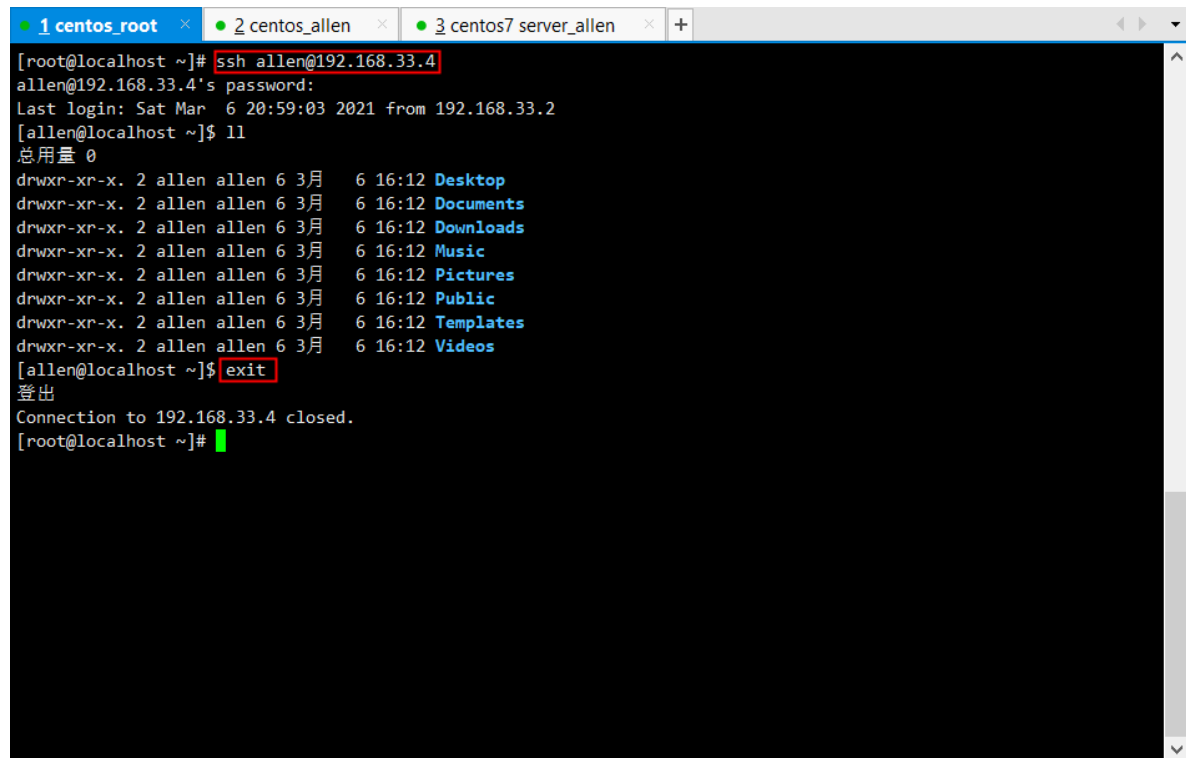


1. SSH

1.1 SSH工作机制

- ssh为Secure Shell（安全外壳协议）的缩写
- 很多ftp、pop和telnet在本质上都是不安全的
- 我们使用的Xshell6就是基于SSH的客户端实现
- SSH的服务端实现为openssh- deamon
- 在Linux上使用ssh命令：`ssh root@192.168.33.3`



```
[root@localhost ~]# ssh allen@192.168.33.4
allen@192.168.33.4's password:
Last login: Sat Mar  6 20:59:03 2021 from 192.168.33.2
[allen@localhost ~]$ ll
总用量 0
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Desktop
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Documents
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Downloads
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Music
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Pictures
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Public
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Templates
drwxr-xr-x. 2 allen allen 6 3月  6 16:12 Videos
[allen@localhost ~]$ exit
登出
Connection to 192.168.33.4 closed.
[root@localhost ~]#
```

- `exit` 登出

1.2 SSH免密码登录

1. 生成秘钥：`ssh-keygen`

```
1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
[root@localhost ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:/Zf3QMkgUemFRSKVJH8qbwxUkNzGjVGgGw1J+nYJQ root@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]---+
|      o+*@X*      |
|      ..oEX=.o    |
|      o+ooX+      |
|      ooB+=.      |
|      .S.+o+ .    |
|      ..+ o .     |
|      . . + .     |
|      . o.        |
|      .           |
+---[SHA256]-----+
[root@localhost ~]#
```

```
+---[SHA256]-----+
[root@localhost ~]# cd /root/.ssh
[root@localhost .ssh]# ll
总用量 12
-rw-r--r--. 1 root root 1679 3月 6 21:15 id_rsa
-rw-r--r--. 1 root root 408 3月 6 21:15 id_rsa.pub
-rw-r--r--. 1 root root 174 3月 6 21:07 known_hosts
[root@localhost .ssh]# cd /root
[root@localhost ~]# ll
总用量 8
-rw-r--r--. 1 root root 1998 3月 6 15:45 anaconda-ks.cfg
-rw-r--r--. 1 root root 2029 3月 6 16:12 initial-setup-ks.cfg
[root@localhost ~]# ll -a
总用量 32
dr-xr-x---. 6 root root 217 3月 6 21:07 .
dr-xr-xr-x. 17 root root 224 3月 6 15:44 ..
-rw-r--r--. 1 root root 1998 3月 6 15:45 anaconda-ks.cfg
-rw-r--r--. 1 root root 142 3月 6 17:15 .bash_history
-rw-r--r--. 1 root root 18 12月 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 12月 29 2013 .bashrc
drwx-----. 4 root root 31 3月 6 17:09 .cache
drwx-----. 4 root root 30 3月 6 17:09 .config
-rw-r--r--. 1 root root 100 12月 29 2013 .cshrc
drwx-----. 3 root root 25 3月 6 16:11 .dbus
-rw-r--r--. 1 root root 2029 3月 6 16:12 initial-setup-ks.cfg
drwx-----. 2 root root 57 3月 6 21:15 .ssh
-rw-r--r--. 1 root root 129 12月 29 2013 .tcshrc
[root@localhost ~]#
```

- 使用默认的SSH密钥生成方式并保存
- 私钥保存在 /root/.ssh/id_rsa 中，隐藏文件
- 公钥保存在 /root/.ssh/id_rsa.pub. 中，隐藏文件

2. 把自己的公钥拷给对方: `ssh-copy-id 192.168.33.4`

```

[root@localhost ~]# clear
[root@localhost ~]# ll -a
总用量 32
dr-xr-x---. 6 root root 217 3月 6 21:07 .
dr-xr-xr-x. 17 root root 224 3月 6 15:44 ..
-rw-----. 1 root root 1998 3月 6 15:45 anaconda-ks.cfg
-rw-----. 1 root root 142 3月 6 17:15 .bash_history
-rw-r--r--. 1 root root 18 12月 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 12月 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 12月 29 2013 .bashrc
drwx-----. 4 root root 31 3月 6 17:09 .cache
drwx-----. 4 root root 30 3月 6 17:09 .config
-rw-r--r--. 1 root root 100 12月 29 2013 .cshrc
drwx-----. 3 root root 25 3月 6 16:11 .dbus
-rw-r--r--. 1 root root 2029 3月 6 16:12 initial-setup-ks.cfg
drwx-----. 2 root root 57 3月 6 21:15 .ssh
-rw-r--r--. 1 root root 129 12月 29 2013 .tcshrc
[root@localhost ~]# cd .ssh
[root@localhost .ssh]# ll
总用量 12
-rw-----. 1 root root 1679 3月 6 21:15 id_rsa
-rw-r--r--. 1 root root 408 3月 6 21:15 id_rsa.pub
-rw-r--r--. 1 root root 174 3月 6 21:07 known_hosts
[root@localhost .ssh]# ssh-copy-id 192.168.33.4
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new ke
ys
root@192.168.33.4's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.33.4'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost .ssh]#

```

- 将192.168.33.3 生成的公钥 发送到了 192.168.33.4 服务器上了

```

1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
[root@localhost .ssh]# ssh 192.168.33.4
Last login: Sat Mar 6 21:26:40 2021 from 192.168.33.4
[root@localhost ~]# cd /home
[root@localhost home]# ll
总用量 4
drwx-----. 15 allen allen 4096 3月 6 21:11 allen
[root@localhost home]# exit
登出
Connection to 192.168.33.4 closed.
[root@localhost .ssh]#

```

- 使用SSH命令直接连接到了 192.168.33.4 服务器上，不需要使用密码了

3. 基于ssh的文件拷贝: `scp abc.txt 192.168.33.4:/root`

```
1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
dr-xr-x---. 6 root root 217 3月 6 21:07 root
drwxr-xr-x. 38 root root 1200 3月 6 16:14 run
lrwxrwxrwx. 1 root root 8 3月 6 15:37 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 3月 6 16:11 sys
drwxrwxrwt. 15 root root 4096 3月 6 21:33 tmp
drwxr-xr-x. 13 root root 155 3月 6 15:37 usr
drwxr-xr-x. 20 root root 282 3月 6 16:11 var
[root@localhost ~]# touch abc.test.txt ← 创建文件
[root@localhost ~]# ll
总用量 20
-rw-r--r--. 1 root root 0 3月 6 21:34 abc.test.txt
lrwxrwxrwx. 1 root root 7 3月 6 15:37 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 3月 6 15:45 boot
drwxr-xr-x. 20 root root 3300 3月 6 16:11 dev
drwxr-xr-x. 134 root root 8192 3月 6 20:58 etc
drwxr-xr-x. 3 root root 19 3月 6 15:44 home
lrwxrwxrwx. 1 root root 7 3月 6 15:37 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 3月 6 15:37 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 3月 6 15:40 opt
dr-xr-xr-x. 240 root root 0 3月 6 16:11 proc
dr-xr-x---. 6 root root 217 3月 6 21:07 root
drwxr-xr-x. 38 root root 1200 3月 6 16:14 run
lrwxrwxrwx. 1 root root 8 3月 6 15:37 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 3月 6 16:11 sys
drwxrwxrwt. 15 root root 4096 3月 6 21:33 tmp
drwxr-xr-x. 13 root root 155 3月 6 15:37 usr
drwxr-xr-x. 20 root root 282 3月 6 16:11 var
[root@localhost ~]# scp abc.test.txt 192.168.33.4 /root ← 拷贝文件到192.168.33.4服务器上 (失败)
cp: 无法获取“192.168.33.4”的文件状态(stat): 没有那个文件或目录
[root@localhost ~]# scp abc.test.txt 192.168.33.4:/root ← 拷贝文件到192.168.33.4服务器上 (成功)
abc.test.txt
100% 0 0.0KB/s 00:00
[root@localhost ~]#
```

4. 基于ssh的目录拷贝：需要加 -r 递归拷贝

- `scp aaa -r 192.168.33.4:/root`

5. 远程执行命令： `ssh 192.168.33.4 "echo hello > /root/test.txt"`

```
1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
[root@localhost ~]# ssh 192.168.33.4 "echo hello > /root/test.txt"
[root@localhost ~]#
```

```
1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
[allen@localhost /]$ cd /root
-bash: cd: /root: Permission denied
[allen@localhost /]$ su root
Password:
[root@localhost /]# cd /root
[root@localhost ~]# ll
total 12
-rw-r--r--. 1 root root  0 Mar  6 21:35 abc.test.txt
-rw----- 1 root root 1998 Mar  6 15:45 anaconda-ks.cfg
-rw-r--r-- 1 root root 2029 Mar  6 16:12 initial-setup-ks.cfg
-rw-r--r-- 1 root root  6 Mar  6 21:40 test.txt
[root@localhost ~]# cat test.txt
hello
[root@localhost ~]# rm abc.test.txt
rm: remove regular empty file 'abc.test.txt'? y
[root@localhost ~]# ll
total 12
-rw----- 1 root root 1998 Mar  6 15:45 anaconda-ks.cfg
-rw-r--r-- 1 root root 2029 Mar  6 16:12 initial-setup-ks.cfg
-rw-r--r-- 1 root root  6 Mar  6 21:40 test.txt
[root@localhost ~]#
```

2. 网络配置

1. 查看当前主机名: `hostname`
2. 修改当前主机名 (重启后无效): `hostname 新的主机名`

```
1 centos_root x 2 centos_allen x 3 centos7 server_allen x +
[root@localhost /]# ssh 192.168.33.4 "echo hello > /root/test.txt"
[root@localhost /]# hostname
localhost.localdomain
[root@localhost /]# hostname centos7
[root@localhost /]# hostname
centos7
[root@localhost /]# exist
bash: exist: 未找到命令...
[root@localhost /]# exit
登出
Connection to 192.168.33.4 closed.
[root@centosluban ~]# hostname
centosluban
[root@centosluban ~]# hostname centos7.luban
[root@centosluban ~]# hostname
centos7.luban
[root@centosluban ~]#
```

3. 修改当前主机名 (重启后有效): `vi /etc/sysconfig/network`

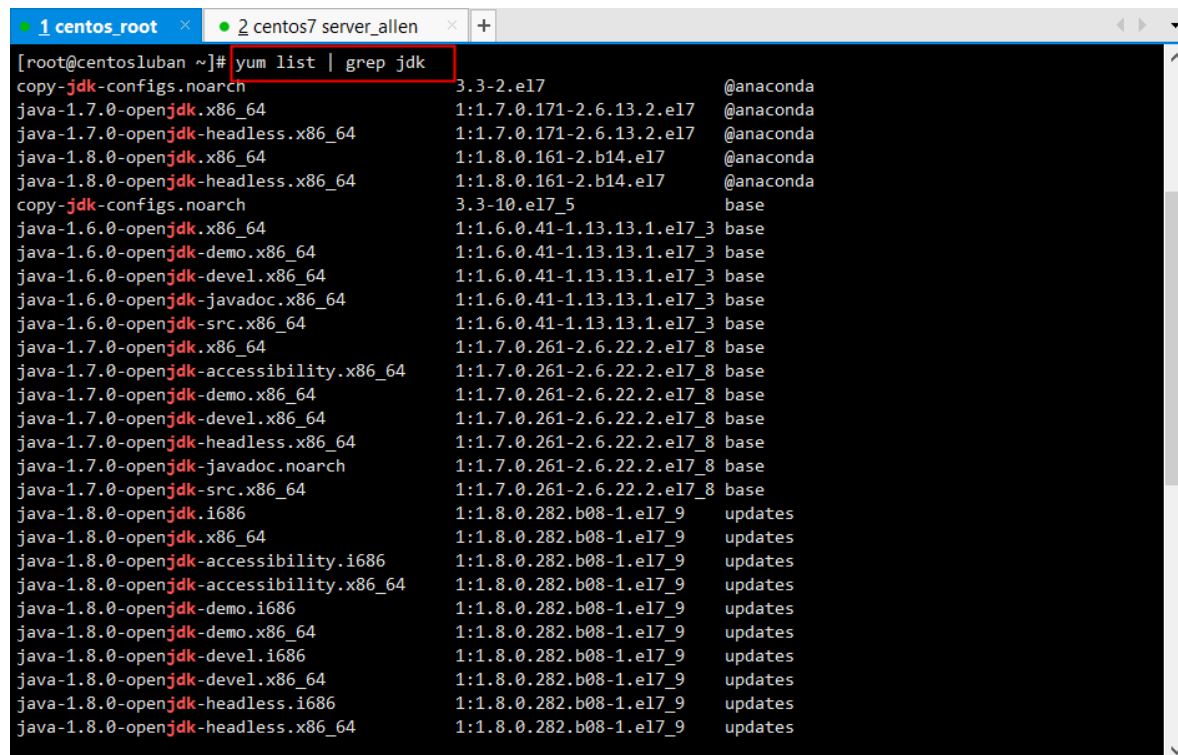
3. yum工具

3.1 介绍

- yum类似于Maven工具，可以从中央仓库下载安装各种软件
- 当某个软件有依赖其它软件时，yum也会自动下载并安装其它软件

3.2 常用命令：

1. 安装软件包：`yum install xxx -y`
- -y 表示免确认
2. 清除本地索引数据：`yum clean all`
3. 查找库中软件包：`yum list | grep xxx`



```
[root@centosluban ~]# yum list | grep jdk
copy-jdk-configs.noarch                               3.3-2.el7 @anaconda
java-1.7.0-openjdk.x86_64                             1:1.7.0.171-2.6.13.2.el7 @anaconda
java-1.7.0-openjdk-headless.x86_64                   1:1.7.0.171-2.6.13.2.el7 @anaconda
java-1.8.0-openjdk.x86_64                             1:1.8.0.161-2.b14.el7 @anaconda
java-1.8.0-openjdk-headless.x86_64                   1:1.8.0.161-2.b14.el7 @anaconda
copy-jdk-configs.noarch                               3.3-10.el7_5 base
java-1.6.0-openjdk.x86_64                             1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-demo.x86_64                       1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-devel.x86_64                     1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-javadoc.x86_64                   1:1.6.0.41-1.13.13.1.el7_3 base
java-1.6.0-openjdk-src.x86_64                       1:1.6.0.41-1.13.13.1.el7_3 base
java-1.7.0-openjdk.x86_64                             1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-accessibility.x86_64             1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-demo.x86_64                     1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-devel.x86_64                     1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-headless.x86_64                 1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-javadoc.noarch                   1:1.7.0.261-2.6.22.2.el7_8 base
java-1.7.0-openjdk-src.x86_64                       1:1.7.0.261-2.6.22.2.el7_8 base
java-1.8.0-openjdk.i686                             1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk.x86_64                             1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-accessibility.i686               1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-accessibility.x86_64             1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-demo.i686                       1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-demo.x86_64                     1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-devel.i686                       1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-devel.x86_64                     1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-headless.i686                   1:1.8.0.282.b08-1.el7_9 updates
java-1.8.0-openjdk-headless.x86_64                 1:1.8.0.282.b08-1.el7_9 updates
```

- | grep 后面带的是查询关键字
- 如，搜索jdk工具，执行指令：`yum list | grep jdk`

4. 列出本地所配置的仓库信息：`yum repolist`
5. 查询yum命令的其它参数：直接敲yum回车

4. 补充命令

1. 解压到当前目录：`tar -zxvf jdk-8u181-linux-x64.tar.gz`
2. 解压到指定目录：`tar -zxvf jdk-8u181-linux-64.tar.gz -C /usr/local/`

5. 软件安装

5.1 安装jdk

1. 使用Xftp6 上传 jdk-8u181-linux-x64.tar.gz 到服务器
 2. 解压到指定目录：
- `tar -zxvf jdk-8u181-linux-x64.tar.gz -C /root/jdk/`

```
1 centos_root x 2 centos7 server_allen x +
[root@centosluban ~]# ll
总用量 200688
-rw-r--r--. 1 root root    1743 12月 11 22:59 anaconda-ks.cfg
-rw-r--r--. 1 root root 10492067 12月 12 23:57 apache-tomcat-8.5.61.tar.gz
drwxr-xr-x. 4 root root     37 12月 12 23:59 apps
-rw-r--r--. 1 root root     32 12月 13 00:22 helloworld.sh
-rw-r--r--. 1 root root    1774 12月 12 00:06 initial-setup-ks.cfg
-rw-r--r--. 1 root root 194990602 12月 12 23:38 jdk-8u211-linux-x64.tar.gz
-rwxr-xr-x. 1 root root    202 12月 13 15:25 myname.sh
drwxr-xr-x. 2 root root     6 12月 12 00:06 公共
drwxr-xr-x. 2 root root     6 12月 12 00:06 模板
drwxr-xr-x. 2 root root     6 12月 12 00:06 视频
drwxr-xr-x. 2 root root     6 12月 12 00:06 图片
drwxr-xr-x. 2 root root     6 12月 12 00:06 文档
drwxr-xr-x. 2 root root     6 12月 12 00:06 下载
drwxr-xr-x. 2 root root     6 12月 12 00:06 音乐
drwxr-xr-x. 2 root root     6 12月 12 00:06 桌面
[root@centosluban ~]# pwd
/root
[root@centosluban ~]# ll apps
总用量 0
drwxr-xr-x. 7 10 143 245 4月 2 2019 jdk1.8
drwxr-xr-x. 9 root root 220 12月 12 23:59 tomcat8.5
[root@centosluban ~]# tar -zxvf jdk-8u211-linux-x64.tar.gz -C /root/jdk/
tar: /root/jdk: 无法 open: 没有那个文件或目录
tar: Error is not recoverable: exiting now
[root@centosluban ~]# mkdir /root/jdk
[root@centosluban ~]# tar -zxvf jdk-8u211-linux-x64.tar.gz -C /root/jdk/
```

3. 配置环境变量

- `vim /etc/profile`
- 在profile文件末尾添加：
 - `export JAVA_HOME=/usr/local/jdk1.8.0_181/`
 - `export PATH=$PATH:$JAVA_HOME/bin`

```
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export JAVA_HOME=/root/apps/jdk1.8/
export PATH=$PATH:$JAVA_HOME/bin/
```

- `PATH=$PATH:$JAVA_HOME/bin/` 中间使用**冒号**分隔

4. 使文件生效:

- `source /etc/profile`

5.2 安装Tomcat

1. 使用Xftp6上传apache-tomcat-8.5.34.tar.gz

2. 解压到指定目录

- `tar -zxvf apache-tomcat-8.5.34.tar.gz -C /root/apps/`

3. 启动

- `[root@localhost bin]# ./startup.sh`

```
1 centos_root x 2 centos7 server_allen x +
[root@centosluban apps]# ll
总用量 0
drwxr-xr-x. 7 10 143 245 4月 2 2019 jdk1.8
drwxr-xr-x. 9 root root 220 12月 12 23:59 tomcat8.5
[root@centosluban apps]# cd tomcat8.5/
[root@centosluban tomcat8.5]# ll
总用量 128
drwxr-x---. 2 root root 4096 12月 13 00:21 bin
-rw-r-----. 1 root root 19318 12月 3 22:07 BUILDING.txt
drwx-----. 3 root root 254 12月 13 00:02 conf
-rw-r-----. 1 root root 5408 12月 3 22:07 CONTRIBUTING.md
drwxr-x---. 2 root root 4096 12月 12 23:59 lib
-rw-r-----. 1 root root 57011 12月 3 22:07 LICENSE
drwxr-x---. 2 root root 197 12月 13 00:02 logs
-rw-r-----. 1 root root 1726 12月 3 22:07 NOTICE
-rw-r-----. 1 root root 3257 12月 3 22:07 README.md
-rw-r-----. 1 root root 7136 12月 3 22:07 RELEASE-NOTES
-rw-r-----. 1 root root 16507 12月 3 22:07 RUNNING.txt
drwxr-x---. 2 root root 30 12月 12 23:59 temp
drwxr-x---. 7 root root 81 12月 3 22:05 webapps
drwxr-x---. 3 root root 22 12月 13 00:02 work
[root@centosluban tomcat8.5]# cd bin/
[root@centosluban bin]# ll
总用量 868
-rw-r-----. 1 root root 36132 12月 3 22:05 bootstrap.jar
-rw-r-----. 1 root root 16655 12月 3 22:05 catalina.bat
-rw-r-----. 1 root root 25121 12月 3 22:05 catalina.sh
-rw-r-----. 1 root root 1664 12月 3 22:07 catalina-tasks.xml
-rw-r-----. 1 root root 2123 12月 3 22:05 ciphers.bat
-rw-r-----. 1 root root 1997 12月 3 22:05 ciphers.sh
-rw-r-----. 1 root root 25287 12月 3 22:05 commons-daemon.jar
-rw-r-----. 1 root root 208136 12月 3 22:05 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 2040 12月 3 22:05 configtest.bat
-rw-r-----. 1 root root 1922 12月 3 22:05 configtest.sh
-rw-r-----. 1 root root 9100 12月 3 22:05 daemon.sh
```

```
1 centos_root x 2 centos7 server_allen x +
-rw-r-----. 1 root root 3257 12月 3 22:07 README.md
-rw-r-----. 1 root root 7136 12月 3 22:07 RELEASE-NOTES
-rw-r-----. 1 root root 16507 12月 3 22:07 RUNNING.txt
drwxr-x---. 2 root root 30 12月 12 23:59 temp
drwxr-x---. 7 root root 81 12月 3 22:05 webapps
drwxr-x---. 3 root root 22 12月 13 00:02 work
[root@centosluban tomcat8.5]# cd bin/
[root@centosluban bin]# ll
总用量 868
-rw-r-----. 1 root root 36132 12月 3 22:05 bootstrap.jar
-rw-r-----. 1 root root 16655 12月 3 22:05 catalina.bat
-rw-r-----. 1 root root 25121 12月 3 22:05 catalina.sh
-rw-r-----. 1 root root 1664 12月 3 22:07 catalina-tasks.xml
-rw-r-----. 1 root root 2123 12月 3 22:05 ciphers.bat
-rw-r-----. 1 root root 1997 12月 3 22:05 ciphers.sh
-rw-r-----. 1 root root 25287 12月 3 22:05 commons-daemon.jar
-rw-r-----. 1 root root 208136 12月 3 22:05 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 2040 12月 3 22:05 configtest.bat
-rw-r-----. 1 root root 1922 12月 3 22:05 configtest.sh
-rw-r-----. 1 root root 9100 12月 3 22:05 daemon.sh
-rw-r-----. 1 root root 2091 12月 3 22:05 digest.bat
-rw-r-----. 1 root root 1965 12月 3 22:05 digest.sh
-rw-r-----. 1 root root 3460 12月 3 22:05 setclasspath.bat
-rw-r-----. 1 root root 3708 12月 3 22:05 setclasspath.sh
-rw-r-----. 1 root root 2020 12月 3 22:05 shutdown.bat
-rw-r-----. 1 root root 1902 12月 3 22:05 shutdown.sh
-rw-r-----. 1 root root 2022 12月 3 22:05 startup.bat
-rw-r-----. 1 root root 1904 12月 3 22:05 startup.sh
-rw-r-----. 1 root root 51204 12月 3 22:05 tomcat-juli.jar
-rw-r-----. 1 root root 421399 12月 3 22:05 tomcat-native.tar.gz
-rw-r-----. 1 root root 4574 12月 3 22:05 tool-wrapper.bat
-rw-r-----. 1 root root 5540 12月 3 22:05 tool-wrapper.sh
-rw-r-----. 1 root root 2026 12月 3 22:05 version.bat
-rw-r-----. 1 root root 1908 12月 3 22:05 version.sh
[root@centosluban bin]#
```



```
1 centos_root x 2 centos7 server_allen x +
-rw-r-----. 1 root root 16655 12月 3 22:05 catalina.bat
-rwxr-x---. 1 root root 25121 12月 3 22:05 catalina.sh
-rw-r-----. 1 root root 1664 12月 3 22:07 catalina-tasks.xml
-rw-r-----. 1 root root 2123 12月 3 22:05 ciphers.bat
-rwxr-x---. 1 root root 1997 12月 3 22:05 ciphers.sh
-rw-r-----. 1 root root 25287 12月 3 22:05 commons-daemon.jar
-rw-r-----. 1 root root 208136 12月 3 22:05 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 2040 12月 3 22:05 configtest.bat
-rwxr-x---. 1 root root 1922 12月 3 22:05 configtest.sh
-rwxr-x---. 1 root root 9100 12月 3 22:05 daemon.sh
-rw-r-----. 1 root root 2091 12月 3 22:05 digest.bat
-rwxr-x---. 1 root root 1965 12月 3 22:05 digest.sh
-rw-r-----. 1 root root 3460 12月 3 22:05 setclasspath.bat
-rwxr-x---. 1 root root 3708 12月 3 22:05 setclasspath.sh
-rw-r-----. 1 root root 2020 12月 3 22:05 shutdown.bat
-rwxr-x---. 1 root root 1902 12月 3 22:05 shutdown.sh
-rw-r-----. 1 root root 2022 12月 3 22:05 startup.bat
-rwxr-x---. 1 root root 1904 12月 3 22:05 startup.sh
-rw-r-----. 1 root root 51204 12月 3 22:05 tomcat-juli.jar
-rw-r-----. 1 root root 421399 12月 3 22:05 tomcat-native.tar.gz
-rw-r-----. 1 root root 4574 12月 3 22:05 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 12月 3 22:05 tool-wrapper.sh
-rw-r-----. 1 root root 2026 12月 3 22:05 version.bat
-rwxr-x---. 1 root root 1908 12月 3 22:05 version.sh
[root@centosluban bin]# startup.sh
bash: startup.sh: 未找到命令...
[root@centosluban bin]# ./startup.sh
Using CATALINA_BASE: /root/apps/tomcat8.5
Using CATALINA_HOME: /root/apps/tomcat8.5
Using CATALINA_TMPDIR: /root/apps/tomcat8.5/temp
Using JRE_HOME: /root/apps/jdk1.8/
Using CLASSPATH: /root/apps/tomcat8.5/bin/bootstrap.jar:/root/apps/tomcat8.5/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@centosluban bin]#
```

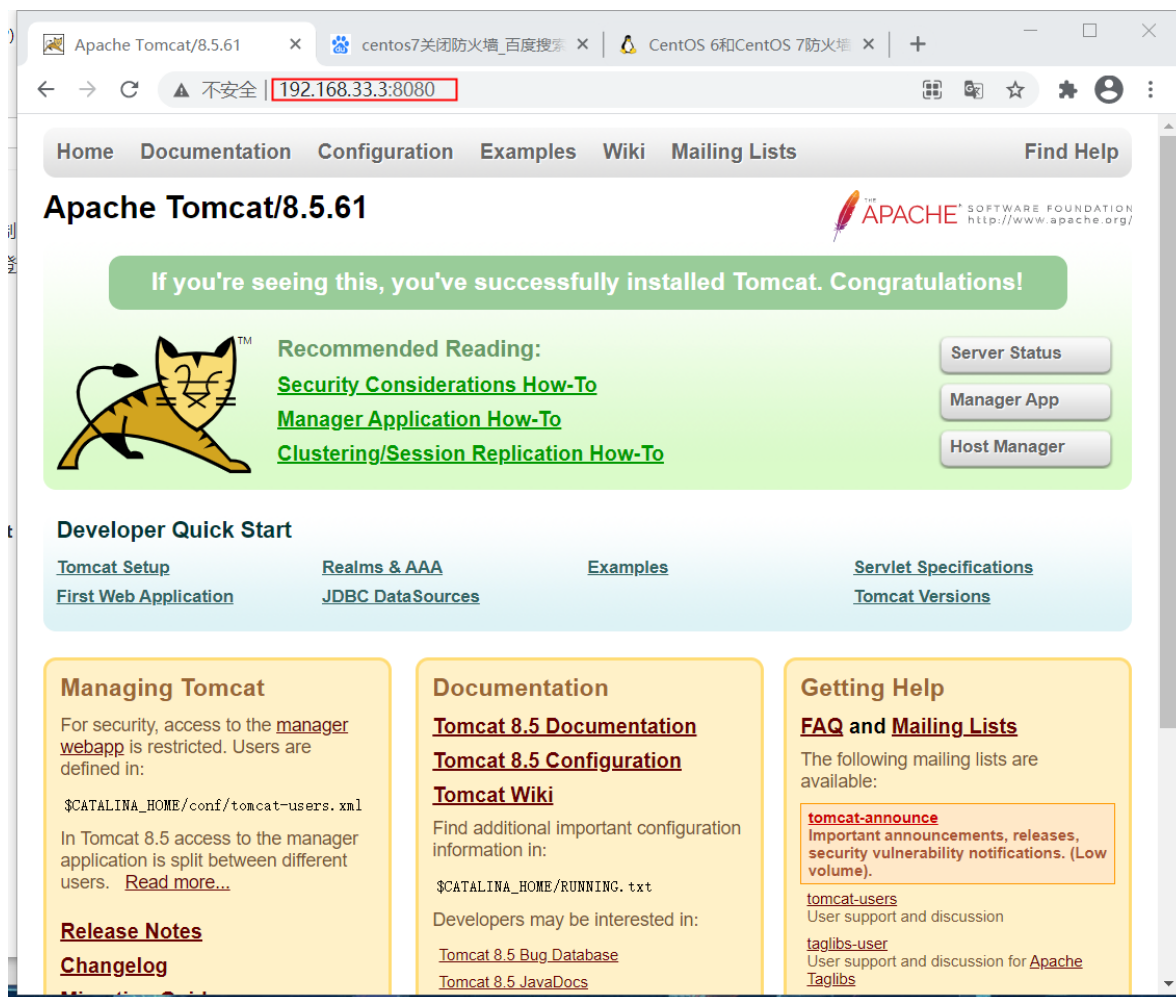
4. 查看监听端口，检查8080

- netstat -nltip

```
-rw-r-----. 1 root root 4574 12月 3 22:05 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 12月 3 22:05 tool-wrapper.sh
-rw-r-----. 1 root root 2026 12月 3 22:05 version.bat
-rwxr-x---. 1 root root 1908 12月 3 22:05 version.sh
[root@centosluban bin]# startup.sh
bash: startup.sh: 未找到命令...
[root@centosluban bin]# ./startup.sh
Using CATALINA_BASE: /root/apps/tomcat8.5
Using CATALINA_HOME: /root/apps/tomcat8.5
Using CATALINA_TMPDIR: /root/apps/tomcat8.5/temp
Using JRE_HOME: /root/apps/jdk1.8/
Using CLASSPATH: /root/apps/tomcat8.5/bin/bootstrap.jar:/root/apps/tomcat8.5/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@centosluban bin]# netstat -nltip
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8080             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8081             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8082             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8083             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8084             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8085             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8086             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8087             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8088             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8089             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8090             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8091             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8092             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8093             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8094             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8095             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8096             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8097             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8098             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8099             0.0.0.0:*               LISTEN      888/sshd
tcp        0      0 0.0.0.0:8100             0.0.0.0:*               LISTEN      888/sshd
tcp6       0      0 :::22                   :::*                     LISTEN      888/sshd
tcp6       0      0 :::80                    :::*                     LISTEN      888/sshd
tcp6       0      0 :::8080                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8081                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8082                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8083                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8084                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8085                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8086                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8087                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8088                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8089                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8090                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8091                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8092                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8093                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8094                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8095                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8096                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8097                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8098                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8099                  :::*                     LISTEN      888/sshd
tcp6       0      0 :::8100                  :::*                     LISTEN      888/sshd
[root@centosluban bin]#
```

5. 关闭防火墙

- `systemctl stop firewalld`
- windows上通过调用 <http://192.168.33.3:8080/> 地址无法访问虚拟机中安装的tomcat，因为防火墙的问题，需要关闭防火墙
- 关闭完服务器的防火墙后，就可以正常访问了



- 但是这样修改后，下次重启服务器后，防火墙还会开启

6. Shell编程

6.1 基本格式

- Shell俗称壳（用来区别于核），是指“为使用者提供操作界面”的软件（命令解析器）
- Shell是用户与内核进行交互操作的一种接口，目前最流行的Shell称为 **bash Shell**
- Shell也是一门编程语言（解释型的编程语言），即**shell脚本（就是在用linux的shell命令编程）**
- 一个系统可以存在多个shell，可以通过 `cat /etc/shells` 命令查看系统中安装的shell，不同的shell可能支持的命令语法是不同的
- `[root@localhost ~]# cat /etc/shells`

1. 代码写在普通文本文件中，通常以.sh为后缀名

```
vi hello.sh
```

```
#!/bin/bash ##表示用哪一种shell解析器来解析执行这个脚本
echo "hello word" ##注释也可以写在这里
##这是一行注释
```

2. 执行脚本：`sh hello.sh`

- 或者给脚本添加x权限，直接执行：`./hello.sh`

```

[root@centosluban ~]# ls
anaconda-ks.cfg          apps          jdk-8u211-linux-x64.tar.gz 公共 视频 文档 音乐
apache-tomcat-8.5.61.tar.gz initial-setup-ks.cfg myname.sh      模板 图片 下载 桌面
[root@centosluban ~]# vi helloworld.sh
[root@centosluban ~]# ll
总用量 200688
-rw-r--r-- 1 root root 1743 12月 11 22:59 anaconda-ks.cfg
-rw-r--r-- 1 root root 10492067 12月 12 23:57 apache-tomcat-8.5.61.tar.gz
drwxr-xr-x 4 root root 37 12月 12 23:59 apps
-rw-r--r-- 1 root root 31 3月 6 23:52 helloworld.sh
-rw-r--r-- 1 root root 1774 12月 12 00:06 initial-setup-ks.cfg
-rw-r--r-- 1 root root 194990602 12月 12 23:38 jdk-8u211-linux-x64.tar.gz
-rwxr--r-- 1 root root 202 12月 13 15:25 myname.sh
drwxr-xr-x 2 root root 6 12月 12 00:06 公共
drwxr-xr-x 2 root root 6 12月 12 00:06 模板
drwxr-xr-x 2 root root 6 12月 12 00:06 视频
drwxr-xr-x 2 root root 6 12月 12 00:06 图片
drwxr-xr-x 2 root root 6 12月 12 00:06 文档
drwxr-xr-x 2 root root 6 12月 12 00:06 下载
drwxr-xr-x 2 root root 6 12月 12 00:06 音乐
drwxr-xr-x 2 root root 6 12月 12 00:06 桌面
[root@centosluban ~]# sh helloworld.sh
hello world
[root@centosluban ~]# chmod +x helloworld.sh
[root@centosluban ~]# ll helloworld.sh
-rwxr-xr-x 1 root root 31 3月 6 23:52 helloworld.sh
[root@centosluban ~]# ./helloworld.sh
hello world
[root@centosluban ~]#

```

6.2 变量

1. 变量=值（例如A=5）

- **注意：等号两侧不能有空格**
- 变量名一般习惯为大写
- 使用变量：\$A

2. 定义变量

```
A=1
```

2. 查看变量

```
echo $A
```

3. 查看当前进程中所有变量

```
set
```

4. 撤销变量

```
unset A
```

5. 声明静态变量，不能unset

```
readonly B=2
```

注意：变量中的值没有类型，全部为字符串

```
1 centos_root x 2 centos7 server_allen x +
[root@centosluban ~]# A=abc
[root@centosluban ~]# echo $A
abc
[root@centosluban ~]# $A+"def"
bash: abc+def: 未找到命令...
[root@centosluban ~]# echo $A+"def"
abc+def
[root@centosluban ~]# echo $A"def"
abcdef
[root@centosluban ~]# echo ${A}def
abcdef
[root@centosluban ~]#
```

作业： 变量a=hello,以下选项哪个可以输出hello luban

- A、 echo \$a+"luban"
- B、 echo a+luban
- C、 echo \$aluban
- D、 echo \$a"luban" E、 echo \${a}luban
- E、 echo \${a}luban

6.3 算数运算

1. 用expr

```
expr $A + $B
```

2. 赋值

```
C=`expr $A + $B`
```

注意中间空格

2. 用 (())

```
((1+2))
```

3. 赋值

```
A=$((1+2))
```

4. 自增

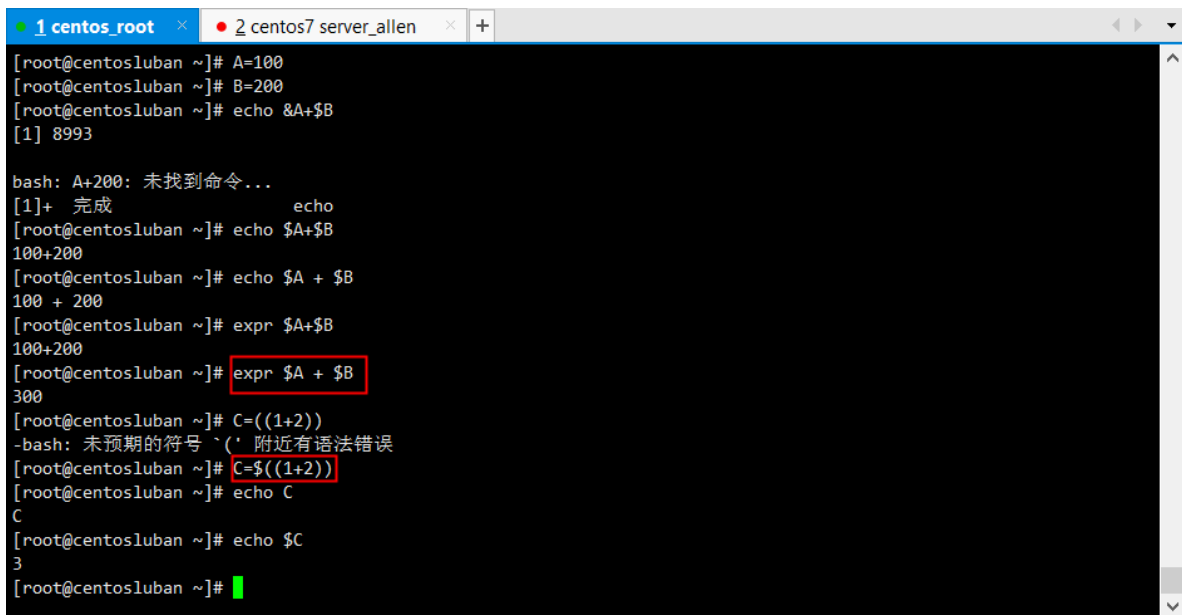
```
count=1
((count++))
echo $count
```

5. 用\$[]

```
a=$((1+2))
echo $a
```

6. 用let

```
i=1
let i++
let i=i+2
```



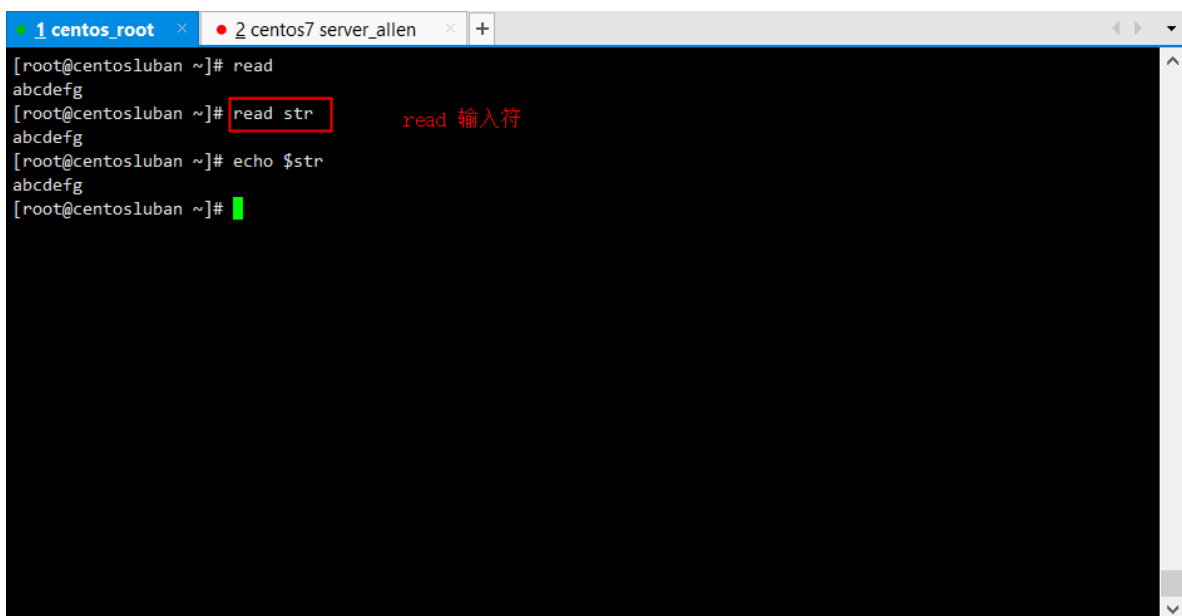
A terminal window with two tabs: '1 centos_root' and '2 centos7 server_allen'. The active tab shows the following commands and output:

```
[root@centosluban ~]# A=100
[root@centosluban ~]# B=200
[root@centosluban ~]# echo &A+$B
[1] 8993

bash: A+200: 未找到命令...
[1]+ 完成                  echo
[root@centosluban ~]# echo $A+$B
100+200
[root@centosluban ~]# echo $A + $B
100 + 200
[root@centosluban ~]# expr $A+$B
100+200
[root@centosluban ~]# expr $A + $B
300
[root@centosluban ~]# C=$((1+2))
-bash: 未预期的符号 `(' 附近有语法错误
[root@centosluban ~]# C=$((1+2))
[root@centosluban ~]# echo C
C
[root@centosluban ~]# echo $C
3
[root@centosluban ~]#
```

6.4 扫描器

- read
- read parm
- read -p "提示信息: "parm



A terminal window with two tabs: '1 centos_root' and '2 centos7 server_allen'. The active tab shows the following commands and output:

```
[root@centosluban ~]# read
abcdefg
[root@centosluban ~]# read str
abcdefg
[root@centosluban ~]# echo $str
abcdefg
[root@centosluban ~]#
```

The command 'read str' is highlighted with a red box, and the text 'read 输入符' is written in red next to it.

6.5 流程控制

1. 语法

if 条件

then

执行代码

elif 条件

then

执行代码

else

执行代码

fi

2. 示例 作业：下去把该示例敲一遍

```
#!/bin/bash
read -p "please input your name:" NAME
if [ $NAME = root ]
then
    echo "Hello, ${NAME},welcome!"
elif [ $NAME = luban ]
then
    echo "Hello,${NAME},welcome!"
else
    echo "SB,get out here!"
fi
```

注意：在shell中，上一句错误不影响执行下一句

作业：以下脚本执行结果：

```
#!/bin/bash
lss
echo "hello"
```

6.6 常用判断运算符

1. 字符串比较：

```
= 字符串是否相等
!= 字符串是否不相等
-z 字符串长度为0返回true
-n 字符串长度不为0返回true
if[ 'aa' = 'bb' ];then echo "ok";else echo "not ok";fi
if[ -n "aa" ];;then echo "ok";else echo "not ok";fi
if[ -z "" ];;then echo "ok";else echo "not ok";fi
```

2. 整数比较：

```
-lt 小于
-le 小于等于
-eq 等于
-gt 大于
-ge 大于等于
-ne 不等于
还可以用转义的数学符号 \<
```

3. 文件判断:

-d 是否为目录

```
if [ -d /bin ];then echo ok;else echo notok;fi
```

-f 是否为文件

```
if [ -f /bin/lis ];then echo ok;else echo notok;fi
```

6.7 循环控制

- 语法:

```
while 表达式
do
command
...
done
```

- 例如:

```
i=1
while((i<3))
do
echo $i
let i++
done
```

6.8 case语句

```
case $i in
start)
echo "starting"
;;
stop)
echo "stoping"
;;
*)
echo "Usage:{start|stop}"
esac
```