Assignment Information			
Name:	Mario Morales	Assignment:	Project 5
Date Submitted:		Course Section:	
Course:	COSN 215		

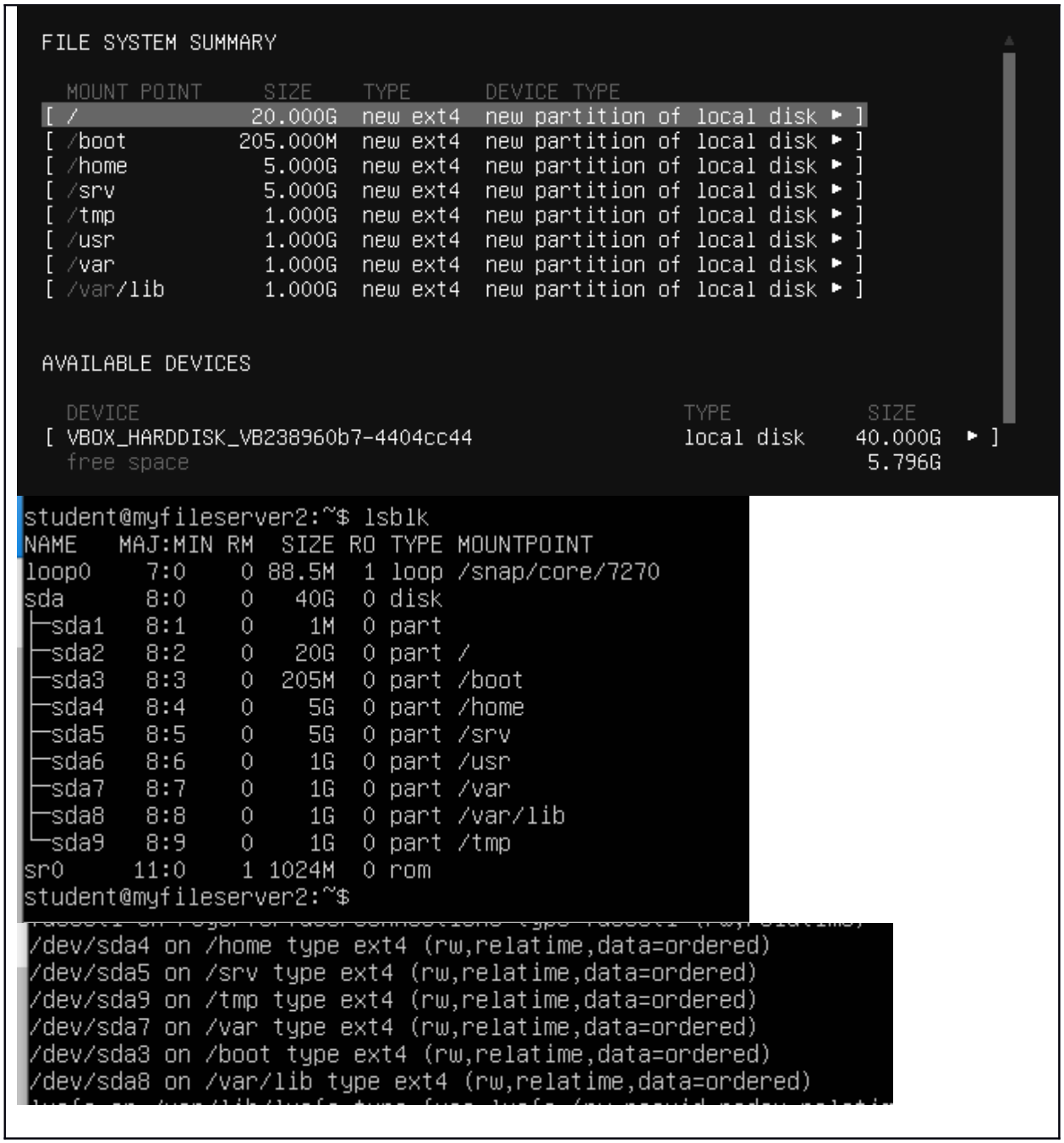
Phase 0 – Linux File System

YOU FOOL!!
You have a server, but you did not partition it in a flexible manner so that you can secure the file system. Take a look at the Linux File System Hierarchical Standard and decide which root-level folders should be their own partitions. What special properties (read-only, encryption, ACL, no-exec, quota, networked) should that partition have?

Directory	Partition(yes/no)	Properties
/bin	No	
/boot	yes	readonly
/cdrom	no	
/dev	no	
/etc	no	
/home	yes	Encryption, networked, quota
/lib	no	
/lib64	no	
/lost+f..	no	
/media	no	
/mnt	no	
/opt	no	
/proc	no	
/root	yes	
/run	no	
/sbin	no	
/snap	no	
/srv	no	
/sys	no	
/tmp	yes	noexec
/usb	no	
/var	yes	Networked, quota

Have your instructor verify before moving on.

Create a new instance of Ubuntu Server named **myFileServer** with the partition scheme you described above. User **student/P@ssw0rd** as the username/password. Verify it with **lsblk** and **mount:**



Find the bash executable. What iNode number it is? When was it last accessed/modified/changed? What does each time mean?

```
File  Machine  View  Input  Devices  Help
student@myfileserver2:~$ ls -id /bin/bash
262172 /bin/bash
student@myfileserver2:~$ stat /bin/bash
  File: /bin/bash
  Size: 1113504         Blocks: 2176       IO Block: 4096   regular file
Device: 802h/2050d      Inode: 262172      Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2019-12-04 00:36:42.365574613 +0000
Modify: 2019-06-06 22:28:15.000000000 +0000
Change: 2019-12-04 00:32:23.341574570 +0000
 Birth: -
student@myfileserver2:~$
```

Phase 1 – Set up file system

Explain the relationship between mount and fstab:

-These partitions and file systems can be listed just issuing `mount`
-Every operating system has a file system table, in Linux fstab happens to be that file.

Based on your partitioning scheme above, which partitions should be mounted noexec? read-only?

Property	Partition
noexec	/tmp
read-only	/boot

Make that happen <u>permanently</u> on **myFileServer** and document the changes here:

```
student@myfileserver:~$ sudo nano /etc/fstab

student@myfileserver2:~$ cat /etc/fstab
UUID=7a0fd887-e3aa-4581-ba0b-d05e8b5e5e90 / ext4 defaults 0 0
UUID=058d4693-4e6f-436c-9e1a-5fee3a71bed5 /boot ext4 ro 0 0
UUID=32f71571-9d11-4bdd-bb22-0220e5a407ea /home ext4 defaults 0 0
UUID=c86d4dd4-82cc-4780-b1b1-fe5f442facb1 /srv ext4 defaults 0 0
UUID=c46ebce4-e213-4efd-be52-c816cf9a3848 /usr ext4 defaults 0 0
UUID=b2b30e71-38ee-4482-a28f-b53b2b3f6b06 /var ext4 defaults 0 0
UUID=91531f6f-f1ed-426e-97a5-466bd08b2526 /var/lib ext4 defaults 0 0
UUID=674d2b15-81f6-4a48-9878-080621ce2a14 /tmp ext4 noexec 0 0
/swap.img       none    swap    sw      0       0
student@myfileserver2:~$ sudo mount -o remount /boot
student@myfileserver2:~$ sudo mount -o remount /tmp
```

Compare and contrast ACLs and POSIX permissions:

For any share point or shared folder or file, POSIX permissions allow you to set permissions only for the Owner, one Group, and Others. ACLs give you the additional option to set permissions for multiple individuals and multiple groups for a shared item. ACLs also have more types of permissions.

We want to make sure that unauthorized people cannot run scripts; that is a huge potential security flaw. What scripting tools exist on your current Linux system:

/bin/bash

Create a new group called **noscripty**. Using ACLs lock them out of using current scripting tools. Document your steps here:

```
student@myfileserver2:~$ sudo groupadd noscripting
student@myfileserver2:~$ _

student@myfileserver2:~$ sudo setfacl -m g:noscripting:--- /bin/bash
student@myfileserver2:~$ getfacl /bin/bash
getfacl: Removing leading '/' from absolute path names
# file: bin/bash
# owner: root
# group: root
user::rwx
group::r-x
group:noscripting:---
mask::r-x
other::r-x

student@myfileserver2:~$ _
```

While this is better than nothing, what is a way that a member of **noscripty** could still execute scripts?

Noscripty could run a different type of language for scripting

Phase 2 – Setup Networked File System

Setup a Samba share for your /home directory of **myFileServer.** Create Samba user accounts as appropriate. Document the process here:

```
student@myfileserver2:~$ sudo apt-get install samba cifs-utils_
```

```
#======================= Share Definitions =======================
#Mario's Shares
[MyShare]
comment = MARIOS SHARES
path = /home
read only = no
guest ok = yes

student@myfileserver2:~$ sudo /etc/init.d/smbd restart
[ ok ] Restarting smbd (via systemctl): smbd.service.
student@myfileserver2:~$
```

XXTRA CRED17

Normaly your **myClient** computer would mount the /home directory of **myFileServer** so users could access their home directories in one central location that you as the administrator could manage. However that is complicated because we didn't create separate partions with **myClient** when we started.

For extra credit, copy the user files from **myClient** to an appropriate directory in **myFileServer** and have **myClient** mount the /home directory of **myFileServer.** Document the process here:

```
student@student-VirtualBox:~$ smbclient -L 192.168.1.107 -U student
Unable to initialize messaging context
Enter WORKGROUP\student's password:

        Sharename       Type      Comment
        ---------       ----      -------
        MyShare         Disk      MARIOS SHARES
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (myfileserver2 server (Samba, Ubu
ntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            MYFILESERVER2
student@student-VirtualBox:~$
```

Terminal

Phase 3 – Quotas

Do file system quotas make Linux more secure? Explain:

Quotas are used to limit the amount of disk space a user or group can use on a filesystem. Without such limits, a user could fill up the machine's disk and cause problems for other

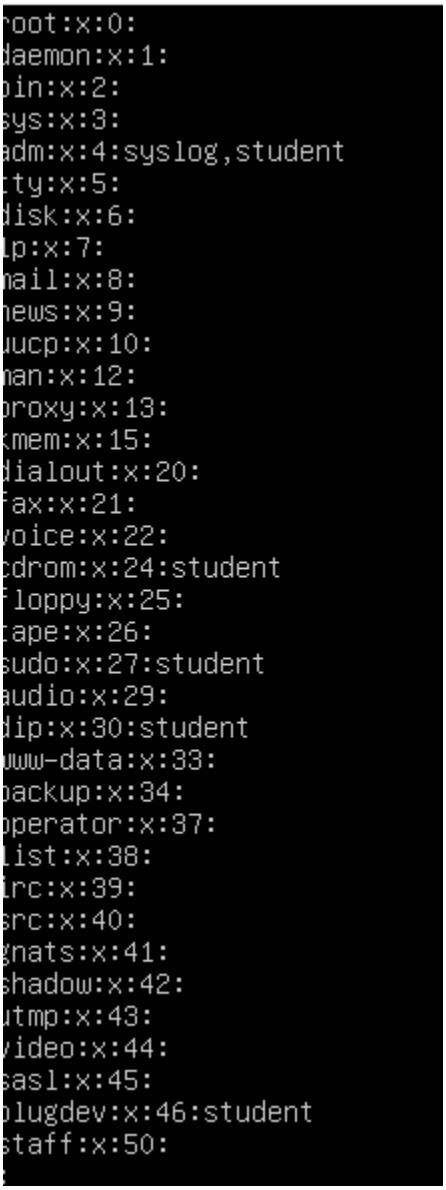
users and services.

Based on your partitioning scheme above, which partitions should have quotas enabled?

Property	Partition
quota	All of them

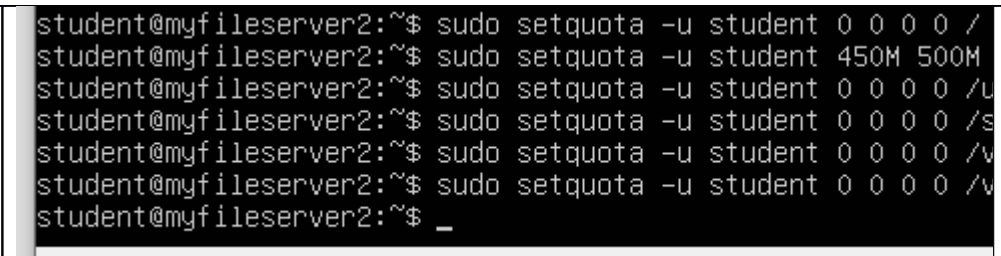
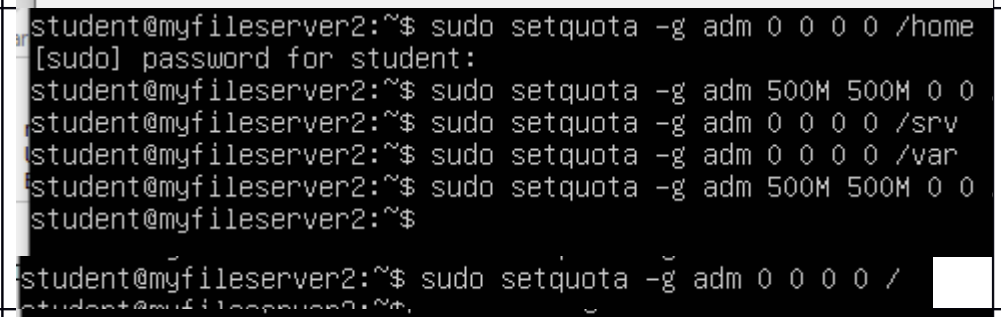
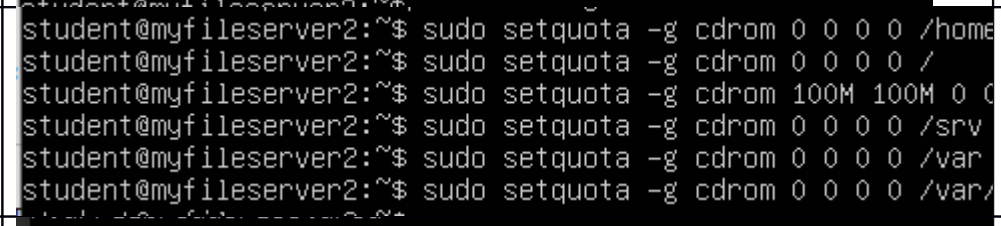
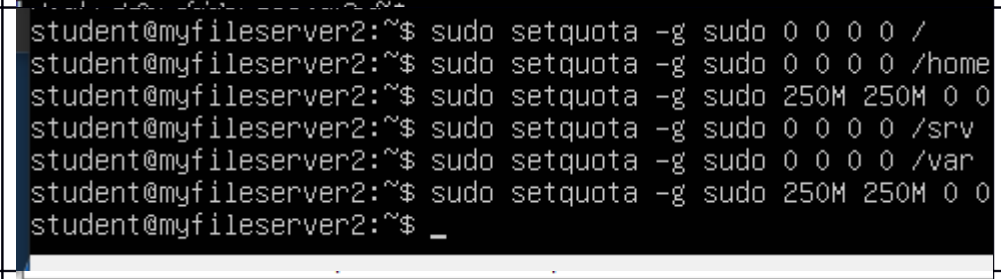
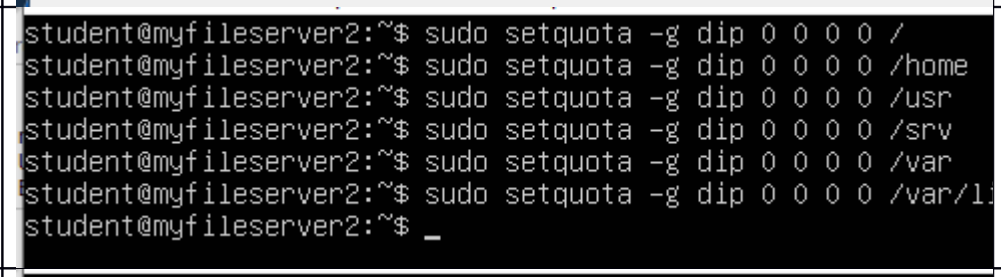
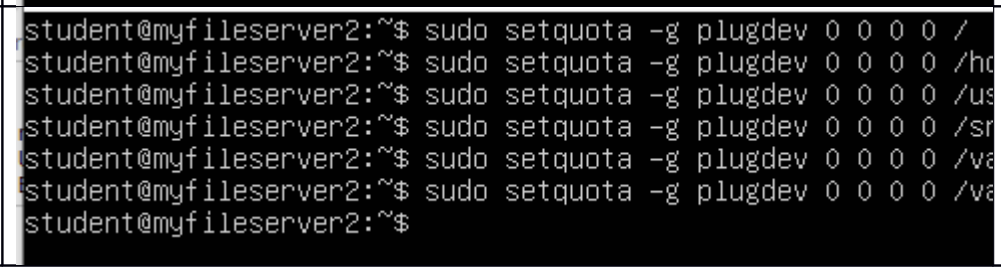
Using the appropriate Linux tools find all users/groups:

```
student@myfileserver2:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
student:x:1000:1000:student:/home/student:/bin/bash
student@myfileserver2:~$ cat /etc/passwd
```



For each user and/or group, what is an appropriate quota amount for each partition you listed above:

User/Group	Partition	Quota
root	95 sudo setquota -u root 0 0 0 0 /home 96 sudo setquota -u root 0 0 0 0 /usr 97 sudo setquota -u root 0 0 0 0 /srv 98 sudo setquota -u root 0 0 0 0 /var student@myfileserver2:~$ sudo setquota -u root 9G 10G 0 0 /	10G

student		500 M
Adm		250 M
Cdrom		100 M
Sudo		250 M
Dip		0
plugdev		0

Implement your quota scheme and document the process here:

```
27  sudo apt update
28  sudo apt install quota
29  clear
30  sudo apt install quota
31  clear
32  quota --version
33  clear
34  find /lib/modules/'uname -r' -type f -name '*quota_v*.ko*'
35  sudo apt install linux-image-extra-virtual
36  find /lib/modules/'uname -r' -type f -name '*quota_v*.ko*'
37  find /lib/modules/`uname -r` -type f -name '*quota_v*.ko*'
38  clear
39  sudo nano /etc/fstab
40  sudo mount -o remount /
41  sudo mount -o remount /boot
42  sudo mount -o remount /tmp
43  clear
44  cat /proc/mounts | grep ' / '
45  clear
46  sudo quotacheck -ugm /
47  ls /
48  clear
49  sudo quotaon -v /
```

```
UUID=7a0fd887-e3aa-4581-ba0b-d05e8b5e5e90 / ext4 usrquota,grpquota 0 0
UUID=058d4693-4e6f-436c-9e1a-5fee3a71bed5 /boot ext4 ro 0 0
UUID=32f71571-9d11-4bdd-bb22-0220e5a407ea /home ext4 usrquota,grpquota 0 0
UUID=c86d4dd4-82cc-4780-b1b1-fe5f442facb1 /srv ext4 usrquota,grpquota 0 0
UUID=c46ebce4-e213-4efd-be52-c816cf9a3848 /usr ext4 usrquota,grpquota 0 0
UUID=b2b30e71-38ee-4482-a28f-b53b2b3f6b06 /var ext4 usrquota,grpquota 0 0
UUID=91531f6f-f1ed-426e-97a5-466bd08b2526 /var/lib ext4 usrquota,grpquota 0 0
UUID=674d2b15-81f6-4a48-9878-080621ce2a14 /tmp ext4 noexec_0 0
/swap.img       none    swap    sw      0       0
```

```
80  sudo quotacheck -ugm /srv
81  sudo quotacheck -ugm /usr
82  sudo quotacheck -ugm /var
83  sudo quotacheck -ugm /var/lib
84  sudo quotacheck -ugm /tmp
```

Use locate to find where groups/users are used
Implment quotas on those partitions see above

Phase 4 – Remote logging

How might remote logging make your Linux network more secure:

in the event of an intrusion, this provides an off site server where log files have been untouched by any attacker. this may be the only way to figure out what has happenend to the system, and aids in identifying the security hole, repairing it, and preventing future intrusions by such means. this helps a security analyst decide whether or not the entire system has been compromised, or just part of it. and this leads me to number three

Configure rsyslog server on **myFileServer** to handle the networks logs, then configure **myServer** to store its log files there. Document the process here:

```
student@myfileserver2:~$ sudo su
root@myfileserver2:/home/student# sudo su -
root@myfileserver2:~# apt update && apt install rsyslog
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 252 kB in 1s (221 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
49 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
rsyslog is already the newest version (8.32.0-1ubuntu4).
rsyslog set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
root@myfileserver2:~#
```

```
root@myfileserver2:~# systemctl start rsyslog
root@myfileserver2:~# systemctl enable rsyslog
Synchronizing state of rsyslog.service with SysV service script with /lib/systemd/systemd-sysv-ins
ll.
Executing: /lib/systemd/systemd-sysv-install enable rsyslog
root@myfileserver2:~# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-12-04 02:06:03 UTC; 1h 6min ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
 Main PID: 980 (rsyslogd)
    Tasks: 4 (limit: 4660)
   CGroup: /system.slice/rsyslog.service
           └─980 /usr/sbin/rsyslogd -n

Dec 04 02:06:02 myfileserver2 systemd[1]: Starting System Logging Service...
Dec 04 02:06:02 myfileserver2 rsyslogd[980]: imuxsock: Acquired UNIX socket '/run/systemd/journal/s
Dec 04 02:06:02 myfileserver2 rsyslogd[980]: rsyslogd's groupid changed to 106
Dec 04 02:06:02 myfileserver2 rsyslogd[980]: rsyslogd's userid changed to 102
Dec 04 02:06:02 myfileserver2 rsyslogd[980]:  [origin software="rsyslogd" swVersion="8.32.0" x-pid=
Dec 04 02:06:03 myfileserver2 systemd[1]: Started System Logging Service.
lines 1-16/16 (END)
```

```
# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

##rules for processing the remote logs
$template RemoteLogs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?RemoteLogs
& ~
```

```
root@myfileserver2:~# systemctl restart rsyslog
root@myfileserver2:~# ss -tulnp | grep "rsyslog"
tcp   LISTEN  0       25                    0.0.0.0:514         0.0.0.0:*      users:(("rsyslogd"
pid=3209,fd=5))
tcp   LISTEN  0       25                       [::]:514            [::]:*      users:(("rsyslogd"
pid=3209,fd=6))
root@myfileserver2:~#
```

```
root@myfileserver2:~# sudo ufw allow 514/tcp
Rules updated
Rules updated (v6)
```

On myServer

```
# forwarding rules, duplicae the whole block!
# Remote logging (we use TCP for reliable delivery)
#
#An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.[  OK  ] Started Daily apt upgrad
#$ActionQueueFileName fwdRule1 # unique name prefix for spool files
#$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
#$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
#$ActionQueueType LinkedList   # run asynchronously
#$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514
*.* @@192.168.1.107:514

# ### end of the forwarding rule ###
```