Maati, Oussama

Linux Server project

Implementing Linux-Based Infrastructure with Virtual Machines

For this project, I successfully set up two virtual machines using VirtualBox: one Ubuntu server and one Ubuntu desktop. To ensure their privacy, I created a NAT network, establishing a dedicated private network for both machines. Additionally, I equipped them with bridged network interfaces, enabling seamless internet access for both virtual machines.

DHCP server :

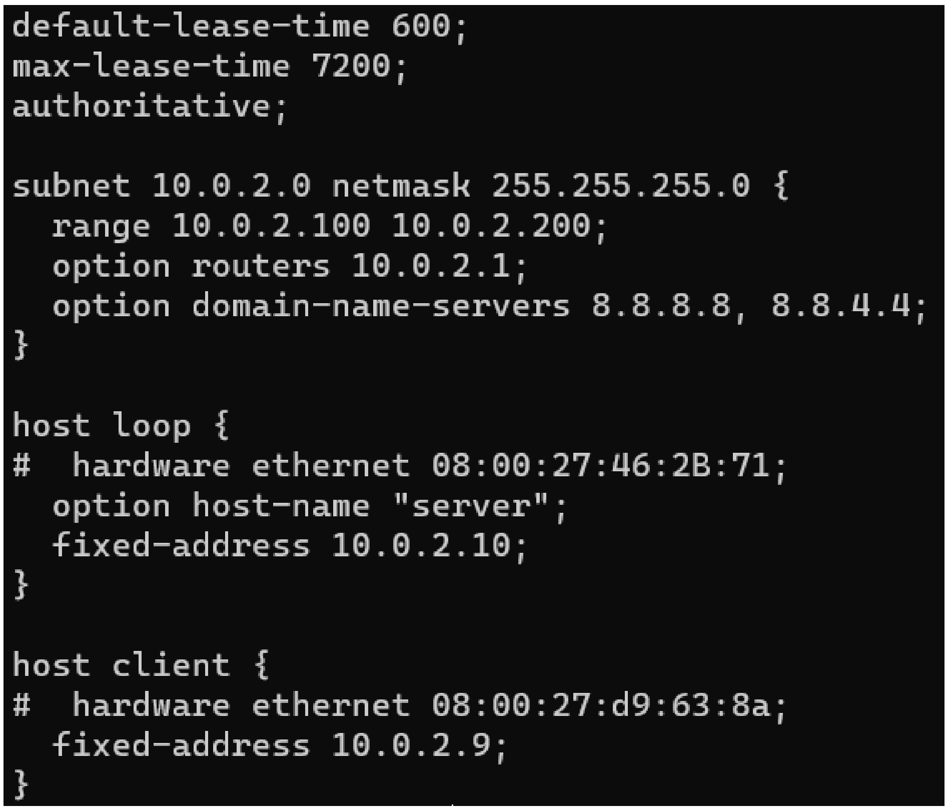
First I install the service

```
$ sudo apt install isc-dhcp-server
```

Then I modify this file ***"/etc/default/isc-dhcp-server"*** and add this :

```
INTERFACESv4="enp0s8"
```

Modify "enp0s8" by your interface where you want to install the dhcp server.Then I configured the "**etc/dhcp/dhcp.conf**" file depending on my NAT network



```
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 10.0.2.0 netmask 255.255.255.0 {
  range 10.0.2.100 10.0.2.200;
  option routers 10.0.2.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}

host loop {
#   hardware ethernet 08:00:27:46:2B:71;
  option host-name "server";
  fixed-address 10.0.2.10;
}

host client {
#   hardware ethernet 08:00:27:d9:63:8a;
  fixed-address 10.0.2.9;
}
```

After that we have to restart the service :

```
$ sudo systemctl start isc-dhcp-server.service
```

```
$ sudo systemctl enable isc-dhcp-server.service
```

And do not forget to permit DHCP service on firewall :

```
$ sudo ufw allow  67/udp
```

```
$ sudo ufw reload
```

```
$ sudo ufw show
```

Then we just need to configure the client in the file **"/etc/network/interfaces"** :

```
auto  enp0s8
```

```
iface enp0s8 inet dhcp
```

After I reboot the system to apply the changes and the client will be on the pool of the DHCP server.

DNS server :

Fist I install the service :

```
$ sudo apt install bind9 bind9-utils bind9-dnsutils -y
```

Then we check is the service is running :

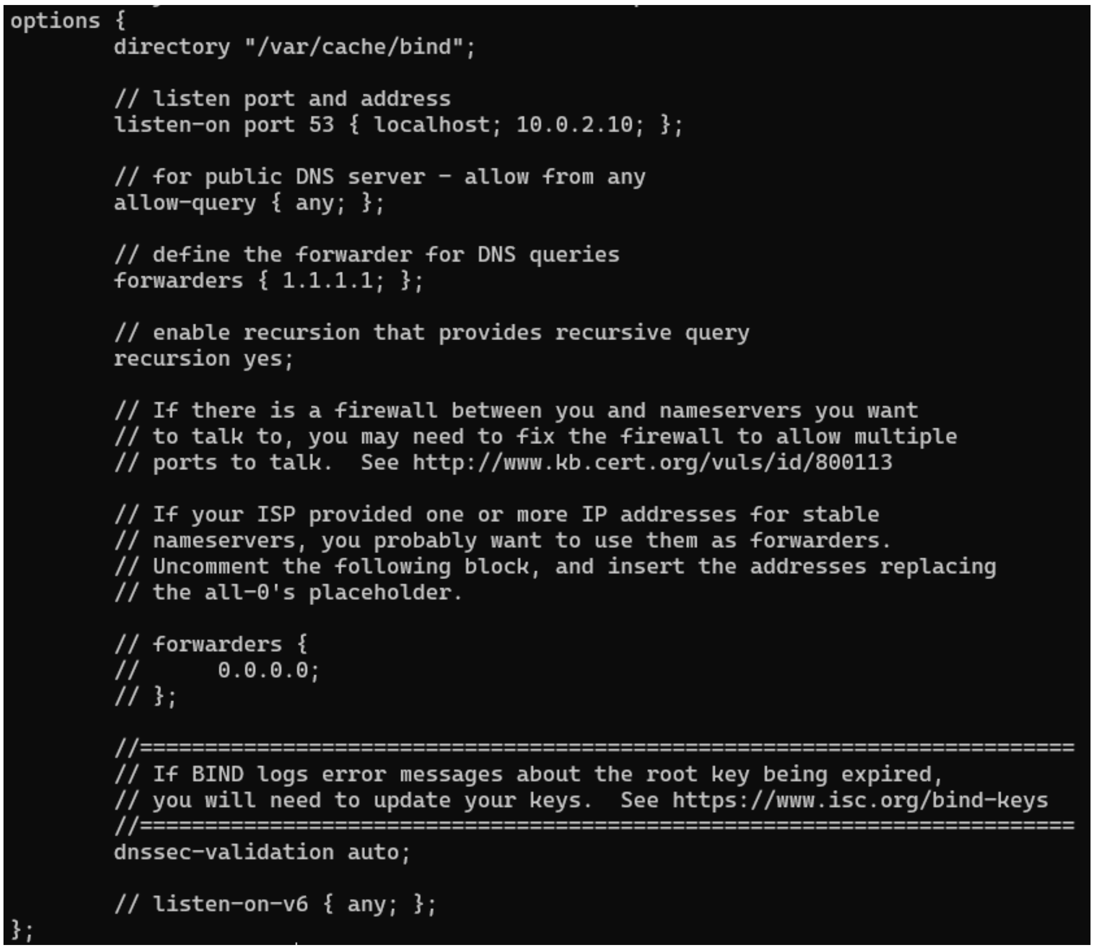
```
$ sudo systemctl status named
```

Now it is time to configure the BIND DNS server (in the same server for dhcp).

Go to this file "***/etc/default/named****" and make sure you have this line :*

```
OPTIONS="-4 -u bind"
```

Now we have to modify the bind configuration file "***/etc/bind/named.conf.options***"



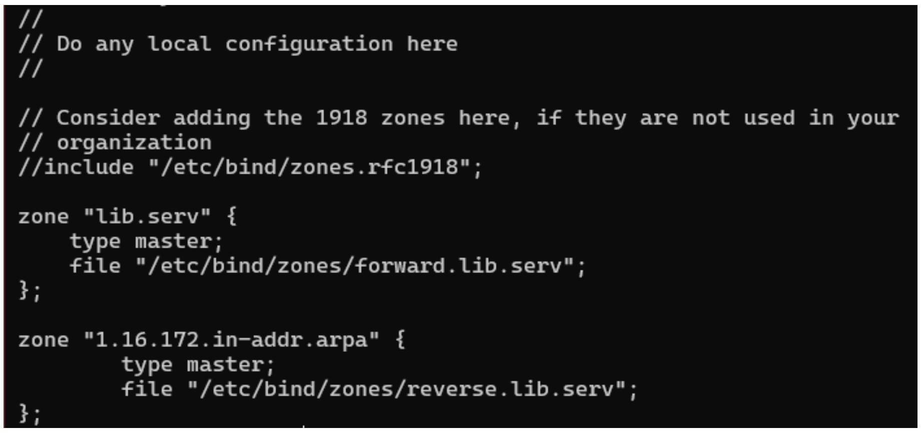
```
options {
        directory "/var/cache/bind";

        // listen port and address
        listen-on port 53 { localhost; 10.0.2.10; };

        // for public DNS server - allow from any
        allow-query { any; };

        // define the forwarder for DNS queries
        forwarders { 1.1.1.1; };

        // enable recursion that provides recursive query
        recursion yes;

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        // listen-on-v6 { any; };
};
```

Make sure you have all this in your file, except the localhost IP that has to match your server IP.

Save the file and make sure there is no error in it with this command.

```
$ sudo named-checkconf
```

After this we will create the zones of our DNS edit the */etc/bind/named.conf.local* file



```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "lib.serv" {
    type master;
    file "/etc/bind/zones/forward.lib.serv";
};

zone "1.16.172.in-addr.arpa" {
        type master;
        file "/etc/bind/zones/reverse.lib.serv";
};
```

There are 2 zones, 1 forward and 1 reverse. With the name lib.serv and the other for the reverse.

After that create the 2 zones files

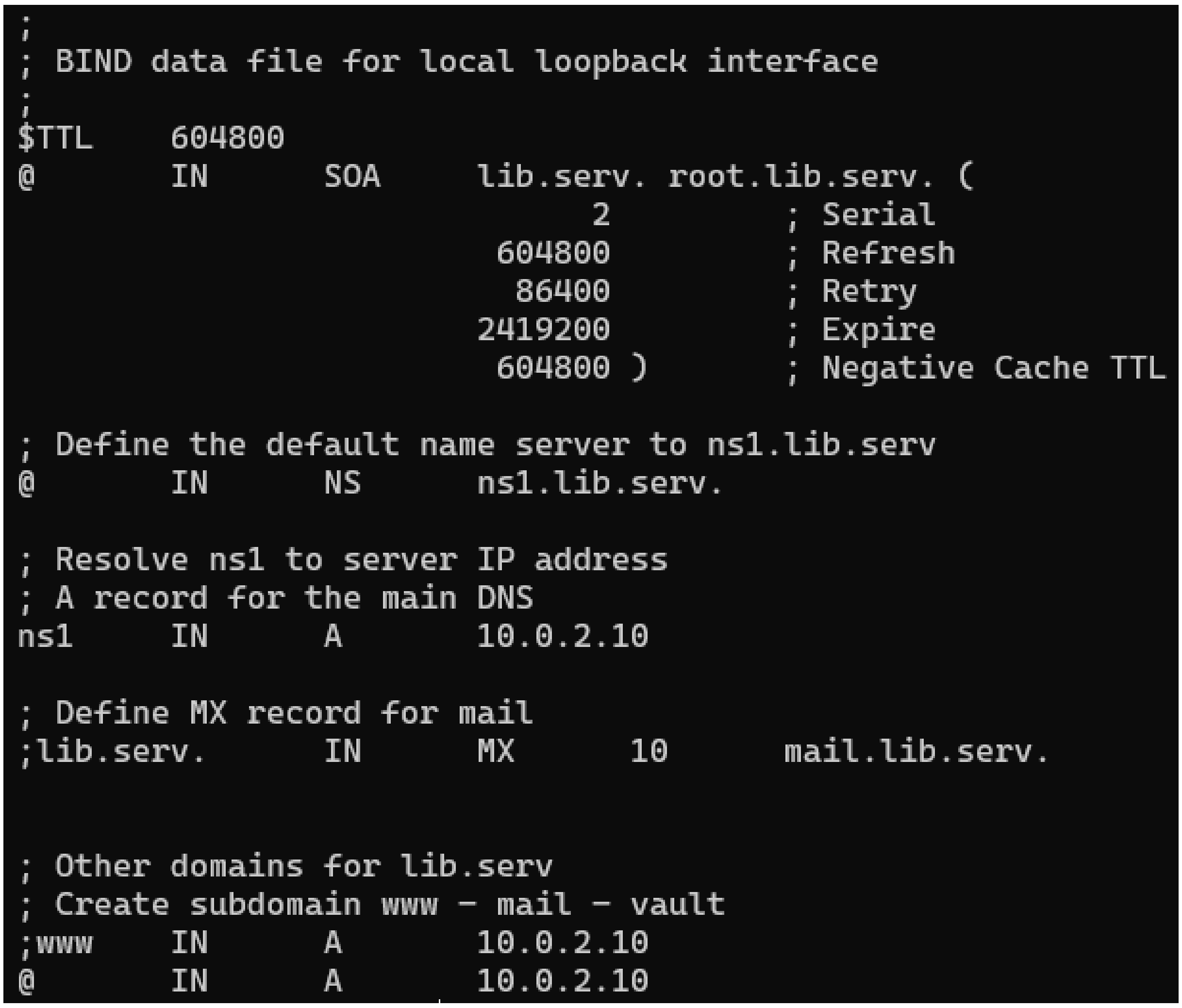
```
$ mkdir -p /etc/bind/zones/
```

```
$ sudo cp /etc/bind/db.local /etc/bind/zones/forward.lib.serv
```

```
$ sudo cp /etc/bind/db.127 /etc/bind/zones/reverse.lib.serv
```

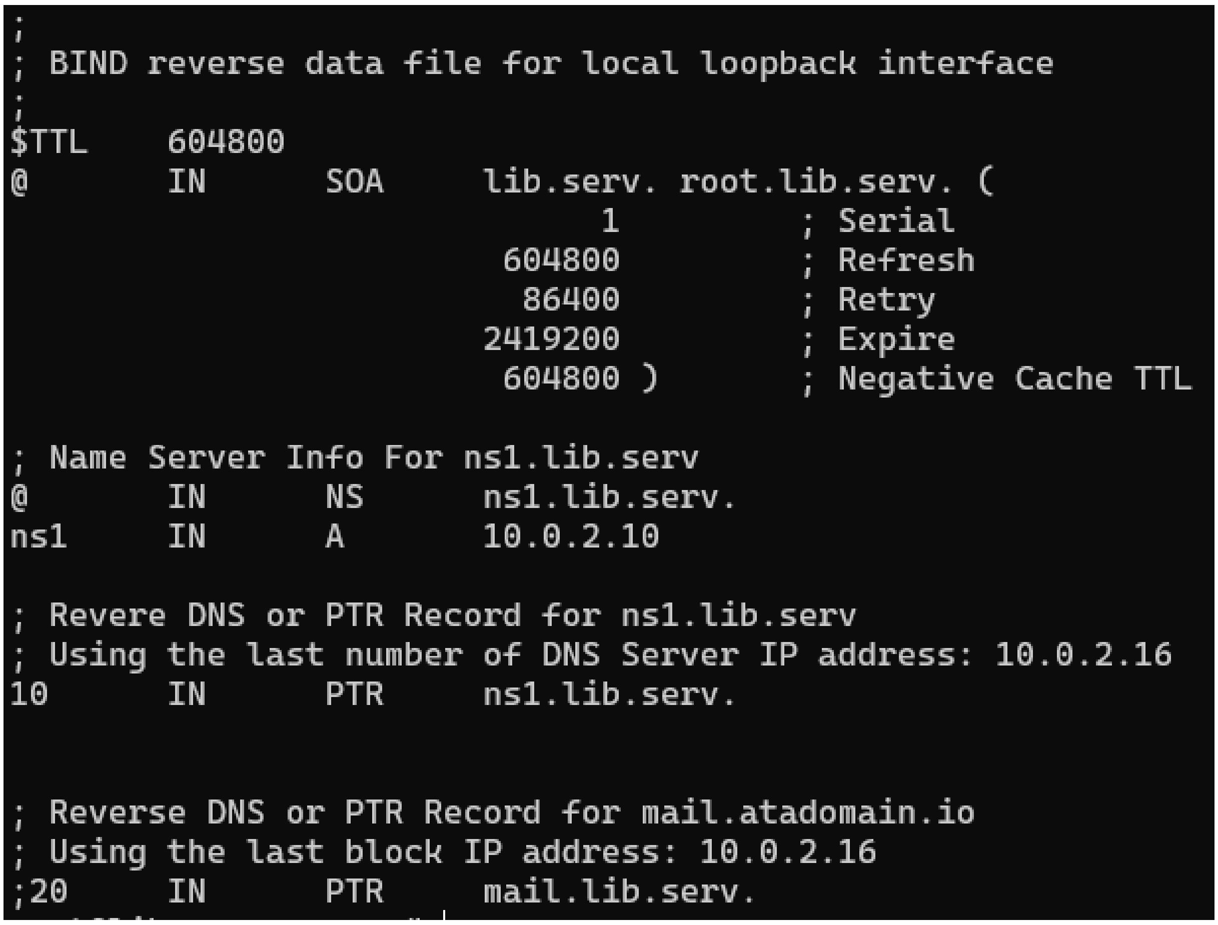
And modify each file, this is how I configured them :

The forward



```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     lib.serv. root.lib.serv. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

; Define the default name server to ns1.lib.serv
@       IN      NS      ns1.lib.serv.

; Resolve ns1 to server IP address
; A record for the main DNS
ns1     IN      A       10.0.2.10

; Define MX record for mail
;lib.serv.      IN      MX      10      mail.lib.serv.


; Other domains for lib.serv
; Create subdomain www - mail - vault
;www    IN      A       10.0.2.10
@       IN      A       10.0.2.10
```

And the reverse



```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     lib.serv. root.lib.serv. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

; Name Server Info For ns1.lib.serv
@       IN      NS      ns1.lib.serv.
ns1     IN      A       10.0.2.10

; Revere DNS or PTR Record for ns1.lib.serv
; Using the last number of DNS Server IP address: 10.0.2.16
10      IN      PTR     ns1.lib.serv.


; Reverse DNS or PTR Record for mail.atadomain.io
; Using the last block IP address: 10.0.2.16
;20     IN      PTR     mail.lib.serv.
```

Again we need to verify if there is no error :

```
$ sudo named-checkconf
```

```
$ sudo named-checkzone lib.serv /etc/bind/zones/forward.lib.serv
```

```
$ sudo named-checkzone lib.serv /etc/bind/zones/reverse.lib.serv
```

And restart the service :

```
$ sudo systemctl restart named
```

```
$ sudo systemctl status named
```

Like for the DHCP, we have to open the port for the DNS in the firewall

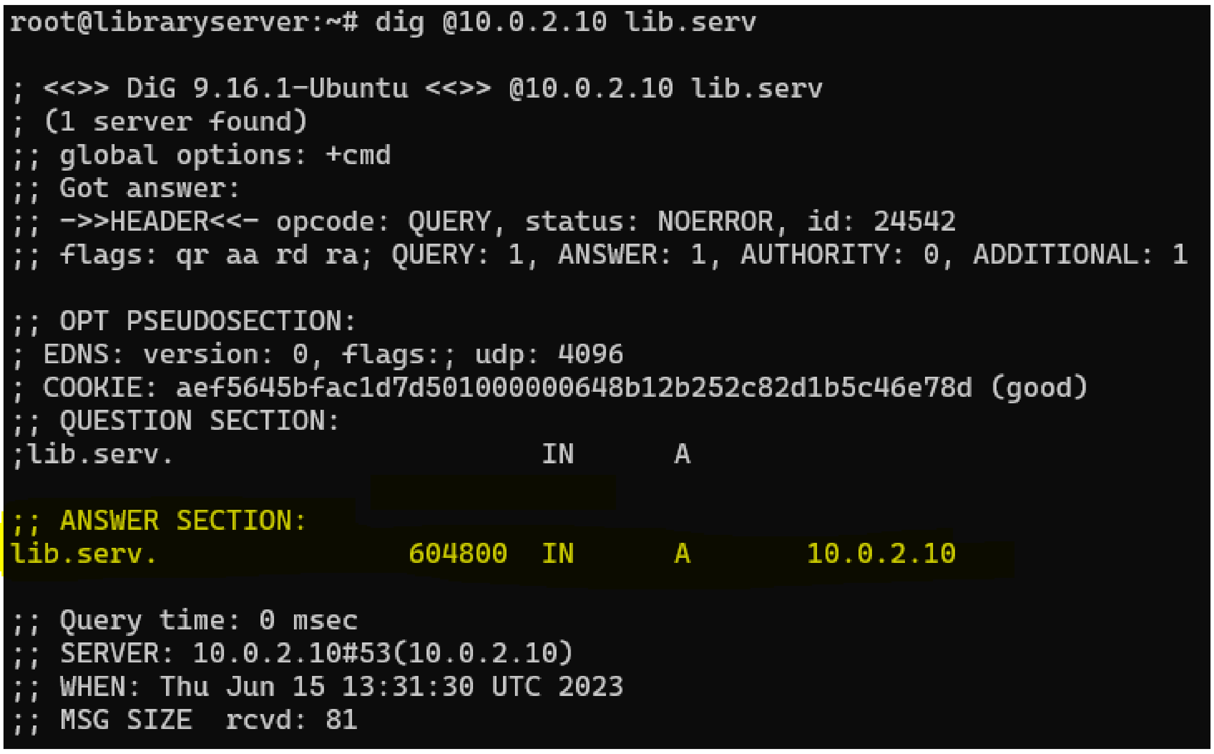
```
$ sudo ufw allow Bind9
```

```
$ sudo ufw status
```

And I also used those command to configure my DNS :

```
$ sudo rndc reconfig
$ sudo rndc managed-keys refresh
```

Finally, we can check with the dig command if everything is working :



```
root@libraryserver:~# dig @10.0.2.10 lib.serv

; <<>> DiG 9.16.1-Ubuntu <<>> @10.0.2.10 lib.serv
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24542
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: aef5645bfac1d7d501000000648b12b252c82d1b5c46e78d (good)
;; QUESTION SECTION:
;lib.serv.                      IN      A

;; ANSWER SECTION:
lib.serv.               604800  IN      A       10.0.2.10

;; Query time: 0 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Thu Jun 15 13:31:30 UTC 2023
;; MSG SIZE  rcvd: 81
```

We can see there is an answer section as supposed.

HTTP+ mariadb GLPI :

First I install the service, the prerequisite and the GPG keys and start the service

```
$ sudo apt install mariadb-server mariadb-client -y
```

```
$ sudo apt install -y software-properties-common
```

```
$ sudo apt-key adv --fetch-keys 'https://mariadb.org/mariadb_release_signing_key.asc'
```

```
$ sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el]
https://mariadb.mirror.liquidtelecom.com/repo/10.6/ubuntu focal main'
```

```
$ sudo apt update && sudo apt install -y mariadb-server mariadb-client
```

```
$ sudo systemctl start mariadb
```

Then we are going to configure it, allow everything in the configuration and then login maraidb with root

```
$ sudo mysql_secure_installation
```

```
$ sudo mariadb -u root -p
```

You are now in maraidb to type sql command and we are going to create an admin user :

```
CREATE USER 'admin_user'@'localhost' IDENTIFIED BY 'secret_password';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

Where admin_user is the username and secret_password is the password,

Optional :

After that I logged in with the adminuser and created a library database with multiple useful table.

```
CREATE DATABASE library;

USE library;
```

```
CREATE TABLE Users (

  UserID INT PRIMARY KEY,

  Name VARCHAR(255) NOT
NULL,

  ContactInfo
VARCHAR(255),

  MembershipStatus
BOOLEAN

);
```

```
CREATE TABLE Books (

  BookID INT PRIMARY KEY,

  Title VARCHAR(255) NOT
NULL,

  Author VARCHAR(255) NOT
NULL,

  PublicationYear INT,

  Publisher VARCHAR(255),

  ISBN VARCHAR(20),

  AvailabilityStatus
BOOLEAN

);
```

```
CREATE TABLE Loans (
 LoanID INT PRIMARY KEY,
 BookID INT,
 UserID INT,
 LoanDate DATE,
 DueDate DATE,
 ReturnDate DATE,
 Fines DECIMAL(10, 2),
 FOREIGN KEY (BookID)
REFERENCES
Books(BookID),
 FOREIGN KEY (UserID)
REFERENCES Users(UserID)
);
```

After that I easily installed GLPI with this script :

```
$ wget https://raw.githubusercontent.com/jr0w3/GLPI_install_script/main/glpi-install.sh && bash
glpi-install.sh
```

And like the previous install we don’t have to forget to open the port in the firewall.

```
$ sudo ufw allow 80
```

```
$ sudo ufw allow 443
```

Then we can easily access GLPI by typing the ip of the server on a web browser.

SSH :

During the installation of the server, Openssh was an install possibility so it is already installed, all I have to do is configuration :

```
$ sudo nano /etc/ssh/ssh_config
```

And make sure there is that line :

```
Port 22
PermitRootLogin yes
AllowUsers your_username@your_server_ip
```

In my case : `AllowUSers servAdmin@10.0.2.10`

Then restart the service and allow the OpenSSH in the firewall

```
$ sudo ufw allow OpenSSH
```

Backup :

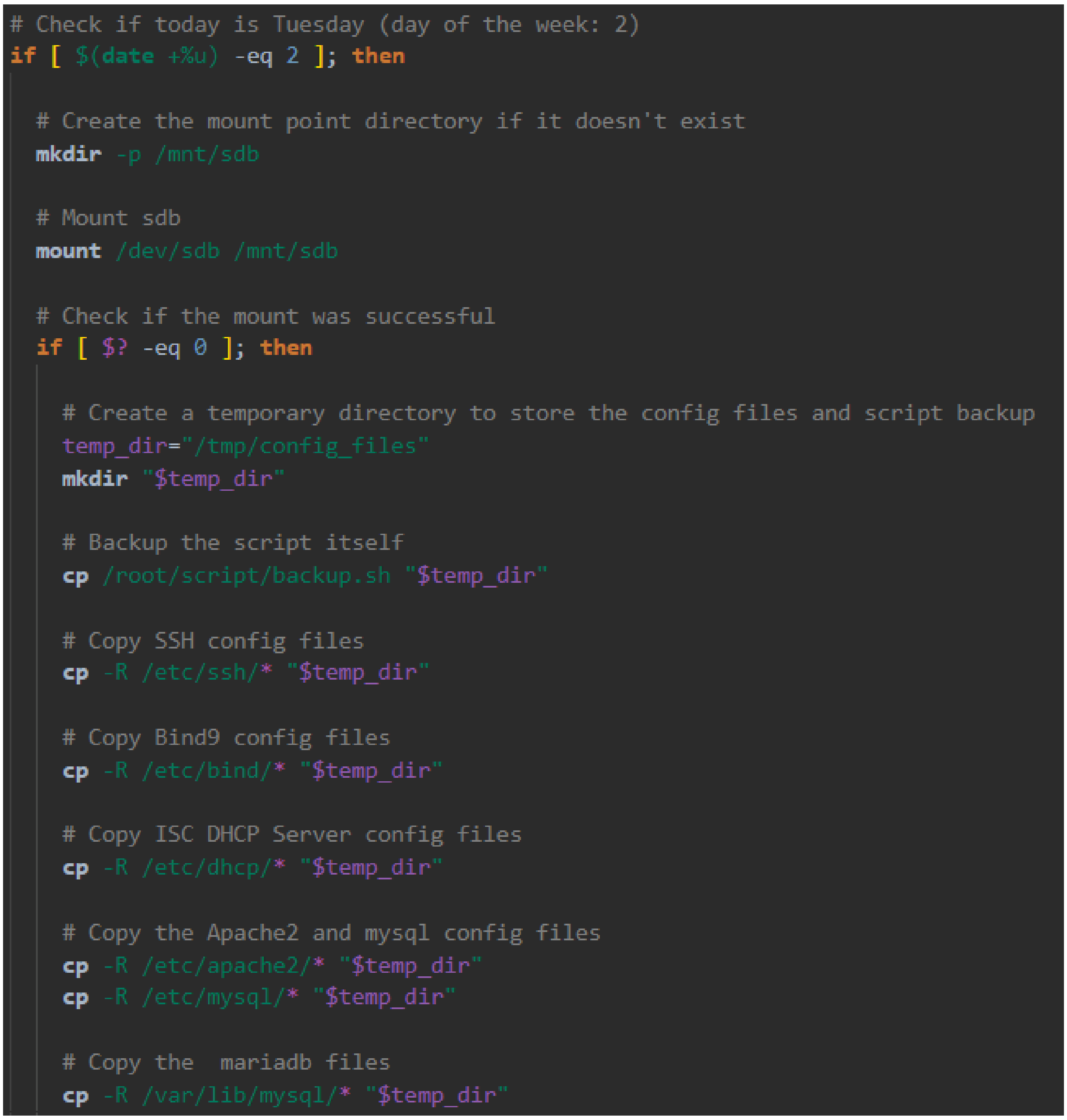
Optional: First I added to my virtual server another disk, so the backup will be mounted to put the files then unmounted for more security.

So for the weekly backup I made a bash script and used crontab so it will be launched at the wanted day of every week.

The first condition check if we are on the wanting day, it can be changed to match which day to launch the script.

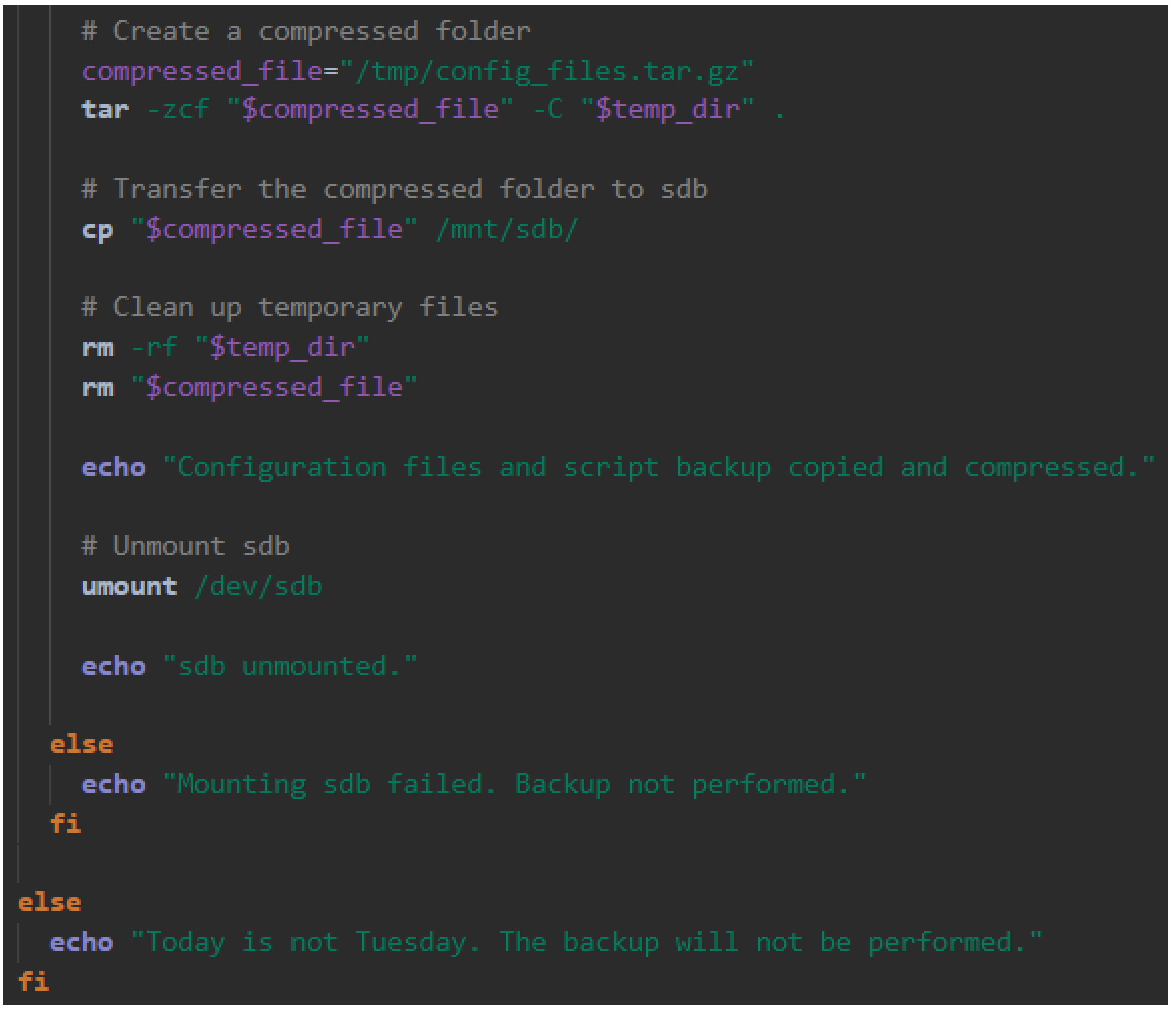
Then the disk where the backup is made and mounted and I create a temporary folder where all the config files will be copied.

Maati, Oussama



```bash
# Check if today is Tuesday (day of the week: 2)
if [ $(date +%u) -eq 2 ]; then

  # Create the mount point directory if it doesn't exist
  mkdir -p /mnt/sdb

  # Mount sdb
  mount /dev/sdb /mnt/sdb

  # Check if the mount was successful
  if [ $? -eq 0 ]; then

    # Create a temporary directory to store the config files and script backup
    temp_dir="/tmp/config_files"
    mkdir "$temp_dir"

    # Backup the script itself
    cp /root/script/backup.sh "$temp_dir"

    # Copy SSH config files
    cp -R /etc/ssh/* "$temp_dir"

    # Copy Bind9 config files
    cp -R /etc/bind/* "$temp_dir"

    # Copy ISC DHCP Server config files
    cp -R /etc/dhcp/* "$temp_dir"

    # Copy the Apache2 and mysql config files
    cp -R /etc/apache2/* "$temp_dir"
    cp -R /etc/mysql/* "$temp_dir"

    # Copy the  mariadb files
    cp -R /var/lib/mysql/* "$temp_dir"
```

Then I compress that folder and copy it in the backup disk, and I don't forget to delete the temporary folder and compressed folder. Also dismount the back disk so it will be inaccessible.



```
        # Create a compressed folder
        compressed_file="/tmp/config_files.tar.gz"
        tar -zcf "$compressed_file" -C "$temp_dir" .

        # Transfer the compressed folder to sdb
        cp "$compressed_file" /mnt/sdb/

        # Clean up temporary files
        rm -rf "$temp_dir"
        rm "$compressed_file"

        echo "Configuration files and script backup copied and compressed."

        # Unmount sdb
        umount /dev/sdb

        echo "sdb unmounted."

    else
        echo "Mounting sdb failed. Backup not performed."
    fi

else
    echo "Today is not Tuesday. The backup will not be performed."
fi
```

After that I use crontab and write a line to launch the script at the wanted time and its location.

```
$ crontab -e
```

```
0 1 * * 2 /root/script/backup.sh
```

So, with this line I make sure the script will be launched every Tuesday at 1AM.

Desktop workstation:

For this part we need to install LibreOffice, GIMP and a web-browser, we can easily do that with the application manger that comes with ubuntu.

Now I am going to move the home folder into another partition for that I first increase the storage's size of my desktop ubuntu VM then I use parted to make a new partition.

```
$ sudo parted
$ select /dev/sda
$ mkpart primary 16.1GB 19GB
$ mkfs.ext4 /dev/sda4
$ quit
```

So here, in the sda partition as I extended my disk from 16.1GB to 19GB, I created a new partition that started from the previous end to the new end of maximm storage, also as the disk was already in 3 partition the new partition is the fourth one.

Now we don't need to forget to update the fstab by first making a linux filesystem and a folder for where to mount the partition :

```
$ sudo fdisk -l
$ sudo mkfs -t ext4 /dev/sda4
$ sudo mkdir /mnt/sda4
```

Then I have to edit the fstab file to update it and add the new partition and its location :

```
$ sudo nano /etc/fstab
```

This line need to be added in the file, it is to select the partition then its location and the type of filesystem and the other parameters can stay by defaults.

```
/dev/sda4   /mnt/sda4   ext4   defaults  0   0
$ sudo mount -a
```

We make sure the partition is mounted

```
sudo mount /dev/sda4 /mnt/sda4
```

Then make a copy of the home folder :

```
mkdir /mnt/sda4/home
sudo cp -a /home/. /mnt/sda4/home
```

And also make sure the folders have been successfully copied

```
sudo diff -r /home /mnt/sda4
```

Then modify the fstab file

```
sudo nano /etc/fstab
```

```
/dev/sda4   /home   ext4   defaults   0   2
```

Run this command to unmont the new partition

```
sudo umount /mnt/new_partition
```

and to remount the new partition as the home directory:

```
sudo mount /home
```

Then I just reboot the system and everything will be in order.

Overall, this project successfully demonstrated the feasibility and benefits of implementing Linux-based infrastructure for the local library. By leveraging virtual machines, network services, web-based management tools, and secure remote access, the proposed solution provides a cost-effective and efficient alternative to traditional Windows-based setups. The Linux-based infrastructure offers stability, security, flexibility, and compatibility with various open-source software solutions, empowering the library to overcome budget constraints and enhance its operations effectively.