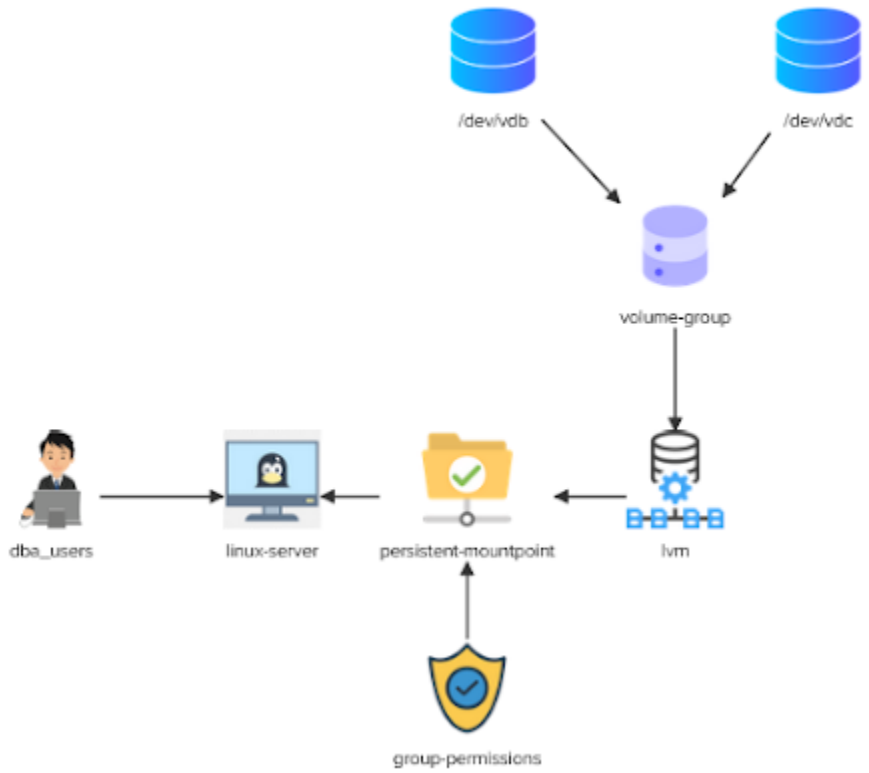
Linux Challenges

Table of Content

Challenge 1

Diagram



Solution:

The database server called centos-host is running short on space! You have been asked to add an LVM volume for the Database team using some of the existing disks on this server.

Task: Install the correct packages that will allow the use of "lvm" on the centos machine.
switch to root user

```
sudo su -
```

check any existing physical / logical volumes

```
pvs
lvs
```

Check the lvm packages

```
rpm -qa | grep lvm
```

Install the required documents

```
yum install -y lvm2
```

Again Check lvm packages

```
rpm -qa |grep lvm
```

Task: Create a Physical Volume for "/dev/vdb"

Create Physical Volume

```
pvcreate /dev/vdb
```

Validate the physical volume

```
pvs
```

Task: Create a Physical Volume for "/dev/vdc"

Create Physical Volume

```
pvcreate /dev/vdc
```

Validate the physical volume

```
pvs
```

Task: Create a volume group called "dba_storage" using the physical volumes "/dev/vdb" and "/dev/vdc"

Create Volume Group

```
vgcreate dba_storage /dev/vdb /dev/vdc
```

Task: Create an "lvm" called "volume_1" from the volume group called "dba_storage". Make use of the entire space available in the volume group.

```
lvcreate -n volume_1 -l 100%FREE dba_storage
```

Validate the logical volume

```
lvs
```

Task: Format the lvm volume "volume_1" as an "XFS" filesystem

```
mkfs.xfs /dev/dba_storage/volume_1
```

Task: Mount the filesystem at the path "/mnt/dba_storage".

Create directory

```
mkdir -p /mnt/dba_storage
```

Mount the directory

```
mount -t xfs /dev/dba_storage/volume_1 /mnt/dba_storage
```

Check the Disk storage

```
df -h /mnt/dba_storage/
```

Task: Make sure that this mount point is persistent across reboots with the correct default options.

Open the fstab file in editor

```
vi /etc/fstab
```

Add the following content

```
/dev/mapper/dba_storage-volume_1 /mnt/dba_storage xfs defaults 0 0
```

Task: Create a group called "dba_users" and add the user called 'bob' to this group

Group Create

```
groupadd dba_users
```

Add **bob** in **dba_users** group

```
usermod -aG dba_users bob
```

Validate

```
id bob
```

Task: Ensure that the mountpoint "/mnt/dba_storage" has the group ownership set to the "dba_users" group

Change only the group ownership

```
chown :dba_users /mnt/dba_storage
```

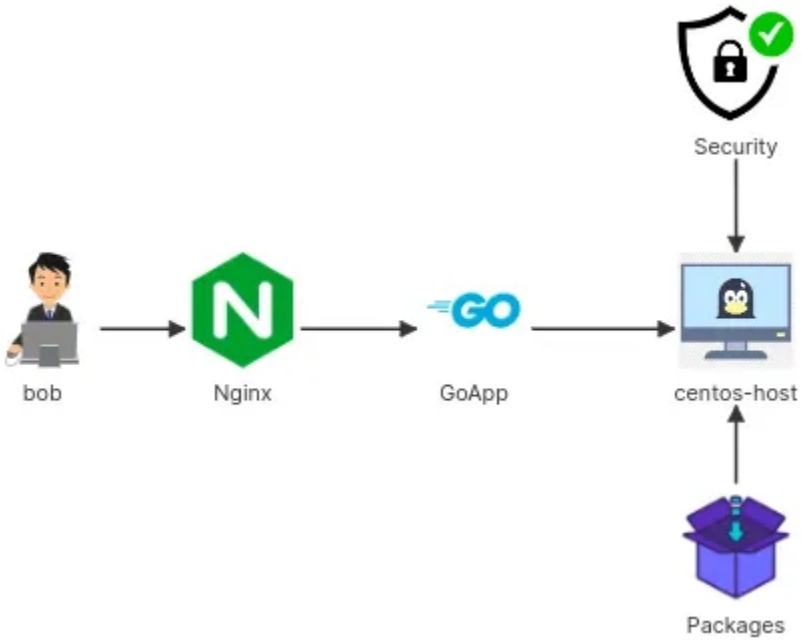
Task: Ensure that the mount point "/mnt/dba_storage" has "read/write" and execute permissions for the owner and group and no permissions for anyone else.

Change Permissions

```
chmod 770 /mnt/dba_storage
```

Challenge 2

Diagram



Solution:

Task:Troubleshoot the issues with “yum/dnf” and make sure you are able to install the packages on “centos-host”.

check the config file

```
cat /etc/resolv.conf
```

found that the nameserver is missing in the file

Add the nameserver in the config file

```
sudo nano /etc/resolv.conf
```

Add the Following content in the config file

```
nameserver 8.8.8.8
nameserver 8.8.4.4
nameserver 1.1.1.1
```

clean command

```
sudo dnf clean all
```

Task: Install “nginx” package and Install “firewalld” package.

After resolved the package manager issue then Install Packages

```
sudo yum install nginx
sudo yum install firewalld
```

Start and Enable Services

```
sudo systemctl start nginx
sudo systemctl enable nginx


sudo systemctl start firewalld
sudo systemctl enable firewalld
```

Task: Add firewall rules to allow only incoming port “22”, “80” and “8081”.

Add Permanent Firewall Rules

```
sudo firewall-cmd --permanent --add-port=22/tcp
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=8081/tcp
```

Reload firewalld

```
sudo firewall-cmd --reload
```

Verify the list

```
sudo firewall-cmd --list-all
sudo firewall-cmd --list-ports
```

Task: Start GoApp by running the “nohup go run main.go &” command from “/home/bob/go-app/” directory, it can take few seconds to start.

Run Go Application

```
cd /home/bob/go-app/
nohup go run main.go &
```

Task: Configure Nginx as a reverse proxy for the GoApp so that we can access the GoApp on port “80”

create nginx file

```
sudo nano /etc/nginx/conf.d/goapp.conf
```

Add the following configuration

```
server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://localhost:8081;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Test nginx

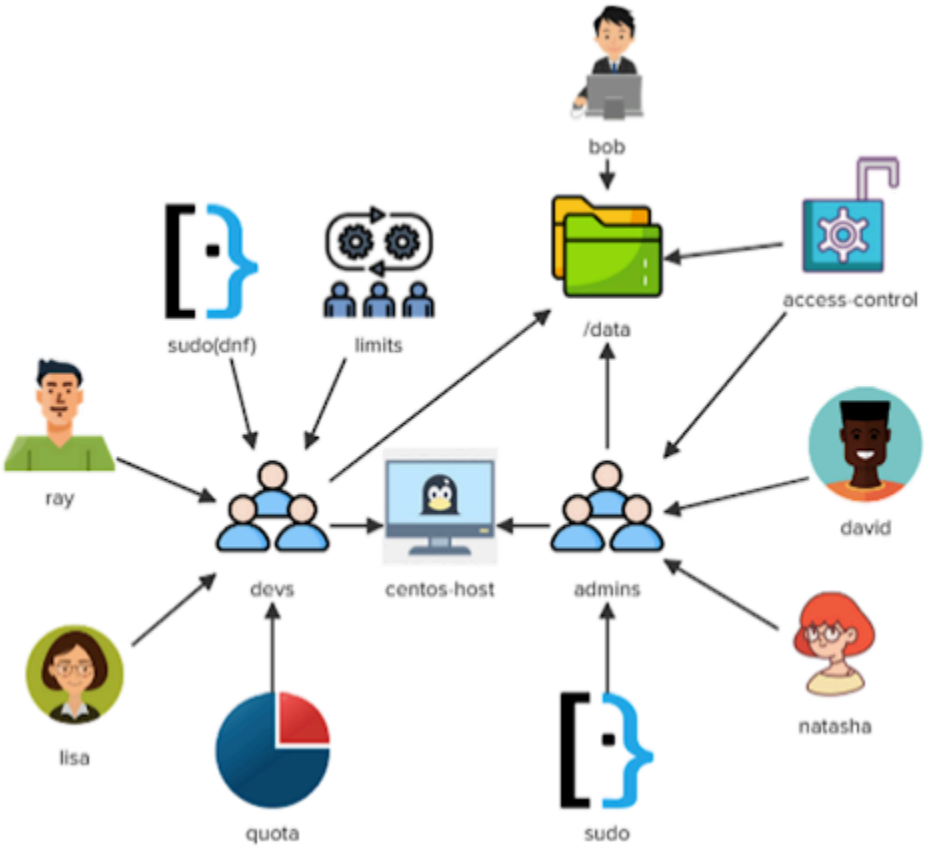
```
sudo nginx -t
```

Restart Nginx

```
sudo systemctl reload nginx
sudo systemctl restart nginx
```

Challenge 3

Diagram



Solution:

Task: Create a group called "*devs*" and Create a group called "*admins*"
Create devs and admin group

```
sudo groupadd devs
sudo groupadd admins
```

Check Group created or not

```
getent group devs
getent group admins
```

Task: Create a user called "david" , change his login shell to "/bin/zsh" and set "D3vUd3raaw" password for this user. Make user "david" a member of "admins" group.

Create user david

```
sudo useradd -s /bin/ash david
```

Set password

```
passwd david
```

Add david user in admins groups

```
sudo usermod -aG admins david
```

Task: Create a user called "natasha" , change her login shell to "/bin/zsh" and set "DwfawUd113" password for this user. Make user "natasha" a member of "admins" group.

Create user natasha

```
sudo useradd -s /bin/ash natasha
```

Set password

```
passwd natasha
```

Add natasha user in admins groups

```
sudo usermod -aG admins natasha
```

Task: Make sure "/data" directory is owned by user "bob" and group "devs" and "user/group" owner has "full" permissions but "other" should not have any permissions.

Check current ownership of **/data** directory

```
ls -lsd /data
```

Change the ownership

```
sudo chown bob:devs /data
```

Change Permission

```
sudo chmod 770 /data
```

Check ownership of /data directory

```
ls -lsd /data
```

Task: Give some additional permissions to "admins" group on "/data" directory so that any user who is a member of the "admins" group has "full permissions" on this directory.

Check Access

```
getfacl /data
```

Allow admin groups members to access the **/data** directory

```
setfacl -m g:admins:rwx /data
```

Task: Make sure all users under "admins" group can run all commands with "sudo" and without entering any password

Open sudoers file

```
visudo
```

Add the following line in the sudoers file

```
%admins ALL=(ALL) NOPASSWD:ALL
```

Task: Configure a "resource limit" for the "devs" group so that this group (members of the group) can not run more than "30 processes" in their session. This should be both a "hard limit" and a "soft limit", written in a single line.

Open **limits.conf** file

```
vi /etc/security/limits.conf
```

Add the following line in the conf file

```
@devs - nproc 30
```

This sets a single-line soft and hard 30 process limit for all members of the group.

Task: Make sure all users under "devs" group can only run the "dnf" command with "sudo" and without entering any password.

Open sudoers file

```
visudo
```

Add the following line in the sudoers file

```
%devs ALL=(ALL) NOPASSWD:/usr/bin/dnf
```

Task: Create a user called "*ray*" , change his login shell to "/bin/sh" and set "D3vU3r321" password for this user. Make user "ray" a member of "devs" group.

Create user ray

```
sudo useradd -s /bin/sh ray
```

Set password

```
passwd ray
```

Add ray user in devs groups

```
sudo usermod -aG devs ray
```

Task: Create a user called "lisa", change her login shell to "/bin/sh" and set "D3vUd3r123" password for this user. Make user "lisa" a member of "devs" group.

Create user ray

```
sudo useradd -s /bin/sh lisa
```

Set password

```
passwd lisa
```

Add ray user in devs groups

```
sudo usermod -aG devs lisa
```

Task : Edit the disk quota for the group called "devs". Limit the amount of storage space it can use (not inodes). Set a "soft" limit of "100MB" and a "hard" limit of "500MB" on "/data" partition.

To Add Disk Quota

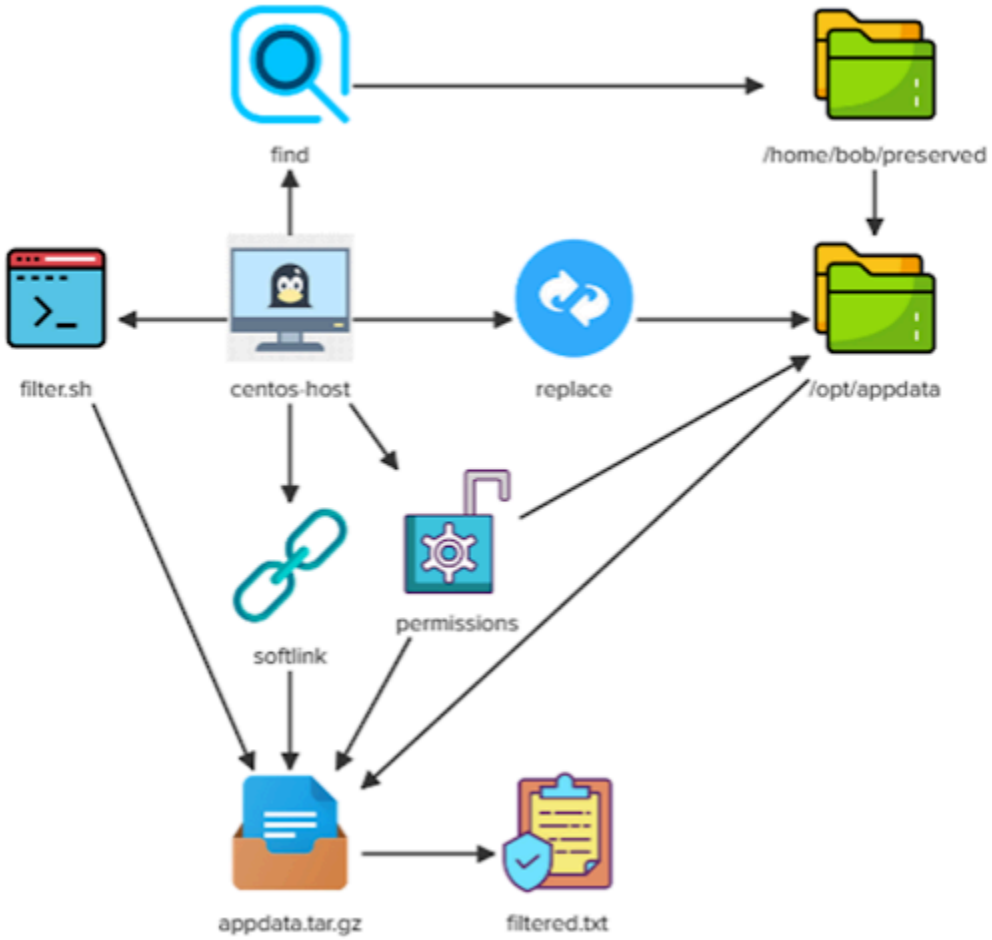
```
sudo setquota -g devs 100M 500M 0 0 /data
```

To see the disk quota

```
sudo quota -g -s  devs /data
```

Challenge 4

Diagram



Solution:

At first switch to **root** user

```
sudo -i
```

Task: Create "/opt/appdata" directory.
list the folders in **/opt**

```
ls -l /opt/
```

appdata directory is not present in the **/opt**.

Create appdata directory in **/opt**

```
cd /opt
mkdir appdata
```

Task: Do not delete any files from "/home/bob/preserved" directory.

Task: Find the "hidden" files in "/home/bob/preserved" directory and copy them in "/opt/appdata/hidden/" directory (create the destination directory if doesn't exist).

Create the directory named **hidden** in the */opt/appdata*

```
mkdir -p /opt/appdata/hidden
```

Check the directory is created or not

```
ls -l /opt/appdata/
```

Find hidden files and copy to */opt/appdata/hidden/*

```
find /home/bob/preserved -type f -name ".*" -exec cp "{}"
/opt/appdata/hidden/ \;
```

Task: Find the "non-hidden" files in "/home/bob/preserved" directory and copy them in "/opt/appdata/files/" directory (create the destination directory if doesn't exist).

Create the directory named **files** in the */opt/appdata*

```
mkdir -p /opt/appdata/files
```

Check the directory is created or not

```
ls -l /opt/appdata/
```

Find non-hidden files and copy to */opt/appdata/files/*

```
find /home/bob/preserved -type f -not -name ".*" -exec cp "{}"
/opt/appdata/files/ \;
```

Task: Find and delete the files in "/opt/appdata" directory that contain a word ending with the letter "t" (case sensitive).

Delete files containing words ending with "t" (case sensitive)

```
rm -f $(find /opt/appdata/ -type f -exec grep -l 't\>' "{}"  \; )
```

or

```
find /opt/appdata -type f -exec grep -l '\<[A-Za-z]*t\>' {} \; | xargs -r
sudo rm -f
```

Task: Change all the occurrences of the word "yes" to "no" in all files present under "/opt/appdata/" directory.

Replace "yes" with "no" in all files

```
find /opt/appdata -type f -name "*" -exec sed -i 's/\byes\b/no/g' "{}" \;
```

or

```
find /opt/appdata -type f -exec sed -i 's/\byes\b/no/g' {} \;
```

Task: Change all the occurrences of the word "raw" to "processed" in all files present under "/opt/appdata/" directory. It must be a "case-insensitive" replacement, means all words must be replaced like "raw , Raw , RAW" etc.

Replace "raw" with "processed" (case-insensitive)

```
find /opt/appdata -type f -name "*" -exec sed -i 's/\braw\b/processed/ig'
"{}" \;
```

or

```
find /opt/appdata -type f -exec sudo sed -i 's/\braw\b/processed/gi' {} \;
```

Task: Create a "tar.gz" archive of "/opt/appdata" directory and save the archive to this file: "/opt/appdata.tar.gz". The "appdata.tar.gz" archive should have the final processed data.

Create tar.gz archive

```
cd /opt
tar -zcf appdata.tar.gz appdata
```

Create a "softlink" called "/home/bob/appdata.tar.gz" of "/opt/appdata.tar.gz" file.

Create softlink

```
ln -s /opt/appdata.tar.gz /home/bob/appdata.tar.gz
```

Task: Add the "sticky bit" special permission on "/opt/appdata" directory (keep the other permissions as it is).

Add sticky bit to /opt/appdata

```
chmod +t /opt/appdata
```

Check permissions

```
ls -l /opt/appdata
```

Task: Make "bob" the "user" and the "group" owner of "/opt/appdata.tar.gz" file.

Change ownership of tar.gz file

```
chown bob:bob /opt/appdata.tar.gz
```

Task: The "user/group" owner should have "read only" permissions on "/opt/appdata.tar.gz" file and "others" should not have any permissions.

Set permissions on tar.gz file (user/group read-only, others none)

```
chmod 440 /opt/appdata.tar.gz
```

Task: Create a script called "/home/bob/filter.sh".

Create filter.sh file

```
cd /home/bob
touch filter.sh
```

Make this file executable

```
chmod +x /home/bob/filter.sh
```

Task: The script should filter the lines from "/opt/appdata.tar.gz" file which contain the word "processed", and save the filtered output in "/home/bob/filtered.txt" file. It must "overwrite" the existing contents of "/home/bob/filtered.txt" file

Add the following in the filter.sh file

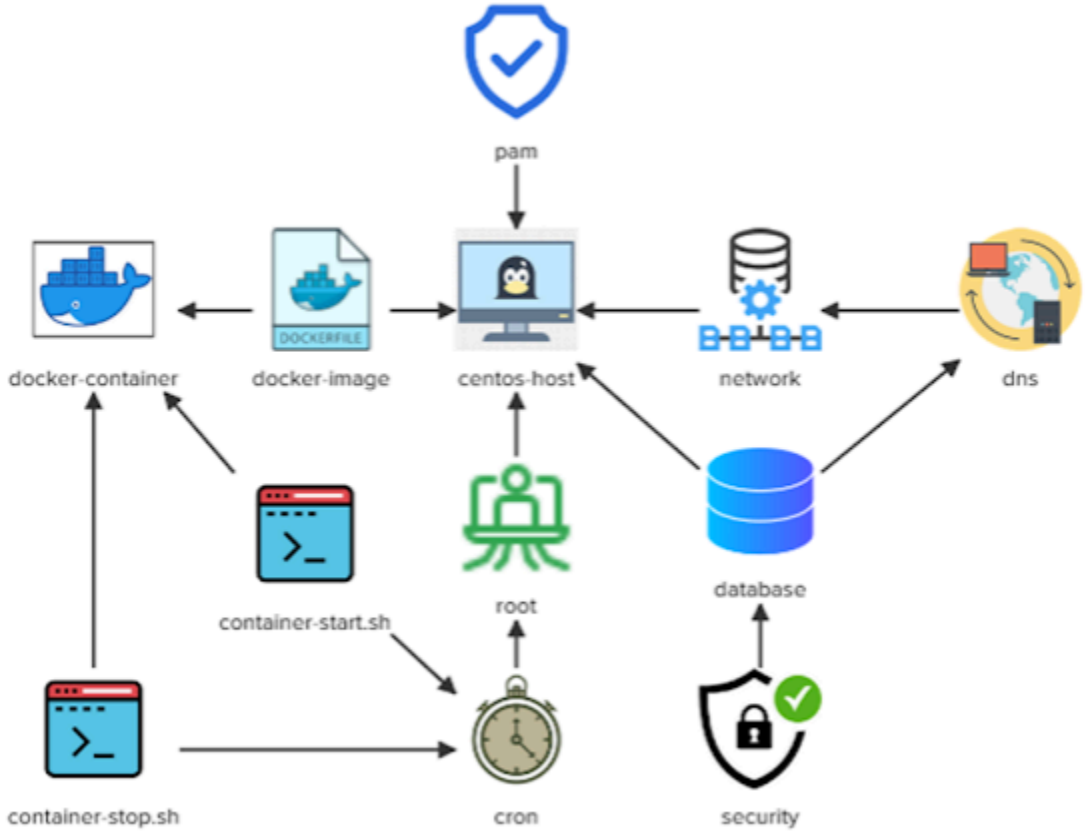
```
#!/bin/bash
tar -xzOf /opt/appdata.tar.gz | grep processed > /home/bob/filtered.txt
```

Execute the script and filter data in filter.txt file

```
cd /home/bob
./filter.sh
```

Challenge 5

Diagram



Solution:

Task: The "root" account is currently locked on "centos-host", please unlock it.
Unlocking root user

```
sudo usermod -U root
```

Switch to root user

```
sudo -i
```

Task: Make user "root" a member of "wheel" group
Adding root user to wheel group

```
usermod -aG wheel root
```

Task: Create a bash script called "container-stop.sh" under "/home/bob/" which should be able to stop the "myapp" container. It should also display a message "myapp container stopped!"

create ***container-stop.sh*** file in the ***/home/bob*** directory

```
cd /home/bob
touch container-stop.sh

vi container-stop.sh
```

Add the following content in the ***container-stop.sh*** file

```
#!/bin/bash

docker stop myapp
echo "myapp container stopped!"
```

Make it executable

```
chmod +x container-stop.sh
```

Task: Create a bash script called "container-start.sh" under "/home/bob/" which should be able to "start" the "myapp" container. It should also display a message "myapp container started!"

create ***container-start.sh*** file in the ***/home/bob*** directory

```
cd /home/bob
touch container-start.sh

vi container-start.sh
```

Add the following content in the ***container-start.sh*** file

```
#!/bin/bash

docker start myapp
echo "myapp container started!"
```

Make it executable

```
chmod +x container-start.sh
```

Task: Add a cron job for the "root" user which should run "container-stop.sh" script at "12am" everyday.
Adding cron job for container-stop.sh

```
crontab -e
```

Add the following in the cronjob file

```
0 0 * * * /home/bob/container-stop.sh
```

Task: Add a cron job for the "root" user which should run "container-start.sh" script at "8am" everyday.
Adding cron job for container-start.sh

```
crontab -e
```

Add the following in the cronjob file

```
0 8 * * * /home/bob/container-start.sh
```

Task: Add a local DNS entry for the database hostname "mydb.kodekloud.com" so that it can resolve to "10.0.0.50" IP address.
Add the entry in the host file for local resolution of dns

```
vi /etc/hosts
```

Add the following in the host file

```
10.0.0.50  mydb.kodekloud.com
```

Task: Add an extra IP to "eth1" interface on this system: 10.0.0.50/24
Adding extra IP to eth1

```
ip address add 10.0.0.50/24 dev eth1
```

Task: Install "mariadb" database server on this server and "start/enable" its service.
Installing MariaDB

```
yum install -y mariadb-server
```

Starting and enabling MariaDB service

```
systemctl start mariadb
systemctl enable mariadb
systemctl status mariadb
```

Task: Set a password for mysql root user to "S3cure#321"
Setting MySQL root password

```
mysqladmin -u root password 'S3cure#321'
```

Task: Pull "nginx" docker image.
Pulling nginx latest Docker image

```
docker pull nginx:latest
```

Task: Create and run a new Docker container based on the "nginx" image. The container should be named as "myapp" and the port "80" on the host should be mapped to the port "80" on the container.
Creating and running myapp container

```
docker run -d --name myapp -p 80:80 nginx
```

Task: Edit the PAM configuration file for the "su" utility so that this utility only accepts the requests from the users that are part of the "wheel" group and the requests from the users should be accepted immediately, without asking for any password.
Configuring PAM for su utility

```
vi /etc/pam.d/su
```

Add the following in the file.

```
#%PAM-1.0
# Allow only users in wheel group
auth       required   pam_wheel.so use_uid
# Allow immediate access without password
auth       sufficient pam_rootok.so
# Disable password authentication
#auth      include    system-auth
account    sufficient pam_succeed_if.so uid = 0 use_uid quiet
account    include    system-auth
session    include    system-auth
```