

RHCE8: RHEL8

RHEL7 数据中心方向

RHEL8 自动化方向

系统管理 1 系统管理 2 系统管理 3

考试地点：创想云教育培训中心

考试时间：6.5 小时（RHCSA 2.5 小时；RHCE 4 小时）

上午：09:00-12:00 下午：13:30-16:30

晚上：18:30-20:30（练习辅导时间）

系统管理 1（RH124）

foundationX 物理机，使用 KVM 虚拟化

VIRTUAL MACHINES PROVIDED

VMNAME	#CPUS	RAM	PLATFORM	
-----	-----	---	-----	
classroom	1	1GB	RHEL 8.0	(instructor only)
bastion	1	1GB	RHEL 8.0	
workstation	2	2GB	RHEL 8.0	
servera	1	1GB	RHEL 8.0	
serverb	1	1GB	RHEL 8.0	
serverc	1	1GB	RHEL 8.0	
serverd	1	1GB	RHEL 8.0	

A NOTE ABOUT HARDWARE REQUIREMENTS

The instructor and each student in this course should have a single machine with multiple CPUs and 16GB of RAM. SSD CPU

```
[kiosk@foundation0 ~]$ rht-vmctl reset all 准备实验环境，恢复虚机快照
Are you sure you want to reset bastion workstation servera serverb serverc
serverd? (y/n) y
Powering off bastion..
Powering off workstation..
Powering off servera..
Powering off serverb..
Powering off serverc..
Powering off serverd..
Resetting bastion.

[kiosk@foundation0 ~]$ rht-vmctl reset all 建议大家每次上课前执行该命令
```

第一章 红帽企业版 LINUX 入门

LINUX 用途，应用非常广泛

LINUX 特点： LINUX 是开源软件 强大命令行接口 模块化操作系统

RHEL8 OpenShift (PAAS) S2I Docker+K8S OpenStack (IAAS)

CentOS 完全免费，易于安装

<https://access.redhat.com> 下载产品 知识库

第二章 访问命令行

登录 LINUX 系统：（设置用户自动登录）

- 1、图形化登录 Login 程序
- 2、字符界面登录 Login 程序

实验室用户名和密码: workstation servera serverb serverc serverd
用户名: root 密码: redhat
用户名: student 密码: student

```
[kiosk@foundation0 ~]$ ssh root@servera  
[root@servera ~]#
```

什么是 SHELL?

SHELL 与操作系统或应用交互的程序

SHELL 是一个命令解释器（翻译），解释用户输入的命令

用户每输入一条命令，SHELL 解释执行命令，立即回应，交互

应用程序

命令解释器（SHELL）

系统核心（内核）

硬件

SHELL 存在于操作系统最外层，负责与用户直接对话，把用户输入的命令解释给操作系统，并处理各种各样的操作系统的输出结果，然后输出结果返回给用户

使用用户名和密码登录到 LINUX 系统之后的所有操作都是由 SHELL 解释并执行

学习 SHELL 编程

SHELL 脚本语言是实现 LINUX/UNIX 系统管理自动化运维必备工作

```
[root@servera ~]# echo $SHELL 默认 SHELL  
/bin/bash
```

```
[root@servera ~]# cat /etc/shells  
/bin/sh  
/bin/bash  
/usr/bin/sh  
/usr/bin/bash  
/usr/bin/tmux  
/bin/tmux
```

Bash 提示符:

```
[root@servera ~]#
```

代表 ROOT 用户登录 SHELL 的提示符（管理员提示符）

```
[student@servera ~]$
```

\$ 代表非 ROOT 用户登录 SHELL 的提示符（普通用户提示符）

什么是 SHELL 脚本?

当命令不在命令行下执行，而是通过一个程序文件来执行

程序文件：SHELL 脚本

```
# 删除日志 版本 1
cd /var/log
cat /dev/null > messages
echo "Logs cleaned up"

# 带有条件的删除日志脚本 版本 2
#!/bin/bash
LOG_DIR=/var/log
ROOT=0
IF [ "uid" -ne $ROOT ]
Then
    Echo "Must be root to run this script"
    Exit 1
fi

cd $LOG_DIR || {
    echo "Cannot change to directory"
    exit 1
}

Cat /dev/null > messages && {
    Echo "Logs cleaned up"
    Exit 0
}
Echo "Logs clean up fail"
Exit 1
```

命令语法: `rht-vmctl`

命令	选项	参数
<code>Useradd</code>	<code>-s</code>	<code>/sbin/nologin</code>

命令：一个可执行程序，可以是脚本，也可以是二进制可执行程序

选项：长选项 `--shell` 短选项 `-s`

参数：命令可以接受东西

```
Useradd -s /sbin/nologin -u 2000 -d /tmp/Natasha
```

```
[root@servera ~]# ls -la            ls -l -a
```

```
[root@servera ~]# ls --shell
```

命令	子命令	参数	
[kiosk@foundation0 ~]\$ rht-vmctl reset all			
fullreset	reset	start	stop
poweroff	save	status	view


```
[root@servera ~]# nmcli
```

agent	device	help	networking
connection	general	monitor	radio


```
[root@servera ~]# systemctl
```

add-requires	is-active	reload-or-restart
add-wants	is-enabled	rescue
cancel	is-failed	reset-failed
cat	isolate	restart
condreload	is-system-running	revert
condrestart	kexec	set-default
condstop	kill	set-environment
daemon-reexec	link	set-property
daemon-reload	list-dependencies	show
default	list-jobs	show-environment
disable	list-machines	start
edit	list-sockets	status
emergency	list-timers	stop
enable	list-unit-files	suspend
exit		list-units
suspend-then-hibernate		
force-reload	mask	switch-root
get-default	poweroff	try-reload-or-restart
halt	preset	try-restart
help	preset-all	unmask
hibernate	reboot	unset-environment
hybrid-sleep	reenable	
import-environment	reload	

BASH SHELL 功能:

1 命令历史

```
[root@servera ~]# history
```

[root@servera ~]# ll .bash_history 在 SHELL 中执行的命令存储在该文件
只有当用户正常关机或重启或注销

```
-rw-----. 1 root root 70 Apr 12 11:08 .bash_history
```

搜索命令历史: 快捷键 `ctrl+r`

```
[root@servera ~]# env |grep HIST
HISTCONTROL=ignoredups
HISTSIZE=1000 默认保存 1000 个命令
```

```
[root@servera ~]# HISTTIMEFORMAT="%F %T "
```

```
[root@servera ~]# cat .bash_profile
# .bash_profile
```

```
export USER_IP=`who -u am i 2> /dev/null |awk '{print $NF}'|sed -e
's/[()]/g`
export HISTTIMEFORMAT="%F %T $USER_IP: `whoami` "
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

```
# User specific environment and startup programs
```

```
PATH=$PATH:$HOME/bin
```

```
export PATH
```

```
[root@servera ~]# source .bash_profile
[root@servera ~]# history
```

```
HISTCONTROL=ignorespace 命令前加空格不记录命令历史
[root@servera ~]# useradd Robert
```

2 命令别名

```
[root@servera ~]# alias p="ps axo pid,comm,nice"
[root@servera ~]# p
```

3 命令补齐 参数补齐 子命令补齐 (只记住前面几个字母的命令)

Tab

```
[root@servera ~]# user
```

```
useradd      userdel      userhelper  usermod      users
[root@servera ~]# passwd
```

使用桌面访问命令行:

红帽 RHEL8 默认桌面 GNOME3 --- Wayland(默认) 提供图形化的框架

RHEL7 X-Window System (默认) 提供图形化的框架

Alt+f2 gnome-terminal

```
[root@servera ~]# rpm -qa |grep bash-completion (提供了命令补齐功能)
bash-completion-2.7-5.el8.noarch
```

快捷方式描述

Ctrl+a 跳到命令行的开头

Ctrl+e 跳到命令行的末尾

Ctrl+u 将光标处的命令行开头的内容清除

Ctrl+k 将光标处的命令行末尾的内容清除

Ctrl+ 跳到命令行中前一字开头

Ctrl+ 跳到命令行中下一字末尾

Ctrl+r 在历史记录列表中搜索某一模式的命令

第三章 从命令行管理文件

```
[root@servera ~]# mandb
```

```
[root@servera ~]# whatis ls
```

```
ls (1) - list directory contents
```

```
ls (lp) - list directory contents
```

```
[root@servera ~]# ls
```

```
anaconda-ks.cfg  original-ks.cfg
```

```
[root@servera ~]# ls -a
```

```
.
anaconda-ks.cfg  .bash_logout  .bashrc  .lessht  .ssh  .v
iminfo
..  .bash_history  .bash_profile  .cshrc  original-ks.cfg  .tcshrc
```

```
[root@servera ~]# whatis pwd 查看当前工作目录
```

```
pwd (1) - print name of current/working directory
```

```
pwd (lp) - return working directory name
```

Linux 系统特点，一切皆文件

/

/boot(系统引导文件) /dev(设备文件) /root(管理员家目录) /home(普通用户家目录) /var(日志) /etc(配置文件) /tmp(临时目录)

```
[root@servera ~]# whatis cp 主要用于复制文件或目录
cp (1)                - copy files and directories
cp (1p)               - copy files
```

```
[root@servera ~]# cp /etc/passwd /tmp/
[root@servera ~]# cp /etc /tmp
cp: -r not specified; omitting directory '/etc'
[root@servera ~]# cp -r /etc /tmp 复制目录
-r, --recursive 递归
-a, --archive 复制目录，保留文件属性
```

Rsync 数据备份工具，增量备份（rsync 算法）

```
[root@servera ~]# whatis rsync
rsync (1)                - a fast, versatile, remote (and local) file-copying
tool
```

rsync 和 cp\tar 备份进行比较，rsync 安全性高、备份速度快、支持增量

```
[root@servera ~]# mount 172.25.254.250:/content /mnt
[root@servera isos]# pwd
/mnt/rhel8.0/x86_64/isos
[root@servera isos]# rsync -P rhel-8.0-x86_64-dvd.iso /tmp
rhel-8.0-x86_64-dvd.iso
-P same as --partial -progress
```

```
[root@servera ~]# whatis mv 文件或目录重命名 剪切
mv (1)                - move (rename) files
mv (1p)               - move files
```

```
[root@servera ~]# mv original-ks.cfg ks
```

```
[root@servera ~]# whatis rm 删除文件
rm (1)                - remove files or directories
rm (1p)               - remove directory entries
```



```
[root@servera ~]# alias |grep -i rm
alias rm='rm -i'
```

-f, --force

-r, -R, --recursive 删除目录

remove directories and their contents recursively

```
[root@servera ~]# rm -rf anaconda-ks.cfg
```

恢复数据: [root@servera ~]# rm -rf anaconda-ks.cfg

```
[root@servera ~]# whatis touch
```

touch (1) - change file timestamps

touch (lp) - change file access and modification times

```
[root@servera ~]# touch /etc/newpw
```

```
[root@servera ~]# whatis mkdir
```

mkdir (1) - make directories

mkdir (lp) - make directories

mkdir (2) - create a directory

mkdir (3p) - make a directory relative to directory file descriptor

```
[root@servera ~]# mkdir /cache
```

-p 确保目录名称存在, 不存在的就新建一个

```
[root@servera ~]# mkdir -p /A1/B1/C1
```

-m, --mode=MODE 设置权限

```
[root@servera ~]# mkdir -m 777 /tools
```

```
[root@servera ~]# ll -d /tools
```

```
drwxrwxrwx. 2 root root 6 Apr 12 13:58 /tools
```

```
[root@servera ~]# mkdir -v /tools/redhat
```

```
mkdir: created directory '/tools/redhat'
```

相对路径和绝对路径

绝对路径 以/开始的路径

```
[root@servera ~]# cd /usr/share/doc
```

```
[root@servera doc]# cd zip/
```

切换当前工作目录

```
[root@servera ~]# cd /usr/share/doc/
```

```
[root@servera doc]# cd
```

```
[root@servera ~]# cd ~student      cd /home/student
```

~表示 home 目录

```
[root@servera student]# ls -a
```

```
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc
```

. 表示当前所在目录

.. 表示目录位置的上一层目录

```
[root@servera student]# cd ..
```

命令行文件管理

cp: 复制文件、目录

mv: 移动文件、目录

rm: 删除文件、目录

mkdir: 创建目录

rmdir: 删除空目录

模式匹配项

* 由零个或更多字符组成的任何字符串

? 任何一个字符

~ 当前用户的主目录

~username username 用户的主目录

~+ 当前工作目录

~- 上一工作目录

[abc...] 括起的类中的任何一个字符

[!abc...] 不在括起的类中的任何一个字符

```
[root@servera ~]# touch {a,b,c,d}.txt
```

大括号中以逗号分割的文件列表进行扩展

```
[root@servera ~]# touch {e..h}.txt
```

顺序文件列表

*表示零个或多个任意字符

```
[root@servera device-mapper]# grep -i vda *
```

` \$() 命令置换

```
[root@servera device-mapper]# echo `hostname`
```

```
servera.lab.example.com
```

```
[root@servera device-mapper]# echo $(hostname)
```

servera.lab.example.com

\ 在 BASH SHELL 解释为转义字符，去除字符特殊意义

```
[root@servera ~]# echo $USER
root
[root@servera ~]# echo \$USER
$USER
```

```
[root@servera ~]# ls -l \
> /home
total 0
drwx-----. 2 devops  devops  62 Apr 12 09:55 devops
drwx-----. 2 king    king    62 Apr 12 11:29 king
drwx-----. 2 robert  robert  62 Apr 12 11:30 robert
drwx-----. 2 student student 83 Apr 12 11:06 student
```

第四章 在红帽 LINUX 中获取帮助

```
[root@servera ~]# whatis ls
ls (1)                - list directory contents
ls (lp)               - list directory contents
```

mandb 创建或更新手册索引

man 是 manual 简写 linux 系统中在线帮助命令
/usr/share/man 目录 所有命令说明文档

- 1 Executable programs or shell commands 普通命令的帮助
 - 2 System calls (functions provided by the kernel) 系统调用 open write
 - 3 Library calls (functions within program libraries) 库的调用开发人员
 - 4 Special files (usually found in /dev) 特殊文件
 - 5 File formats and conventions eg /etc/passwd 文件的格式，解释说明文件各的字段的含义
 - 6 Games 游戏
 - 7 Miscellaneous (including macro packages and conventions), e.g. man(7), 宏，变量
groff(7)
 - 8 System administration commands (usually only for root) 系统管理用的命令
- 1 5 8

man [章节] 命令

/string 搜索字符串, 支持正则

n 下一个要搜索的字符串

N 上一个要搜索的字符串

q 退出

-k, --apropos

Equivalent to apropos. Search the short manual page descriptions for

keywords and display any matches. See apropos(1) for details.

man -k keyword 使用关键词搜索文档

man + 命令 158

man l cat cat 命令用法

man -a cat 所有章节

man -k passwd 查询包含 passwd 的帮助文件

[root@servera ~]# ls -help

LINUX 提供了两种命令

1 SHELL 的内部命令, 如: echo shell 内部本身提供的命令

2 SHELL 的外部命令 如:ls shell 本身不提供, 但它是通过 shell 调用

[root@servera ~]# which echo

/usr/bin/echo

[root@servera ~]# mv /usr/bin/echo /tmp

[root@servera ~]# echo Welcome to CLOUDSHELLEDU Training

Welcome to CLOUDSHELLEDU Training

[root@servera ~]# ls

-bash: /usr/bin/ls: No such file or directory

[root@servera ~]# echo \$PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@servera ~]#

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/tmp

[root@servera ~]# ls

a. txt b.txt c.txt d.txt e.txt f.txt

外部命令 ls -help

内部命令 help ls

man bash 查询所有内部命令

man 不区分内部命令和外部命令

```
[root@servera ~]# 11
/etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
-rw-r--r--. 1 root root 471 Apr 12 09:55
/etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
[root@servera ~]# man ifcfg-Wired_connection_1
No manual entry for ifcfg-Wired_connection_1
[root@servera ~]# rpm -qf /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
file /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1 is not
owned by any package
[root@servera ~]# rpm -qf /etc/sysconfig/network-scripts
NetworkManager-1.14.0-14.el8.x86_64
[root@servera ~]# rpm -qd NetworkManager-1.14.0-14.el8.x86_64
/usr/share/doc/NetworkManager/AUTHORS
/usr/share/doc/NetworkManager/CONTRIBUTING
/usr/share/doc/NetworkManager/NEWS
/usr/share/doc/NetworkManager/README
/usr/share/doc/NetworkManager/TODO
/usr/share/doc/NetworkManager/examples/server.conf
/usr/share/man/man1/nm-online.1.gz
/usr/share/man/man1/nmcli.1.gz
/usr/share/man/man5/NetworkManager.conf.5.gz
/usr/share/man/man5/nm-settings-ifcfg-rh.5.gz
/usr/share/man/man5/nm-settings-keyfile.5.gz
/usr/share/man/man5/nm-settings.5.gz
/usr/share/man/man5/nm-system-settings.conf.5.gz
/usr/share/man/man7/nmcli-examples.7.gz
/usr/share/man/man8/NetworkManager.8.gz
```

--query 查询

-f file -qf 查询某一个文件属于哪个软件包

--doc

-qd 查询某一个软件包的帮助文件

man ls pinfo ls

读取/usr/share/doc 中的文档

红帽客户门户

<https://access.redhat.com>

红帽产品文档

<https://access.redhat.com/documentation/>

使用 sosreport 生成 SoS 报告

```
[root@server0 ~]# sosreport  
/var/tmp/sosreport-servera-2020-04-12-zuswtqx.tar.xz
```

如何分析 sosreport 生成的文件 --->xsos

内核版本 系统运行时间 内存使用情况 磁盘信息 网络信息 进程信息

第五章 创建、查看和编辑文本文件

将一个程序的文本输出重定向到文件或另一程序

标准输出： 显示器

正确输出

错误输出

标准输入： 键盘，扫描仪

```
[root@servera ~]# ll /dev/std*  
lrwxrwxrwx. 1 root root 15 Apr 12 09:53 /dev/stderr -> /proc/self/fd/2  
lrwxrwxrwx. 1 root root 15 Apr 12 09:53 /dev/stdin -> /proc/self/fd/0  
lrwxrwxrwx. 1 root root 15 Apr 12 09:53 /dev/stdout -> /proc/self/fd/1  
2 代表错误输出  
0 代表标准输入  
1 代表正确输出
```

```
[root@servera ~]# ls -l > txt  
[student@servera ~]$ find /etc -name passwd  
find: â€™/etc/pki/rsyslogâ€™: Permission denied  
find: â€™/etc/dhcpâ€™: Permission denied  
/etc/pam.d/passwd  
/etc/passwd  
find: â€™/etc/lvm/archiveâ€™: Permission denied  
find: â€™/etc/lvm/backupâ€™: Permission denied  
find: â€™/etc/lvm/cacheâ€™: Permission denied  
find: â€™/etc/polkit-1/rules.dâ€™: Permission denied  
find: â€™/etc/polkit-1/localauthorityâ€™: Permission denied  
find: â€™/etc/sssâ€™: Permission denied  
find: â€™/etc/grub.dâ€™: Permission denied  
find: â€™/etc/auditâ€™: Permission denied  
find: â€™/etc/firewalldâ€™: Permission denied  
find: â€™/etc/sudoers.dâ€™: Permission denied
```

```
[student@servera ~]$ find /etc -name passwd > 1.txt 2> 2.txt
```

```
[student@servera ~]$ find /etc -name passwd &> all.txt
```

```
[root@servera scripts]# cat ping.sh
#!/bin/bash
for i in $(seq 1 254)
do
    ping -c 1 172.25.250.$i &> /dev/null
    if [ $? -eq 0 ];then
        echo "172.25.250.$i active"
    fi
done
```

/dev/null 字符设备文件，黑洞文件，通常用于丢弃不需要的数据的输出

```
curl http://www.cloudshelledu.com 2> /dev/null
```

清除日志：

```
cat /dev/null > /var/log/mylog
```

定时任务：

/ * * * * /bin/sh /scripts/backup.sh 2> /dev/null 所有输入信息（正确和错误）都重定向到黑洞文件进行丢弃

管道：

```
Command1|command2|command3|command4
```

注意：

Command1 必须是正确的输出

为什么用管道：

把多个命令组合在一起，完成复杂的操作

```
mysqldump -u root -p "redhat123" users|gzip -9|ssh username@ip "cat > /backup/$(date +%F).usersdb.gz"
```

统计目录中有多少个子目录

```
[root@servera etc]# ls -l |cut -c 1|grep "d"|wc -l
97
```

ls -l 长格式查看文件属性

cut -c 1 提取第一个字符

grep d 获取文件类型是 d 的

合并两个文件内容：

```
[root@servera public]# cat a.txt |paste -d: - b.txt > ab.txt
[root@servera public]# cat ab.txt
```

```
111:aaaa
222:bbbb
333:cccc
444:dddd
555:eeee
666:ffff
```

统计系统中有多少用户 UID=0

```
[root@servera ~]# cat /etc/passwd | awk -F: '($3==0)' {print $1}'
Root
```

```
[root@servera ~]# cat /proc/cpuinfo | grep "physical id"|wc -l
1
```

```
[root@servera ~]# cat /proc/cpuinfo | grep "core id"|wc -l
1
```

```
[root@servera ~]# exit
```

logout

Connection to servera closed.

查询物理 CPU 个数

```
[kiosk@foundation0 ~]$ cat /proc/cpuinfo | tee cpu.txt | grep "physical id"|wc -l
```

4 4 个 socket, 每个 socket 一个核心

查询 CPU 核数

```
[kiosk@foundation0 ~]$ cat /proc/cpuinfo | grep "core id"|wc -l
4
```

```
[root@servera ~]# ifconfig | grep 172.25.250.10|awk ' {print $2}'
172.25.250.10
```

```
[root@servera ~]# grep 'bash$' /etc/passwd |wc -l
5
```

```
[root@servera ~]# ls
a.txt b.txt c.txt d.txt e.txt f.txt g.txt h.txt txt
```

```
[root@servera ~]#
```

```
[root@servera ~]# ls |rm -rf
```

```
[root@servera ~]# ls
```

```
a.txt b.txt c.txt d.txt e.txt f.txt g.txt h.txt txt
```

```
[root@servera ~]# ls |xargs rm -rf
```

```
[root@servera ~]# ls
```

```
[root@servera ~]#
```

```
[root@servera ~]# ls -l >> a.txt 追加写入重定向的信息
[root@servera ~]# grep student < /etc/passwd
student:x:1000:1000:Student User:/home/student:/bin/bash
```

```
[root@servera ~]# cat /tmp/yyy
y
[root@servera ~]# yum install vsftpd < /tmp/yyy
```

```
[root@servera ~]# touch /etc/myhosts
[root@servera ~]# cat >> /etc/myhosts <<EOF  End of file
> 192.168.0.1    www.cloudshelledu.com
> 192.168.0.2    ftp.cloudshelledu.com
> 192.168.0.3    print.cloudshelledu.com
> EOF
[root@servera ~]# cat /etc/myhosts
192.168.0.1    www.cloudshelledu.com
192.168.0.2    ftp.cloudshelledu.com
192.168.0.3    print.cloudshelledu.com
```

用法：系统自动部署

```
[kiosk@foundation0 ~]$ less /content/ks/foundation.ks
```