentOS-Media.repo
-rw-r--r-- 1 root root 1331 8月 30 2017 CentOS-Sources.repo
-rw-r--r-- 1 root root 3830 8月 30 2017 CentOS-Vault.repo
[root@zhangzhihu ~]#
```

- (1)文件中以"#"开头的行是注释行。
- (2)**repositoryid**：用于指定一个仓库，必须保证此值的唯一性。比如**base**、**update**、**centosplus**等。
- (3)**name**：用于指定易读的仓库名称。
- (4)**baseurl**：用于指定本仓库的 URL，可以是如下 3 种类型：
 http：用于指定远程 HTTP 协议的源。
 ftp：用于指定远程 FTP 协议的源。
 file：用于本地镜像或 NFS 挂装文件系统。
- (5)**mirrorlist**：用于指定仓库的镜像站点列表。
- (6)**enabled**：用于指定是否使用本仓库，默认值为 1，即可用。
- (7)**gpgcheck**：用于指定是否检查软件包的 GPG 签名。
- (8)**gpgkey**：用于指定 GPG 签名文件的 URL。

▼ 配置软件仓库—配置 YUM 仓库 (1)

(1) CentOS 的 YUM仓库:仓库配置文件 /etc/yum.repos.d/*.repo 配置了 yum 命令在安装和查询软件时连接的 YUM 仓库地址，而 CentOS 的 YUM 仓库存放在 CentOS 的镜像站点中。

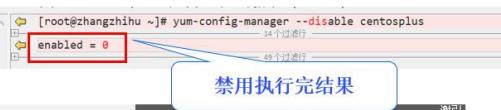
<http://www.centos.org/download/mirrors/> 提供了 CentOS 的镜像站点列表。并且可以在 <http://mirrorstatus.centos.org/>查看镜像站点的状态。

官方仓库有四个：**base (发行包)**、**update (基本base的升级包)**、**extras (扩展包)**、**centosplus(增强功能)**。

(2) 仓库的启用和禁用：

方法一：修改配置文件中enabled。使用vi编辑器进行修改。

方法二：使用**yum-config-manager**。

启用 centosplus :	yum-config-manager --enable centosplus	禁用 centosplus :	yum-config-manager --disable centosplus
			
启用执行完结果		禁用执行完结果	

(3) 配置仓库镜像站点

在仓库配置文件 `CentOS-Base.repo` 中默认配置了 `mirrorlist` 的 URL 地址，且默认安装启用了 `yum-plugin-fastestmirror` 插件，因此执行 `yum` 命令时默认会从镜像地址列表中选择一个速度最快的镜像地址，并从此地址获取软件包。

并非所有的国内镜像都在 CentOS 的镜像站点列表中，所以可以使用 `baseurl` 直接指定最近或最快的镜像仓库。为了加快更新，在确保更新服务器及线路良好的情况下，在 `baseurl` 中只指定一个 URL 即可。

使用 `baseurl` 配置仓库地址还用于如下情况：

- 指定局域网中的一个本地镜像地址。
- 在 `yum` 客户配置了代理时，为了避免代理缓存更多的不同镜像站点的相同内容。为了加快更新，国内用户可以修改仓库配置文件使用国内的镜像站点。

常用国内的镜像站点 (HTTP/HTTPS) :

1. 清华大学镜像站 <https://mirror.tuna.tsinghua.edu.cn/>
2. 阿里云镜像站 <https://developer.aliyun.com/mirror/>
3. 网易云镜像站 <http://mirrors.163.com/>
4. 华中科技大学镜像站 <http://mirror.hust.edu.cn/>
5. 北京交通大学镜像站 <https://mirror.bjtu.edu.cn/>
6. 中国科学技术大学镜像站 <https://mirror.bjtu.edu.cn/>
7. 搜狐镜像站 <http://mirrors.sohu.com/>

▼ 配置软件仓库—配置 YUM 仓库 (2)

配置仓库镜像站点步骤：

(1) 指到YUM仓库配置文件目录(`/etc/yum.repos.d/`)：

`cd /etc/yum.repos.d/`

(2) 备份原始base配置文件 (`CentOS-Base.repo`文件)：

`cp CentOS-Base.repo CentOS-Base.repo.orig`

(3) 注释`mirrorlist` 配置行：

`sed -i "s/^mirrorlist/#mirrorlist/g" CentOS-Base.repo`

(4) 启用`baseurl`配置行：

`sed -i "s/^#baseurl/baseurl/g" CentOS-Base.repo`

(5) 修改镜像站的URL：

`sed -i "s/mirror.centos.org/mirrors.sohu.com/g" CentOS-Base.repo`

注意：`sed -i` 就是直接对文本文件进行操作的。`sed -i 's/原字符串/新字符串/' /home/1.tx`

注：修改配置文件的内容，也可以进行 vi编辑器进行。建议使用`sed -i`命令，替换字符串。

▼ 配置软件仓库—配置后文件内容

```
[base]
name=CentOS $releasever - Base
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
baseurl=http://mirrors.sohu.com/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS $releasever - Updates
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra
baseurl=http://mirrors.sohu.com/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful
[extras]
name=CentOS $releasever - Extras
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras&infra=$infra
baseurl=http://mirrors.sohu.com/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS $releasever - Plus
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus&infra=$infra
baseurl=http://mirrors.sohu.com/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

▼ 使用 yum 安装软件

▼ yum基本命令

▼ yum是YUM系统的字符界面管理工具

▼ 语法格式

- yum 【全局参数】命令【命令参数】

▼ 全局参数

▼ -y

- 对于yum命令的所有提问都回答yes

▼ -C

- 只利用本地缓存，不从远程仓库下载文件

▼ --enablerepo=REPO

- 临时启用指定的名为REPO的仓库

▼ --disablerepo=REPO

- 临时禁用指定名为REPO的仓库
- ▼ -insallroot=PATH
- 指定安装软件时的根目录，主要用于为chroot环境安装软件
- ▼ 常用语法
- ▼ yum check-update
 - 检查可更新的所有软件包
- ▼ yum update
 - 下载跟新系统已经安装的所有软件包
- ▼ yum upgrade
 - 大规模的版本升级
- ▼ yum install<packages>
 - 安装新软件包
- ▼ yum update<packages>
 - 更新指定的软件包
- ▼ yum remove<packages>
 - 移除指定的软件包
- ▼ yum list installed
 - 列出资源库中已经安装的rpm包
- ▼ yum工具安装软件
- ▼ yum install git lftp
 - 安装指定的软件包
- ▼ yum list installed git|grep git
 - 查看软件安装情况
- ▼ yum update git lftp
 - 升级指定的软件包
- ▼ yum remove git lftp
 - 卸载指定的软件包

