Linux Assignments

Question section

1. **Password-less ssh setup between 2 ec2 instances?**
2. **Create and delete user?**
3. **(self-made) Elevate a newly added user privilege to full root access?**
4. **(self-made) Elevate a newly added user privilege to specific sudo command access only?**
5. **What is swap memory without attaching new volume? Create/edit swap partition.**
6. **Change file ownership.**
7. **Local/HTTP hosted repo setup for yum update without internet?**
8. **FTP setup?**
9. **What is temporary user level and global level environment/system variables?**
10. **Setup a mount point and mount temporary and permanent.**
11. **What is soft and hard link? Set them up.**
12. **What is iptables?**
13. **Setup a cronjob.**
14. **NFS setup?**
15. **Change hostname permanently?**
16. **SFTP setup?**
17. **Samba setup?**
18.

NFS ref:
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/nfs-serverconfig

https://www.techrunnr.com/setup-nfs-server-on-amazon-ec2/

NOTE: OS used - RedHat OS Family

Answer section

Q-12. Setup hostnames permanently?

```
[root@srv ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
172.31.82.141 srv.windows.bigscam
172.31.88.207 clnt.windows.bigscam
```

```
[root@srv ~]# cat /etc/hostname
srv.windows.bigscam
[root@srv ~]# reboot
[root@srv ~]# hostname -f
srv.windows.bigscam
```

Q-1. Setup passwordless ssh between two ec2 instances?

```
Let’s make passwordless ssh between ec2 users
In Srv:
$ ssh-keygen
$ cd .ssh ; ls
authorized_keys  id_rsa  id_rsa.pub
copy content of srv’s id_rsa.pub and append it to clnt’s
~/.ssh/authorized_keys
And vice versa (i.e., from clnt -> srv, same step)
tests/verify:
from srv -> clnt:
[ec2-user@srv .ssh]$ ssh ec2-user@clnt.linux.com
The authenticity of host 'clnt.linux.com (172.31.88.207)'
can't be established.
ECDSA key fingerprint is
SHA256:GwH+kZzX7cOf5pjPtQV+wUgyDZnHLX29OJVrUdP0jMw.
ECDSA key fingerprint is
MD5:1c:97:80:4c:71:76:0a:7d:d5:c4:e0:bf:c2:55:47:d3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'clnt.linux.com,172.31.88.207'
(ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@clnt ~]$
from clnt -> srv
[ec2-user@clnt ~]$ ssh ec2-user@srv.linux.com
The authenticity of host 'srv.linux.com (172.31.82.141)' can't
be established.
ECDSA key fingerprint is
SHA256:A8/ks4E/rQ0t5Hqdoyw8q3VH/yo2xiF1JWiDFijsu5M.
ECDSA key fingerprint is
MD5:7e:0b:34:1f:b2:83:54:5d:e9:a4:dc:61:62:ea:7d:31.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'srv.linux.com,172.31.82.141'
(ECDSA) to the list of known hosts.
```

```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@srv ~]$
```

Q.2. Create and delete user?

```
[root@srv ec2-user]# id testuser
id: testuser: no such user
[root@srv ec2-user]# useradd testuser
[root@srv ec2-user]# passwd testuser
Changing password for user testuser.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
test/verify:
[root@srv ec2-user]# ls /home/
ec2-user  testuser
[root@srv ec2-user]# tail -1 /etc/passwd
testuser:x:1001:1001::/home/testuser:/bin/bash
[root@srv ec2-user]# sudo su - testuser
[testuser@srv ~]$ pwd
/home/testuser
[testuser@srv ~]$ touch abc.txt
[testuser@srv ~]$ ls
abc.txt
```

Q -4. Change file ownership?

```
[root@srv testuser]# pwd
/home/testuser
[root@srv testuser]# echo 'hello from root in /home/testuser'
>  hello.txt
[root@srv testuser]# cat hello.txt
hello from root in /home/testuser
[root@srv testuser]# ll
total 4
-rw-r--r-- 1 root root 34 Jun 21 11:20 hello.txt
[root@srv testuser]# chown testuser:testuser hello.txt
[root@srv ~]$ sudo su - testuser
[root@srv testuser]# ll
total 4
-rw-r--r-- 1 testuser testuser 34 Jun 21 11:20 hello.txt
[testuser@srv ~]$ cat hello.txt
hello from root in /home/testuser
```

Q-4.B. Change file and fodler ownership?

```
$ chown <username_who_gets_ownership>  <file>
$ chown -R <username_who_gets_ownership>  <folder>
```

Q-14. sftp setup?

```
setup:
======
srv:
----
# yum install openssh-server
# mkdir -p /var/sftp/data
# groupadd sftpusers
# useradd -d /var/sftp/data/sftp-user1 -s /sbin/nologin -g
sftpusers sftp-user1
# passwd sftp-user1    <-- redhat
# cd /var/sftp
# ll -d data
# chown -R root:root data
# chmod 700 data
# ls -l data/
output: root root.... sftp-user1
# chown -R root:sftpusers sftp-user1
# chmod 750 sftp-user1
# cd sftp-user1
# mkdir upload
# chown -R sftp-user1:sftusers upload
# chmod 700 upload

# cd /etc/ssh
# vi sshd_config
Port 2222
PasswordAuthentication yes

Match Group sftpusers    <-- match group name for sftp
     ChrootDirectory %h     <-- %h for home dir
     ForceCommand internal-sftp    <-- it's a command only
non-sudo commands can run
     AllowTcpForwarding no    <-- don't allow sftp's tcp
session forwarding - security reasons
     X11Forwarding no  <-- x11/gui forwarding disabled -
security reasons

save and exit the sshd_config file.

# service sshd restart
```

optional troubleshootings:
journalctl -xe

optional SELinux troubleshoot/allow sftp through SELINUX policy, without disabling selinux:
semanage port -a -t ssh_port_t -p tcp

service sshd restart

note: what the hell is internal-sftp command doing here?
https://serverfault.com/questions/660160/openssh-difference-between-internal-sftp-and-sftp-server

verify:
=======
srv:
----
cd upload ; echo 'hello from srv' > srv.txt

clnt:

$ sftp -oPort=2222 sftp-user1@serverIP
provide ssh password
sftp> ls
output: upload

$ echo 'hello from client' > clnt.txt

$ sftp -oPort=2222 sftp-user1@serverIP
sftp> cd uplaod
sftp> put client.txt
Uploading.......
sftp>
sftp> !cat client.txt
hello from client
sftp> get srv.txt
Fetching /upload/srv.txt to srv.txt
/upload/srv.txt   100%
18   3.1KB/s   00:00
sftp> exit
[ec2-user@clnt ~]$ ls
client.txt srv.txt
[ec2-user@clnt ~]$ cat srv.txt
hello from server

note: run sftp> !commands

```
srv:
----
# cd upload/
# ls
output: client.txt ,  srv.txt
```

Q-9. What is soft and hard link? Set them up.

Q-10. What is iptables?

All about iptables: https://medium.com/skilluped/what-is-iptables-and-how-to-use-it-781818422e52

iptables states: https://serverfault.com/questions/371316/iptables-difference-between-new-established-and-related-packets

basic intro: https://docs.rackspace.com/support/how-to/allow-web-traffic-in-iptables/

25-most used iptables commands: https://crm.vpscheap.net/index.php?rp=%2Fknowledgebase%2F29%2F25-Most-Frequently-Used-Linux-IPTables-Rules-Examples.html

```
iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -m conntrack --ctstate
ESTABLISHED -j ACCEPT
```

Q-3. Swap memory/partition configuration, from existing disk space/without mounting a new volume?

https://aws.amazon.com/premiumsupport/knowledge-center/ec2-memory-swap-file/

```
$ sudo dd if=/dev/zero of=/swapfile bs=128M count=8
$ sudo chmod 600 /swapfile
$ sudo mkswap /swapfile
$ file swapfile
$ lsblk -d -fs /swapfile
output: error- /swapfile: not a block device

$ sudo blkid /swapfile
/swapfile: UUID="7089e460-a8ea-468f-9c2d-8c9ece6e3ba4"
TYPE="swap"

$ sudo cat >> /etc/fstab
```

/swapfile swap swap deafults 0 0 <-- UUID for swap not gonna work.

$ init 6
verify: $ free -mh

temporary swap: # swapon -a ; # swapon -s # free -m

Q-12: NFS setup?
NFS setup:

SERVER conf., as root:
disable SELINUX: # setenforce 0 ; getenforce
yum install nfs-utils -y
systemctl start,enable,status nfs-service.service

troubleshoot: # rpcinfo -p | grep nfs

mkdir -p /mnt/shared/srv <--hostname of server
chown -R nobody: /mnt/shared/srv
chmod -R 777 /mnt/shared/srv/
systemctl restart nfs-server.service

vi /etc/exports <-- to export the nfs share, so that the client systems can access it, use subnetIPaddr to include multiple clients
/mnt/shared/srv CLIENT-IP(rw,sync,no_all_squash,root_squash)

use the exportfs command to export the shared folder:
exportfs -arv
output: exporting ClientIP:/mnt/shared/srv
exportfs -s

(optional steps: firewall)
iptables -F
iptables -L
OR
allow nfs servcie through firewall and reload the firewalld:
yum install firewalld
systemctl start,enable,status firewalld
firewall-cmd --permanent --add-service=nfs
output: success
firewall-cmd --permanent -add-service=rpcbind
firewall-cmd --permanent -add-service=mountd
firewall-cmd --reload
==========

client conf. Starts - as root:

```
# setenforce 0 ; getenforce
(sometimes the following installation may not neccessary, i
had only nfs-utils pkg in client system pre-installed) # yum
install nfs-utils nfs4-acl-tools
# showmount -e SERVERIP
output: Export list for serverIP
/mnt/shared/srv clientIP

# mkdir -p /mnt/shared/clnt
# ls /mnt
# mount -t nfs serverIP:/mnt/shared/srv /mnt/shared/clnt
# mount | grep -i nfs
Output: sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs
(rw,relatime)
SERVERip:/mnt/shared/srv on /mnt/shared/clnt type nfs4
(rw,relatime,vers=4.1,rsize=131072,wsize=131072,namlen=255,har
d,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.31.21.1
10,local_lock=none,addr=172.31.29.168)

# permannent mount - works after reboot
# vi /etc/fstab
serverIP:/mnt/shared/srv  /mnt/shared/clnt nfs defaults 0 0
==================================
Testing:
--------
In client:
# cd /mnt/shared/clnt/
# cat > client_nfsfile
Hello from client!
^C

In server:
# ll /mnt/shared/srv
# cat client_nfsfile
output: Hello from client!
# cat > server_nfsfile
Hello from server!
^C

In client:
# /mnt/shared/clnt
# cat server_nfsfile
output: Hello from server!
```

Now, you can share files between nfs server and client, and do anything you want with the files.

Q-4. Elevate/Escalate a newly added user's privilege to specific sudo command access only?
https://www.techrepublic.com/article/how-to-quickly-give-users-sudo-privileges-in-linux/

```
# useradd half-admin
# passwd half-admin
# sudo su – half-admin

admin $ sudo yum update
output: [sudo] password for half-admin:
half-admin is not in the sudoers file.  This incident will be reported.
You have mail in /var/spool/mail/root

# cat /var/spool/mail/root        ← security threat mail spool log/report/notification
From root@srv.localdomain  Thu Jun 23 16:57:52 2022
Return-Path: <root@srv.localdomain>
X-Original-To: root
Delivered-To: root@srv.localdomain
Received: by ip-172-31-38-76.localdomain (Postfix, from userid 0)
     id 3F49D8C1946; Thu, 23 Jun 2022 16:57:52 +0000 (UTC)
To: root@srv.localdomain
From: half-admin@srv.localdomain
Auto-Submitted: auto-generated
Subject: *** SECURITY information for srv ***
Message-Id: <20220623165752.3F49D8C1946@srv.localdomain>
Date: Thu, 23 Jun 2022 16:57:52 +0000 (UTC)

srv : Jun 23 16:57:52 : half-admin : user NOT in sudoers ;
TTY=pts/0 ; PWD=/home/half-admin ; USER=root ;
COMMAND=/bin/yum update
```

Now let's make half-admin an half admin user to be able to run yum commands only:

```
# visudo
Cmnd_Alias YUM_UPDATE = /usr/bin/yum      ← in appropriate section
half-admin ALL=(ALL) YUM_UPDATE      ← at the EOF
```

Test:

```
# sudo su - half-admin
```

```
Last login: Thu Jun 23 16:57:46 UTC 2022 on pts/0
[half-admin@srv ~]$ sudo yum update
[sudo] password for half-admin:
Loaded plugins: extras_suggestions, langpacks, priorities,
update-motd
amzn2-core                                                | 3.7
kB     00:00
No packages marked for update
[half-admin@srv ~]$ echo 'hi from half-admin' > abc
[half-admin@srv ~]$ sudo cp -v abc /abc
[sudo] password for half-admin:
Sorry, user half-admin is not allowed to execute '/bin/cp -v
abc /abc' as root on srv.
  ← no security notifications/spool mail received – crazy!
```

Q-3. Elevate a newly added user privilege to full root access?

```
# visudo    or   #  vi /etc/sudoers    ← visudo is recommended
<username>     ALL=(ALL)     ALL            ← at EOF
```

```
Verify:
[fulladmin@srv ~]$ yum update                ← works
[fulladmin@srv ~]$ echo 'hi from fulladmin' > hi
[fulladmin@srv ~]$ sudo cp -v hi /hi
‘hi’ -> ‘/hi’
[fulladmin@srv ~]$ cat /hi
hi from fulladmin
```

Q-18. FTP setup?

In srv:

```
# yum install vsftpd ftp -y
# vi /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
## Uncomment ##
ascii_upload_enable=YES
ascii_download_enable=YES
ftpd_banner=Welcome to UNIXMEN FTP service.
# Add at the end of this file
use_localtime=YES      <-- to avoid any possible upcoming NTP
syncing issues

# systemctl start,restart ,enable,status vsftpd.service
#useradd shareme ; passwd shareme
```

In client:

```
$ sudo yum install ftp -y
$ echo ‘hi from client’ > client.txt
```

```
[ec2-user@clnt ~]$ ftp
ftp> open srv.aws.com
Connected to srv.aws.com (172.31.34.51).
220 Welcome to ftp service hosted in LYNUX
Name (srv.aws.com:ec2-user): shareme
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put client.txt
local: client.txt remote: client.txt
227 Entering Passive Mode (172,31,34,51,40,101).
150 Ok to send data.
226 Transfer complete.
18 bytes sent in 0.000475 secs (37.89 Kbytes/sec)
In Srv: # cat /home/shareme/client.txt
```

Q-17. Samba setup? And (optionally) protect samba against sambacry vulnerability?

Scalable smb setup: https://linuxize.com/post/how-to-install-and-configure-samba-on-centos-7/

https://www.techrepublic.com/article/how-to-protect-samba-from-the-likes-of-the-sambacry-exploit/

simple smb setup: https://www.redhat.com/sysadmin/samba-file-sharing