

Task1:

Part1: LVM: Tuesday

1. Create disk from vm
2. Create partition
`# fdisk /dev/sdb`
3. Initialize partition as physical volume
`# pvcreate /dev/sdb1`
4. Create volume group (VG) with 16M extend size from physical volume (/dev/sdb1)
`# vgcreate -s 16M vg /dev/sdb1`
5. Create logical volume (lv) on volume group with extend size 50
`# lvcreate -l 50 vg -n lv`
6. Format logical volume as ext4 file system
`# mkfs -t ext4 /dev/vg/lv`
7. Go to mnt directory (directory for mount file system)
`# cd /mnt`
8. Create data directory
`mnt# mkdir data`
9. Go to fstab file
`# vi /etc/fstab`
10. Add this on the last line in fstab file to make mount
`/dev/vg/lv /mnt/data ext4 defaults 0 0`
11. Mount the logical volume under /mnt/data
`# mount -a`

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   19G  0 part
   ├─centos-root 253:0    0   17G  0 lvm  /
   └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb          8:16    0 40.8G  0 disk
├─sdb1       8:17    0 40.8G  0 part
│   └─vg-lv   253:2    0   800M  0 lvm  /mnt/data
sr0         11:0    1 1024M  0 rom
```

Part2: Users, groups and permissions: cd Tuesday

2.1)

1. Create user1

```
# useradd user1
```

2. Change uid for user1

```
# usermod -u 601 user1
```

3. Set password: redhat to user1

```
# Passwd user1
```

4. Make the user non-interactive (no ssh access to server)

```
# usermod -s /sbin/nologin user1
```

```
user1:x:601:1003::/home/user1:/sbin/nologin
user2:x:1002:1004::/home/user2:/bin/bash
user3:x:1003:1004::/home/user3:/bin/bash
```

```
[root@localhost ~]# usermod -s /sbin/nologin user1
[root@localhost ~]# usermod -s /sbin/nologin user1
usermod: no changes
[root@localhost ~]# su user1
This account is currently not available.
[root@localhost ~]#
```

2.2)

1. Create TrainingGroup

```
# groupadd TrainingGroup
```

2. Add user1 to TrainingGroup

```
# usermod -g TrainingGroup user1
```

```
[root@localhost ~]# groups user1
user1 : TrainingGroup
[root@localhost ~]#
```

```
[root@localhost ~]# id user1
uid=601(user1) gid=1003(TrainingGroup) groups=1003(TrainingGroup)
[root@localhost ~]# id user1 && finger user1
```

2.3)

1. Create adminGroup

```
# groupadd adminGroup
```

2. Create user2

```
# useradd user2
```

3. Set password to user2 (password: redhat)

```
# passwd user2
```

4. add user2 in group

```
# usermod -g adminGroup user2
```

5. Create user3

```
# useradd user3
```

6. Set password to user3 (password: redhat)

```
# passwd user3
```

7. add user3 in group

```
# usermod -g adminGroup user3
```

```
[root@localhost ~]# groups user2 user3
user2 : adminGroup
user3 : adminGroup
```

8. Give user3 root permission:

```
# setfacl -m u:user3:rwX /
```

Part3: SSH: Wednesday

1. In the first server:

- 1.1. Generate the ssh key

```
# ssh-keygen
```

- 1.2. Copy the value of key in the id_rsa.pub file in second server

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub root@10.10.10.30
```

Note: where the 10.10.10.30 is ip address for second server

- 1.3. Make connection to with second server

```
# ssh root@10.10.10.40
```

Part4: Permission: Tuesday

1. To copy fstab file to admin:

```
# cp /etc/fstab /var/tmp/
```

2. To change the owner of fstab file to adminGroup group to control permission and access for the fstab file

```
# chgrp adminGroup /var/tmp/fstab
```

3. Set user1 could read, write and modify it

```
# setfacl -m u:user1:rwX /var/tmp/fstab
```

4. Set user2 can't do any permission.

```
# setfacl -m u:user2:--- /var/tmp/fstab
```

```
[root@localhost tmp]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: adminGroup
user::rw-
user:user1:rwX
user:user2:---
group::r--
mask::rwX
other::r--
```

Part5: Permission: Tuesday

1. Change the mode to enforcing from config file

- 1.1. Open the selinux config file

```
# vi /etc/selinux/config
```

- 1.2. Change the SELINUX value to enforcing

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected.
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Part6: Bash script and processes: Wednesday

1. go to tmp directory

```
# cd tmp
```

2. Open crontab and edit on it

```
2.1. tmp# crontab -e
```

```
2.2. Write in file:
```

```
*/10 * * * * /tmp/myscript.sh
```

Note: this means the /tmp/myscript.sh file run every 10 min

3. Open myscript file to write on it the code I need to run

```
3.1. tmp# vi myscript.sh
```

```
3.2. Write this code:
```

```
#!/bin/bash
```

```
sleep 120&      Note: this line used to sleep this process 2min in the background
```

```
date
```

```
cat /root/file.txt
```

4. Change mode of script file to allowing you to run it as a script by executing

```
tmp# chmod +x myscript.sh
```

5. Run myscript file in the background

```
tmp# ./myscript.sh &
```

6. To show the process details

```
tmp# ps -u
```

7. To kill the process

```
tmp# Kill <PID for this process>
```

```

[root@localhost tmp]# ./myscript.sh
Sun Jun 18 17:47:21 EDT 2023
anasNimer
[root@localhost tmp]# ps -u
USER      PID %CPU %MEM    USZ    RSS TTY      STAT START   TIME COMMAND
root        441  0.0  0.1 191988 2388 tty1    S    17:23   0:00 su -
root        448  0.0  0.1 115548 2112 tty1    S    17:23   0:00 -bash
root        608  0.0  0.1 192036 2456 tty1    S    17:26   0:00 su - user2
root        756  0.0  0.1 191988 2392 tty1    S    17:27   0:00 su -
root        763  0.0  0.1 115544 2084 tty1    S    17:27   0:00 -bash
root        835  0.0  0.1 192036 2456 tty1    S    17:28   0:00 su - user1
root       1131  0.0  0.1 191988 2388 tty1    S    17:34   0:00 su -
root       1139  0.0  0.1 115548 2120 tty1    S    17:34   0:00 -bash
root       1600  0.0  0.0 115544 1484 tty1    Ss   Jun15   0:00 -bash
root       1835  0.0  0.0 108056   360 tty1    S    17:47   0:00 sleep 120
root       1844  0.0  0.0 155452 1872 tty1    R+   17:47   0:00 ps -u
root      31378  0.0  0.1 192036 2460 tty1    S    16:57   0:00 su - user2
root      31749  0.0  0.1 191988 2392 tty1    S    17:03   0:00 su -
root      31760  0.0  0.1 115544 2096 tty1    S    17:03   0:00 -bash
root      31910  0.0  0.1 192036 2456 tty1    S    17:06   0:00 su - user2
root      32285  0.0  0.1 191988 2388 tty1    S    17:12   0:00 su - user3
root      32506  0.0  0.1 191988 2392 tty1    S    17:16   0:00 su -
root      32512  0.0  0.1 115544 2084 tty1    S    17:16   0:00 -bash
root      32554  0.0  0.1 192036 2456 tty1    S    17:17   0:00 su - user3
[root@localhost tmp]# kill 1835

```

Part7: Yum Repo

1. Install tmux

```
# yum install tmux
```

2. Install httpd & mysql

```
2.1. # yum install httpd
```

```
2.2. # yum install mysql-server
```

3. Create local yum repository

```
3.1 . # yum install createrepo
```

```
3.2. # yum install yum-utils
```

```
3.3. # cd /var/www/html
```

```
3.4. # mkdir repo
```

```
3.5. # vi /etc/yum.repos.d/local.repo
```

and write on it:

```
[local]
```

```
name= local repo
```

```
baseurl=file:/// root/repo
```

```
enabled=1
```

```
gpgcheck=0
```

4. Install the packages from url

```
# wget https://repo.zabbix.com/zabbix/4.4/rehl/7/x86_64/zabbix-agent-4.4.10-1.el7.x86_64.rpm
```

Note: install all type of packages 4.4.10-1.el7

5. Create repository on the current directory
createrepo .
6. Disable all other repositories and keep only the new repo
6.1. # yum-config-manager --disable /*
6.2. #yum-config-manager --enable repo
7. Install zabbix rpms from the new repo
yum install zabbix zabbix-web php zabbix-server zabbix-agent

Part8: Network management: Wednesday

1. Add port 443,80 and make the changes permanent (active every time not temporary)
1.1. firewall-cmd --zone=public --add-port=443 --permanent
1.2. firewall-cmd --zone=public --add-port=80 --permanent
2. you need to reload firewall to make the permanent active
firewall-cmd --reload
3. Add ssh service
firewall-cmd --zone=public --add-service=ssh
4. Reload the changes
firewall-cmd --reload
5. Block ssh connection
firewall-cmd --add-rich-rule='rule family=ipv4 source address="172.20.10.4" service name="ssh" reject'
6. **In another VM test the ssh block connection**
ssh 172.20.10.3

Note: the output is connection refused

```
ssh: connect to host 172.20.10.3 port 22: Connection refused
you have new mail in /var/spool/mail/root
```

Part9: Cronjob: Wednesday

1. Go to tmp directory

```
# cd tmp
```

2. Open crontab and edit on it

- 2.1. tmp# crontab -e

- 2.2. Write in file:

```
30 1 * * * /tmp/filescript.sh
```

Note: this means the /tmp/filescript.sh file run at 1:30 AM every day

3. Create filescript.sh file

```
tmp# touch filescript.sh
```

4. Open filescript file to write on it the code I need to run

- 3.1. tmp# vi filescript.sh

- 3.2. Write this code:

```
#!/bin/bash
```

```
time=$(date)
```

```
user=$(who)
```

```
echo "${time} - ${user}" >> file.txt
```

5. Change mode of script file to allowing you to run it as a script by executing

```
tmp# chmod +x filescript.sh
```

6. Run filescript file in the background

```
tmp# ./filescript.sh
```

7. cat file.txt

```
Wed Jun 14 07:58:32 EDT 2023 - root
Wed Jun 14 08:00:00 EDT 2023 - root
```


Part10 Mariadb:

1. install mariadb from the local repo that was created in yum Repo section

1.1. yum install zabbix-proxy-mysql.x86_64

1.2. yum install zabbix-server-mysql.x86_64

1.3. yum install mariadb

2. Start and enable mariadb server

systemctl start mariadb

systemctl enable mariadb

3. open ports in iptables from mariadb

iptables -A INPUT -p tcp --dport 3306 -j ACCEPT

4. To change login password in mariadb

mysqladmin -u root password

Note: Write my password anas@1234

5. Open mariadb

5.1. mysql -u root -p

5.2. Enter the password anas@1234

6. Create user and database

6.1. sudo mysql -u root -p

6.2. to create database and user write:

CREATE DATABASE mydb;

// create user set name is anas and set password password

CREATE USER 'anas'@'localhost' IDENTIFIED BY 'password';

// give all privileges on the "mydb" database to the user "anas" when connecting from the "localhost" host. The privileges include the ability to create tables, insert data...

GRANT ALL PRIVILEGES ON mydb.* TO 'anas'@'localhost';

//changes made in the previous commands are immediately applied

FLUSH PRIVILEGES;

EXIT;

7. Connect to database using the user was created in step 6

7.1. `mysql -u anas -p`

7.2. Write the password **password**

8. Use DataBase that i created

8.1. use mydb;

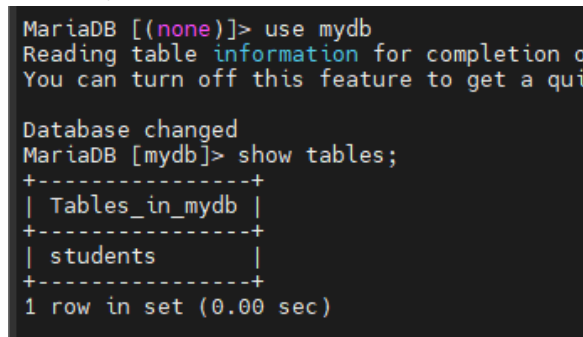
8.2. Write on it:

```
CREATE TABLE students (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    firstName VARCHAR(15),  
    lastName VARCHAR(15),  
    programEnrolled VARCHAR(20),  
    expectedGraduationYear INT,  
    studentNumber VARCHAR(15)  
);
```

```
INSERT INTO students (firstName, lastName, programEnrolled, expectedGraduationYear, studentNumber)  
VALUES  
('Allen', 'Brown', 'mechanical', 2017, '110-001'),  
('David', 'Brown', 'mechanical', 2017, '110-002'),  
('Mary', 'Green', 'mechanical', 2017, '110-003'),  
('Dennis', 'Green', 'electrical', 2018, '110-004'),  
('Joseph', 'Black', 'electrical', 2018, '110-005'),  
('Dennis', 'Black', 'electrical', 2018, '110-006'),  
('Ritchie', 'Salt', 'computer science', 2020, '110-007'),  
('Robert', 'Salt', 'computer science', 2020, '110-008'),  
('David', 'Suzuki', 'computer science', 2020, '110-009'),  
('Mary', 'Chen', 'computer science', 2020, '110-010');
```

9. To show table I have

9.1. Show tables;



```
MariaDB [(none)]> use mydb  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
MariaDB [mydb]> show tables;  
+-----+  
| Tables_in_mydb |  
+-----+  
| students        |  
+-----+  
1 row in set (0.00 sec)
```

9.2. Describe students

```
MariaDB [mydb]> describe students;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
firstName	varchar(15)	YES		NULL	
lastName	varchar(15)	YES		NULL	
programEnrolled	varchar(20)	YES		NULL	
expectedGraduationYear	int(11)	YES		NULL	
studentNumber	varchar(15)	YES		NULL	

6 rows in set (0.00 sec)

9.3. Select * from students; (to show value that I was insert it)

```
MariaDB [mydb]> select * from students;
```

id	firstName	lastName	programEnrolled	expectedGraduationYear	studentNumber
1	Allen	Brown	mechanical	2017	110-001
2	David	Brown	mechanical	2017	110-002
3	Mary	Green	mechanical	2017	110-003
4	Dennis	Green	electrical	2018	110-004
5	Joseph	Black	electrical	2018	110-005
6	Dennis	Black	electrical	2018	110-006
7	Ritchie	Salt	computer science	2020	110-007
8	Robert	Salt	computer science	2020	110-008
9	David	Suzuki	computer science	2020	110-009
10	Mary	Chen	computer science	2020	110-010

10 rows in set (0.00 sec)