ECOLE NATIONALE
SUPÉRIEURE
D'INFORMATIQUE

الجـــمهوريـــة الجزائريـــة الديمـــقراطيــــة الشـعـبيـة
وزارة التعليـــم العــــالي والبـــحث العلمـــي
People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Virtualization and Cloud Computing, VCL

2nd Year Specialty SIQ G02, 2CS SIQ2

LAB5A Report

Containerization with Linux LXC

Studied by:

HEDDADJI Nour El Imane

SOLTANI Meriem

E-mails :

jn_heddadji@esi.dz

jm_soltani@esi.dz

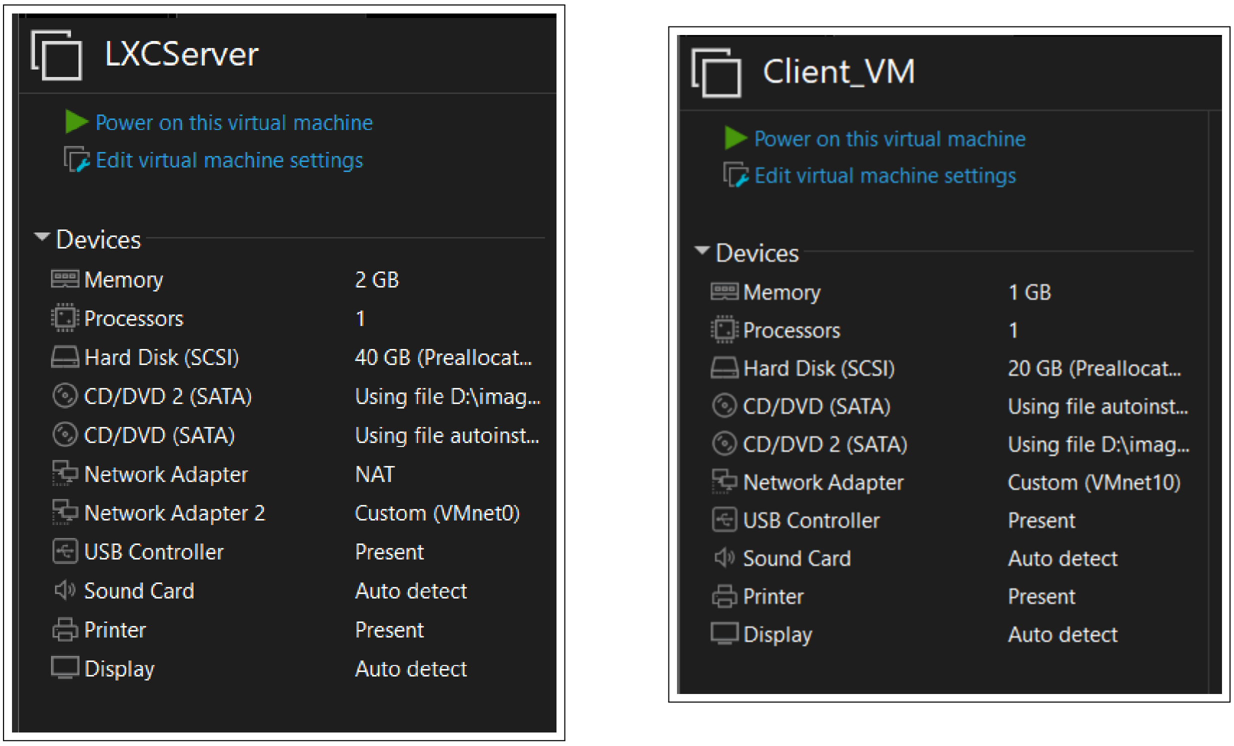
Contents

1. LXC Installation

1.1. Preparation

To conduct the lab, we set up:

1. **LXC Server** : This virtual machine will act as the LXC container engine.
2. **Client VM** : For testing.



LXCServer

Power on this virtual machine
Edit virtual machine settings

Devices

Memory	2 GB
Processors	1
Hard Disk (SCSI)	40 GB (Preallocat...
CD/DVD 2 (SATA)	Using file D:\imag...
CD/DVD (SATA)	Using file autoinst...
Network Adapter	NAT
Network Adapter 2	Custom (VMnet0)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

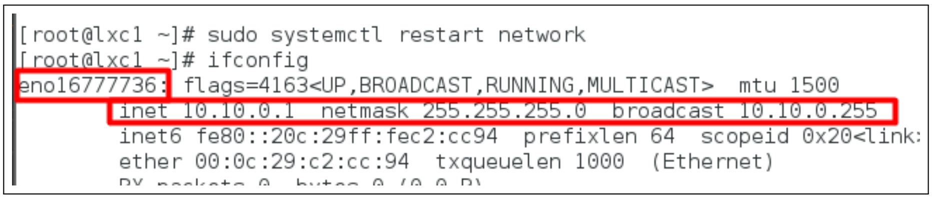
Client_VM

Power on this virtual machine
Edit virtual machine settings

Devices

Memory	1 GB
Processors	1
Hard Disk (SCSI)	20 GB (Preallocat...
CD/DVD (SATA)	Using file autoinst...
CD/DVD 2 (SATA)	Using file D:\imag...
Network Adapter	Custom (VMnet10)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Figure 1: Configuration



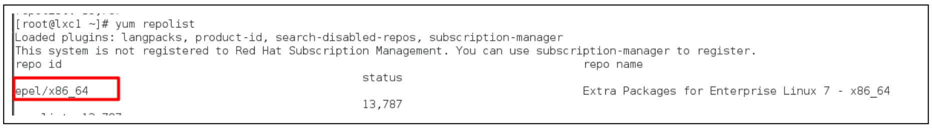
```
[root@lxc1 ~]# sudo systemctl restart network
[root@lxc1 ~]# ifconfig
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.0.1  netmask 255.255.255.0  broadcast 10.10.0.255
        inet6 fe80::20c:29ff:fec2:cc94  prefixlen 64  scopeid 0x20<link:
        ether 00:0c:29:c2:cc:94  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
```

1.2. Activating the EPEL repository

EPEL, or Extra Packages for Enterprise Linux, is a community-maintained repository that adds extra software to (RHEL) and its derivatives, like CentOS. It provides additional high-quality, open-source packages, expanding the software options available for users.

- LXC virtualization is not provided by the ISO image repository of Linux 7.
- LXC is supplied through the EPEL repositories. Therefore, the EPEL repository must be activated first from the internet.

```

 wget http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
 rpm -ivh epel-release-latest-7.noarch.rpm
 yum repolist
```



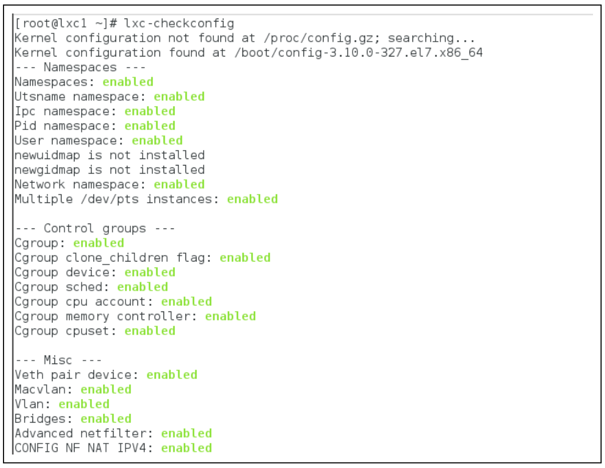
```
[root@lxc1 ~]# yum repolist
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
repo id                                                                           repo name
                                                  status
epel/x86_64                                                                       Extra Packages for Enterprise Linux 7 - x86_64
                                                  13,787
```

1.3. Installing Dependencies

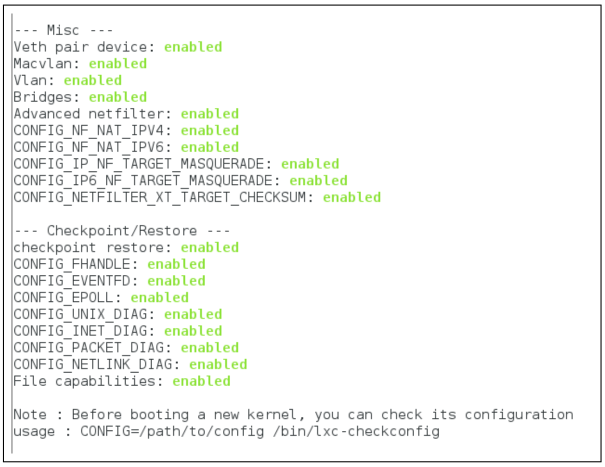
```

 sudo yum install epel-release
 sudo yum install --skip-broken debootstrap perl libvirt
```

1.4. Installing LXC Virtualization Solution

```

 sudo yum install lxc lxc-templates
 sudo yum install --skip-broken debootstrap perl libvirt
 sudo systemctl start lxc.service
 sudo systemctl start libvirtd
 # Check the status of the LXC service
 sudo systemctl status lxc.service
 # Verify the state of the LXC kernel virtualization by executing the following command:
 lxc-checkconfig
```



```
[root@lxc1 ~]# lxc-checkconfig
Kernel configuration not found at /proc/config.gz; searching...
Kernel configuration found at /boot/config-3.10.0-327.el7.x86_64
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
newuidmap is not installed
newgidmap is not installed
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
Bridges: enabled
Advanced netfilter: enabled
CONFIG NF NAT IPV4: enabled
```

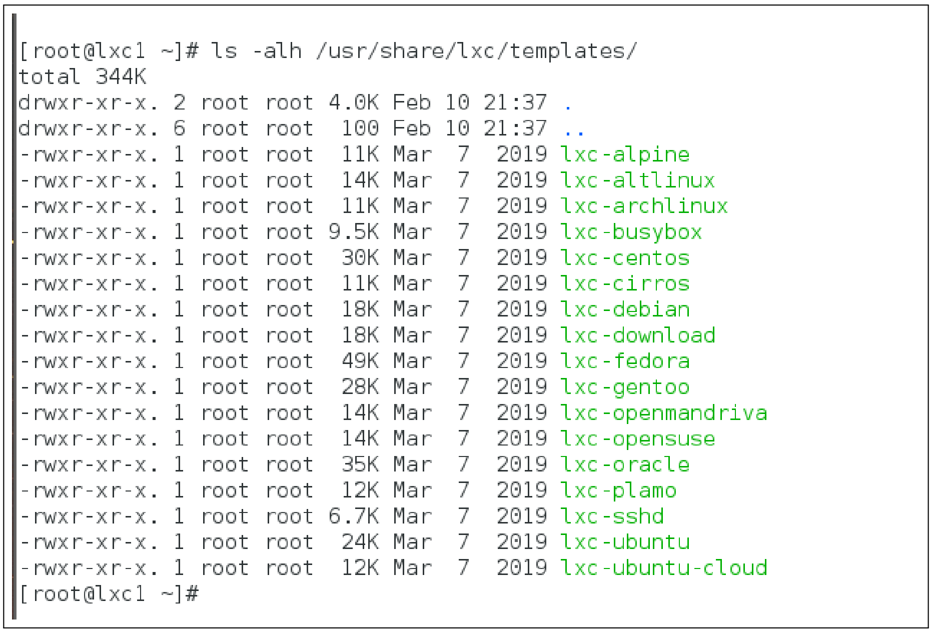


```
--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
Bridges: enabled
Advanced netfilter: enabled
CONFIG_NF_NAT_IPV4: enabled
CONFIG_NF_NAT_IPV6: enabled
CONFIG_IP_NF_TARGET_MASQUERADE: enabled
CONFIG_IP6_NF_TARGET_MASQUERADE: enabled
CONFIG_NETFILTER_XT_TARGET_CHECKSUM: enabled

--- Checkpoint/Restore ---
checkpoint restore: enabled
CONFIG_FHANDLE: enabled
CONFIG_EVENTFD: enabled
CONFIG_EPOLL: enabled
CONFIG_UNIX_DIAG: enabled
CONFIG_INET_DIAG: enabled
CONFIG_PACKET_DIAG: enabled
CONFIG_NETLINK_DIAG: enabled
File capabilities: enabled

Note : Before booting a new kernel, you can check its configuration
usage : CONFIG=/path/to/config /bin/lxc-checkconfig
```

2. LXC containers

2.1. List Available LXC Templates

```
    ls -alh /usr/share/lxc/templates
```



```
[root@lxc1 ~]# ls -alh /usr/share/lxc/templates/
total 344K
drwxr-xr-x. 2 root root 4.0K Feb 10 21:37 .
drwxr-xr-x. 6 root root  100 Feb 10 21:37 ..
-rwxr-xr-x. 1 root root  11K Mar  7  2019 lxc-alpine
-rwxr-xr-x. 1 root root  14K Mar  7  2019 lxc-altlinux
-rwxr-xr-x. 1 root root  11K Mar  7  2019 lxc-archlinux
-rwxr-xr-x. 1 root root 9.5K Mar  7  2019 lxc-busybox
-rwxr-xr-x. 1 root root  30K Mar  7  2019 lxc-centos
-rwxr-xr-x. 1 root root  11K Mar  7  2019 lxc-cirros
-rwxr-xr-x. 1 root root  18K Mar  7  2019 lxc-debian
-rwxr-xr-x. 1 root root  18K Mar  7  2019 lxc-download
-rwxr-xr-x. 1 root root  49K Mar  7  2019 lxc-fedora
-rwxr-xr-x. 1 root root  28K Mar  7  2019 lxc-gentoo
-rwxr-xr-x. 1 root root  14K Mar  7  2019 lxc-openmandriva
-rwxr-xr-x. 1 root root  14K Mar  7  2019 lxc-opensuse
-rwxr-xr-x. 1 root root  35K Mar  7  2019 lxc-oracle
-rwxr-xr-x. 1 root root  12K Mar  7  2019 lxc-plamo
-rwxr-xr-x. 1 root root 6.7K Mar  7  2019 lxc-sshd
-rwxr-xr-x. 1 root root  24K Mar  7  2019 lxc-ubuntu
-rwxr-xr-x. 1 root root  12K Mar  7  2019 lxc-ubuntu-cloud
[root@lxc1 ~]#
```

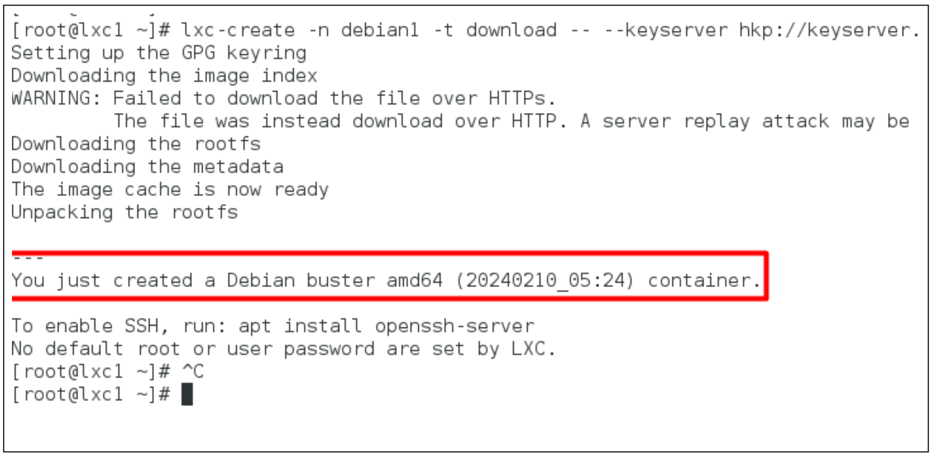
2.2. Creating LXC Containers

The command syntax for creating a new LXC container is

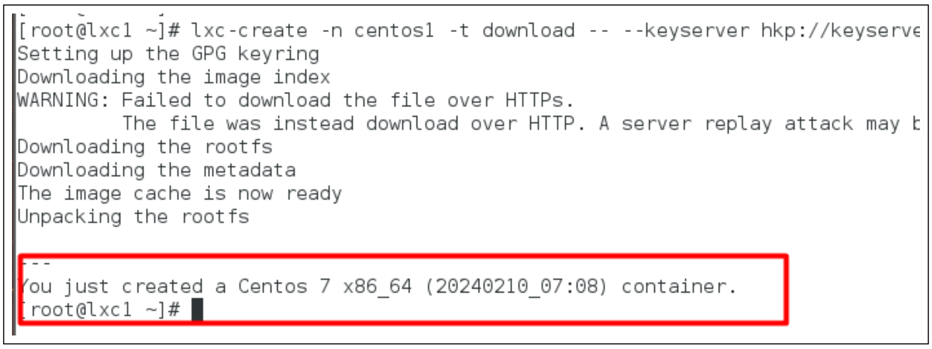
```

    lxc-create -n container_name -t container_template
```

Container Name	Command
debian1	lxc-create -n debian1 -t download – –keyserver hkp://keyserver.ubuntu.com:80 -d debian -r buster -a amd64
centos1	lxc-create -n centos1 -t download – –keyserver hkp://keyserver.ubuntu.com:80 -d centos -r 7 -a amd64
ubuntu1	lxc-create -n ubuntu1 -t download – –keyserver hkp://keyserver.ubuntu.com:80 -d ubuntu -r xenial -a amd64

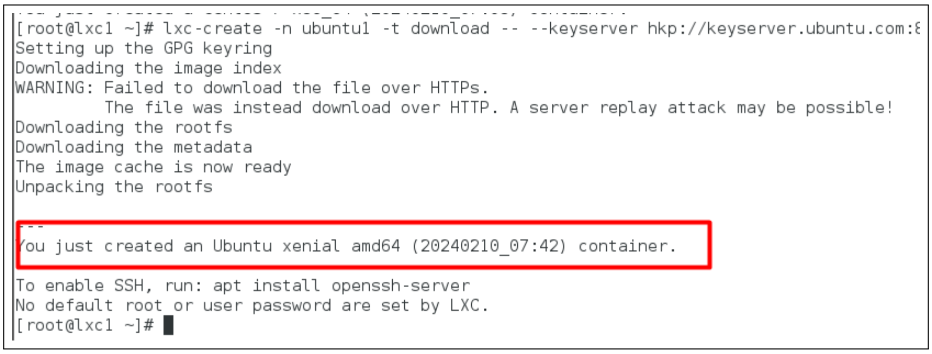
Table 1: LXC Container Creation Commands



```
[root@lxc1 ~]# lxc-create -n debian1 -t download -- --keyserver hkp://keyserver.
Setting up the GPG keyring
Downloading the image index
WARNING: Failed to download the file over HTTPs.
         The file was instead download over HTTP. A server replay attack may be
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created a Debian buster amd64 (20240210_05:24) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
[root@lxc1 ~]# ^C
[root@lxc1 ~]#
```



```
[root@lxc1 ~]# lxc-create -n centos1 -t download -- --keyserver hkp://keyserve
Setting up the GPG keyring
Downloading the image index
WARNING: Failed to download the file over HTTPs.
         The file was instead download over HTTP. A server replay attack may t
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created a Centos 7 x86_64 (20240210_07:08) container.
[root@lxc1 ~]#
```

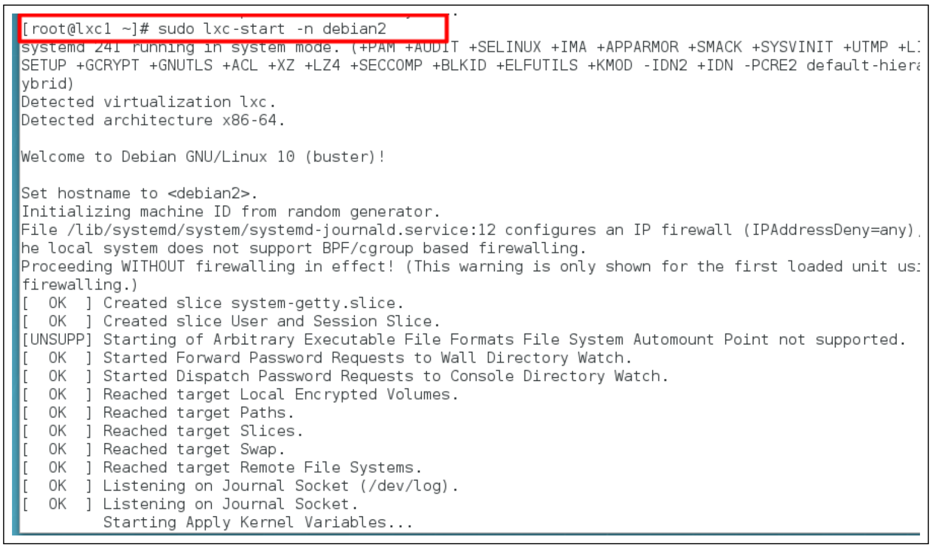


```
[root@lxc1 ~]# lxc-create -n ubuntu1 -t download -- --keyserver hkp://keyserver.ubuntu.com:8
Setting up the GPG keyring
Downloading the image index
WARNING: Failed to download the file over HTTPs.
         The file was instead download over HTTP. A server replay attack may be possible!
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created an Ubuntu xenial amd64 (20240210_07:42) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
[root@lxc1 ~]#
```

3. Managing Containers: Starting, Opening, and Stopping

3.1. Start and Open Containers

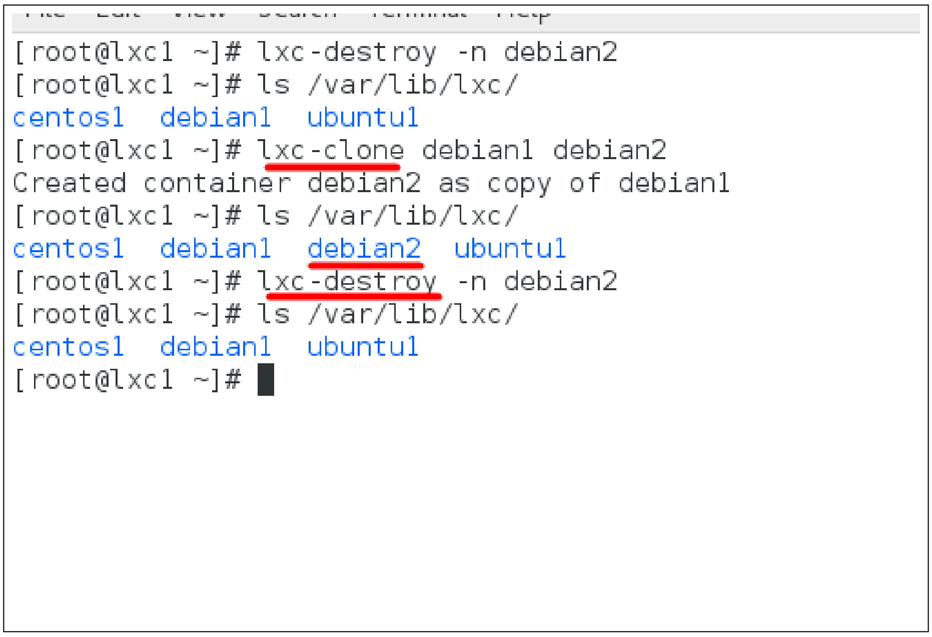
```

    # Start the newly created container in the background
    lxc-start -n debian1 -d
    # Open a console to the container:
    lxc-console -n debian1

.
```



```
[root@lxc1 ~]# sudo lxc-start -n debian2
systemd 241 running in system mode. (+PAM +AUDIT +SELINUX +IMA +APPARMOR +SMACK +SYSVINIT +UTMP +L
SETUP +GCRYPT +GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS +KMOD -IDN2 +IDN -PCRE2 default-hier
ybrid)
Detected virtualization lxc.
Detected architecture x86-64.

Welcome to Debian GNU/Linux 10 (buster)!

Set hostname to <debian2>.
Initializing machine ID from random generator.
File /lib/systemd/system/systemd-journald.service:12 configures an IP firewall (IPAddressDeny=any),
he local system does not support BPF/cgroup based firewalling.
Proceeding WITHOUT firewalling in effect! (This warning is only shown for the first loaded unit us
firewalling.)
[  OK  ] Created slice system-getty.slice.
[  OK  ] Created slice User and Session Slice.
[UNSUPP] Starting of Arbitrary Executable File Formats File System Automount Point not supported.
[  OK  ] Started Forward Password Requests to Wall Directory Watch.
[  OK  ] Started Dispatch Password Requests to Console Directory Watch.
[  OK  ] Reached target Local Encrypted Volumes.
[  OK  ] Reached target Paths.
[  OK  ] Reached target Slices.
[  OK  ] Reached target Swap.
[  OK  ] Reached target Remote File Systems.
[  OK  ] Listening on Journal Socket (/dev/log).
[  OK  ] Listening on Journal Socket.
         Starting Apply Kernel Variables...
```

3.2. Clone and Destroy Containers

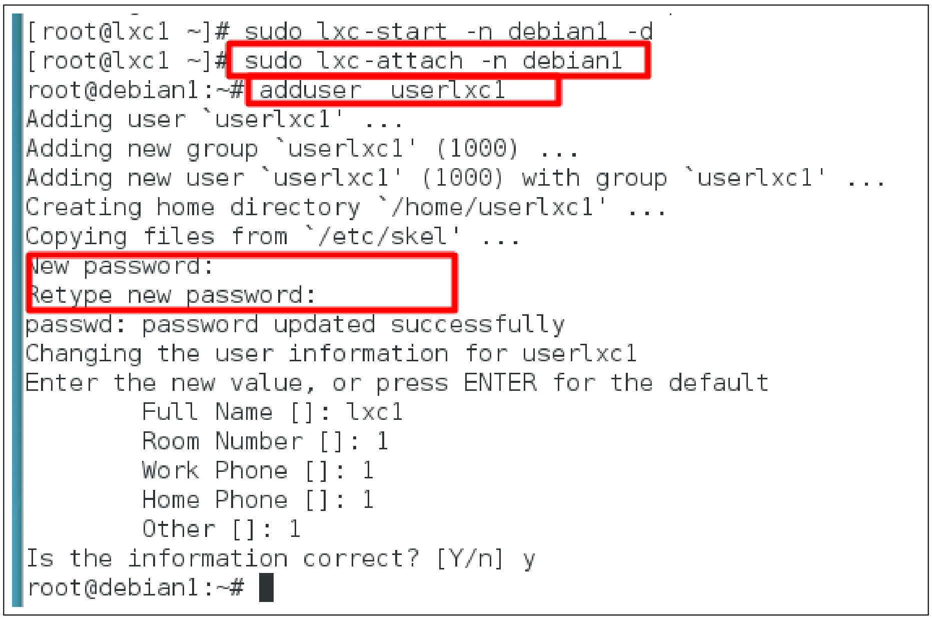
```

    # Clone a container:
    lxc-clone debian1 debian2
    # Destroy (delete) a container:
    lxc-destroy -n debian2
.
```



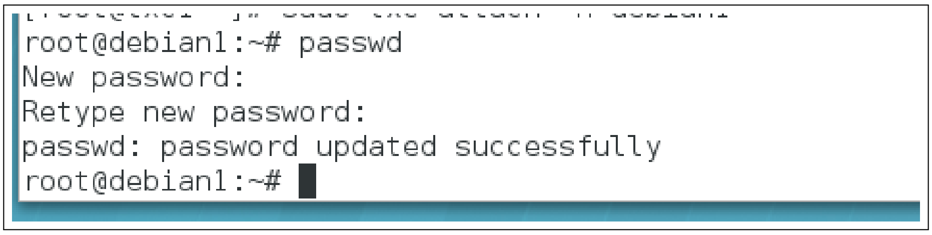
```
[root@lxc1 ~]# lxc-destroy -n debian2
[root@lxc1 ~]# ls /var/lib/lxc/
centos1  debian1  ubuntu1
[root@lxc1 ~]# lxc-clone debian1 debian2
Created container debian2 as copy of debian1
[root@lxc1 ~]# ls /var/lib/lxc/
centos1  debian1  debian2  ubuntu1
[root@lxc1 ~]# lxc-destroy -n debian2
[root@lxc1 ~]# ls /var/lib/lxc/
centos1  debian1  ubuntu1
[root@lxc1 ~]# 
```

3.3. Setting passwords

```

    passwd
.
```



```
[root@lxc1 ~]# sudo lxc-start -n debian1 -d
[root@lxc1 ~]# sudo lxc-attach -n debian1
root@debian1:~# adduser  userlxc1
Adding user `userlxc1' ...
Adding new group `userlxc1' (1000) ...
Adding new user `userlxc1' (1000) with group `userlxc1' ...
Creating home directory `/home/userlxc1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for userlxc1
Enter the new value, or press ENTER for the default
        Full Name []: lxc1
        Room Number []: 1
        Work Phone []: 1
        Home Phone []: 1
        Other []: 1
Is the information correct? [Y/n] y
root@debian1:~# 
```

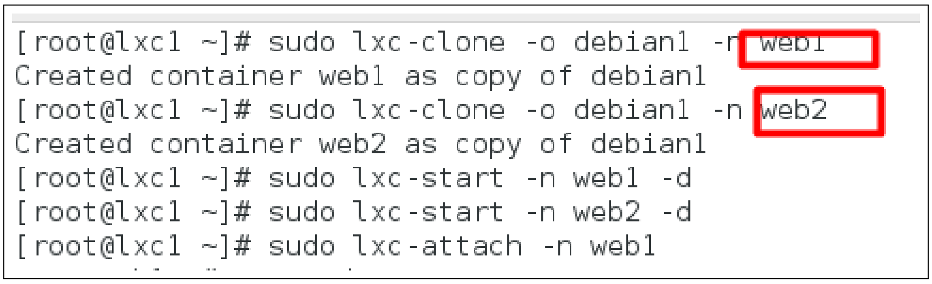


```
root@debian1:~# passwd
New password:
Retype new password:
passwd: password updated successfully
root@debian1:~#
```

4. Hosting two Apache (httpd) web servers on separate LXC containers

4.1. Cloning existing Debian container

```

    # Clone a container:
    sudo lxc-clone -o debian1 -n web1
    sudo lxc-clone -o debian1 -n web2
.
```

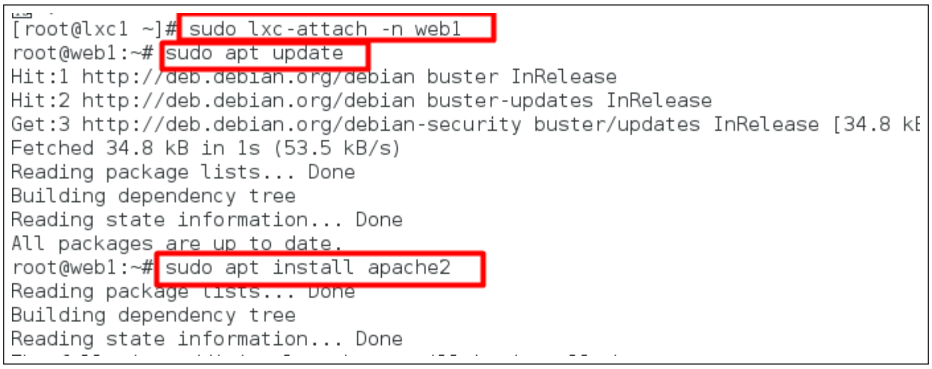


```
[root@lxc1 ~]# sudo lxc-clone -o debian1 -n web1
Created container web1 as copy of debian1
[root@lxc1 ~]# sudo lxc-clone -o debian1 -n web2
Created container web2 as copy of debian1
[root@lxc1 ~]# sudo lxc-start -n web1 -d
[root@lxc1 ~]# sudo lxc-start -n web2 -d
[root@lxc1 ~]# sudo lxc-attach -n web1
```

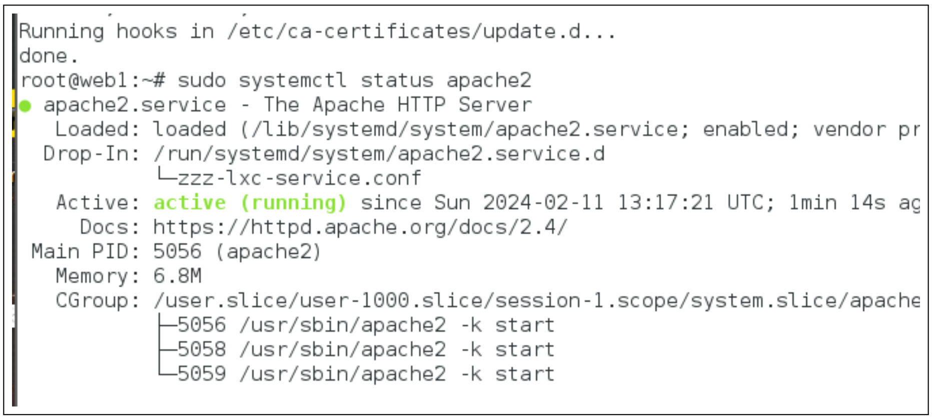
4.2. Installing Apache (httpd) on each container

```

    sudo lxc-start -n web1 -d
    sudo lxc-start -n web2 -d
    sudo lxc-attach -n web1
    sudo apt update
    sudo apt install apache2
    # Verify
    sudo systemctl status apache2
    exit
    sudo lxc-attach -n web2
    sudo apt update
    sudo apt install apache2
    sudo systemctl status apache2
    exit

.
```

Container web1

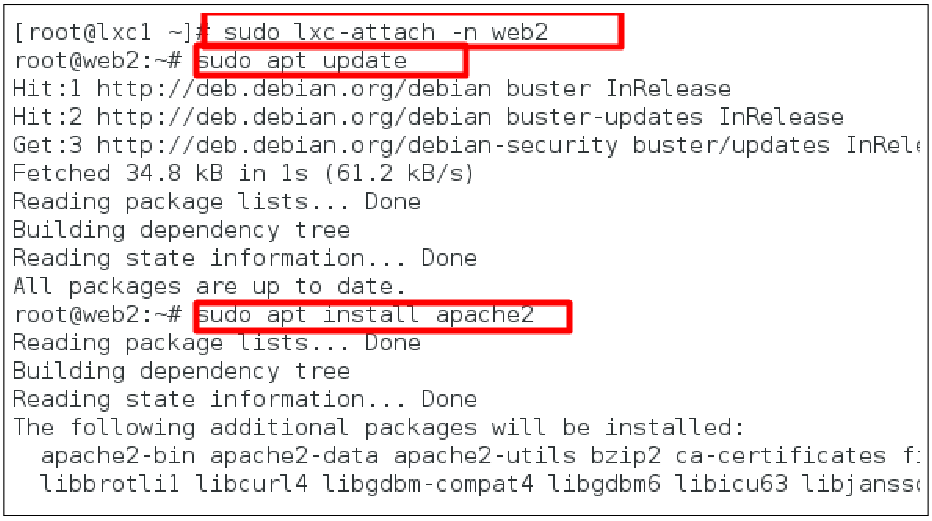


```
[root@lxc1 ~]# sudo lxc-attach -n web1
root@web1:~# sudo apt update
Hit:1 http://deb.debian.org/debian buster InRelease
Hit:2 http://deb.debian.org/debian buster-updates InRelease
Get:3 http://deb.debian.org/debian-security buster/updates InRelease [34.8 kE
Fetched 34.8 kB in 1s (53.5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@web1:~# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

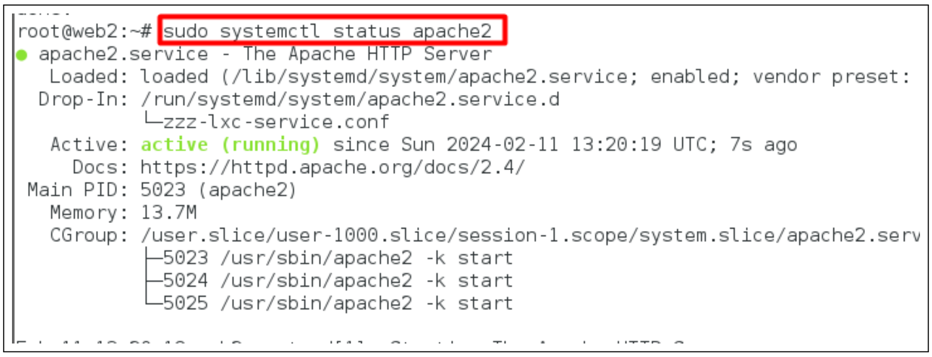


```
Running hooks in /etc/ca-certificates/update.d...
done.
root@web1:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pr
  Drop-In: /run/systemd/system/apache2.service.d
           └─zzz-lxc-service.conf
   Active: active (running) since Sun 2024-02-11 13:17:21 UTC; 1min 14s ag
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 5056 (apache2)
   Memory: 6.8M
   CGroup: /user.slice/user-1000.slice/session-1.scope/system.slice/apache
           ├─5056 /usr/sbin/apache2 -k start
           ├─5058 /usr/sbin/apache2 -k start
           └─5059 /usr/sbin/apache2 -k start
```

Container web2



```
[root@lxc1 ~]# sudo lxc-attach -n web2
root@web2:~# sudo apt update
Hit:1 http://deb.debian.org/debian buster InRelease
Hit:2 http://deb.debian.org/debian buster-updates InRelease
Get:3 http://deb.debian.org/debian-security buster/updates InRel
Fetched 34.8 kB in 1s (61.2 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@web2:~# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 ca-certificates f
  libbrotli1 libcurl4 libgdbm-compat4 libgdbm6 libicu63 libjanss
```

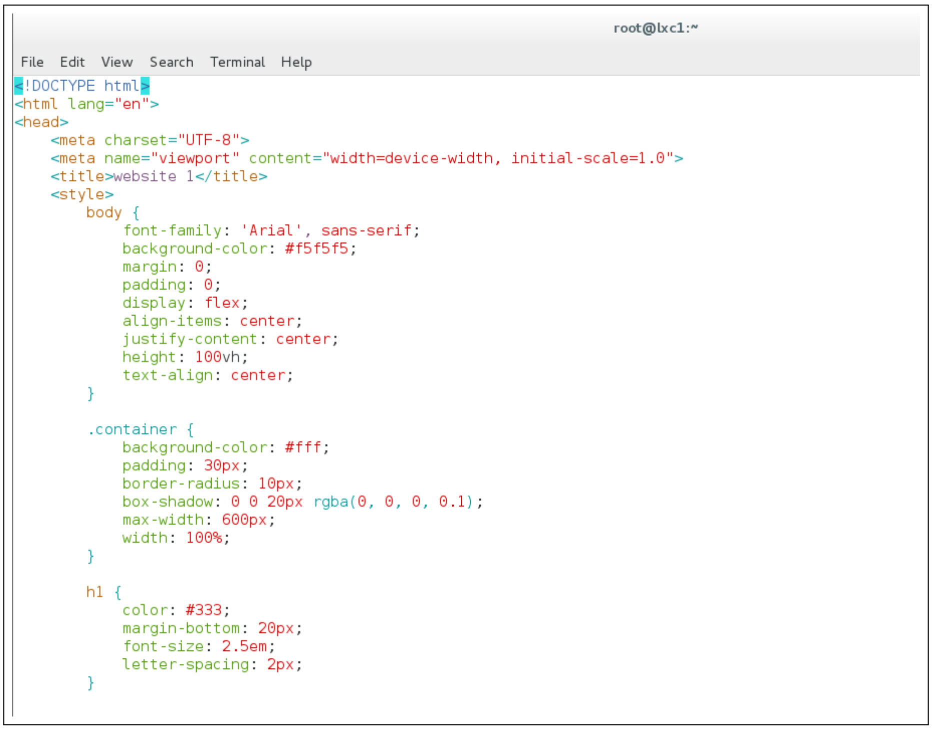


```
root@web2:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /run/systemd/system/apache2.service.d
           └─zzz-lxc-service.conf
   Active: active (running) since Sun 2024-02-11 13:20:19 UTC; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 5023 (apache2)
   Memory: 13.7M
   CGroup: /user.slice/user-1000.slice/session-1.scope/system.slice/apache2.serv
           ├─5023 /usr/sbin/apache2 -k start
           ├─5024 /usr/sbin/apache2 -k start
           └─5025 /usr/sbin/apache2 -k start
```

4.3. Creating the websites

We will use a personal template for web1 and the default one for web2.

```

    sudo lxc-attach -n web1
    # Create folder for web1
    sudo mkdir /var/www/html/web1
    vi /var/www/html/web1/index.html
    # insert the html code for website 1
    exit
    sudo mkdir /var/www/html/web2
    vi /var/www/html/web2/index.html
    # we will keep the default page
    exit

.
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>website 1</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
            text-align: center;
        }

        .container {
            background-color: #fff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
            max-width: 600px;
            width: 100%;
        }

        h1 {
            color: #333;
            margin-bottom: 20px;
            font-size: 2.5em;
            letter-spacing: 2px;
        }
```

You can access the full code on the GitHub repository: GitHub Repository

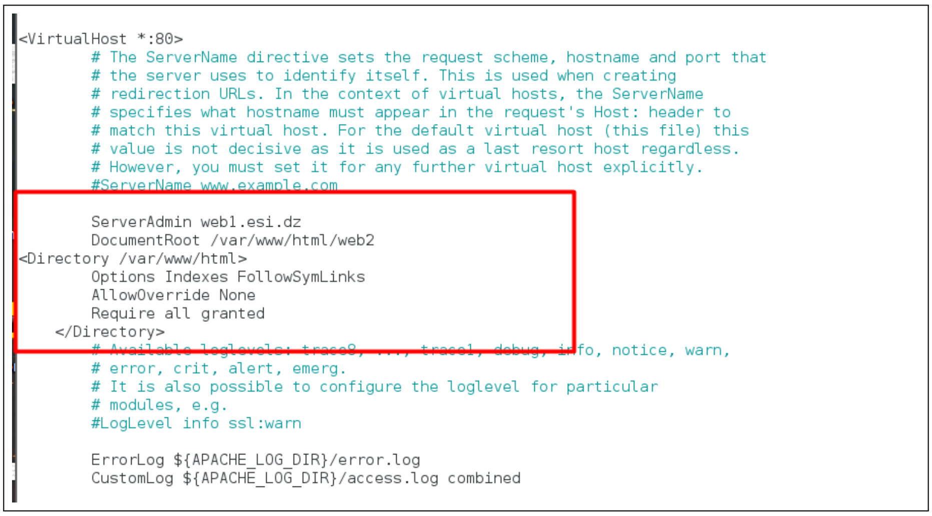
4.4. Configuring Apache for different sites

```

    sudo lxc-attach -n web1
    //Firewall
    apt-get install ufw
    ufw allow 80
    # Edit the Apache configuration file
    apt-get install nano
    nano /etc/apache2/sites-available/000-default.conf
    service apache2 restart

.
```

Let's add section to the 000-default.conf and ddit the hosts file to include the mapping for web1.esi.dz and web2.esi.dz using the command **sudo nano /etc/hosts**

```

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/web1
     ServerName web1.esi.dz
#-----------
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted

</Directory>
```

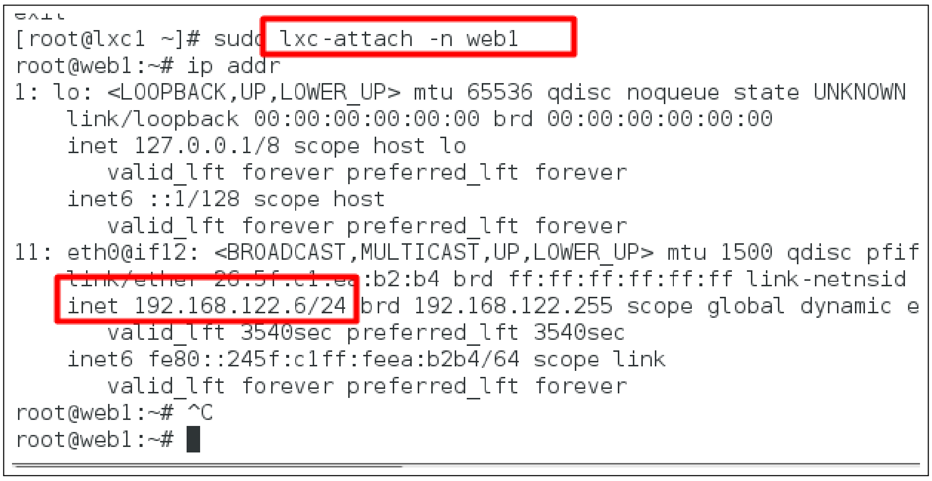
Listing 1: Apache Configuration for web1



```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin web1.esi.dz
        DocumentRoot /var/www/html/web2
<Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

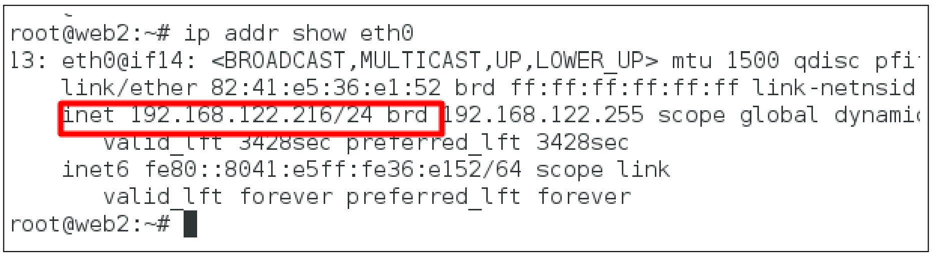
4.5. Accessing the container

```

    sudo lxc-attach -n web1
    //get the ip adress of the container
    ip addr
.
```



```
exit
[root@lxc1 ~]# sudo lxc-attach -n web1
root@web1:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
11: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfif
    link/ether 26:5f:c1:ea:b2:b4 brd ff:ff:ff:ff:ff:ff link-netnsid
    inet 192.168.122.6/24 brd 192.168.122.255 scope global dynamic e
       valid_lft 3540sec preferred_lft 3540sec
    inet6 fe80::245f:c1ff:feea:b2b4/64 scope link
       valid_lft forever preferred_lft forever
root@web1:~# ^C
root@web1:~#
```

Figure 2: Container web1 addr

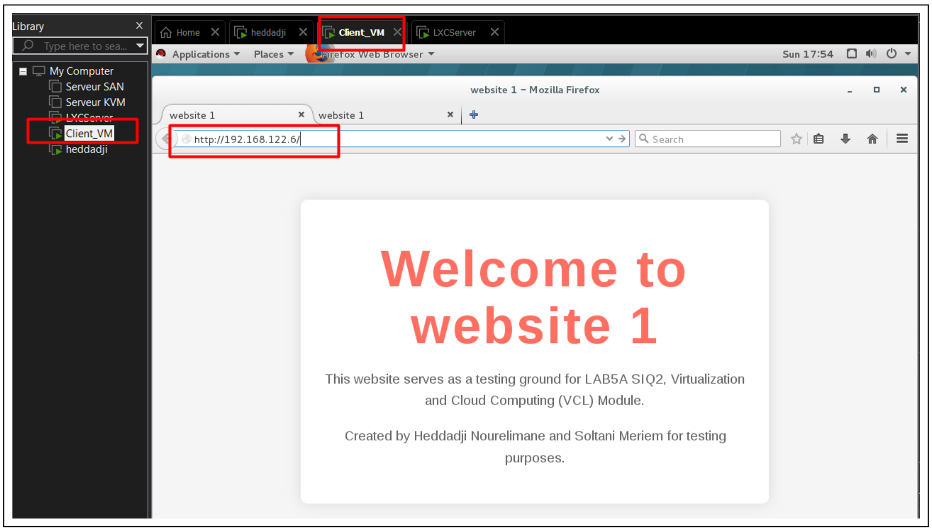


```
root@web2:~# ip addr show eth0
13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfi
    link/ether 82:41:e5:36:e1:52 brd ff:ff:ff:ff:ff:ff link-netnsid
    inet 192.168.122.216/24 brd 192.168.122.255 scope global dynami
       valid_lft 3428sec preferred_lft 3428sec
    inet6 fe80::8041:e5ff:fe36:e152/64 scope link
       valid_lft forever preferred_lft forever
root@web2:~#
```

Figure 3: Container web2 addr

Let's Test in vm client :

Conatiner web1

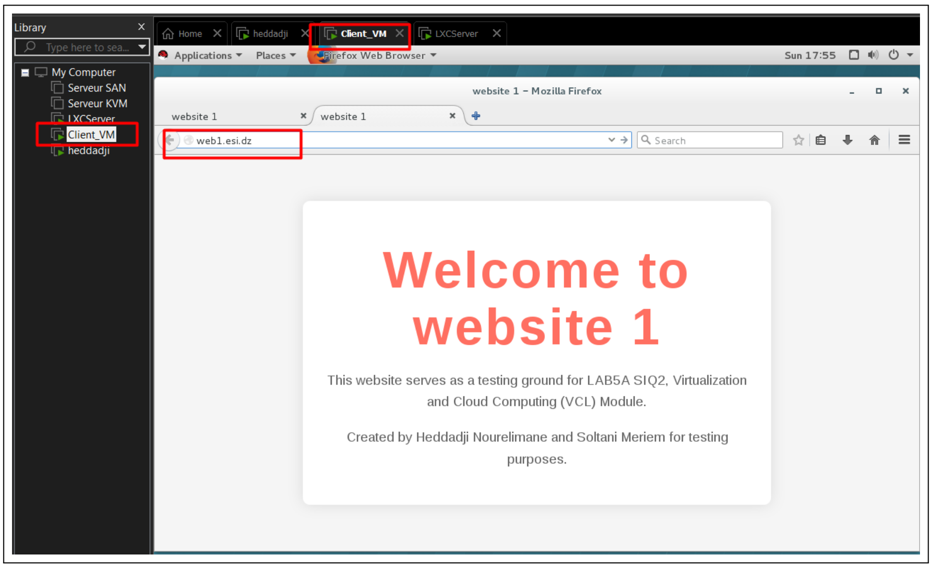


http://192.168.122.6/

Welcome to website 1

This website serves as a testing ground for LAB5A SIQ2, Virtualization and Cloud Computing (VCL) Module.

Created by Heddadji Nourelimane and Soltani Meriem for testing purposes.



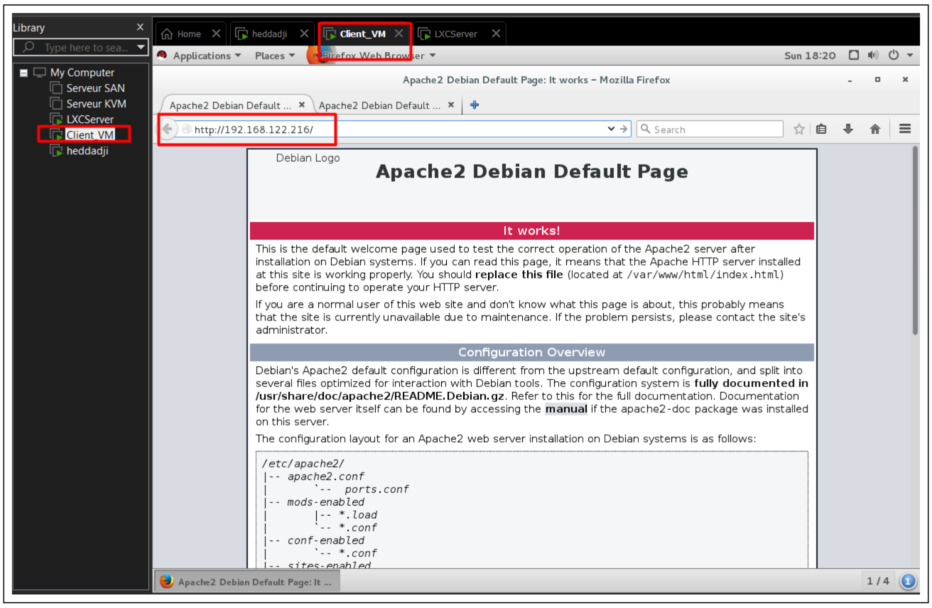
web1.esi.dz

Welcome to website 1

This website serves as a testing ground for LAB5A SIQ2, Virtualization and Cloud Computing (VCL) Module.

Created by Heddadji Nourelimane and Soltani Meriem for testing purposes.

Container web2



Debian Logo

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

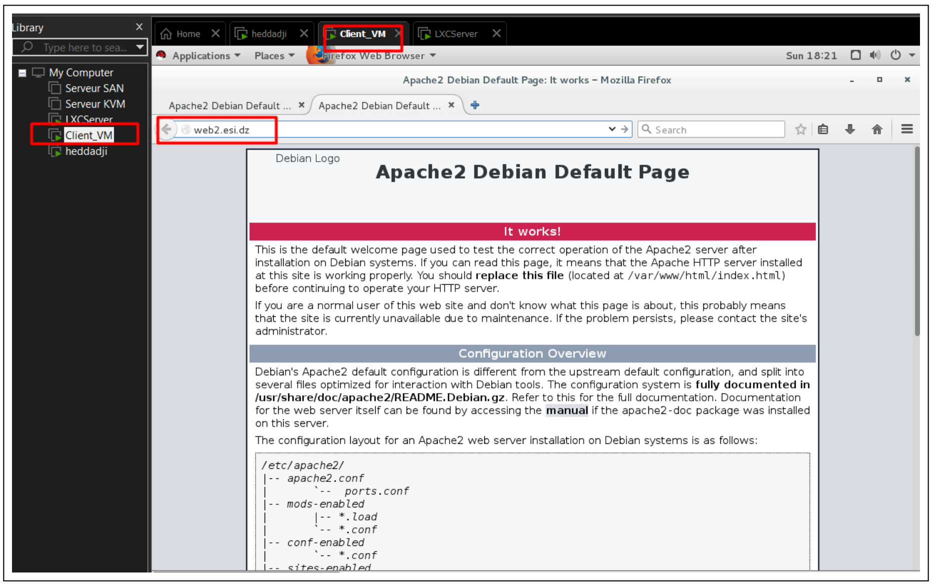
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```



Debian Logo

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```