# 云计算应用管理

# 一. 什么是shell脚本

提前设计可执行语句，用来完成任务的文件

-- 解释型程序

-- 顺序，批量执行

## 1. shell脚本的一般组成

1) #! 环境声明，一下代码由哪个程序进行解释

2) # 注释文本

3) 可执行代码

e.g. :

1)输出当前红帽系统的版本信息

2)输出当前使用的内核版本

3)输出当前系统的主机名

[root@server0 ~]# vim /root/hello.sh

#!/bin/bash

echo hello world

cat /etc/redhat-release

uname -r

hostname

ifconfig | head -2

[root@server0 ~]# /root/hello.sh

## 2. 简单脚本技巧

1) /dev/null: 黑洞设备

2) 重定向输出

> :只收集前面命令的正确输出信息

2> :只收集前面命令的错误输出信息

&> :收集前面命令的正确与错误信息

3) 单引号' '

取消所有特殊字符的意义,原样输出

4) 反撇号 ` ` 或 $( )

将命令的输出,直接参与另一条命令的运行,作为另一条命令的参数

5) 造数工具: {起始..结束}

{1..20}:制造1到20之间连续的数字

# 二. 变量

## 1. 基本说明

定义：以不变的名称存放的可能会变化的值

语法：变量名＝变量值

作用：1) 方便以故定名称重复使用某个值

2) 提高对任务需求，运行环境变化的使用能力

说明：-- 等号两边不要有空格

　-- 变量名只能由字母／数字／下滑先组成，区分大小写

　 　-- 变量名不能以数字开头，不要使用关键字和特殊字符

　-- 若指定的变量名已存在，相当与为此变量重新赋值

查看／应用变量：

引用变量值：$变量名

查看变量值：echo $变量名

echo ${变量名}

## 2. 常用系统变量

环境变量：

变量名为大写字母，由系统定义并且赋值完成

e.g. : USER=当前登录的用户名

位置变量：

由系统定义并赋值完成；在执行脚本时提供的命令行参数

e.g. : 表示为$n, n为序号

　$1、$2、.. .. ${10}、${11}、.. ..

预定义变量：

由系统定义并赋值完成

e.g. : $# 已加载的位置变量的个数

　 $? 程序退出后的状态，0表示正常，其他值异常

# 三. 条件测试

[ 测试表达式 ] 每一部分都要有空格

## 1. 检查文件状态

-e: 文档存在为真

-d: 文档存在,且必须为目录才为真

-f: 文档存在,且必须为文件才为真

e.g. : [ -e /etc ]; echo $? # 0

## 2. 比较整数大小

-gt: 大于

-ge: 大于等于

-eq: 等于

-ne: 不等于

-lt: 小于

-le: 小于等于

e.g. : [ 1 -eq 2 ]; echo $? # 1

## 3. 字符串比对

==: 一致为真

!=: 不一致为真

e.g. : [ hello != world ]; echo $? # 0

# 四. read / if / for语句

## 1. read语句

提高用户体验度，产生交互

作用：

1) 直接产生交互,等待用户输入

2) 记录用户在键盘上的输入

3) 将记录的内容,赋值给一个变量储存

选项：

-p : ＂屏幕提示信息＂

语法：

read -p "屏幕提示信息" 变量名

e.g. :

read -p '请您输入要创建的用户名:' a

## 2. if语句

if双分支处理：

if [条件测试]; then

命令序列1

else

命令序列2

fi

if多分支处理:

if [条件测试1];　then

命令序列1

elif [条件测试2];then

命令序列2

elif [条件测试3];then

命令序列3

.........

else

命令序列n

fi

## 3. for循环语句

作用：遍历/列表式循环；根据变量的不同取值，重复执行处理

语法：

for 变量名 in 值列表

do

命令序列

done

# 四. 系统安全保护

SELinux安全机制

SELinux概述：

-- 全称Security-Enhanced Linux

-- 美国NSA国家安全全局主导开发，一套增强Linux系统安全的强制访问操作体系

-- 集成到Linux内核(2.6及以上)中运行

-- RHEL7基于SELinux体系针对用户、进程、目录和文件提供了预设的保护策略,以及管理工具

SELinux运行模式：

enforcing: 强制

permissive: 宽松

disabled: 彻底禁用

所有状态变成disabled(彻底禁用),都要经历重启系统

SELinux切换：

1. 临时切换: setenforce 1|0

2. 固定配置: /etc/selinux/config 文件

e.g. : SELinux状态永久配置文件

vim /etc/selinux/config

SELINUX=permissive

常用命令：

getenforec : 查看SELinux当前系统状态

setenforec 0 : 修改SELinux状态，为宽松模式

# 五. 自定义命令

## 1. Linux命令字的来源

命令字：可执行程序的路径

别名：在用户环境中，为一个较为复杂的，需要经常使用的命令行所起的短名称；可用来替换普通命令

## 2. alias别名设置

查看别名：alias [别名名称]

定义别名：alias 别名名称="实际执行的命令行"

取消别名：unalias [别名名称]

e.g. : 为hostname定义别名为h\_n

alias h\_n="hostname"

## 3. 用户初始化文件

1) 用户个性化配置

配置文件：~/.bashrc

2) 全局环境配置

配置文件：/etc/bashrc

e.g. : 给studetn用户配置别名

vim /home/student/.bashrc

alias hello="echo hello world"

# 六. 搭建服务

## 常见的互联网协议

http: 超文本传输协议 默认端口:80

FTP: 文本传输协议 默认端口:21

https: 安全的超文本传输协议 默认端口:443

DNS: 域名解析协议 默认端口:53

telnet: 远程管理协议 默认端口:23

tftp: 简单的文本传输协议 默认端口:69

smtp: 发邮件协议 默认端口:25

pop3: 收邮件协议 默认端口:110

snmp: 简单的网络管理协议 默认端口:161

## 搭建http服务

http协议:超文本传输协议,专用于传输页面内容

1. 安装httpd软件包

yum -y install httpd

2.重启httpd服务

systemctl restart httpd

systemctl enable httpd

3.本机测试

firefox 172.25.0.11

4.书写自己的页面

默认存放文件路径: /var/www/html/

默认网页文件名称: index.html

vim /var/www/html/index.html

<marquee><font color=red><h1>hello world</h1>

滚动 字体颜色 标题字体

## 搭建FTP服务

FTP:文件传输协议

1. 安装vsftpd软件

yum -y install vsftpd

2. 重启vsftpd服务

systemctl restart vsftpd

systemctl enable vsftpd

3. 默认FTP服务共享位置

/var/ftp/

4. 本机验证

firefox ftp://172.25.0.11

# 七. 防火墙策略管理

作用：隔离，进行入站过滤

常见防火墙类型：硬件防火墙和软件防火墙

## 1. firewalld服务基础

Linux的防火墙体系：

系统服务：firewalld

管理工具：

firewall-cmd : 命令行模式

firewall-config ：图形化界面模式

e.g. : firewall-config & # 图像化界面

预设安全区域：

根据所在的网络场所区分,预设保护规则集

– public: 仅允许访问本机的sshd dhcp ping

– trusted: 允许任何访问

– block: 拒绝任何来访请求(明确拒绝回应)

– drop: 丢弃任何来访的数据包(直接丢弃不给回应,节省服务器资源)

防火墙的判定规则：

-- 查看请求数据包,中客户端来源IP地址,查看自己所有区域规则,如果某一个区域有该源IP地址的规则,则进入该区域

-- 进入默认区域public

## 2. 防火墙列表查看

firewall-cmd --list-all-zones

firewall-cmd --list-all [--zone=区域名] # 查看区域详细规则

firewall-cmd --get-services

firewall-cmd --get-default-zone # 查看默认区域

firewall-cmd --get-zones

## 3. 防火墙默认区修改

e.g. : 修改默认区域为block

firewall-cmd --set-default-zone=block

效果：不可以通信,有回应

e.g. : 修改默认区域为drop

firewall-cmd --set-default-zone=drop

效果：不可以通信,没有回应

## 4. 在默认区域添加协议

e.g. : 添加允许的协议http

firewall-cmd --zone=public --add-service=http

e.g. : 添加允许的协议ftp

firewall-cmd --zone=public --add-service=ftp

e.g. : 查看区域详细规则

firewall-cmd --zone=public --list-all

## 5. 防火墙永久策略

e.g. :

firewall-cmd --permanent --zone=public --add-service=http

firewall-cmd --permanent --zone=public --add-service=ftp

firewall-cmd --zone=public --list-all

e.g. : 重新加载防火墙所有配置

firewall-cmd --reload

e.g. : 单独拒绝某个ID的访问

firewall-cmd --zone=block --add-cource=172.25.10

firewall-cmd --zone=block --list-all

## 6. 实现本机的端口映射

-- 本地应用的端口重定向(端口1 --> 端口2)

-- 从客户机访问　端口1 的请求，自动映射到本机　端口2

e.g. : 访问以下两个地址可以看到相同的页面

http://172.25.0.11:1234 ---> http://172.25.0.11:80

第一步：

firewall-cmd --permanent --zone=public

--add-forward-port=port=1234:proto=tcp:toport=80

第二步：

firewall-cmd --reload

第三步：

firewall-cmd --zone=public --list-all

第四步：

本机测试：

firefox 172.25.0.11:80 # 可以访问

firefox 172.25.0.11:1234 # 无法访问

虚拟机：

firefox 172.25.0.11:80 # 可以访问

firefox 172.25.0.11:1234 # 可以访问

# 八．iSCSI网络磁盘

## 1. 概念：

Internet SCSI，网际SCSI接口（默认端口：3260）

作用：一种基于C/S架构的虚拟磁盘技术；服务器提供磁盘空间，客户机连接并当成本地磁盘使用。

## 2. iSCSI磁盘构成

backstore，后端存储

对应到服务端提供实际存储空间的设备，需要起一个管理作用

target，磁盘组

是客户端的访问目标，作为一个框架，由多个lun组成

ISCSI Qualified Name(iqn)名称规范

用来识别target磁盘组，也用来识别客户身份

格式：iqn.xxxx.倒叙域名：自定义标识

e.g. :

iqn.2019-05.com.example:server0

lun,逻辑单元

每一个lun需要关联到某一个后端存储设备，在客户端会视为一块虚拟硬盘

## 3. 发布iSCSI磁盘

1）安装targetcli软件包

yum -y install targetcli

2）配置

a. 建立backstore，后端存储

targetcli

/backstores/block create name=xhj dev=/dev/vdb1

b. 建立target磁盘组

/iscsi create iqn.2019-05.com.example:server0

c. 进行lun关联

/iscsi/iqn.2019-05.com.example:server0/tpg1/luns \

create /backstores/block/xhj

d. 设置访问控制ACL，设置客户端声明的名字

/iscsi/iqn.2019-05.com.example:server0/tpg1/acls \

create iqn.2019-05.com.example:hello

e. 指定本机开放共享服务的IP地址及端口

/iscsi/iqn.2019-05.com.example:server0/tpg1/portals \

create 172.25.0.11

f. 重启服务

systemctl restart target

systemctl enable target

## 4. 访问iSCSI磁盘

a. 安装客户端软件

yum -y install iscsi-initiator-utils

b. 配置客户端声明的名字

vim /etc/iscsi/initiatorname.iscsi

InitiatorName= iqn.2019-05.com.example:hello

c. 客户端刷新声明名字的服务

systemctl restart iscsid

d. 发现服务端共享存储

Linux查看命令帮助的方法：man iscsiadm

全文查找/example，按n向下跳转匹配，看到EXAMPLES

iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 –discover

e. 使用服务端共享存储

systemctl restart iscsi

# 九. 配置NFS共享

## 1. 简介：

概念：Network File System，网络文件系统

作用：为客户机提供共享使用的文件夹

协议：NFS（TCP/UDP 2049）、RPC（TCP/UDP 111）

所需软件包：nfs-utils

系统服务：nfs-server

## 2. 普通NFS配置

1）环境设置

firewall-cmd --set-default-zone=trusted

2）虚拟机server构建NFS共享

a．安装nfs-utils软件包

yum -y install nfs-ntils

b．修改配置文件

vim /etc/exports

文件夹路径 \*（ro）

作用：将共享文件给所有客户端，可以只读方式访问

配置文件格式：文件夹路径 客户机地址（权限）...

c．重启服务

systemctl restart nfs-server

systemctl enable nfs-server

2）虚拟机desktop访问NFS共享

a．显示服务端共享路径

showmount -e 客户端IP地址

b.1 挂载到本机

mount 服务端IP：/共享文件 挂载点

b.2 开机自动挂载

vim /ect/fstab

服务端IP：/共享文件 挂载点 nfs defaults,\_netdev 0 0

说明：\_netdev声明网络设备，Linux先部署网络所有参数，再进行挂载该设备

c．检测

mount -a

df -h

# 十. HTTP服务基础

## 1. Web通信基本概念

-- 基于B/S（Browser/Server）架构的网页服务

-- 服务端提供网页

-- 浏览器下载并显示网页

-- Hyper Text Transfer Protocol（HTTP），超文本传输协议

-- Hyper Text Markup Language（HTML），超文本标记语言

## 2. 提供Web的服务软件

http（Apache），NGINX，Tomcat

软件包：httpd

系统服务：httpd

提供的默认配置：

Listen：监听地址：端口80

ServerName：本地点注册的DNS名称（空缺）

DocumentRoot：网页根目录（/var/www/html）

## 3. 构建基本的Web服务

虚拟机server

1）安装httpd

2）重启httpd服务

3）设置httpd开机自启

4）书写一个网站内容

5）修改访问路径配置文件

vim /etc/httpd/conf/httpd.conf

DocumentRoot=“/var/www/myweb”

6）重启httpd

虚拟机desktop

依次访问server

3. 说明

-- DocumentRoot：设置存放网页文件的路径（网页文件的根目录）

-- 实际路径与网络路径：

客户端：firefox IP:80 --> 服务端httpd -->

主配置 DocumentRoot /var/www/myweb --> /var/www/myweb

-- Web访问控制，默认继承父目录访问控制

-- 访问控制的配置字段：

<Directory “访问路径”>

Require all granted # 允许所有人访问

Require all IP地址 # 仅允许该IP访问

Require all denied # 拒绝所有人访问

</Directroy>

## 4. 虚拟Web主机

由一台服务器提供多个不同的Web页面

1）区分方式

-- 基于域名的虚拟机

-- 基于端口的虚拟机

-- 基于IP地址的虚拟机

2）配置文件

/etc/httpd/conf/httpd.conf # 主配置文件

/etc/httpd/conf.d/\*.conf # 调用配置文件

3）搭建虚拟Web

vim /etc/httpd/conf.d/new.conf # 调用配置文件

<VirtualHost \*:80> # 启动虚拟Web主机功能

ServerName “server0.example.com” # 设置访问的域名

DocumentRoot “/var/www/qq” # 设置存放网页文件目录

</VrirualHost>

systemctl restart httpd # 重启httpd服务

4）说明：

一旦使用了虚拟Web主机功能，所有的网站都必须使用虚拟Web主机实现；

如果直接使用IP地址进行访问，默认会将第一个虚拟Web主机页面内容呈现。

## 5. 文件夹权限

-- 针对DocumentRoot网页目录的权限控制

-- 使用<Directory>配置区段

-- 每个文件夹自动继承其父目录的ACL访问权限

-- 除非针对子目录有明确设置

# 十一. 部署动态网站

## 1. 静态动态网站区别

静态网站的运行

服务端的原始网页 = 浏览器访问到的网页

-- 由Web服务软件处理所有请求

-- 文本（txt/html）、图片（jgp/png）等静态资源

动态网站的运行

服务端的原始网页 ≠ 浏览器访问到的网页

-- 由Web服务软件接收请求，动态程序转后端模块处理

-- PHP网页、python网页、JSP网页...

## 2. semanage命令

功能：查询与修改SELinux默认目录的安全上下文。

格式：semanage [参数] [选项]

选项：

-l：查询

fcontext：主要在安全上下文方面

-a：增加，可以增加一些目录的默认安全上下文类型

-m：修改

-d：删除

e.g. : 添加端口

semanage port -a –t http\_port\_t -p tcp 8909

## 3. chcon命令

功能：修改文件的安全向下文

格式：chcon [选项] [参数] [文件]

选项：

--reference：参照对象

-f：强制执行

-R: 递归地修改对象的安全上下文

-r：修改安全上下文角色的配置

-t：修改安全上下文类型的配置

-u：修改安全上下文用户的配置

-v：显示冗长的信息

-l，--range：修改安全上下文中的安全级别

e.g. : 将/webroot目录的安全上下文改为；和/var/www目录一样

chcon -R --referenece=/var/www /webroot/

ls -Zd 文件：查看文件安全上下文

## 4. 部署Python页面步骤：

第一步：方便用户的访问，页面跳转（页面别名；地址重写）

vim /etc/httpd/conf.d/virtual.conf

Listen 8909

<VirtualHost \*:8909>

ServerName "域名"

DocumentRoot "访问父目录"

WsgiScriptAlias / Python页面路径

</VirtualHost>

e.g.

-- 端口的优先级最高

-- 默认将第一个虚拟Web主机内容呈现

第二步：安装mod\_wsgi软件包，专用于翻译Python页面

第三步：SElinux非默认端口的开放

semanage port -at http\_port\_t -t tcp 8909

第四步：重启服务；测试

# 十二. 安全Web服务，实现https

## 1. PKI公钥基础设施

Public Key Infrastructure，公钥基础设施

-- 公钥：主要用来加密数据

-- 私钥：主要用来解密数据（与相应的公钥匹配）

-- 数字证书：证明拥有者的合法性、权威性（单位名称、有效期、颁发机构...）

-- Certificate Authority，数字证书授权中心：负责证书的申请、审核、颁发、鉴定...

## 2. 搭建基本的https

第一步：下载相应的证书

a. 部署网站证书

/etc/pki/tls/certs/server0.crt

b. 部署根证书

/etc/pki/tls/certs/example-ca.crt

c. 部署私钥（解密数据）

/etc/pki/tls/private/server0.key

第二步：安装支持安全软件mod\_ssl

第三步：修改配置文件

vim /etc/httpd/conf.d/ssl.conf

a. 修改默认路径

DocumentRoot “/var/www/html”

b. 修改默认服务名

ServerName "server0.example.com"

c. 指定网站证书的路径及名称

SSlCertificateFile /etc/pki/tls/certs/server0.crt

d. 指定解密的私钥

SSLCertificateKeyFile /etc/pki/tls/private/server0.key

e. 指定根证书

SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt

第四步：重启http服务

第五步：验证

firefox <https://server0.example.com>

# 十三. 网络参数配置

## 1. 设置永久主机名

e.g. : 修改/etc/hostname配置文件

echo Julian > /etc/hostname

## 2. 配置ipv4 IP地址，子网掩码，网关地址

32个二进制位，点分隔的十进制数

1）查看识别的网卡名称

nmcli connection show

2）修改地址、子网掩码、网关地址

nmcli connection 修改 “识别的网卡名”

ipv4.方法 手动配置IP地址 ipv4.地址 “IP地址/子网掩码 网关地址”

每次开机自动启动

e.g. :

nmcli connection modify "System eth0"

ipv4.method manual ipv4.addresses "172.25.0.11/24 172.25.0.254"

connection.autoconnect yes

3）激活配置

nmcli conection up "System eth0"

## 3. 配置ipv6地址

128个二进制位，冒号分隔的十六进制数

每段内连续的前置0可省略、连续的多个：可简化为：：

1）查看识别的网卡名称

nmcli connection show

2）修改地址、子网掩码

nmcli connection 修改 “识别的网卡名”

ipv6.方法 手动配置IP地址 ipv6.地址 “IP地址/子网掩码”

每次开机自动启动

e.g. :

nmcli connection modify "System eth0"

ipv4.method manual ipv4.addresses "2003:ac19::306/64"

connection.autoconnect yes

3）激活配置

nmcli conection up "System eth0"

4）测试ping

ping6 2003:ac19::306

# 十四. 链路聚合（聚合链路 网卡绑定）

作用：提高网卡设备的可靠性，提供冗余

## 1. 建立虚拟网卡team0

nmcli connection 添加 类型 team

配置文件名 team0

网卡名 team0

每次开机自动启用 链路聚合管理方式 热备份方式

e.g. :

nmcli connection add type team

con-name team0

ifname team

autoconnect yes config '{"runner":{"name":"activebackup"}'

说明：

-- 参考man teamd.conf全文所搜/example

-- 生成配置文件位置：/etc/sysconfig/network-scripts/

## 2. 添加成员

nmcli connection 添加 类型 team-成员

配置文件名 team0-1

网卡名 eth1

主设备 team0

e.g. :

nmcli connection add type team-slave

con-name team0-1

ifname eth1

master team0

## 3. 配置IP地址

nmcli connection modfiy team0

ipv4.method manual

ipv4.addresses 192.168.177.1/24

connection.autoconnect yes

## 4. 激活

nmcli connection up team0

e.g. : 专用于查看team0信息

teamdctl team0 state

e.g. : 禁用eth2网卡

ifconfig eth2 down

e.g. : 如果无法激活，删除所有配置重新来过

nmcli connection delete team0

nmcli connection delete team0-1

nmcli connection delete team0-2

# 十五. parted分区工具，进行gpt分区模式

[root@A ~]# parted /dev/vdb

(parted) mktable gpt #指定分区表类型为gpt

(parted) print #输出分区表信息

(parted) mkpart #划分新的分区

分区名称？ []? haha #分区名称随意写

文件系统类型？ [ext2]? ext4 #文件系统随意些,不起实际作用

起始点？ 0 #分区起始点

结束点？ 2G #分区结束点

警告: The resulting partition is not properly aligned

for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择ignore忽略

(parted) unit GB #使用GB最为单位

(parted) mkpart

分区名称？ []? haha

文件系统类型？ [ext2]? ext4

起始点？ 2G

结束点？ 5G

(parted) print

(parted) quit

# 十六. 基础邮件服务

## 1. 基本功能

-- 为用户提供电子邮箱存储空间(用户名@邮件域名)

-- 处理用户发出的邮件 ----> 传递给收件服务器

-- 处理用户收到的邮件 ----> 投递到邮箱

## 2. 部署postfix邮件服务器

1）安装postfix软件包

2）修改配置文件：/etc/postfix/main.cf

myorigin = server0.example.com

inet\_interfaces = all

mydestination = server0.example.com

3）重启服务

4）测试

发邮件：

mail -s “邮件标题” -r 发件人 收件人

e.g. :

mail -s "First Test" -r yg xln

收邮件：

mail -u 用户名

e.g. :

mail -u xln