

# 服务器

---

关闭虚拟机: `sudo shutdown -h now` 需要有sudo权限

自动补全文件名: `cd` 部分文件名+`tab`键

服务器鼠标支持: 安装gpm `sudo apt install -y gpm`

清华镜像源: <https://mirrors.tuna.tsinghua.edu.cn/ubuntu/>

本机IP: 10.203.195.146

## Lab1

---

安装虚拟机

## 服务器与桌面通过内网连接

---

服务器ip地址 10.0.2.5

主机ip地址: 10.0.2.15

## LAMP

---

安装

```
1 | sudo apt install lamp-server^
```

在命令的结尾添加^符号表示该安装包是一个用于同时安装多个包的元包。但是，如果要删除这个元包，可能会删除许多不应该删除的依赖项。可以使用元包^安装LAMP堆栈，但不能将其作为元包删除。

测试

获取ip地址

```
1 | ifconfig
```

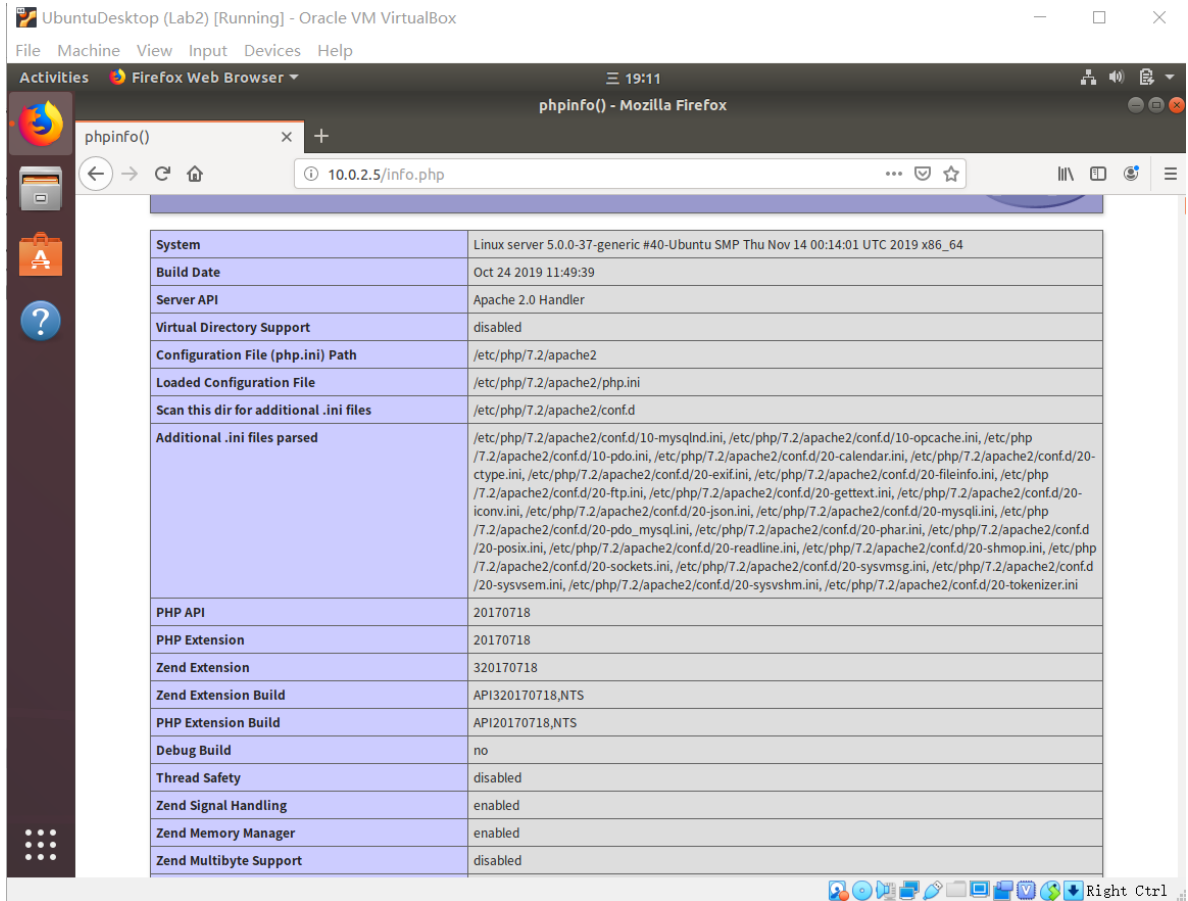
```
1 | sudo pico或者nano /var/www/html/info.php
2 | <?php
3 | phpinfo();
4 | ?>
```

按 `ctrl+x` 进入操作界面保存文件

在桌面查看:

在浏览器中打开:

```
1 | serverIP/info.php
```



## Lab2

### snapshots屏幕快照

### DokuWiki

- 下载

```
1 | cd /tmp
2 | wget https://download.dokuwiki.org/src.dokuwiki/dokuwiki-stable.tgz
```

- 解压

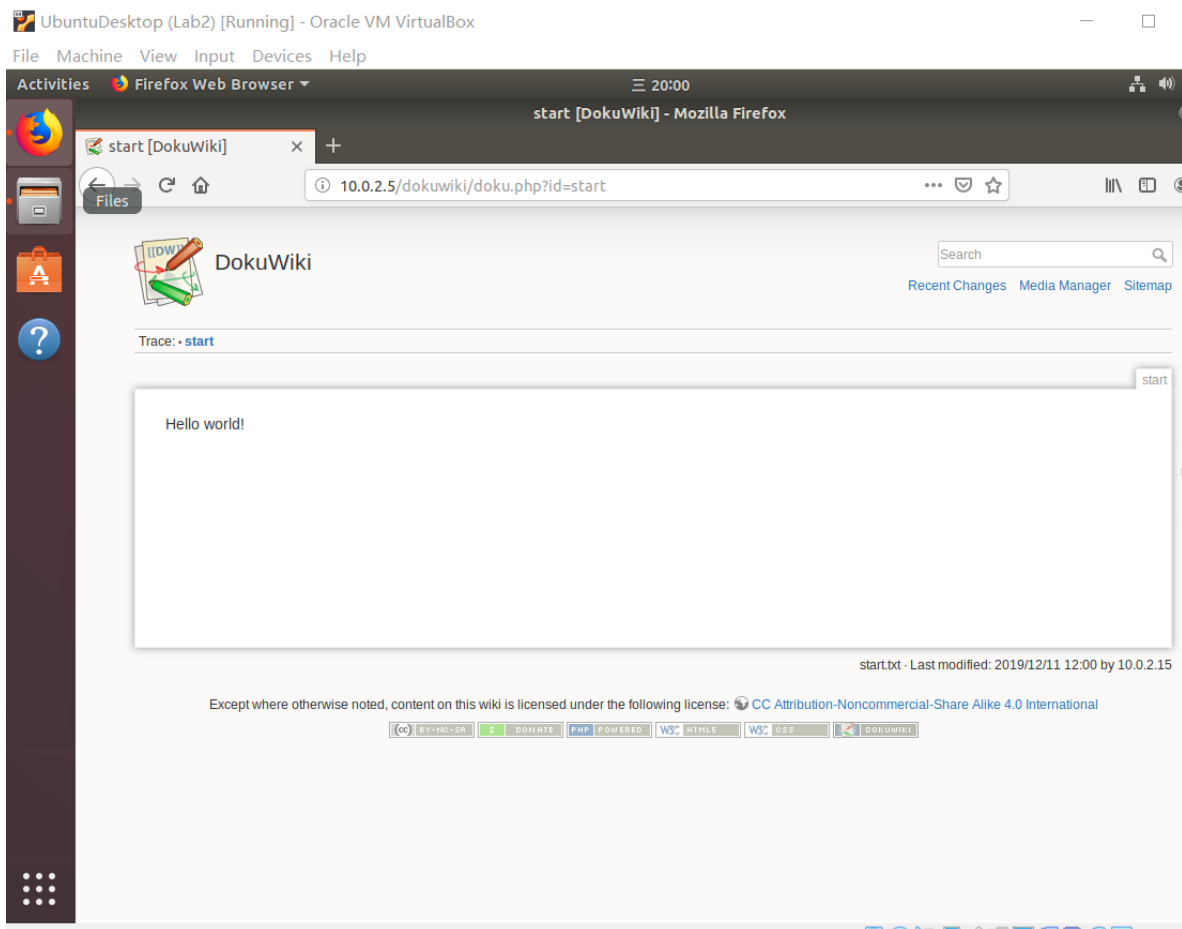
```
1 | tar -zxvf dokuwiki-stable.tgz
```

- 将所有文件移动到webspaace目录

```
1 | sudo mv dokuwiki-*/ /var/www/html/dokuwiki
2 | sudo chown -Rh www-data:www-data /var/www/html/dokuwiki
```

打开并添加笔记

```
1 | serverip/dokuwiki/
```



## man pages

命令的说明手册

查看某个命令

1 | man chown

section 3 包含的说明——库调用

```
1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions eg /etc/passwd
6 Games
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]
```

chown 设置文件所有者，如chown root (root组)

chmod 700 设置文件访问权限，如 chmod 700 (读写执行)

chmod +x (+表示增加，x表示执行，-同理)

## 用户和组

```
abc123@server:/tmp$ sudo addgroup shark
Adding group `shark' (GID 1001) ...
Done.
```

查看添加的组

`tail` 表示查看文件末尾

```
abc123@server:/tmp$ tail /etc/group
mlocate:x:111:
rdma:x:112:
ssh:x:113:
landscape:x:114:
systemd-coredump:x:999:
abc123:x:1000:
lxd:x:998:
mysql:x:115:
ssl-cert:x:116:
shark:x:1001:
```

将一个用户添加从组（主组只能有一个，从组可以有多个）

`-a` 追加，`-G` 是从组（`-g` 是主组）

使用脚本添加用户和组

为.sh脚本文件设置权限

添加用户后的shark组

```
abc123@server:/$ sudo usermod -a -G shark test3
abc123@server:/$ sudo usermod -a -G shark test4
abc123@server:/$ grep shark /etc/group
shark:x:1001:test3,test4
```

## Cron定时器

定时器日期顺序：分时日月周

修改定时器为9am Monday:

```
GNU nano 3.2 /tmp/crontab.eMIeqC/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
* 9 * * 1 grep shark /etc/group 2>&1 > /tmp/sharks
```

查看主机定时任务：

```
abc123@server:/$ crontab -u abc123 -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* 9 * * 1 grep shark /etc/group 2>&1 > /tmp/sharks
```

## Lab3

---

抓包

### 设置时区

---

查看当前时区文件中的内容

`more` 分页查看文本（`cat` 升级版）

```
abc123@server:/$ more /etc/timezone
Etc/UTC
```

修改时区

再次查看时区文件中的内容

```
abc123@server:/$ more /etc/timezone
Asia/Shanghai
abc123@server:/$ date
Wed 11 Dec 2019 08:48:41 PM CST
```

### 抓包

---

在桌面上安装ssh

### tcpdump

监听桌面ping服务器的ICMP或者ARP包

```

abc123@server:/$ sudo tcpdump -c 8 "icmp or arp"
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
21:00:06.075694 IP 10.0.2.15 > server: ICMP echo request, id 4106, seq 1, length 64
21:00:06.075744 ARP, Request who-has 10.0.2.15 tell server, length 28
21:00:06.076052 ARP, Reply 10.0.2.15 is-at 08:00:27:f2:fd:58 (oui Unknown), length 46
21:00:06.076059 IP server > 10.0.2.15: ICMP echo reply, id 4106, seq 1, length 64
21:00:11.241971 ARP, Request who-has tell server, length 28
21:00:11.242200 ARP, Reply _gateway is-at 52:54:00:12:35:00 (oui Unknown), length 46
21:00:11.299821 ARP, Request who-has server tell 10.0.2.15, length 46
21:00:11.299845 ARP, Reply server is-at 08:00:27:f5:4e:33 (oui Unknown), length 28
8 packets captured
8 packets received by filter
0 packets dropped by kernel

```

服务器MAC地址见上面红线

## wireshark

### ICMP

#### request包:

服务器ping桌面

Wireshark - Packet 1 - packages.pcapng

Frame 1: 98 bytes on wire (784 bits), 90 bytes captured (720 bits) on interface 0

Ethernet II, Src: PcsCompu\_f5:4e:33 (08:00:27:f5:4e:33), Dst: PcsCompu\_f2:fd:58 (08:00:27:f2:fd:58)

Destination: PcsCompu\_f2:fd:58 (08:00:27:f2:fd:58)

Source: PcsCompu\_f5:4e:33 (08:00:27:f5:4e:33)

Type: IPv4 (0x0800)

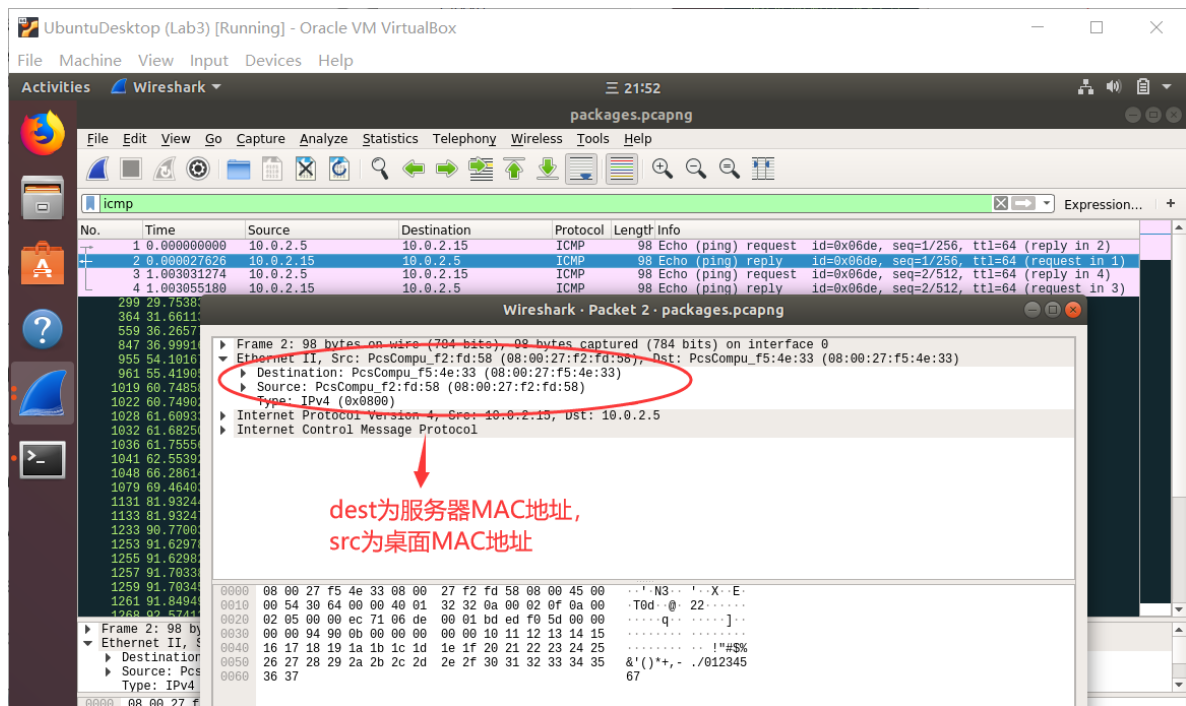
Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.15

Internet Control Message Protocol

dest为桌面MAC地址,  
src为服务器MAC地址

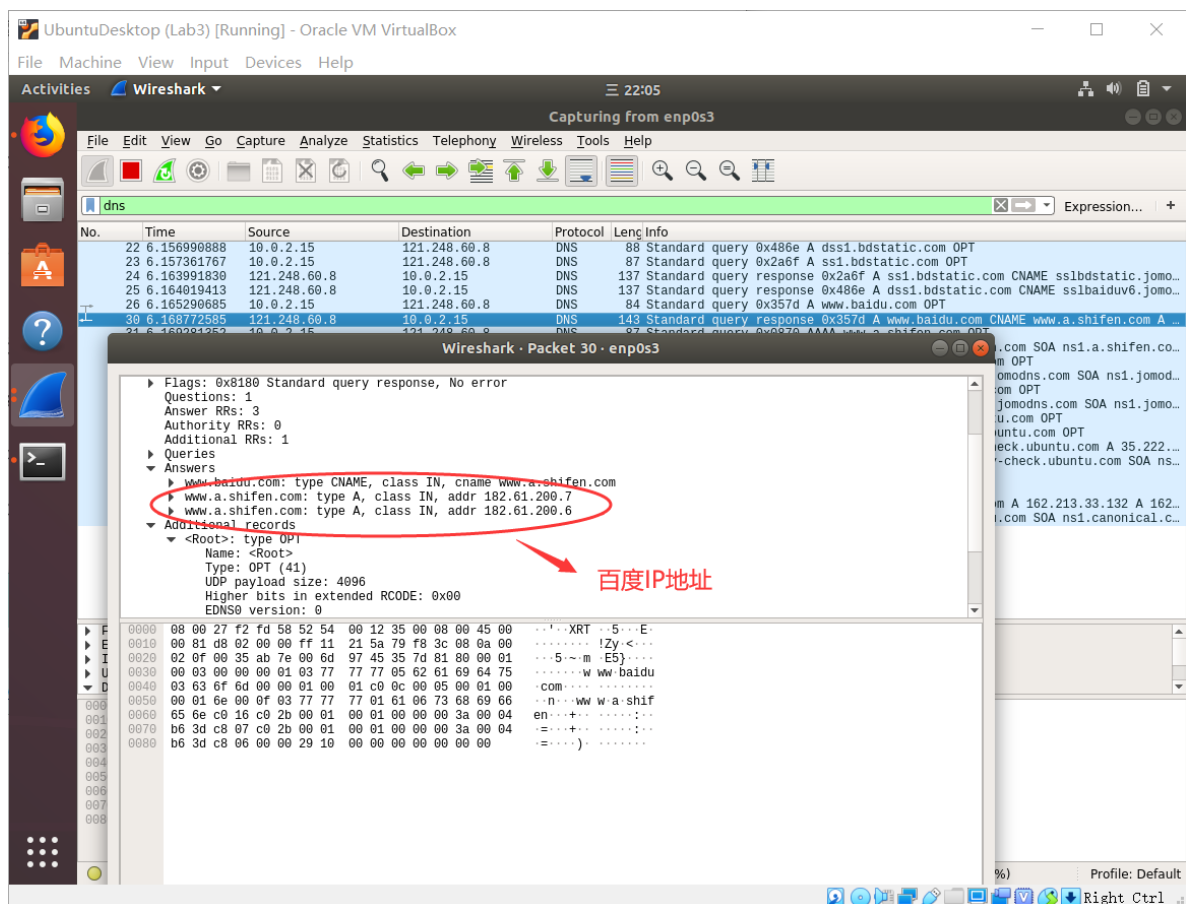
#### reply包:

服务器ping桌面



## DNS

### response包:

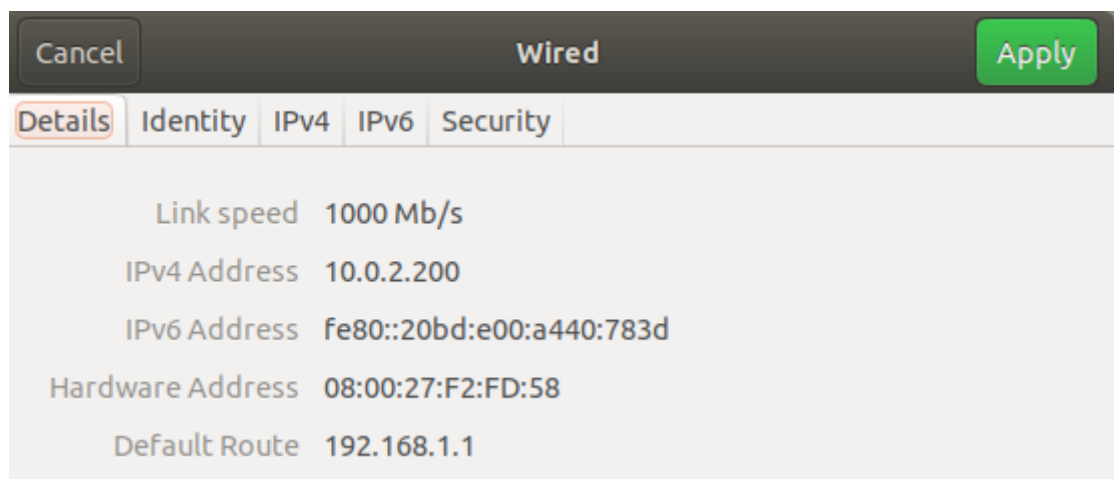
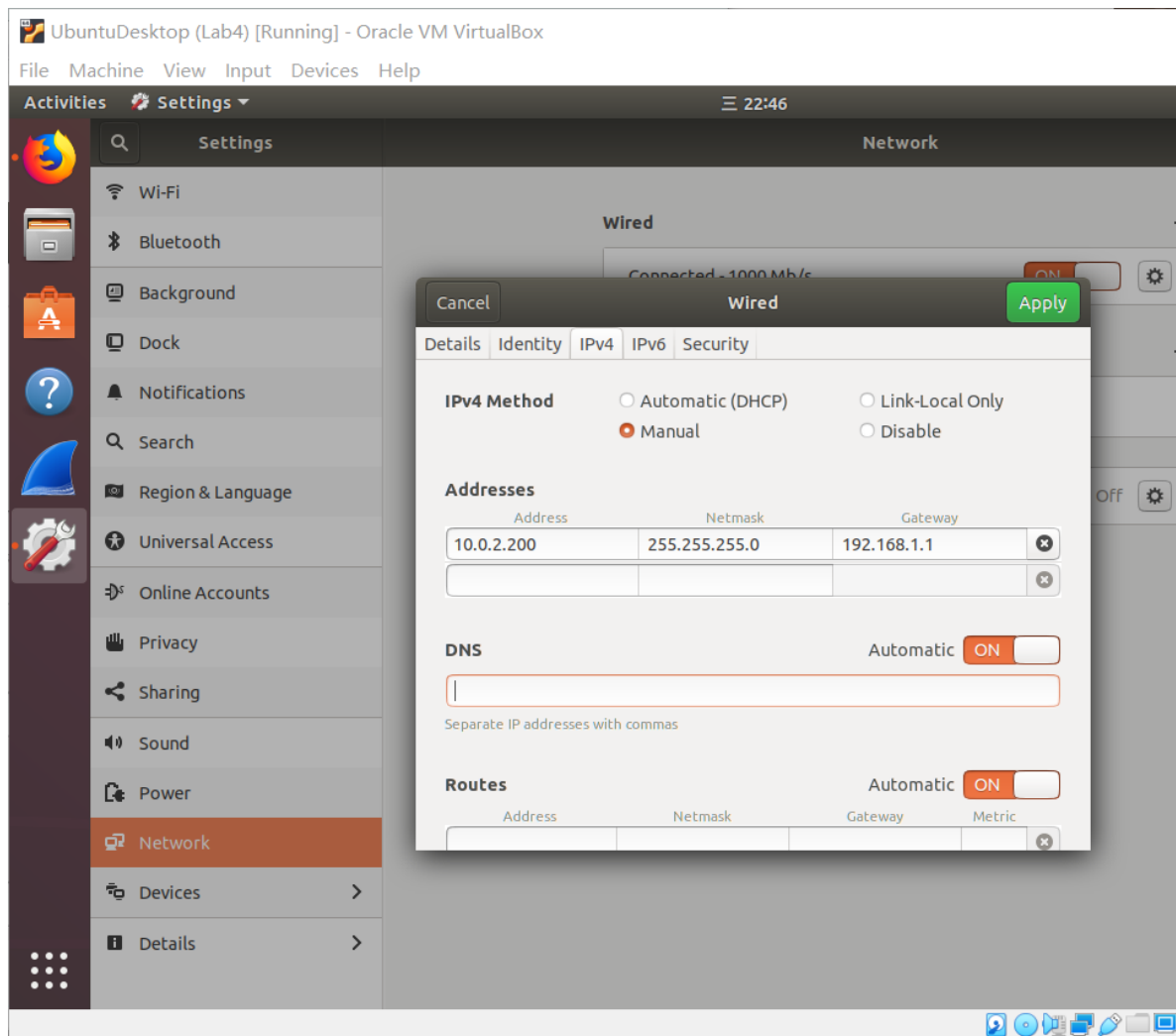


## Lab4

日志

## logs和正则表达式 (Regex)

修改桌面IP地址



打开DokuWiki进行操作

### 查看日志统计GET次数:

| 符号(pipe line)的作用: 连接两个命令, 将前一个命令的输出结果作为后一个命令的输入。

grep 命令: 查找符合条件的文本, 支持正则表达式。

wc -l: wc: word count统计功能; -l: 表示统计行数

修改前和修改后仅统计IP地址改变后的GET次数:

```
abc123@server:/var/log/apache2$ grep "GET /dokuwiki" access.log |grep "10.0.2.200"|wc -l
0
abc123@server:/var/log/apache2$ grep "GET /dokuwiki" access.log |grep "10.0.2.200"|wc -l
8
```



统计所有GET次数：

```
abc123@server:/var/log/apache2$ grep "GET /dokuwiki" access.log |wc -l
150
```

## Syslog

### 确定虚拟机同步到哪个服务器

查找12月份的同步记录

```
abc123@server:/var$ cat /etc/resolv.conf |grep 'nameserver'
nameserver 127.0.0.53
abc123@server:/var$ grep "pool server" /var/log/syslog |grep ^Dec
Dec 11 15:00:40 server ntpd[2448]: Soliciting pool server 78.46.102.180
Dec 11 15:00:41 server ntpd[2448]: Soliciting pool server 84.16.73.33
Dec 11 15:00:41 server ntpd[2448]: Soliciting pool server 108.59.2.24
Dec 11 15:00:42 server ntpd[2448]: Soliciting pool server 162.159.200.1
Dec 11 15:00:43 server ntpd[2448]: Soliciting pool server 193.182.111.141
Dec 11 15:00:45 server ntpd[2448]: Soliciting pool server 124.108.20.1
Dec 11 15:00:45 server ntpd[2448]: Soliciting pool server 119.28.206.193
Dec 11 15:00:45 server ntpd[2448]: Soliciting pool server 91.189.89.198
Dec 11 15:00:46 server ntpd[2448]: Soliciting pool server 91.189.89.199
Dec 11 15:00:46 server ntpd[2448]: Soliciting pool server 193.182.111.12
Dec 11 15:00:46 server ntpd[2448]: Soliciting pool server 185.255.55.20
Dec 11 15:00:47 server ntpd[2448]: Soliciting pool server 91.189.91.157
Dec 11 15:00:47 server ntpd[2448]: Soliciting pool server 116.203.151.74
Dec 11 15:00:47 server ntpd[2448]: Soliciting pool server 193.182.111.142
Dec 11 15:00:48 server ntpd[2448]: Soliciting pool server 91.189.94.4
Dec 11 15:00:48 server ntpd[2448]: Soliciting pool server 119.28.183.184
Dec 11 15:00:48 server ntpd[2448]: Soliciting pool server 94.237.64.20
Dec 11 15:00:49 server ntpd[2448]: Soliciting pool server 2606:4700:f1::1
```

`^xxx` 表示正则表达式开头为xxx的行

这些服NTP务器均为虚拟机同步的服务器

### 查找用户：

/etc/passwd文件的内容如下：

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4::sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
...
```

每一行是由分号分隔的字串组成，每行格式如下：

```
username: password: uid: gid: gecoss: homedir: shell
用户名: 密码: 用户ID: 组ID: 用户全名: 主目录: 登录shell
gecos是通用电子计算机操作系统的缩写，是Bell实验室中的一台大型主机。
```

由于使用shadow影子密码系统，passwd文件中可用账户的密码全部为“x”

[详细信息](#)

awk 可输出每行的指定列，用 `{print $列号}` 指令实现

```
1 #输出第一列和第三列,-F分割,-F:以:分割
2 awk -F: '{print $1, $3}'
```

### grep的表达式规则:

- `^` #锚定行的开始 如: `^grep` 匹配所有以grep开头的行。
- `\$` #锚定行的结束 如: `grep\$` 匹配所有以grep结尾的行。
- `.` #匹配一个非换行符的字符 如: `gr.p` 匹配gr后接一个任意字符, 然后是p。
- `*` #匹配零个或多个先前字符 如: `*grep` 匹配所有有一个或多个空格后紧跟grep的行。
- `.` #一起用代表任意字符。
- `[]` #匹配一个指定范围内的字符, 如 `[Gg]rep` 匹配Grep和grep。
- `[^]` #匹配一个不在指定范围内的字符, 如: `[^A-FH-Z]rep` 匹配不包含A-R和T-Z的一个字母开头, 紧跟rep的行。
- `\(.\)` #标记匹配字符, 如 `(love)`, love被标记为1。
- `\<` #锚定单词的开始, 如: `\<grep` 匹配包含以grep开头的单词的行。
- `\>` #锚定单词的结束, 如 `grep\>` 匹配包含以grep结尾的单词的行。
- `x\{m\}` #重复字符x, m次, 如: `o{5}` 匹配包含5个o的行。
- `x\{m,\}` #重复字符x,至少m次, 如: `o{5,}` 匹配至少有5个o的行。
- `x\{m,n\}` #重复字符x, 至少m次, 不多于n次, 如: `o{5,10}` 匹配5--10个o的行。
- `\w` #匹配文字和数字字符, 也就是[A-Za-z0-9], 如: `G\w*p` 匹配以G后跟零个或多个文字或数字字符, 然后是p。
- `\W` #\w的反置形式, 匹配一个或多个非单词字符, 如点号句号等。
- `\b` #单词锁定符, 如: `\bgrep\b` 只匹配grep。

查找开头为元音[aeiou]的用户名(第一列)和uid(第三列)并排序

使用 `sort -u` 指令排序

### 详细信息

-F: 以: 为分隔符

```
1 grep ^[aeiou] /etc/passwd |awk -F: '{print $1, $3}' |sort -u
```

```
abc123@server:~$ grep ^[aeiou] /etc/passwd |awk -F: '{print $1, $3}' |sort -u
abc123 1000
eeeeeeee 1008
iiiiiiii 1009
irc 39
oooooooo 1010
uucp 10
uuid 106
uuuuuuuu 1011
```

### 日志命令:

安装 openssh

```
1 | sudo apt-get install openssh-server
```

启动openssh

查看是否开启

```
1 | ps -e | grep ssh
```

看到有ssh字样，说明已启动，如果没有就手动启动

```
1 | /etc/init.d/ssh start
```

配置ssh-server，配置文件位于/etc/ssh/sshd\_config，默认端口为22，为了安全，一般自定义为其他端口，然后重启

```
1 | sudo /etc/init.d/ssh resart
```

遇到问题：

Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

解决方案——添加googleDNS服务器：

```
1 | sudo vim /etc/resolv.conf
```

在原来的nameserver后添加

```
1 | nameserver 8.8.8.8
```

保存退出，然后更新设置

```
1 | sudo apt-get update
```

然后再次安装openssh

## 虚拟机和windows系统连接

编写ifoddthenlog.sh脚本

```
1 | #!/bin/bash
2 | # Adds an account to a special group 'odds' if the passed username has a
   | uid
3 | # that is odd, and logs to syslog directly
4 | THE_GROUP="odds"
5 | usege(){
6 |     echo "USAGE: $0 username"
7 |     exit 2
8 | }
9 |
10 | #MAIN
11 |
12 | # check for argument.
13 | if [ $# -ne 1 ] ; then
14 |     echo "One argument expected."
```

```

15 usage
16 fi
17
18 # Run in root
19 if [ $EUID -ne 0 ]; then
20 echo "Must run $0 as root."
21 exit 2
22 fi
23
24 #Obtain uid
25
26 UIDIN=`/bin/grep ^$1: /etc/passwd | awk -F: '{print $3}`
27
28 #Check if odd
29 if [ -n $UIDIN ] && [ $((UIDIN%2)) -eq 1 ]; then
30 logger -t $0 "$1 has uid $UIDIN which is odd. Adding to group."
31 sudo /usr/sbin/usermod -a -G $THE_GROUP $1
32 else
33 logger -t $0 "$1 was not added to the $THE_GROUP group."
34 fi
35 exit 0

```

创建文件后运行前应添加权限，x表示可执行

```
1 | sudo chmod +x ifoddthenlog.sh
```

```

ifoddthenlog.sh
abc123@server:~$ sudo ./ifoddthenlog.sh test1
[sudo] password for abc123:
sudo: ./ifoddthenlog.sh: command not found
abc123@server:~$ sudo chmod +x ifoddthenlog.sh
abc123@server:~$ sudo ./ifoddthenlog.sh test1
abc123@server:~$ sudo ./ifoddthenlog.sh test2
abc123@server:~$ sudo ./ifoddthenlog.sh test3
abc123@server:~$ sudo ./ifoddthenlog.sh test4

```

执行脚本后查看syslog中odds:

```

abc123@server:~$ sudo ./ifoddthenlog.sh test1
abc123@server:~$ sudo ./ifoddthenlog.sh test3
abc123@server:~$ sudo ./ifoddthenlog.sh test2
abc123@server:~$ sudo ./ifoddthenlog.sh test4
abc123@server:~$ sudo ./ifoddthenlog.sh test5
abc123@server:~$ tail /var/log/syslog
Dec 12 18:42:58 server ./ifoddthenlog.sh: test3 has uid 1003 which is odd. Adding to group.
Dec 12 18:43:01 server ./ifoddthenlog.sh: test4 was not added to the odds group.
Dec 12 18:44:10 server ntpd[784]: 209.97.168.88 local addr 10.0.2.5 -> <null>
Dec 12 18:44:55 server systemd[1]: Started ntp-systemd-netif.service.
Dec 12 18:44:55 server systemd[1]: ntp-systemd-netif.service: Succeeded.
Dec 12 18:46:08 server ./ifoddthenlog.sh: test1 has uid 1001 which is odd. Adding to group.
Dec 12 18:46:13 server ./ifoddthenlog.sh: test3 has uid 1003 which is odd. Adding to group.
Dec 12 18:46:16 server ./ifoddthenlog.sh: test2 was not added to the odds group.
Dec 12 18:46:19 server ./ifoddthenlog.sh: test4 was not added to the odds group.
Dec 12 18:46:22 server ./ifoddthenlog.sh: test5 has uid 1005 which is odd. Adding to group.
abc123@server:~$ sudo grep ^odds: /etc/group
odds:x:1013:test1,test3,test5

```

可以看到对test1, 3, 5的操作被写入log中

## Lab5

搭建DNS服务器

# 安装 BIND

## 设置缓存/代理域名服务器

修改/etc/bind/named.conf.options

添加SEU DNS和谷歌Public DNS

### 测试查询语句

查询一个不存在的域名 uniseu.edu

```
1 | dig uniseu.edu
```

```
abc123@server:~$ dig uniseu.edu

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> uniseu.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 1777
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;uniseu.edu.                IN      A

;; Query time: 10 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 12:05:47 CST 2019
;; MSG SIZE rcvd: 39
```

发现查询时间为10毫秒

间隔5~10秒再次发起查询

```
abc123@server:~$ dig uniseu.edu

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> uniseu.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 11165
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;uniseu.edu.                IN      A

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 12:07:11 CST 2019
;; MSG SIZE rcvd: 39
```

查询时间变为0ms。

原因是因为第二次查询实际上是缓存结果

## 设置静态IP

查询LAN路由IP

```
1 | route -n
```

```
abc123@server:/etc/bind$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          10.0.2.1        0.0.0.0          UG    100    0      0 enp0s3
10.0.2.0         0.0.0.0         255.255.255.0    U     0      0      0 enp0s3
10.0.2.1         0.0.0.0         255.255.255.255 UH    100    0      0 enp0s3
```

将服务器的IP改为10.0.2.100

```
1 | sudo pico /etc/netplan/50-cloud-init.yaml
```

注意不能打tab只能打空格

```
1 | network:
2 |   ethernets:
3 |     enp0s3:
4 |       addresses:
5 |         - 10.0.2.100/24
6 |       gateway4: 10.0.2.1
7 |       nameservers:
8 |         addresses: [10.0.2.100]
9 |       optional: true
10 |   version: 2
11 |
```

```
1 | sudo netplan apply
```

```
abc123@server:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          10.0.2.1        0.0.0.0          UG     0      0      0 enp0s3
10.0.2.0         0.0.0.0         255.255.255.0    U     0      0      0 enp0s3
```

继续修改桌面的IP和DNS和Gateway

## 为abc123.test设置转发区域文件

修改 `./etc/bind/zones/rfc1918`

```
1 | zone "abc123.test"{
2 |   type master;
3 |   file "/etc/bind/db.abc123.test";
4 | };
5 |
6 | zone "2.0.10.in-addr.arpa"{
7 |   type master;
8 |   notify no;
9 |   file "/etc/bind/db.10";
10 | };
```

复制文件db.local为db.abc123.test, 然后设置新的转发区域文件

修改SOA的序号 (serial) 和其他的内容

```
1 ;
2 ; This is the forward zone for abc123.test internal domain ;
3 ;
4 $TTL      604800
5 @         IN      SOA      server.abc123.test. asl.seu.edu.cn. (
6                                     2019112001      ; Serial
7                                     604800          ; Refresh
8                                     86400           ; Retry
9                                     2419200         ; Expire
10                                    604800 )         ; Negative Cache TTL
11 ;
12 @         IN      NS       server.abc123.test.
13 @         IN      A        127.0.0.1
14 @         IN      AAAA     ::1
15
16 gateway   IN      A        10.0.2.1
17 server    IN      A        10.0.2.100
18 desktop   IN      A        10.0.2.200
19
20 dns       IN      CNAME     server
21 www       IN      CNAME     server
22
```

同理修改reverse zone file:

```
1 ;
2 ; This is the reverse zone for abc123.test internal domain ;
3 ;
4 $TTL      604800
5 @         IN      SOA      server.abc123.test. asl.seu.edu.cn. (
6                                     2019112001      ; Serial
7                                     604800          ; Refresh
8                                     86400           ; Retry
9                                     2419200         ; Expire
10                                    604800 )         ; Negative Cache TTL
11 ;
12 @         IN      NS       server.abc123.test.
13 1         IN      PTR      gateway.abc123.test.
14 100       IN      PTR      server.abc123.test.
15 200       IN      PTR      desktop.abc123.test.
```

检查设置, 然后重启服务器和桌面

## 测试abc123.test域的DNS

### 测试forward zone

测试

```
1 | dig desktop.abc123.test
```

```

abc123@server:~$ dig desktop.abc123.test
; <<>> DiG 9.11.5-P1-ubuntu2.6-Ubuntu <<>> desktop.abc123.test
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27316
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;desktop.abc123.test.          IN      A
;
; ANSWER SECTION:
desktop.abc123.test.          7154    IN      A      10.0.2.200
;
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 14:18:15 CST 2019
;; MSG SIZE rcvd: 64

```

可以看到desktop.abc.123的信息

也可以指定DNS来查询主域名服务器

```
1 | dig desktop.abc123.test @server.abc123.test
```

```

abc123@server:~$ dig desktop.abc123.test @server.abc123.test
; <<>> DiG 9.11.5-P1-ubuntu2.6-Ubuntu <<>> desktop.abc123.test @server.abc123.test
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42773
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ce38e3b24b23d5483fdc67485df47f4e82b131f143ae07be (good)
;; QUESTION SECTION:
;desktop.abc123.test.          IN      A
;
; ANSWER SECTION:
desktop.abc123.test.          604800  IN      A      10.0.2.200
;
; AUTHORITY SECTION:
abc123.test.                  604800  IN      NS      server.abc123.test.
;
; ADDITIONAL SECTION:
server.abc123.test.           604800  IN      A      10.0.2.100
;
;; Query time: 0 msec
;; SERVER: 10.0.2.100#53(10.0.2.100)
;; WHEN: Sat Dec 14 14:21:02 CST 2019
;; MSG SIZE rcvd: 129
abc123@server:~$

```

可以看到查询结果 (answer section) 和认证结果 (authority section)

## 测试reverse zone

```
1 | dig -x 10.0.2.200 @server.abc123.test
```



```

abc123@server:~$ dig -x 10.0.2.200 @server.abc123.test

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> -x 10.0.2.200 @server.abc123.test
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21397
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 70dd65f60574a7d9abc52fb65df48d0b14c3ae24d991bc7f (good)
;; QUESTION SECTION:
;200.2.0.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
200.2.0.10.in-addr.arpa. 604800 IN      PTR      gateway.abc123.test.

;; AUTHORITY SECTION:
2.0.10.in-addr.arpa.    604800 IN      NS       server.abc123.test.

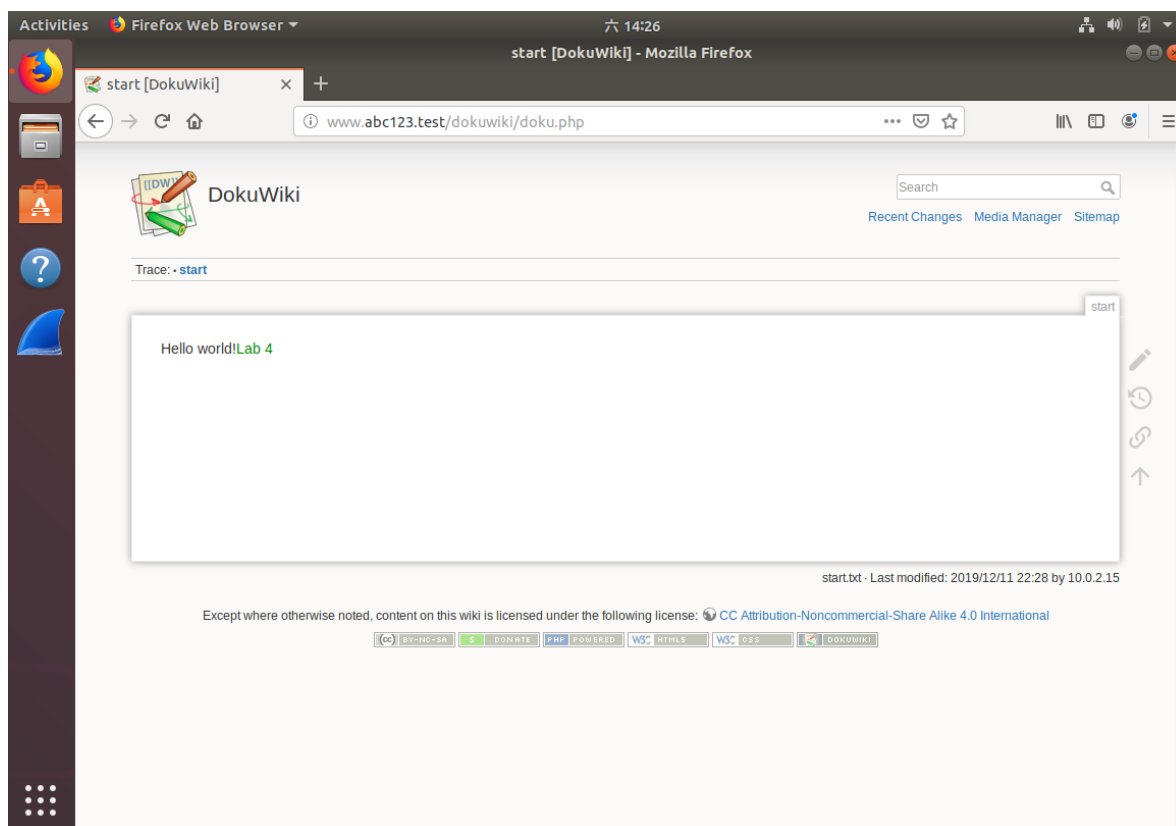
;; ADDITIONAL SECTION:
server.abc123.test.    604800 IN      A        10.0.2.100

;; Query time: 0 msec
;; SERVER: 10.0.2.100#53(10.0.2.100)
;; WHEN: Sat Dec 14 15:19:39 CST 2019
;; MSG SIZE rcvd: 150

```

## 在桌面访问dokuwiki

用桌面打开 [www.abc123.test/dokuwiki](http://www.abc123.test/dokuwiki)



发现可以打开

## 其他dig命令

## 通过域名服务器查看域名地址

```
1 | dig @seic8.seu.edu.cn www.seu.edu.cn
```

@后为指定的服务器

```
abc123@server: ~$ dig @seic8.seu.edu.cn www.seu.edu.cn

; <<>> DiG 9.11.5-P1-lubuntu2.6-Ubuntu <<>> @seic8.seu.edu.cn www.seu.edu.cn
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54124
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.seu.edu.cn.                IN      A

;; ANSWER SECTION:
www.seu.edu.cn.                3600    IN      CNAME   widc142.seu.edu.cn.
widc142.seu.edu.cn.           3600    IN      A        58.192.118.142

;; Query time: 6 msec
;; SERVER: 202.119.24.18#53(202.119.24.18)
;; WHEN: Sat Dec 14 14:32:29 CST 2019
;; MSG SIZE rcvd: 81
```

## 通过IP地址查看对应域名

```
1 | dig -x 121.248.60.55
```

```
abc123@server: ~$ dig -x 121.248.60.55

; <<>> DiG 9.11.5-P1-lubuntu2.6-Ubuntu <<>> -x 121.248.60.55
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1887
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;55.60.248.121.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
55.60.248.121.in-addr.arpa. 3600    IN      PTR      voidc55.seu.edu.cn.

;; Query time: 82 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 14:34:16 CST 2019
;; MSG SIZE rcvd: 87
```

```
1 | dig -x 8.8.8.8
```

```

abc123@server:~$ dig -x 8.8.8.8

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51360
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.    21295   IN      PTR      dns.google.

;; Query time: 3734 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 14:37:20 CST 2019
;; MSG SIZE  rcvd: 73

```

查看特殊类型的DNS信息，如查看邮件DNS地址

```
1 | dig -t MX seu.edu.cn
```

```

abc123@server:~$ dig -t MX seu.edu.cn

; <<>> DiG 9.11.5-P1-1ubuntu2.6-Ubuntu <<>> -t MX seu.edu.cn
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45593
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;seu.edu.cn.                  IN      MX

;; ANSWER SECTION:
seu.edu.cn.                  412     IN      MX      1 voidc50.seu.edu.cn.

;; Query time: 278 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 14 14:39:40 CST 2019
;; MSG SIZE  rcvd: 63

```

则东大邮箱 mail.seu.edu.cn 内部host为 voidc50.sdu.edu.cn

[dig命令详解](#)

## Lab6

## 服务器添加磁盘

### 安装RAID

识别新建的磁盘

```
1 | lsblk -o NAME,SIZE,FSTYPE,MOUNTPOINT
```

创建RAID5 array

```
1 | sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb  
/dev/sdc /dev/sdd
```

检查磁盘状态

```
1 | cat /proc/mdstat
```

创建并挂载(mount)文件系统

```
1 | sudo mkfs.ext4 /dev/md0
```

创建一个挂载点来附加新文件系统

```
1 | sudo mkdir /mnt/md0
```

挂载文件系统

```
1 | sudo mount /dev/md0 /mnt/md0
```

可以通过下面命令来检查新空间

```
1 | df -h
```

保存RAID阵列布局，以便在启动时自动重新组装阵列，将阵列配置附加到 `/etc/mdadm/mdadm.conf`

```
1 | sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf  
2 | sudo update-initramfs -u
```

```
abc123@server:~$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf  
ARRAY /dev/md0 metadata=1.2 name=server:0 UUID=e58ca0a5:4287ca86:d2ac1825:bb3f1947  
abc123@server:~$ sudo update-initramfs -u  
update-initramfs: Generating /boot/initrd.img-5.0.0-37-generic
```

查看“`/etc/mdadm/mdadm.conf`”的配置内容

```

abc123@server:/etc/mdadm$ cat mdadm.conf
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#
# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
#DEVICE partitions containers

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays

# This configuration was auto-generated on Tue, 16 Apr 2019 19:46:02 +0000 by mkconf
ARRAY /dev/md0 metadata=1.2 name=server:0 UUID=e58ca0a5:4287ca86:d2ac1825:bb3f1947

```

将新的文件系统挂载选项添加到/etc/fstab文件中，以便在启动时自动挂载

```

1 | echo '/dev.md0 /mnt/md0 ext4 defaults,nofail,sicard 0 0' | sudo tee -a
   | /etc/fstab

```

```

abc123@server:/$ echo '/dev.md0 /mnt/md0 ext4 defaults,nofail,sicard 0 0' | sudo tee -a /etc/fstab
[sudo] password for abc123:
/dev.md0 /mnt/md0 ext4 defaults,nofail,sicard 0 0

```

查看“/etc/fstab”文件，可以看到已经添加 mnt/md0

```

root@server:/mnt/md0# cat /etc/fstab
UUID=b9937854-5640-4a6e-99d8-c7c33da7b75c / ext4 defaults 0 0
/swap.img none swap sw 0 0
/dev.md0 /mnt/md0 ext4 defaults,nofail,sicard 0 0

```

在共享文件夹下创建文件

```

1 | cd /mnt/md0
2 | sudo pico readme.txt

```

```

root@server:/mnt/md0# ls
readme.txt

```

## 创建NFS（Network File System）网络文件系统

安装NFS sever

```

1 | sudo apt install nfs-kernel-server

```

向 etc/exports 文件中添加内容

```

1 | sudo pico /etc/exports

```

在 exports 中添加：

```
1 | /mnt/md0 *(rw,sync,no_root_squash)
```

开启NFS server

```
1 | sudo systemctl start nfs-kernel-server.service
```

可以看到目录里面已有mdt文件

```
abc123@server:/$ ls
bin    etc    initrd.img    lib32    lost+found    opt    run    srv    tmp    vmlinuz
boot   exports  initrd.img.old  lib64    media    proc    sbin    swap.img  usr    vmlinuz.old
dev    home    lib    libx32    mnt    root    snap    sys    var
```

## 从桌面虚拟机连接NFS共享

以下均在桌面操作

安装NFS客户端

```
1 | sudo apt install nfs-common
```

检查共享列表

```
1 | sudo apt install nfs-common
```

```
root@desktop:/home/vana# sudo showmount -e server.abc123.test
Export list for server.abc123.test:
/mnt/md0 *
```

看到 /mnt/md0 已被共享

安装NFS共享

创建一个挂载点，即本机与服务器同步的文件夹为 /share/md0

```
1 | sudo mkdir -p /share/md0
```

```
root@desktop:/# ls
bin    dev    initrd.img    lib64    mnt    root    share    swapfile    usr
boot   etc    initrd.img.old  lost+found  opt    run    snap    sys    var
cdrom  home  lib    media    proc    sbin    srv    tmp    vmlinuz
```

挂载NFS共享，将 /share/md0 和 /mnt/md0 关联

```
1 | sudo mount -t nfs server.abc123.test:/mnt/md0 /share/md0
```

查看共享文件夹下的文件，应该可以看到在服务器上创建的 readme.txt

```
1 | ll /share/md0
2 | #或者
3 | ls /share/md0
```

```
root@desktop:/# ll /share/md0
total 12
drwxr-xr-x 2 root root 4096 12月 15 17:39 ./
drwxr-xr-x 3 root root 4096 12月 15 17:20 ../
-rw-r--r-- 1 root root  28 12月 15 17:39 readme.txt
```

从桌面系统向共享文件夹添加文件 fromdesktop.txt

```
root@desktop:/share/md0# pico fromdesktop.txt
root@desktop:/share/md0# ls
fromdesktop.txt  readme.txt
```

以下操作在服务器上进行:

在服务器查看共享文件夹中的文件是否被同步

```
root@server:/mnt/md0# ll
total 16
drwxr-xr-x 2 root root 4096 Dec 15 17:52 ./
drwxr-xr-x 3 root root 4096 Dec 15 11:52 ../
-rw-r--r-- 1 root root  24 Dec 15 17:52 fromdesktop.txt
-rw-r--r-- 1 root root  28 Dec 15 17:39 readme.txt
```

设置开机加载共享文件

```
1 | sudo pico /etc/fstab
```

在 fstab 中添加

```
1 | server.abc123.test:/mnt/md0 /share/md0 nfs defaults,user,exec 0 0
```

## 在服务器VM上模拟RAID阵列中的磁盘故障

检查阵列状态

```
1 | sudo mdadm -D /dev/md0
```

检查阵列的一个磁盘状态(盘名sdd)

```
1 | sudo mdadm -E /dev/sdd
```

模拟磁盘故障

关机, 删除myDisk2.vdi, 重启

强制重装磁盘

```
1 | sudo mdadm --stop /dev/md0
2 | sudo mdadm --assemble --force /dev/md0
```

重新查看磁盘阵列状态

```
1 | sudo mdadm -D /dev/md0
```

```

abc123@server:~$ sudo mdadm -D /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Sun Dec 15 11:46:24 2019
    Raid Level : raid5
    Array Size : 2093056 (2044.00 MiB 2143.29 MB)
  Used Dev Size : 1046528 (1022.00 MiB 1071.64 MB)
    Raid Devices : 3
  Total Devices : 2
 Persistence : Superblock is persistent

 Update Time : Sun Dec 15 12:10:53 2019
      State : clean, degraded
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
 Spare Devices : 0

```

Number	Major	Minor	RaidDevice	State	
—	0	0	0	removed	
1	8	16	1	active sync	/dev/sdb
3	8	32	2	active sync	/dev/sdc

可以看到只剩下两个磁盘 (sdb, sdc) 状态为clean和degrade

重新查看NFS共享文件夹中的文件

```

abc123@server:/mnt/md0$ ll
total 16
drwxr-xr-x 2 root root 4096 Dec 15 17:52 ./
drwxr-xr-x 3 root root 4096 Dec 15 11:52 ../
-rw-r--r-- 1 root root  24 Dec 15 17:52 fromdesktop.txt
-rw-r--r-- 1 root root  28 Dec 15 17:39 readme.txt

```

```

abc123@server:/mnt/md0$ cat readme.txt
This is my storage on RAID5

```

文件完好并且可以访问

向磁盘阵列添加一个新的磁盘替代损坏的磁盘

```
1 | 1sb1k
```



```

abc123@server: ~$ lsblk -o NAME, SIZE, FSTYPE, MOUNTPOINT
NAME        SIZE FSTYPE      MOUNTPOINT
loop0       89.1M squashfs    /snap/core/8268
loop1       89.1M squashfs    /snap/core/8213
loop2       53.9M squashfs    /snap/lxd/10601
loop3       54.9M squashfs    /snap/lxd/12631
sda          10G
├── sda1       1M
└── sda2      10G ext4      /
sdb          1G
sdc          1G linux_raid_member
├── md0        2G ext4
sdd          1G linux_raid_member
├── md0        2G ext4
sr0         73.6M iso9660

```

可以看到新添加的磁盘为sdb

将新添加的磁盘加入磁盘阵列替代出错的磁盘

```
1 | sudo mdadm --add /dev/md0 /dev/sdb
```

再次查看磁盘阵列状态

```
1 | sudo mdadm -D /dev/md0
```

```

abc123@server: ~$ sudo mdadm -D /dev/md0
/dev/md0:
        Version : 1.2
        Creation Time : Sun Dec 15 11:46:24 2019
        Raid Level : raid5
        Array Size : 2093056 (2044.00 MiB 2143.29 MB)
        Used Dev Size : 1046528 (1022.00 MiB 1071.64 MB)
        Raid Devices : 3
        Total Devices : 3
        Persistence : Superblock is persistent

        Update Time : Sun Dec 15 19:44:30 2019
        State : clean
        Active Devices : 3
        Working Devices : 3
        Failed Devices : 0
        Spare Devices : 0

```

Number	Major	Minor	RaidDevice	State	
4	8	16	0	active sync	/dev/sdb
1	8	32	1	active sync	/dev/sdc
3	8	48	2	active sync	/dev/sdd

磁盘状态变为clean，已合并

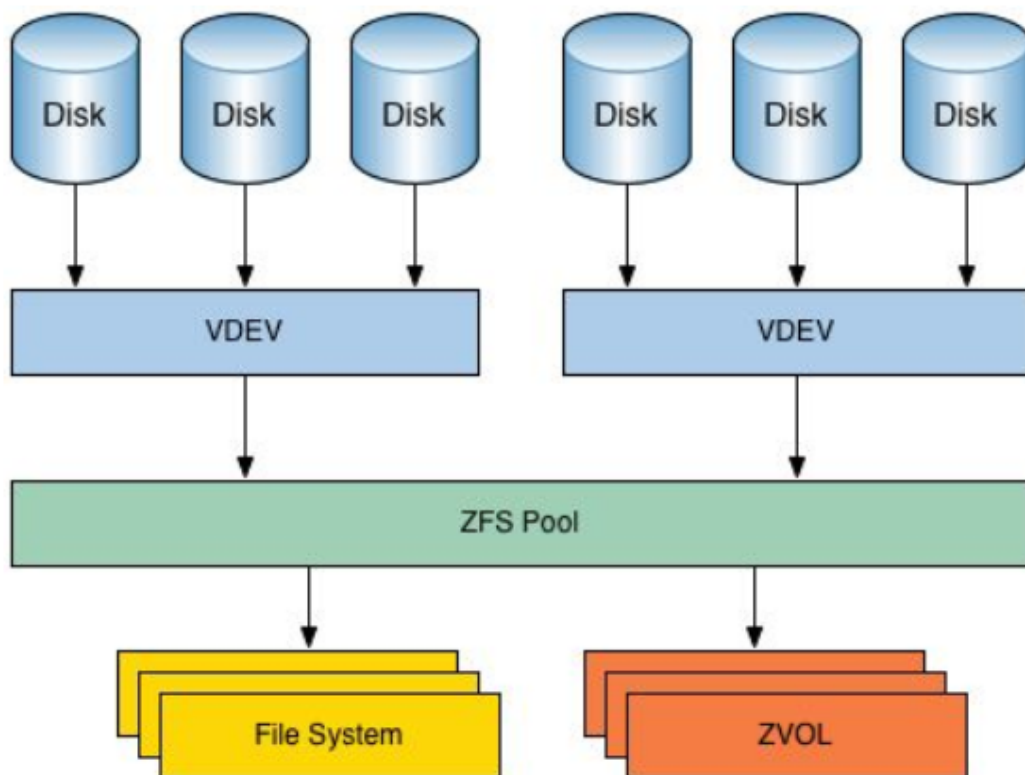
## Lab7

Zettabyte File System，也叫动态文件系统（Dynamic File System）

ZFS是一个组合文件系统和逻辑卷管理器，它可以充分利用可用磁盘的能力。ZFS 可以创建跨越一系列硬盘或池的文件系统。

[ZFS详细介绍](#)

结构



## 向服务器添加硬盘

## 安装ZFS

```
1 | sudo apt install zfsutils-linux
```

## 创建ZFS pool存储池

查看创建的磁盘，分别为efgh

```
1 | lsblk -o NAME,SIZE,FSTYPE,MOUNTPOINT
```

```

abc123@server:~$ lsblk -o NAME, SIZE, FSTYPE, MOUNTPOINT
NAME        SIZE FSTYPE      MOUNTPOINT
loop0       54.9M squashfs    /snap/lxd/12631
loop1       89.1M squashfs    /snap/core/8268
loop2       89.1M squashfs    /snap/core/8213
loop3       53.9M squashfs    /snap/lxd/10601
sda          10G
├── sda1       1M
└── sda2      10G ext4      /
sdb          1G linux_raid_member
└── md0        2G ext4
sdc          1G linux_raid_member
└── md0        2G ext4
sdd          1G linux_raid_member
└── md0        2G ext4
sde          1G
sdf          1G
sdg          2G
sdh          2G
sr0         73.6M iso9660

```

用两个1GB的磁盘(e,f)作为镜像VDEV创建存储池（RAID0）

```
1 | sudo zpool create myZpool mirror /dev/sde /dev/sdf
```

其中“myZpool”为镜像池的名字

查看ZFS的zpool

```
1 | zpool status
```

查看新添加的空间

```
1 | df -h
```

其他指令

```

1 | zpool get health myZpool#健康状态为ONLINE
2 | zpool get size,free,allocated myZpool#查看其他特性

```

## 创建文件系统（dataset）

文件存储在ZFS池中的数据集（dataset）中。数据集将继续在池中放置文件，直到池被填满。可以在dataset上设置配额

创建dataset

```

1 | sudo zfs create myZpool/test1
2 | sudo zfs create myZpool/test2
3 | sudo zfs create myZpool/test3

```

查看dataset

```
1 | zfs list
```

```
abc123@server:~$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
myZpool                             160K   880M   24K    /myZpool
myZpool/test1                       24K   880M   24K    /myZpool/test1
myZpool/test2                       24K   880M   24K    /myZpool/test2
myZpool/test3                       24K   880M   24K    /myZpool/test3
```

每个dataset都被自动挂载到其挂载点，并具有对存储池的完全访问权

为文件设置权限以便可执行

```
1 | sudo chmod 777 /myZpool/test*
```

修改之前创建的脚本文件“test3”使之生成10条随机数据(访问.bat 文件需要sudo，使用 `sudo su`)

```
1 | cd /myZpool/test3
2 | sudo su
3 | for i in {1..10}; do dd if=/dev/urandom of=file$i.dat bs=1024 count=$RANDOM;
4 | done
```

```
root@server:/myZpool/test3# ls
file10.dat file1.dat file2.dat file3.dat file4.dat file5.dat file6.dat file7.dat file8.dat file9.dat
```

可以看到增加了10个随机文件

再次查看datasets

```
abc123@server:~$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
myZpool                             150M   730M   24K    /myZpool
myZpool/test1                       24K   730M   24K    /myZpool/test1
myZpool/test2                       24K   730M   24K    /myZpool/test2
myZpool/test3                       149M   730M  149M    /myZpool/test3
```

test3数据集占用的空间增加

## 对dataset压缩

查看压缩状态

```
1 | zfs get -r compression myZpool
```

```
root@server:/myZpool/test3# zfs get -r compression myZpool
NAME                                PROPERTY  VALUE      SOURCE
myZpool                             compression  off        default
myZpool/test1                       compression  off        default
myZpool/test2                       compression  off        default
myZpool/test3                       compression  off        default
```

默认关闭压缩

对test1开启压缩，类型是lz4

```
1 | sudo zfs set compression=lz4 myZpool/test1
2 | zfs get -r compression myZpool
```

```
root@server:/myZpool/test3# sudo zfs set compression=lz4 myZpool/test1
root@server:/myZpool/test3# zfs get -r compression myZpool
```

NAME	PROPERTY	VALUE	SOURCE
myZpool	compression	off	default
myZpool/test1	compression	lz4	local
myZpool/test2	compression	off	default
myZpool/test3	compression	off	default

对test1和test2分别生成一个1000000行的txt文件，测试压缩效果

test1

```
1 | cd /myZpool/test1
2 | sudo su
3 | for i in {1..1000000}; do echo "Line $i: This is a line of text." >>
   | textfile.txt; done
```

test2

```
1 | cd /myZpool/test2
2 | sudo su
3 | for i in {1..1000000}; do echo "Line $i: This is a line of text." >>
   | textfile.txt; done
```

再次查看datasets

```
1 | zfs list
```

```
root@server:/myZpool/test2# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
myZpool	190M	690M	24K	/myZpool
myZpool/test1	5.01M	690M	5.01M	/myZpool/test1
myZpool/test2	35.3M	690M	35.3M	/myZpool/test2
myZpool/test3	149M	690M	149M	/myZpool/test3

发现压缩后的dataset占用的空间小于未压缩的

### 问题

随机生成10个dat文件，test1占用的空间竟然比test2还大。。

```
root@server:/myZpool# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
myZpool	483M	397M	24K	/myZpool
myZpool/test1	168M	397M	168M	/myZpool/test1
myZpool/test2	166M	397M	166M	/myZpool/test2
myZpool/test3	149M	397M	149M	/myZpool/test3

### 查看压缩率

```
1 | zfs get compressratio myZpool/test1
```

```
root@server:/myZpool/test2# zfs get compressratio myZpool/test1
NAME                PROPERTY          VALUE  SOURCE
myZpool/test1       compressratio     7.07x  -
```

7.07倍压缩率

## 快照

ZFS快照是数据集或整个池的只读副本。它保存了ZFS文件系统在某个时间点的状态，这个时间点可以在以后回滚。可以从快照中提取文件，而不需要执行完整的回滚。

在执行快照之前创建一个文件

```
1 | cd /myZpool/test1
2 | echo `date` >> myfile.txt
3 | cat myfile.txt
```

```
root@server:/myZpool/test1# cat myfile.txt
Sun 15 Dec 2019 09:21:59 PM CST
```

### 创建快照并查看

```
1 | sudo zfs snapshot -r myZpool/test1@snapshot1
2 | zfs list -t snapshot
```

```
root@server:/myZpool/test1# sudo zfs snapshot -r myZpool/test1@snapshot1
root@server:/myZpool/test1# zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
myZpool/test1@snapshot1             0B    -     5.01M  -
```

删除 `textfile` 文件

```
1 | rm textfile.txt
```

从快照中提取需要恢复的文件副本（快照可以只恢复想要的文件）

```
1 | cp .zfs/snapshot/snapshot1/textfile.txt textfile.txt
```

发现之前误删除的文件已恢复

```
root@server:/myZpool/test1# cp .zfs/snapshot/snapshot1/textfile.txt textfile.txt
root@server:/myZpool/test1# ls
myfile.txt  textfile.txt
```

在 `myfile` 文件添加修改时间

```
1 | echo `date` >> myfile.txt
2 | cat myfile.txt
```

```
root@server:/myZpool/test1# cat myfile.txt
Sun 15 Dec 2019 09:21:59 PM CST
Sun 15 Dec 2019 09:25:52 PM CST
```

撤回修改——提取快照中需要恢复的文件副本

```
1 | cp .zfs/snapshot/snapshot1/myfile.txt myfile.txt
2 | cat myfile.txt
```

```
root@server:/myZpool/test1# cp .zfs/snapshot/snapshot1/myfile.txt myfile.txt
root@server:/myZpool/test1# cat myfile.txt
Sun 15 Dec 2019 09:21:59 PM CST
```

发现文件已经被覆盖

回滚dataset

```
1 | sudo zfs rollback myZpool/test1@snapshot1
```

## 扩充pool的容量

添加其他的2个VDEV磁盘（g和h）到镜像池中

```
1 | sudo zpool add myZpool mirror /dev/sdg /dev/sdh
2 | zpool status
3 | zfs list
```

```
root@server:/myZpool/test1# zpool status
pool: myZpool
state: ONLINE
scan: none requested
config:
```

	NAME	STATE	READ	WRITE	CKSUM
	myZpool	ONLINE	0	0	0
	mirror-0	ONLINE	0	0	0
	sde	ONLINE	0	0	0
	sdf	ONLINE	0	0	0
	mirror-1	ONLINE	0	0	0
	sdg	ONLINE	0	0	0
	sdh	ONLINE	0	0	0

```
root@server:/myZpool/test1# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
myZpool                             195M  2.65G   24K    /myZpool
myZpool/test1                       10.0M  2.65G   5.01M  /myZpool/test1
myZpool/test2                       35.3M  2.65G   35.3M  /myZpool/test2
myZpool/test3                       149M  2.65G   149M   /myZpool/test3
```

可以看到又增加了一个新的镜像mirror-1



## 测试可靠性

从pool删除磁盘，模拟磁盘错误

```
1 | sudo zpool detach myZpool /dev/sde
2 | zpool status
```

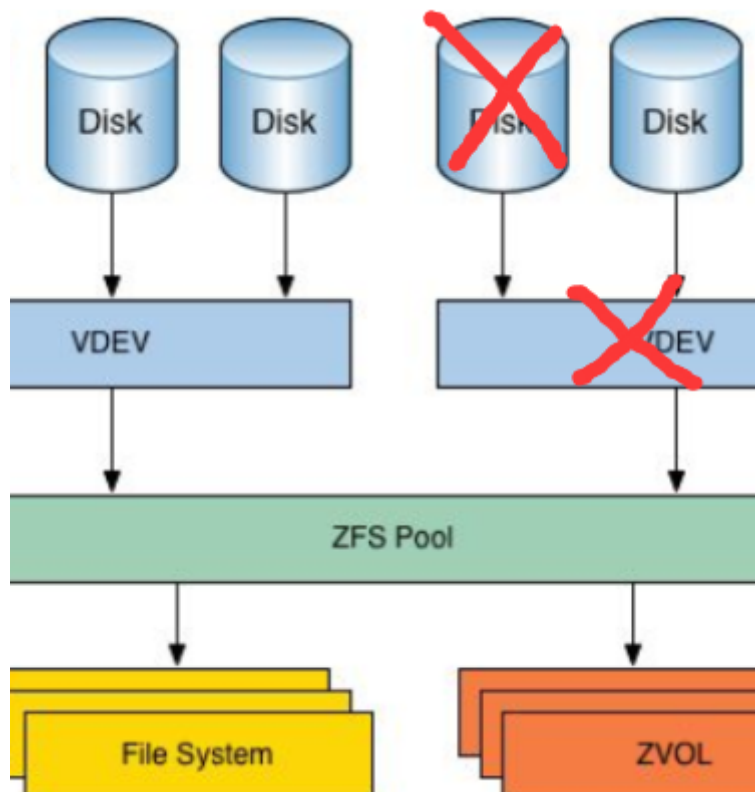
```
root@server:/myZpool/test1# zpool status
pool: myZpool
state: ONLINE
scan: none requested
config:

    NAME        STATE        READ  WRITE  CKSUM
    myZpool      ONLINE       0     0     0
      sdf        ONLINE       0     0     0
        mirror-1 ONLINE       0     0     0
          sdg     ONLINE       0     0     0
          sdh     ONLINE       0     0     0

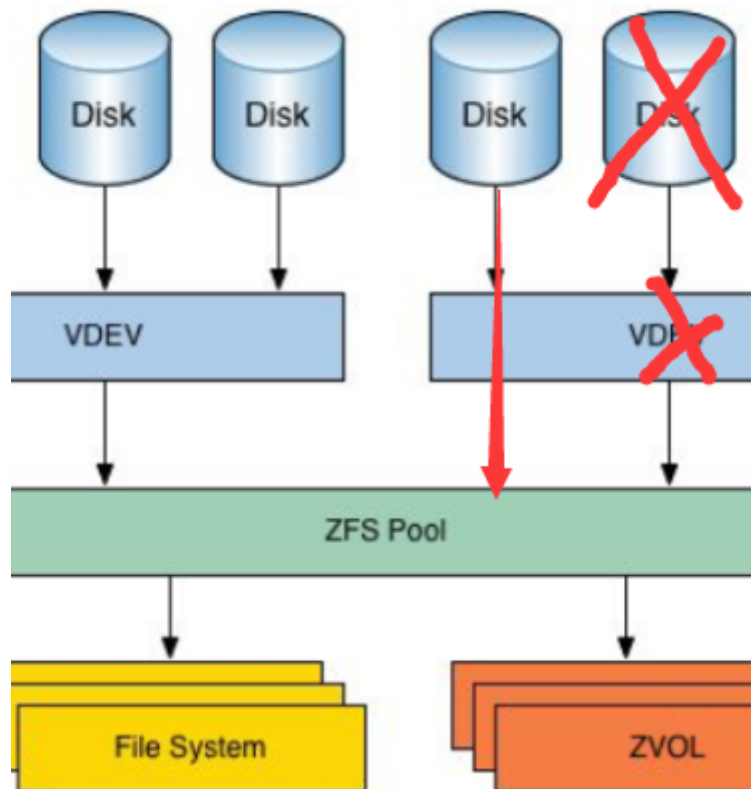
errors: No known data errors
```

mirror-0消失，只剩下sdf，，但是以然可以对文件进行操作。

myZpool直接指向myZpool，与mirror-1，同层







将sdf变为新的mirror

重写sde, 大小变为0

```
1 | sudo dd if=/dev/zero of=/dev/sde bs=1M count=1024
```

将sdf变为新的mirror

```
1 | sudo zpool attach myzpool /dev/sdf /dev/sde
2 | zpool status
```

```
root@server:/myZpool/test1# sudo zpool attach myZpool /dev/sdf /dev/sde
root@server:/myZpool/test1# zpool status
  pool: myZpool
 state: ONLINE
  scan: resilvered 195M in 0h0m with 0 errors on Sun Dec 15 22:01:23 2019
config:
```

NAME	STATE	READ	WRITE	CKSUM
myZpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
sdf	ONLINE	0	0	0
sde	ONLINE	0	0	0
mirror-1	ONLINE	0	0	0
sdg	ONLINE	0	0	0
sdh	ONLINE	0	0	0

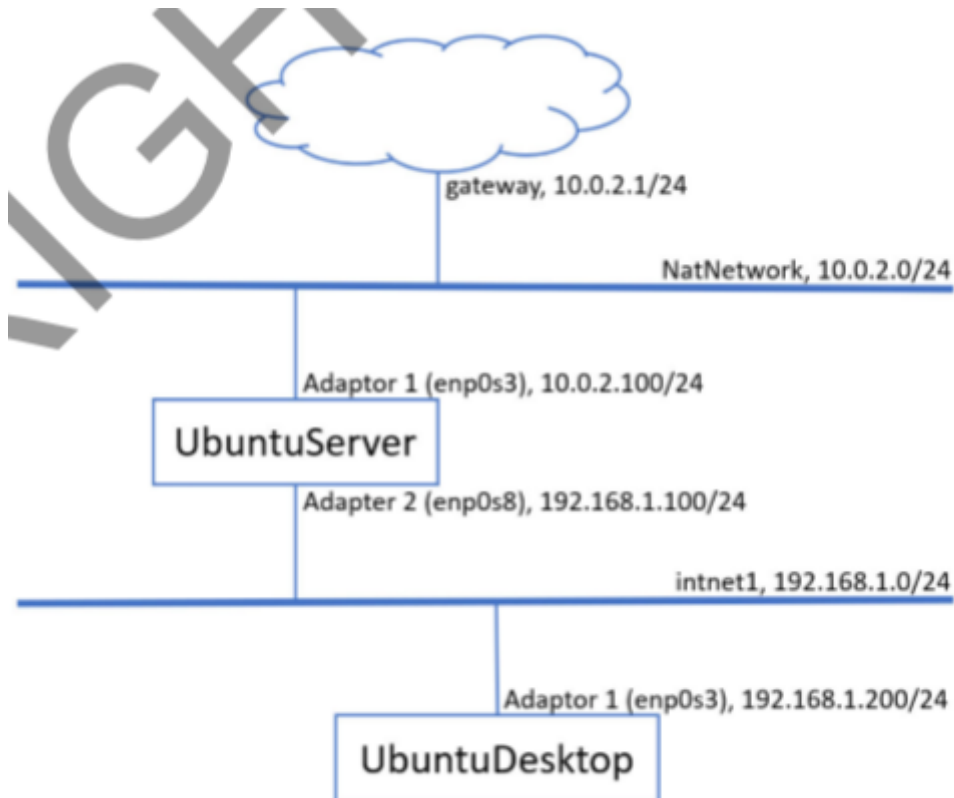
errors: No known data errors

## Lab8

路由设置。桌面连不上网 `iptables -F`

## 添加内网

为服务器增加一个网卡，内网连接；将桌面网卡改为内网连接，然后将服务器与桌面通过内网相连。



## 设置网卡

命令

- 1 `ip link` #查找网卡名
- 2 `netplan apply` #更新网络设置
- 3 `ping` #使用ip地址连接其他主机
- 4 `dig` #尝试获取网址或IP信息

## 查找网卡名

```
abc123@server:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
   link/ether 08:00:27:f5:4e:33 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
   link/ether 08:00:27:85:e4:e9 brd ff:ff:ff:ff:ff:ff
```

两张网卡分别为 `enp0s3` `enp0s8`，需要修改s8

按照Lab5的步骤3（设置静态IP）进行IP修改（在s3后面添加s8的信息即可）

```

1 network:
2     ethernets:
3         enp0s8:
4             addresses:
5                 - 192.186.1.100/24
6             gateway4: 10.0.2.1
7             nameservers:
8                 addresses: [10.0.2.100]
9             optional: true
10    version: 2

```

网关地址指向能联网的方向的下一跳地址，不是指网卡连接的网络的下一跳地址。

主机的网关应该是相连路由器的接口

### ping结果

服务器ping桌面

```

abc123@server:~$ ping 192.168.1.200
PING 192.168.1.200 (192.168.1.200) 56(84) bytes of data.
64 bytes from 192.168.1.200: icmp_seq=1 ttl=64 time=0.366 ms
64 bytes from 192.168.1.200: icmp_seq=2 ttl=64 time=0.878 ms
64 bytes from 192.168.1.200: icmp_seq=3 ttl=64 time=0.467 ms
64 bytes from 192.168.1.200: icmp_seq=4 ttl=64 time=0.641 ms
^C
--- 192.168.1.200 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 0.366/0.588/0.878/0.194 ms

```

桌面ping服务器

```

vana@desktop:~$ ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.592 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.588 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.621 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=0.532 ms
^C
--- 192.168.1.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.532/0.583/0.621/0.036 ms

```

## 设置NAT路由

### 桌面ping百度

```

vana@desktop:~$ ping www.baidu.com
PING www.baidu.com (39.156.68.79) 56(84) bytes of data.
64 bytes from www.baidu.com (39.156.68.79): icmp_seq=1 ttl=43 time=28.6 ms
64 bytes from www.baidu.com (39.156.68.79): icmp_seq=2 ttl=43 time=30.1 ms
^C
--- www.baidu.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 28.675/29.423/30.171/0.748 ms

```

## 重新配置DNS服务器

```

1 ;
2 ; This is the forward zone for abc123.test internal domain ;

```

```

3 ;
4 $TTL      604800
5 @         IN      SOA      server.abc123.test. asl.seu.edu.cn. (
6                               2019112001      ; Serial
7                               604800          ; Refresh
8                               86400           ; Retry
9                               2419200         ; Expire
10                              604800 )         ; Negative Cache TTL
11 ;
12 @         IN      NS       server.abc123.test.
13 @         IN      A        127.0.0.1
14 @         IN      AAAA     ::1
15
16 gateway   IN      A        10.0.2.1
17 server     IN      A        10.0.2.100
18 desktop    IN      A        192.168.1.200 #这里修改
19
20 dns        IN      CNAME    server
21 www        IN      CNAME    server

```

### 服务器通过域名ping桌面

```

abc123@server:~$ ping -c 4 desktop.abc123.test
PING desktop.abc123.test (192.168.1.200) 56(84) bytes of data.
64 bytes from 192.168.1.200 (192.168.1.200): icmp_seq=1 ttl=64 time=0.366 ms
64 bytes from 192.168.1.200 (192.168.1.200): icmp_seq=2 ttl=64 time=0.485 ms
64 bytes from 192.168.1.200 (192.168.1.200): icmp_seq=3 ttl=64 time=0.623 ms
64 bytes from 192.168.1.200 (192.168.1.200): icmp_seq=4 ttl=64 time=0.445 ms

--- desktop.abc123.test ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 0.366/0.479/0.623/0.096 ms

```

### 桌面通过域名ping服务器

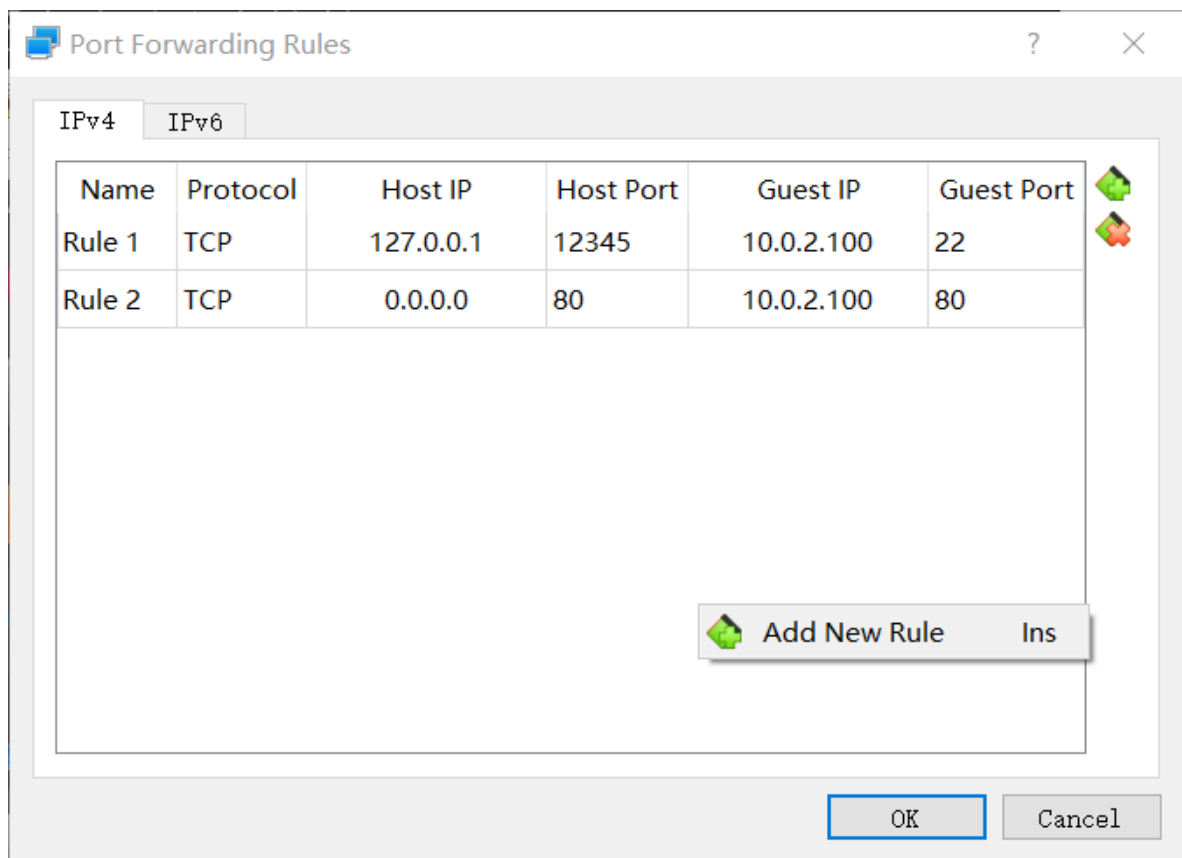
```

vana@desktop:~$ ping -c 4 server.abc123.test
PING server.abc123.test (10.0.2.100) 56(84) bytes of data.
64 bytes from server.abc123.test (10.0.2.100): icmp_seq=1 ttl=64 time=0.482 ms
64 bytes from server.abc123.test (10.0.2.100): icmp_seq=2 ttl=64 time=0.900 ms
64 bytes from server.abc123.test (10.0.2.100): icmp_seq=3 ttl=64 time=0.439 ms
64 bytes from server.abc123.test (10.0.2.100): icmp_seq=4 ttl=64 time=0.829 ms

--- server.abc123.test ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
rtt min/avg/max/mdev = 0.439/0.662/0.900/0.205 ms

```

## 在主机上启用端口转发



从主机访问dokuwiki

