

Linux_day02

笔记本:	Linux		
创建时间:	2018/9/16 13:31	更新时间:	2018/10/11 22:26
作者:	155307642@qq.com		
URL:	https://blog.csdn.net/u011277123/article/details/54846422		

课程大纲:

- SSH
- yum工具
- Shell编程

一、SSH

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- daemon。
在linux上使用ssh

```
ssh root@192.168.33.88
```

2、SSH免密码登录

生成密钥:

```
ssh-keygen
```

把自己的公钥拷给对方:

```
ssh-copy-id 192.168.33.3
```

基于ssh的文件拷贝:

```
scp abc.txt 192.168.33.3:/root
```

基于ssh的目录拷贝: 需要加-r递归拷贝

```
scp aaa -r 192.168.33.3:/root
```

远程执行命令:

```
ssh 192.168.33.3 "echo hello > /root/hello.txt"
```

作业: 配置两台linux服务器之间免密码登录

二、网络配置

1、查看主机名

```
hostname
```

2、修改主机名(重启后无效)

```
hostname luban
```

3、修改主机名(重启后有效)

```
vi /etc/sysconfig/network
```

三、yum工具

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。

常用命令：

安装软件包：yum install xxx -y -y表示免确认

清除本地索引数据：yum clear all

查找库中软件包：yum list | grep xxx

列出本地所配置的仓库信息：yum repolist

其它参数：直接敲yum回车

如：搜索jdk工具

```
yum list | grep jdk
```

四、补充命令

解压到当前目录：

```
tar -zxvf jdk-8u181-linux-x64.tar.gz
```

解压到指定目录：

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /usr/local/
```

五、软件安装

1、安装jdk

1) 使用Xftp6上传jdk-8u181-linux-x64.tar.gz

2) 解压到指定目录

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /usr/local/
```

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
```

4) 文件生效

```
source /etc/profile
```

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 /usr/local/
```

3) 启动

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

4) 查看监听端口，检查8080

```
netstat -nltp
```

5) 关闭防火墙

作业：安装jdk和tomcat

六、Shell编程

1、基本格式

Shell俗称壳（用来区别于核），是指“为用户提供操作界面”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
```

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
```

执行脚本：

```
sh hello.sh
```

或者给脚本添加x权限，直接执行

```
./hello.sh
```

2、变量

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

定义变量

```
A=1
```

查看变量

```
echo $A
```

查看当前进程中所有变量

```
set
```

撤销变量

```
unset A
```

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。

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

3、算数运算

1) 用expr

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

赋值

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

注意中间空格

2) 用()

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

赋值

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

自增

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

3) 用\$[]

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

4) 用let

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

4、扫描器

read

read parm

read -p "提升信息: "parm

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、常用判断运算符

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
```

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 \<

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
```

7、循环控制

1、语法

while 表达式

do

command

...

done

例如：

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

done

8、case语句

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