

Virtualización y Consolidación de Servidores

Trabajo Práctico Final

Año 2023

Profesor: Ing. Luis María Carriles

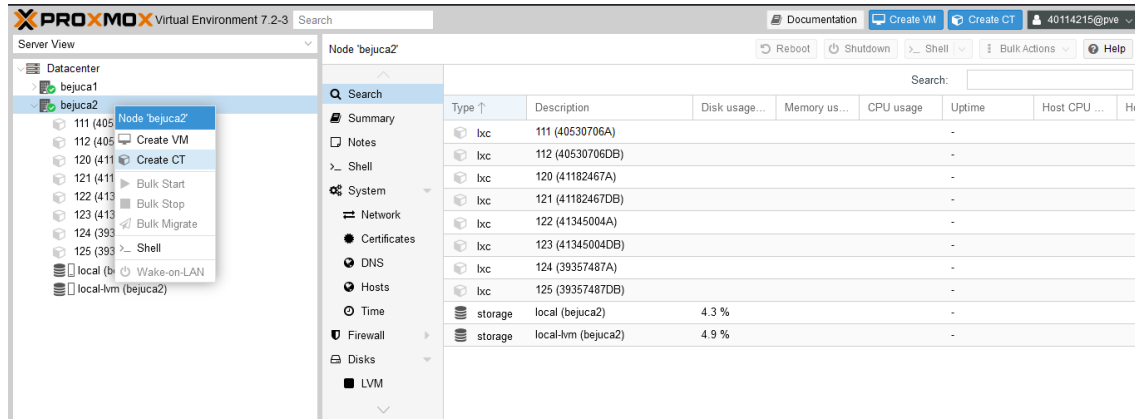


Alumno: Tomás Issolio

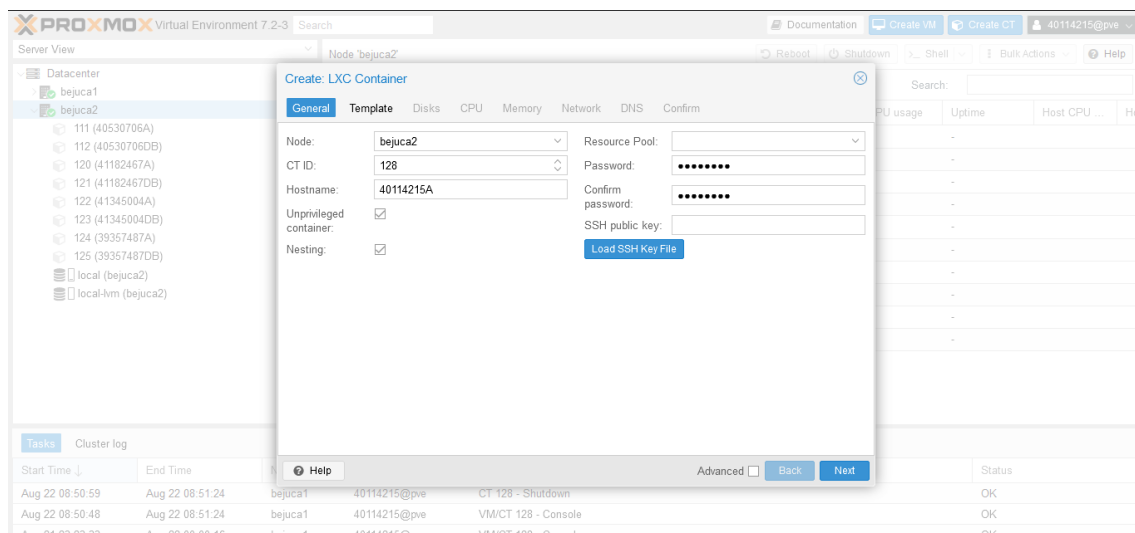
Legajo: 50049

Primera parte: Creando los contenedores

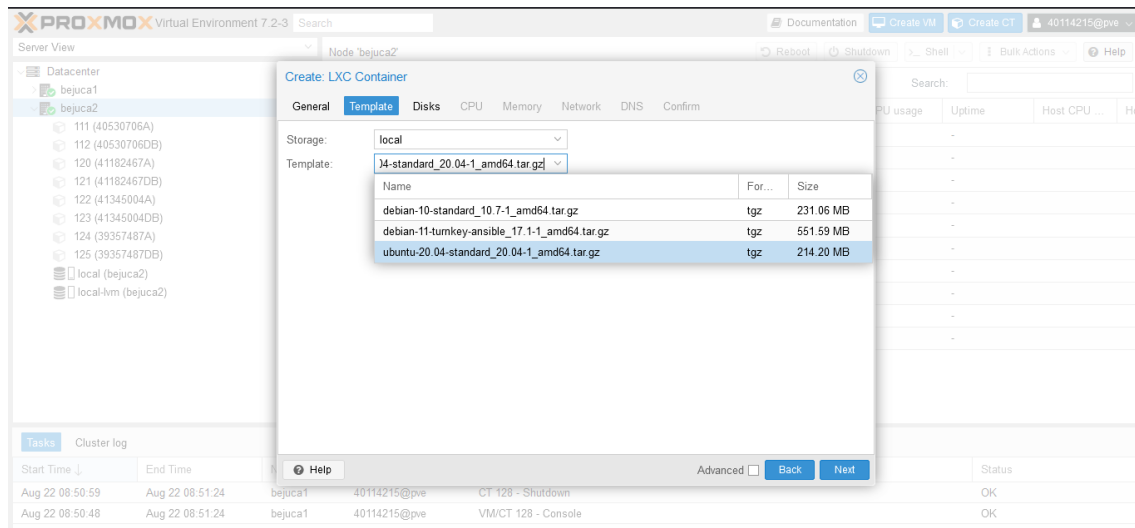
i) Creamos un nuevo contenedor “Create CT”. En este caso, en el nodo “bejuca2”.



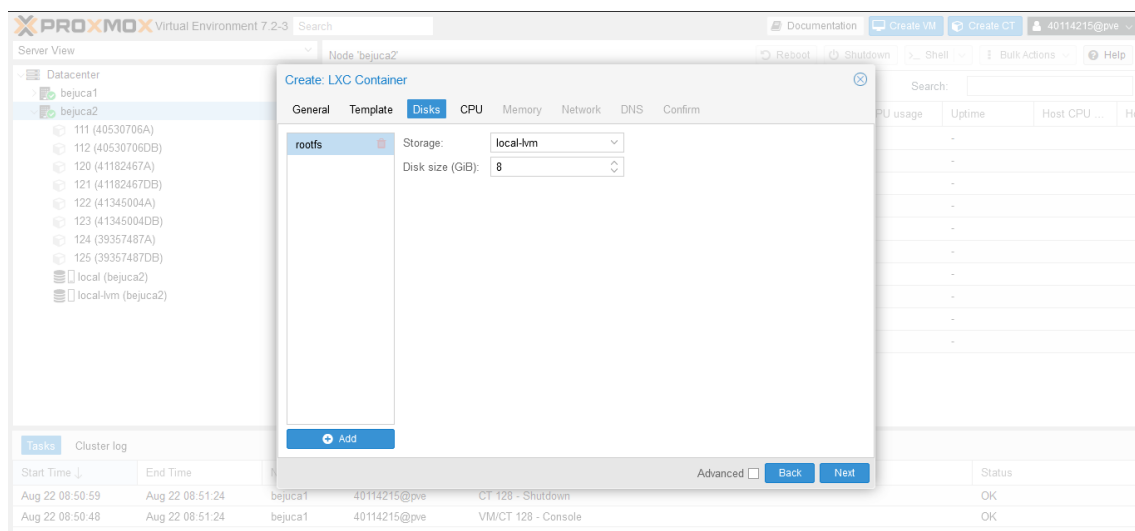
ii) Asignamos nombre al contenedor y creamos la contraseña de ingreso (será la que se use al iniciar el contenedor; el usuario será “root” por defecto).



iii) Elijo como S.O para el contenedor a Ubuntu, por considerarlo la opción más amigable para interacción con el usuario entre las disponibles.



iv) Se configura el contenedor con los 8 GB de almacenamiento, como fue especificado en el enunciado del trabajo.



vi) Se configura el contenedor con 1 procesador, como fue especificado en el enunciado del trabajo.

Create: LXC Container

General

Template

Disks

CPU

Memory

Network

DNS

Confirm

Cores:

1

Help

Advanced ☐

Back

Next

vii) Se configura el contenedor con 128 MB de memoria RAM, como fue especificado en el enunciado del trabajo; además, se agrega un margen de 128 MB para swapping.

Create: LXC Container

General

Template

Disks

CPU

Memory

Network

DNS

Confirm

Memory (MiB):

128

Swap (MiB):

128

Help

Advanced ☐

Back

Next

viii) Se configura el contenedor para tener dirección IP asignada mediante DHCP, como fue especificado en el enunciado del trabajo.

Create: LXC Container

General

Template

Disks

CPU

Memory

Network

DNS

Confirm

Name:

eth0

MAC address:

auto

Bridge:

vmbr0

VLAN Tag:

no VLAN

Rate limit (MB/s):

unlimited

Firewall:

☒

IPv4:

☐ Static ☒ DHCP

IPv4/CIDR:

Gateway (IPv4):

IPv6:

☐ Static ☒ DHCP ☐ SLAAC

IPv6/CIDR:

Gateway (IPv6):

?

 Help

Advanced ☐

Back

Next

ix) La sección DNS se deja en vacío.

Create: LXC Container

General
Template
Disks
CPU
Memory
Network
DNS
Confirm

DNS domain: use host settings

DNS servers: use host settings

Advanced
Back
Next

x) Se observa que los datos del contenedor sean correctos y se procede a confirmar su creaci3n.

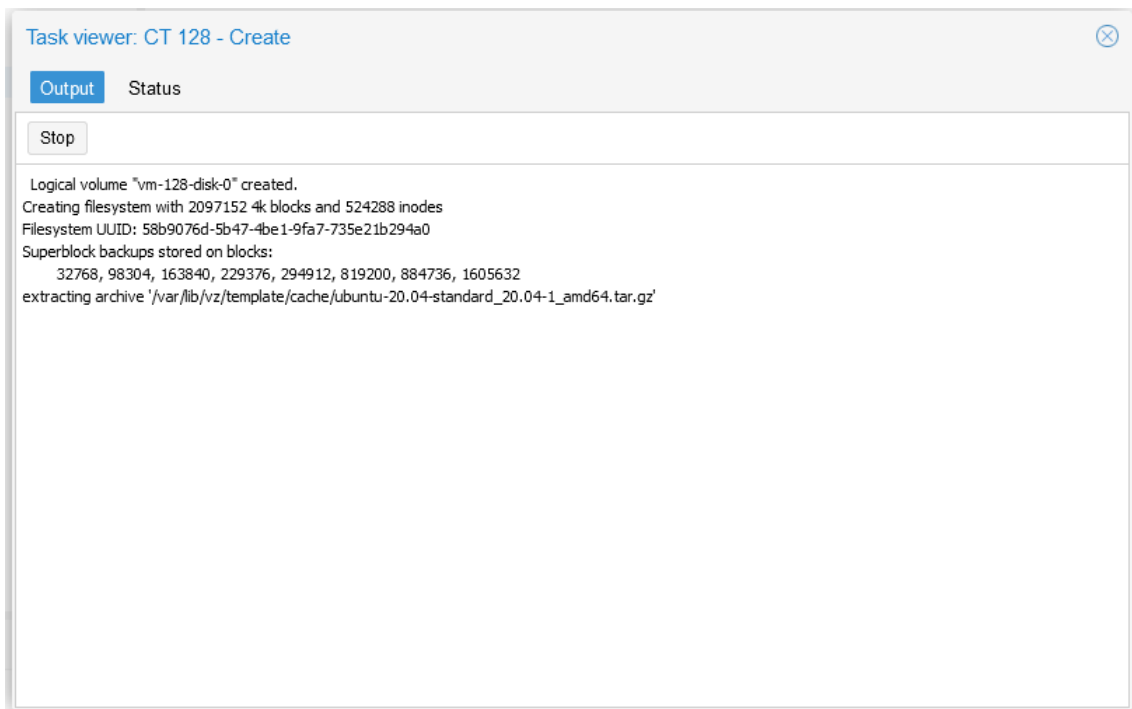
Create: LXC Container

General
Template
Disks
CPU
Memory
Network
DNS
Confirm

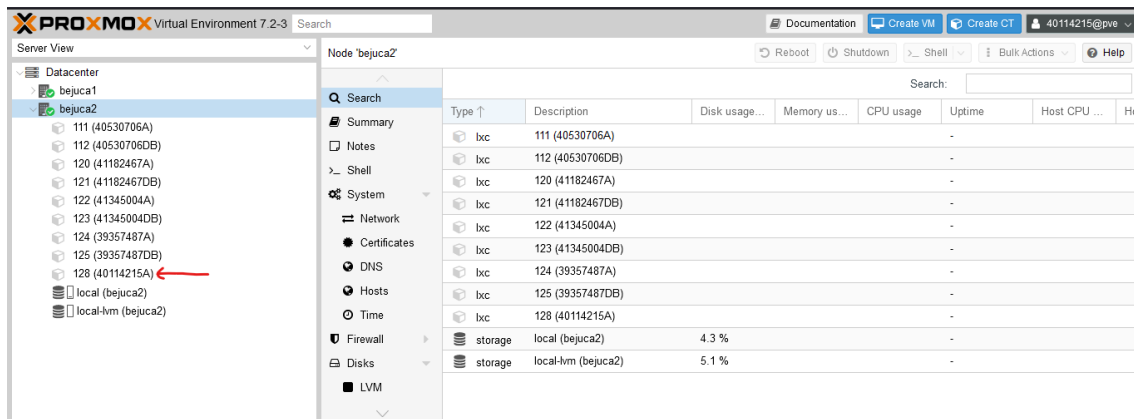
Key ↑	Value
cores	1
features	nesting=1
hostname	40114215A
memory	128
net0	name=eth0,bridge=vibr0,firewall=1,ip6=dhcp,ip=dhcp
nodename	bejuca2
ostemplate	local:vztmpl/ubuntu-20.04-standard_20.04-1_amd64.tar.gz
pool	
rootfs	local-lvm:8
swap	128
unprivileged	1
vmid	128

☐ Start after created

Advanced
Back
Finish



xi) Podemos ver que el contenedor ha sido creado.



xii) Repetimos el proceso para el contenedor correspondiente a la base de datos.

Create: LXC Container

General
Template
Disks
CPU
Memory
Network
DNS
Confirm

Key ↑	Value
cores	1
features	nesting=1
hostname	40114215DB
memory	128
net0	name=eth0,bridge=vbr0,firewall=1,ip6=dhcp,ip=dhcp
nodename	bejuca2
ostemplate	local:vztmpl/ubuntu-20.04-standard_20.04-1_amd64.tar.gz
pool	
rootfs	local-lvm:8
swap	128
unprivileged	1
vmid	129

☐ Start after created

Advanced ☐
Back
Finish

PROXMOX Virtual Environment 7.2-3

Documentation
Create VM
Create CT
40114215@pve

Server View
Node 'bejuca2'
Reboot
Shutdown
Shell
Bulk Actions
Help

Datacenter
bejuca1
bejuca2
111 (40530706A)
112 (40530706DB)
120 (41182467A)
121 (41182467DB)
122 (41345004A)
123 (41345004DB)
124 (39357487A)
125 (39357487DB)
128 (40114215A)
129 (40114215DB) ←
local (bejuca2)
local-lvm (bejuca2)

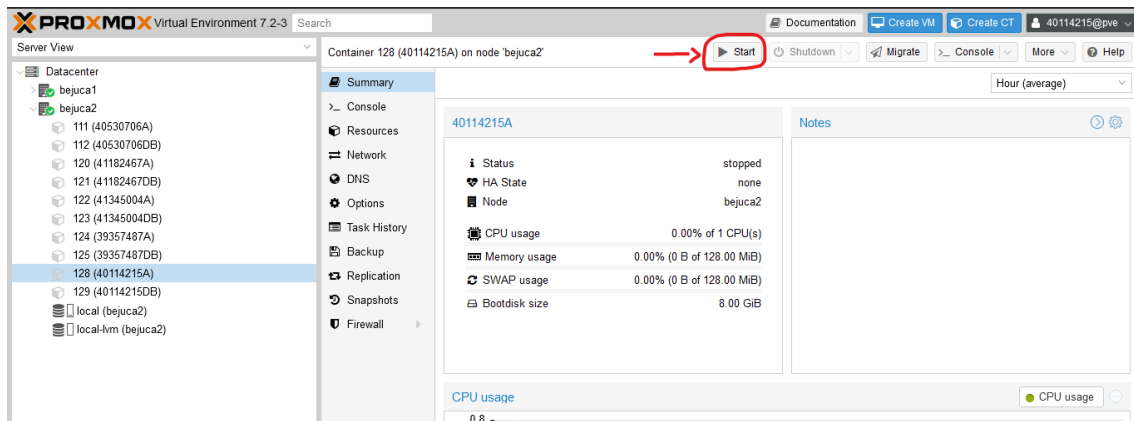
Search
Summary
Notes
Shell
System
Network
Certificates
DNS
Hosts
Time
Firewall
Disks
LVM

Type ↑
Description
Disk usage...
Memory us...
CPU usage
Uptime
Host CPU ...
Ho

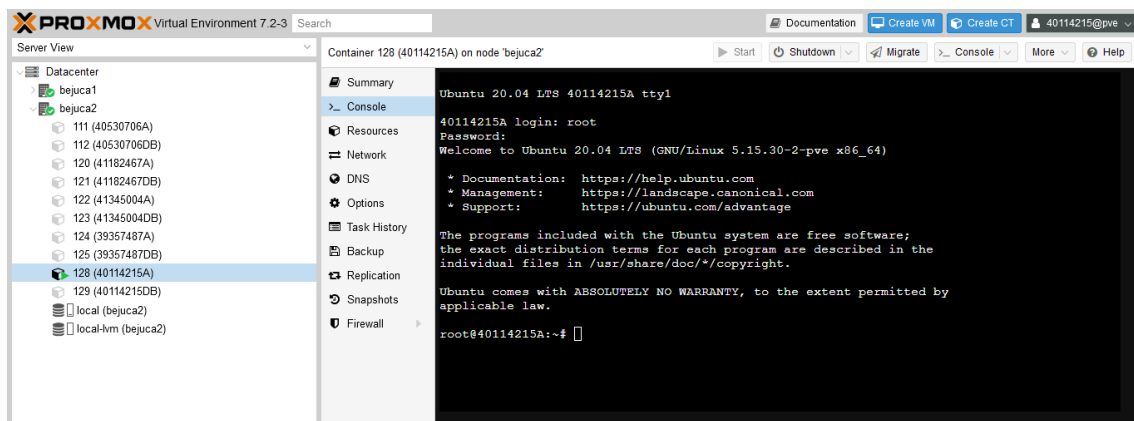
lxc	111 (40530706A)				-	
lxc	112 (40530706DB)				-	
lxc	120 (41182467A)				-	
lxc	121 (41182467DB)				-	
lxc	122 (41345004A)				-	
lxc	123 (41345004DB)				-	
lxc	124 (39357487A)				-	
lxc	125 (39357487DB)				-	
lxc	128 (40114215A)				-	
lxc	129 (40114215DB)				-	
storage	local (bejuca2)	4.3 %			-	
storage	local-lvm (bejuca2)	5.3 %			-	

Segunda parte: Trabajando con el contenedor del blog personal

i) Se inicia la máquina virtual.



ii) Se procede a abrir la consola y a registrarse con usuario y contraseña (la configurada al crear el contenedor).

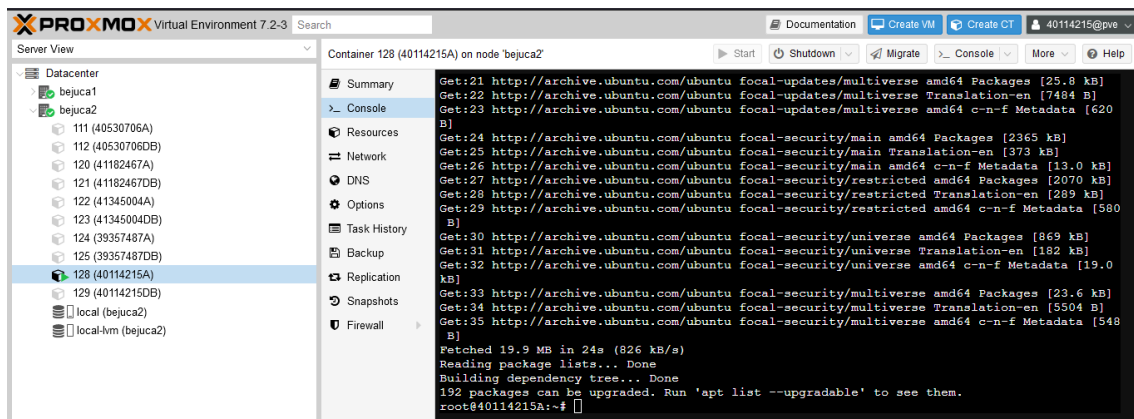
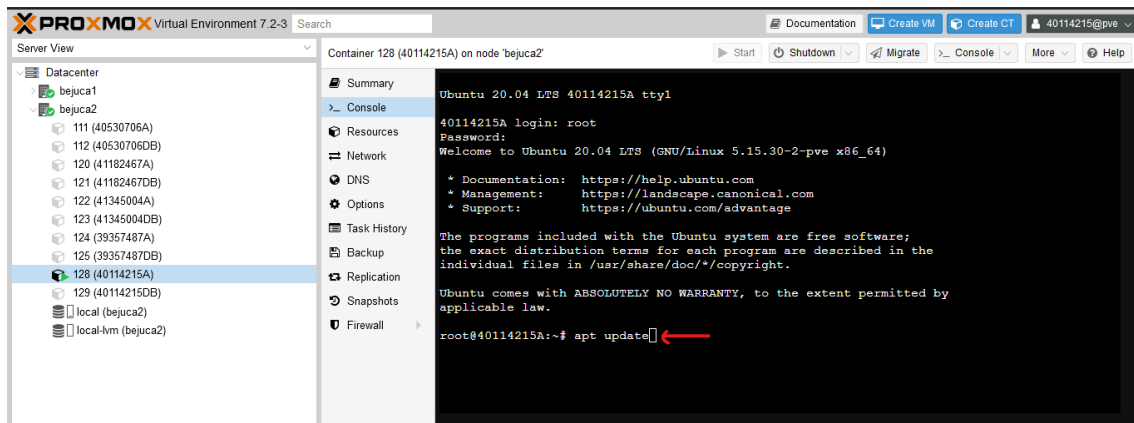


iii) Se procede a ejecutar el comando "apt update".

El comando APT (Advanced Package Tool por sus siglas en inglés que traducen Herramienta Avanzada de Empaquetado) es un elemento de línea de comandos creado por el proyecto Debian con el objetivo de permitirle a los usuarios gestionar y administrar los paquetes de sus distribuciones de Linux Debian.

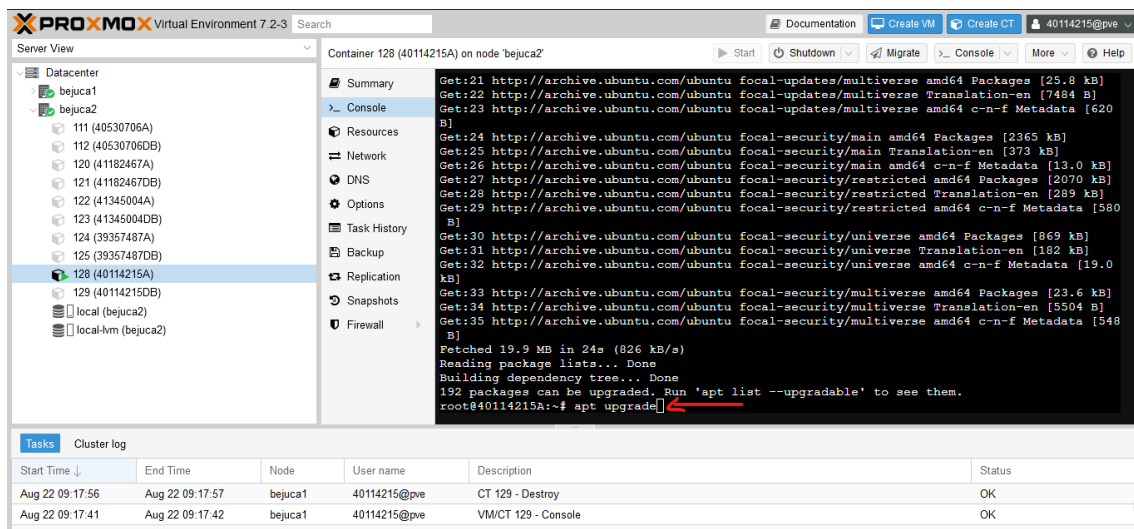
Dentro de las funciones principales de este comando, se puede encontrar que Linux APT te permite instalar, actualizar, eliminar paquetes o programas del sistema en Linux, entre otras funciones. Los repositorios APT Linux hacen referencia a una colección de paquetes Deb que pueden ser entendidos por toda la familia de herramientas apt (es decir apt-get y demás). Contar con este tipo de repositorios, permite realizar la instalación, eliminación, actualización y demás operaciones de paquetes, a través del uso de paquetes individuales o grupos de paquetes.

En concreto, el comando apt update se utiliza para actualizar la lista de paquetes disponibles en los repositorios de software de tu sistema. Cuando ejecutas "apt update", el sistema contacta los servidores de los repositorios configurados en tu sistema y verifica si hay actualizaciones disponibles para los paquetes que has instalado o que podrías querer instalar. Sin embargo, "apt update" no instala ni actualiza ningún paquete en sí, solo actualiza la información sobre las versiones disponibles.



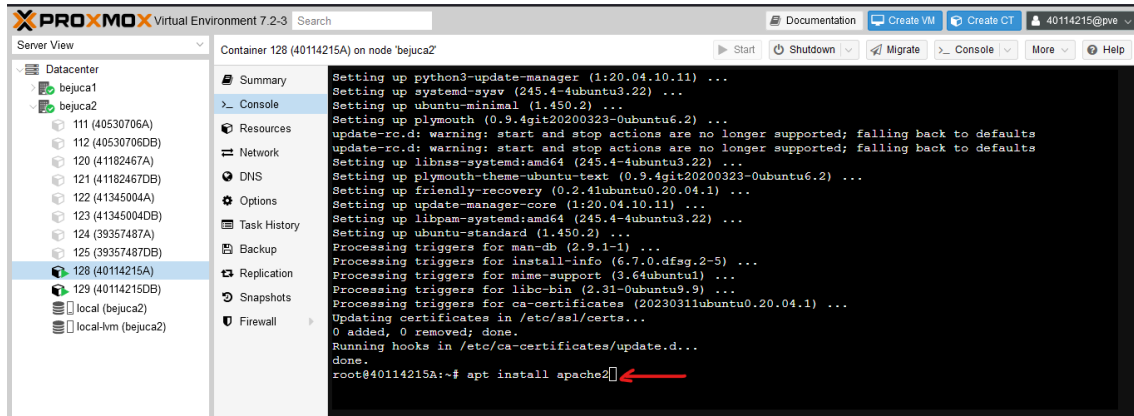
iv) Se procede a ejecutar el comando “apt upgrade”.

Una vez se ha actualizado la lista de paquetes con "apt update", el comando "apt upgrade" se puede utilizar para efectivamente instalar las actualizaciones disponibles. Este comando instalará las versiones más recientes de los paquetes que ya están instalados en tu sistema.



v) Se instala apache2 mediante el comando “apt install apache2”.

Apache es un software de servidor web, encargado de aceptar peticiones (requests) de visitantes y de devolverles la información solicitada en forma de páginas web. Es responsable de que el servidor en el que el sitio se encuentra almacenado pueda comunicarse con el dispositivo que el visitante está usando. Es lo que conecta el hardware del visitante con el del servidor. Apache constituye el software en el que el servidor web corre, el software detrás del servidor.

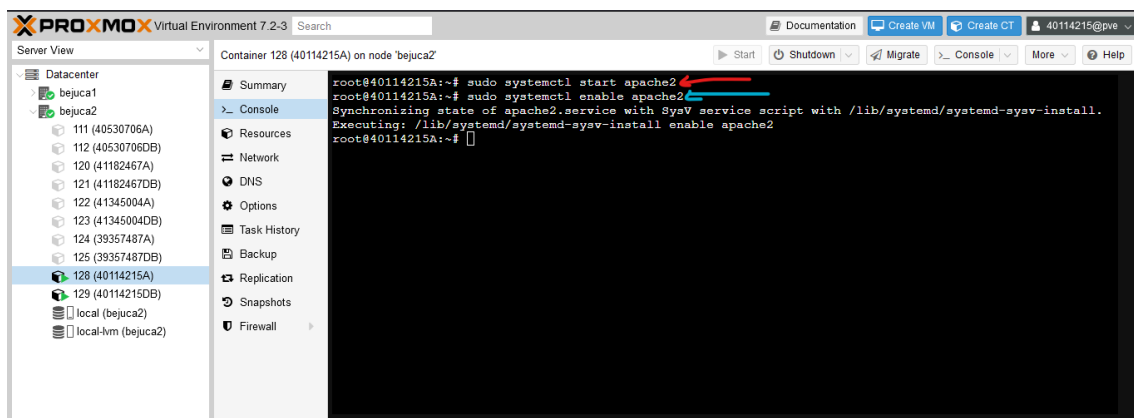


The screenshot shows the Proxmox Virtual Environment interface. On the left, the 'Server View' sidebar lists a datacenter with two hosts: 'bejuca1' and 'bejuca2'. Under 'bejuca2', several VMs are listed, including VM 128 (40114215A) which is selected. The main panel displays the 'Console' of VM 128. The terminal output shows the installation of various packages: python3-update-manager, systemd-sysv, ubuntu-minimal, plymouth, update-rc.d, libnss-systemd, libpam-systemd, and ubuntu-standard. The prompt 'root@40114215A:~# apt install apache2' is entered, with a red arrow pointing to it.

vi)

Se procede a ejecutar los comandos “systemctl start apache2” y “systemctl enable apache2”.

“systemctl start apache2” se utiliza para iniciar el servicio del servidor web Apache inmediatamente. Cuando se ejecute este comando, Apache se iniciará y comenzará a atender las solicitudes entrantes. Por su parte, “systemctl enable apache2” se utiliza para habilitar el servicio del servidor web Apache para que se inicie automáticamente en el arranque del sistema. Una vez habilitado el servicio Apache con este comando, el sistema asegurará que Apache se inicie cada vez que se inicie o reinicie el servidor. Esto es útil para asegurarse de que el servidor web esté disponible incluso después de un reinicio del sistema.



The screenshot shows the same Proxmox interface as before. The terminal in VM 128 now shows the execution of two commands: 'root@40114215A:~# sudo systemctl start apache2' and 'root@40114215A:~# sudo systemctl enable apache2'. A red arrow points to the first command. The output shows the state of the 'apache2.service' being synchronized with the SysV script and then enabled.

vii)

- Pero para poder acceder al contenedor del blog de forma remota y subir los archivos necesarios para el blog, necesitamos saber la dirección IP del contenedor y es el puerto por el cual se levantará.
- Para conocer la dirección IP podemos usar 2 comandos: “ifconfig” o “ip address”. Como puede verse en las imágenes, ambos comandos en Ubuntu generan el mismo resultado.

The screenshot shows the Proxmox VE interface with the console of container 128 (40114215A) on node 'bejuca2'. The user has run the 'ifconfig' command, displaying the following output:

```

root@40114215A:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.77.216 netmask 255.255.255.0 broadcast 192.168.77.255
    inet6 fe80::687b:bcff:fe9a:3c2c prefixlen 64 scopeid 0x20<link>
    ether 6a:7b:bc:9a:3c:2c txqueuelen 1000 (Ethernet)
    RX packets 840 bytes 79998 (79.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 38 bytes 4965 (4.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@40114215A:~#

```

The bottom of the screenshot shows a task log table:

Start Time	End Time	Node	User name	Description	Status
Aug 28 10:30:40		bejuca1	40114215@pve	VM/CT 128 - Console	
Aug 28 10:30:23	Aug 28 10:30:27	bejuca2	40114215@pve	CT 128 - Start	OK

The screenshot shows the Proxmox VE interface with the console of container 128 (40114215A) on node 'bejuca2'. The user has run several commands to install xdg-utils and configure the network:

```

root@40114215A:~/var/www/html# xdg-open blog_personal.html
Command 'xdg-open' not found, but can be installed with:
apt install xdg-utils

root@40114215A:~/var/www/html# cd ..
root@40114215A:~/var/www# cd ..
root@40114215A:~/var# cd ..
root@40114215A:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
2: eth0@if69: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 6a:7b:bc:9a:3c:2c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.77.216/24 brd 192.168.77.255 scope global dynamic eth0
        valid_lft 346sec preferred_lft 346sec
    inet6 fe80::687b:bcff:fe9a:3c2c scope link
        valid_lft forever preferred_lft forever

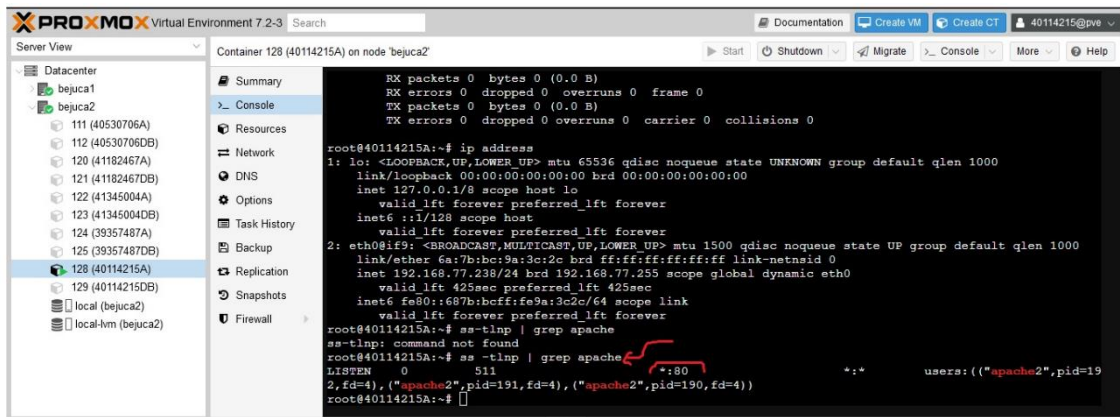
root@40114215A:~# ^C
root@40114215A:~#

```

The bottom of the screenshot shows the same task log table as the previous image:

Start Time	End Time	Node	User name	Description	Status
Aug 28 10:30:40		bejuca1	40114215@pve	VM/CT 128 - Console	
Aug 28 10:30:23	Aug 28 10:30:27	bejuca2	40114215@pve	CT 128 - Start	OK

- Cuando el software del servidor web Apache2 está instalado en un sistema operativo Ubuntu de forma predeterminada, suele escuchar en el puerto 80 para conexiones HTTP y en el puerto 443 para conexiones HTTPS. Como nuestra conexión se establecerá usando el protocolo HTTP, el puerto debería ser el 80. Esto fue comprobado, como puede verse en la imagen a continuación, mediante el comando “ss -tmp | grep apache”.



viii)

Paso siguiente pasamos a instalar un firewall para reforzar la seguridad en un sistema informático. Ayuda a controlar y filtrar el tráfico de red entrante y saliente, lo que ayuda a proteger el sistema contra amenazas y ataques desde Internet y redes locales.

Para para instalar el programa "ufw" ("Uncomplicated Firewall"), se utiliza el comando "apt install ufw". Ufw es una interfaz de línea de comandos que simplifica la configuración y administración del firewall en sistemas basados en Debian (como el caso de los sistemas Ubuntu). Mediante consola, se puede ver que se nos informa que ufw ya se encontraba en su versión mas nueva.

Paso siguiente, se ejecuta el comando "ufw allow 80/tcp", el cual se utiliza para permitir el tráfico entrante en el puerto 80 utilizando el protocolo TCP a través del firewall gestionado por "Uncomplicated Firewall" (UFW).

Por último, se activa el firewall mediante el comando "ufw enable".


```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
2: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 6a:7b:bc:9a:3c:2c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.77.238/24 brd 192.168.77.255 scope global dynamic eth0
        valid_lft 425sec preferred_lft 425sec
    inet6 fe80::687b:bcff:fe9a:3c2c/64 scope link
        valid_lft forever preferred_lft forever
root@40114215A:~# ss -tlnp | grep apache
ss-tlnp: command not found
root@40114215A:~# ss -tlnp | grep apache
LISTEN 0 511 *:80 *: * users:((("apache2",pid=19
2,fd=4)),("apache2",pid=191,fd=4)),("apache2",pid=190,fd=4))
root@40114215A:~# apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@40114215A:~#
```

```
valid_lft forever preferred_lft forever
2: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 6a:7b:bc:9a:3c:2c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.77.238/24 brd 192.168.77.255 scope global dynamic eth0
        valid_lft 425sec preferred_lft 425sec
    inet6 fe80::687b:bcff:fe9a:3c2c/64 scope link
        valid_lft forever preferred_lft forever
root@40114215A:~# ss -tlnp | grep apache
ss-tlnp: command not found
root@40114215A:~# ss -tlnp | grep apache
LISTEN 0 511 *:80 *: * users:((("apache2",pid=19
2,fd=4)),("apache2",pid=191,fd=4)),("apache2",pid=190,fd=4))
root@40114215A:~# apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@40114215A:~# ufw allow 80/tcp
Rules updated (v6)
root@40114215A:~# ufw enable
Firewall is active and enabled on system startup
root@40114215A:~#
```

ix)

El blog está programado utilizando html y css en un entorno externo al contenedor. Para incorporarlo al mismo, se utilizó un repositorio de github.

Inicialmente se instalan los paquetes para trabajar con git en Ubuntu.

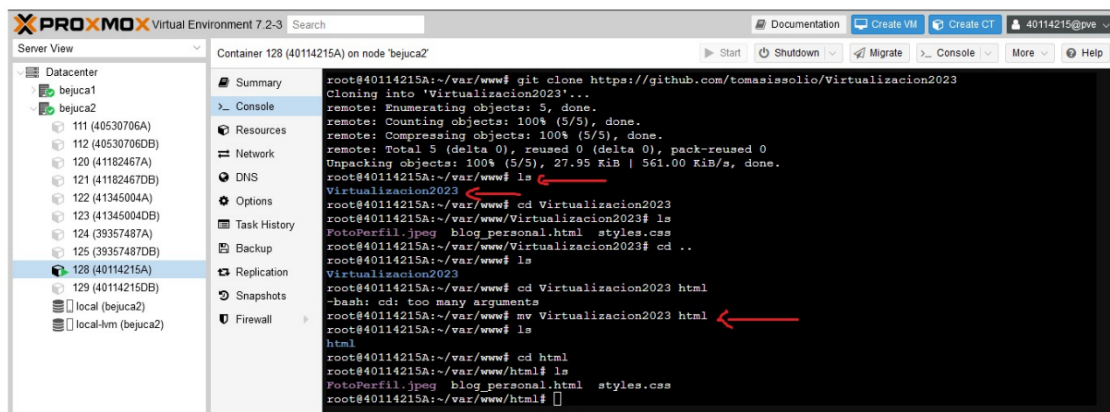
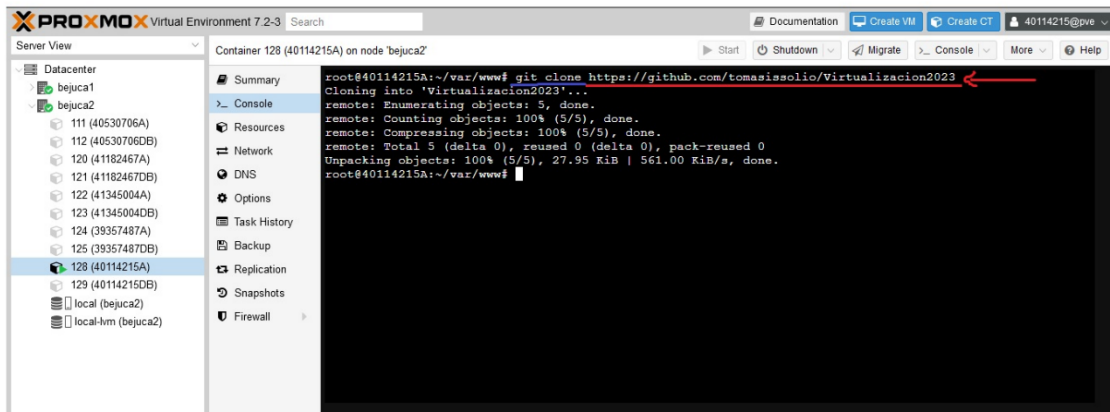
```
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@40114215A:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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

root@40114215A:~# apt-get install git-all
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core cvs cvsps dbus-user-session
  dconf-gsettings-backend dconf-service dhcpcd-elisa-helper elpa-dash elpa-github elpa-git-commit
  elpa-let-alist elpa-magit elpa-magit-popup elpa-treepsy elpa-with-editor emacs emacs-bin-common
  emacs-common emacs-el emacs-gtk emacs-sen-common fontconfig fonts-droid-fallback fonts-noto-mono
  fonts-urw-base35 ghostscript git git-cvs git-daemon-run git-doc git-el git-email git-gui git-man
  git-mediawiki git-svn gitk gitweb glib-networking glib-networking-common glib-networking-services
  gsettings-desktop-schemas gsfonts gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme
  imagemagick-6-common libalgorithm-c3-perl libasound2 libasound2-data libatk-bridge2.0-0 libatk1.0-0
  libatk1.0-data libatspi2.0-0 libauthen-sasl-perl libavahi-client3 libavahi-common-data
  libavahi-common3 libb-hooks-endofscope-perl libb-hooks-op-check-perl libcairo-gobject2 libcairo2
```

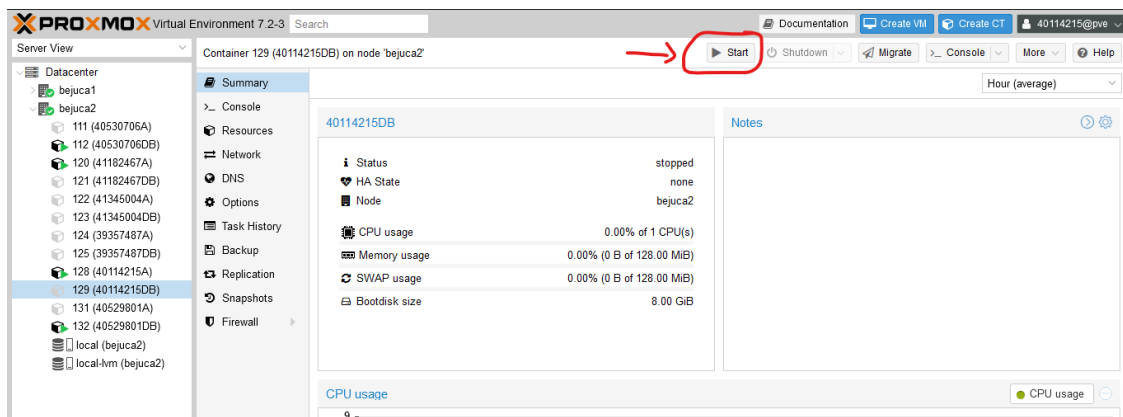
```
libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
libhttp-negotiate-perl libidn11 libijs-0.35 libio-html-perl libio-socket-ssl-perl libjbig0
libjbig2dec0 libjpeg-turbo8 libjpeg8 libjson-glib-1.0-0 libjson-glib-1.0-common libjson-perl
libjson-xs-perl liblcms2-2 liblqr-1-0 libltdl7 liblwp-mediatypes-perl liblwp-protocol-http-perl
libm17n-0 libmagickcore-6.q16-6 libmagickwand-6.q16-6 libmailtools-perl libmediawiki-api-perl
libmodule-implementation-perl libmodule-runtime-perl liboro-compat-perl libnamespace-autoclean-perl
libnamespace-clean-perl libnet-dns-perl libnet-dns-sec-perl libnet-domain-tld-perl libnet-http-perl
libnet-ip-perl libnet-libidn-perl libnet-smtp-ssl-perl libnet-ssleay-perl libopenjp2-7 libotf0
libpackage-attach-perl libpackage-attach-xs-perl libpadwalker-perl libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpaper-utils libpaper-perl libparams-classify-perl libparams-util-perl
libparams-validate-perl libparams-validationcompiler-perl libperl4-corelibs-perl libpixman-1-0
libproxyv5 libreadonly-perl libref-util-perl libref-util-xs-perl librest-0.7-0 librole-tiny-perl
librsvg2-2 librsvg2-common libaerf-1-1 libasoup-gnome2.4-1 libasoup2.4-1 libspecio-perl
libsub-exporter-perl libsub-exporter-progressive-perl libsub-identify-perl libsub-install-perl
libsub-name-perl libsub-quote-perl libsvn-perl libsvn1 libtcl8.6 libterm-readkey-perl libthai-data
libthai0 libtiff5 libtimedate-perl libtk8.6 libtry-tiny-perl libtypes-serialiser-perl liburi-perl
libutmper0 libutf8proc2 libvariable-magic-perl libwayland-client0 libwayland-cursor0
libwayland-egl1 libwebp6 libwebpmux3 libwww-perl libwww-robotrules-perl libxcb-render0 libxkbcommon0
libxss1 libxstring-perl libyaml-libyaml-perl libyaml-perl lib7n-db patch perl-openssl-defaults
poppler-data runit tcl tcl8.6 tk tk8.6 ubuntu-mono xbitmaps xterm
0 upgraded, 242 newly installed, 0 to remove and 0 not upgraded.
Need to get 91.0 MB of archives.
After this operation, 407 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Luego nos ubicamos en la carpeta html generada por Apache2, path: /var/www/html; se clona en ella el repositorio donde se encuentran los archivos que se desea manejar, y se cambia el nombre inicial de la carpeta que los contiene a "html". Este cambio permitirá que los elementos cargados sean reconocidos por Apache (el software de servidor web) y puedan ponerse a disposición de los visitantes del blog.

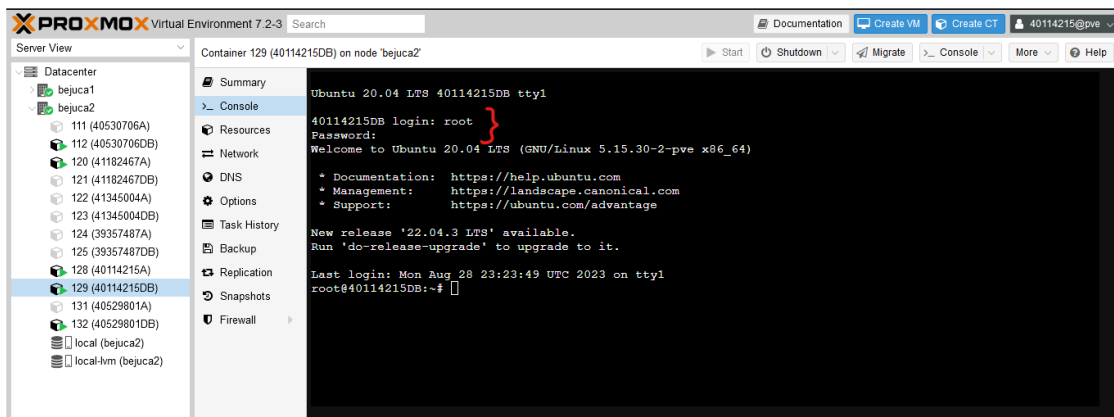


Tercera parte: Trabajando con el contenedor de la DB

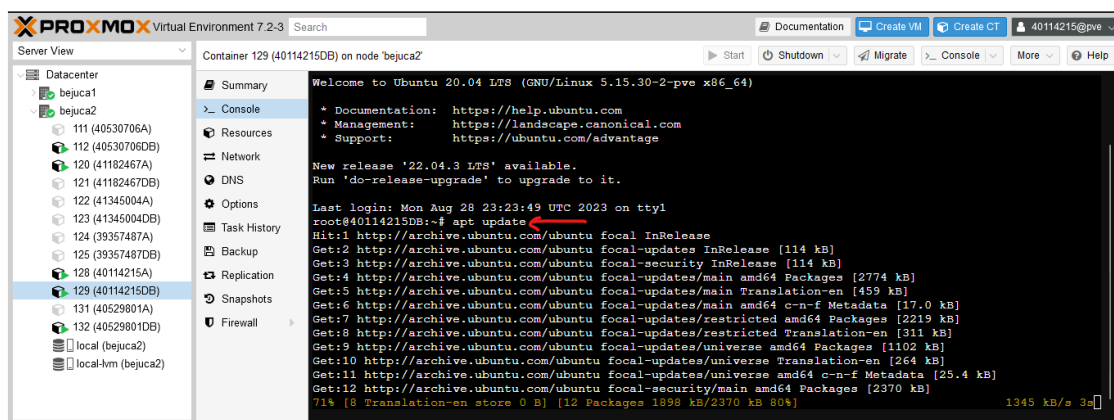
i) Se inicia la máquina virtual.



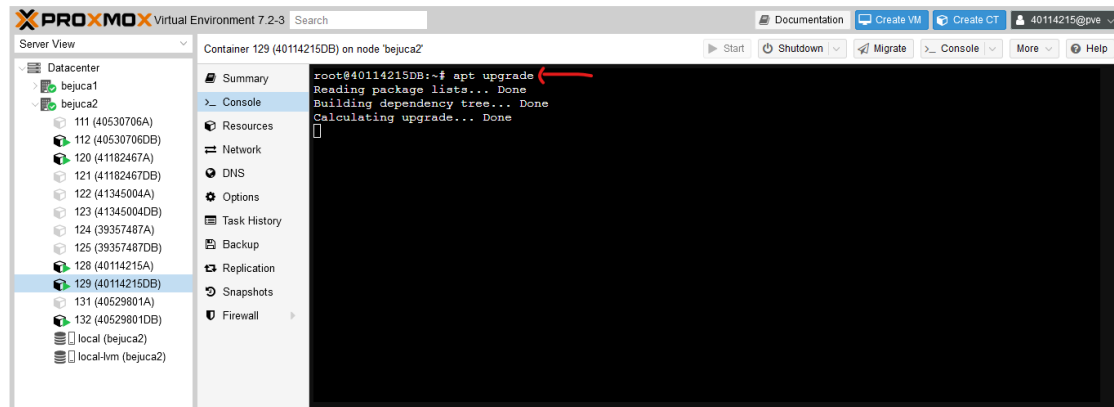
ii) Se procede a abrir la consola y a registrarse con usuario y contraseña (la configurada al crear el contenedor).



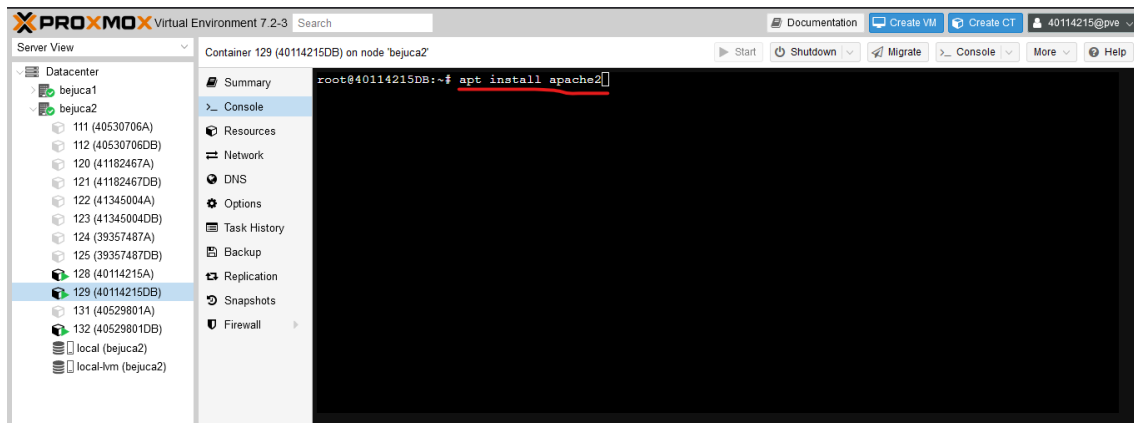
iii) Se procede a ejecutar el comando “apt update”.



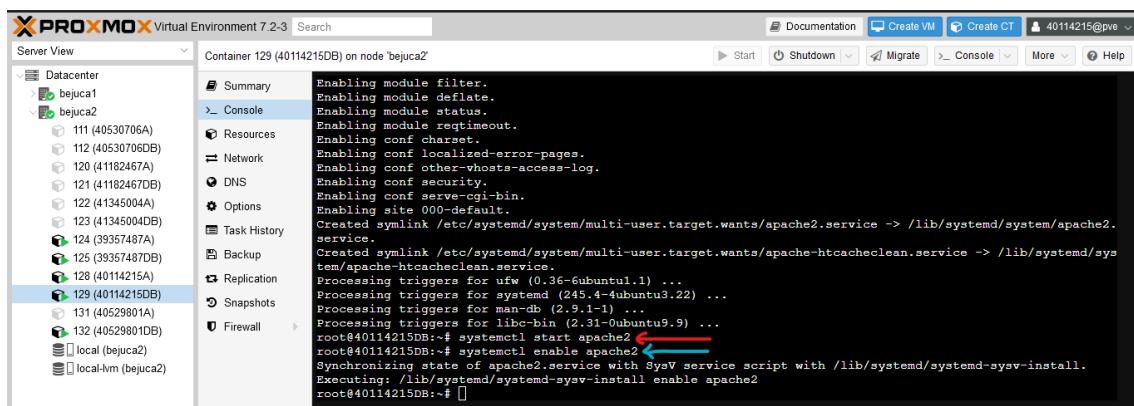
iv) Se procede a ejecutar el comando “apt upgrade”.



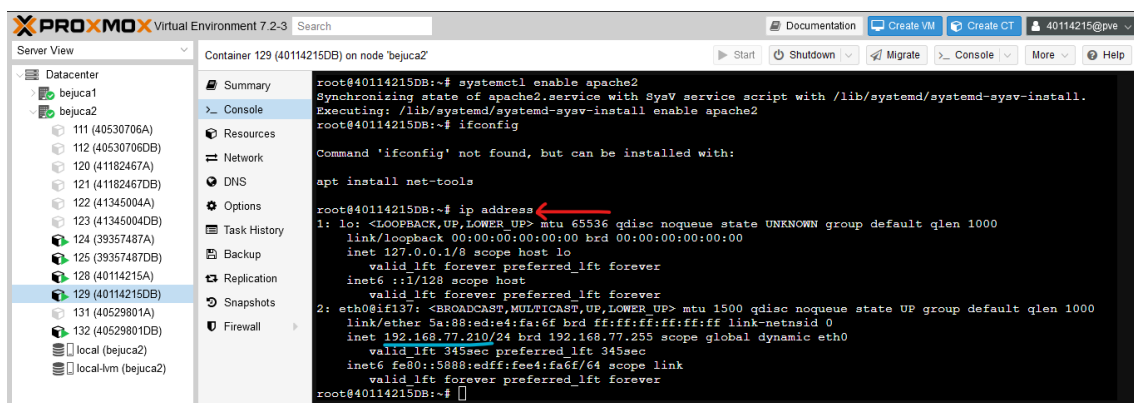
v) Se instala apache2 mediante el comando “apt install apache2”.



vi) Se procede a ejecutar los comandos “systemctl start apache2” y “systemctl enable apache2”.

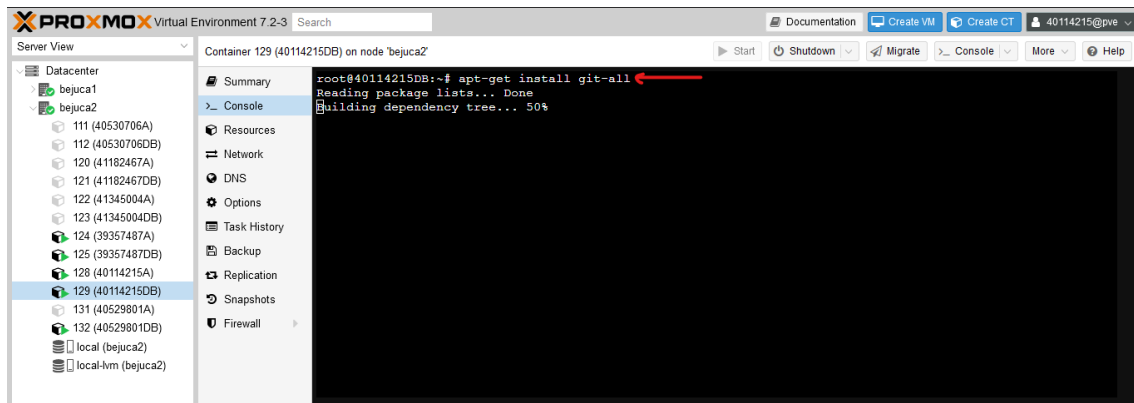


vii) Para poder acceder al contenedor del blog de forma remota y subir los archivos necesarios para el blog, necesitamos saber la dirección IP del contenedor y es el puerto por el cual se levantará. Para conocer la dirección IP usamos el comando: “ip address”.

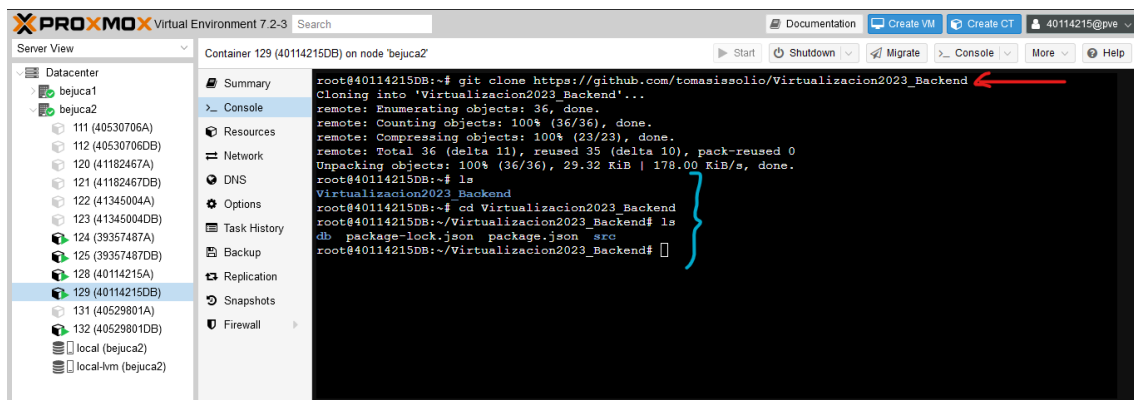


viii)

Se instalan los paquetes para trabajar con GIT, con el comando “apt-get install git-all”

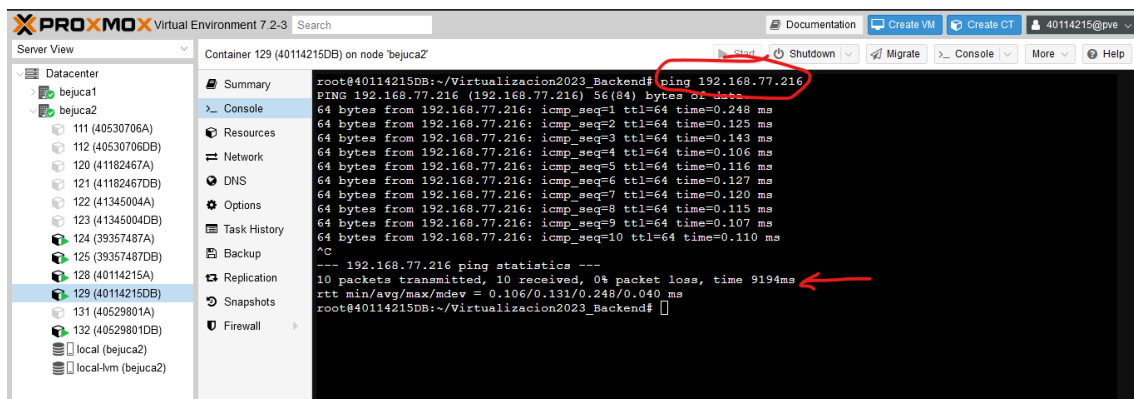


ix) Se clona el repositorio donde se encuentra la Rest API mediante el comando “git clone”.
Luego se puede ver cómo se accedió a la carpeta importada para ver los archivos que en ella se encuentran.



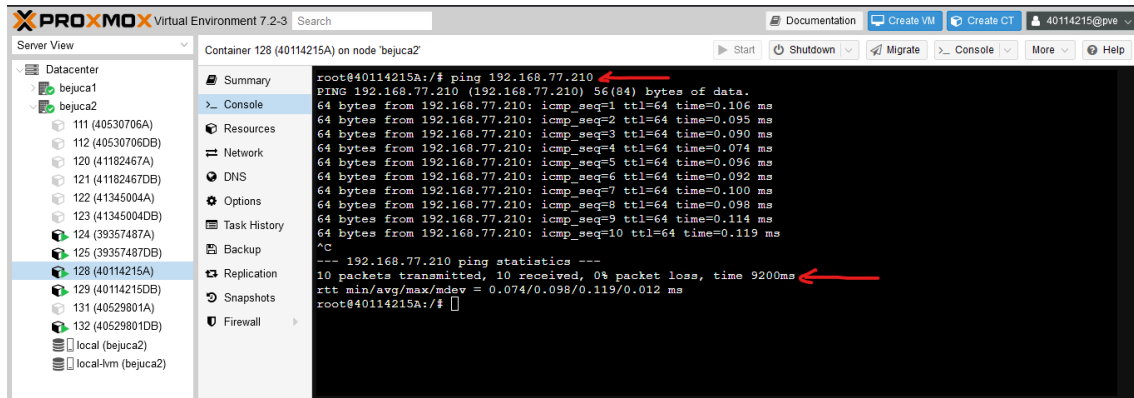
Cuarta parte: Comprobando accesibilidad entre contenedores

i) Hago un ping del contenedor de la base de datos al contenedor del blog (con ip 192.168.77.216)



Resultado: Todos los paquetes enviados hacia al contenedor A fueron recibidos, no se produjeron pérdidas (nótese que ambos contenedores se están ejecutando).

i) Hago un ping del contenedor del blog al contenedor de la base de datos (con ip 192.168.77.210)



The screenshot shows the Proxmox Virtual Environment interface. On the left, a tree view shows the 'Datacenter' containing two hosts: 'bejuca1' and 'bejuca2'. Under 'bejuca2', there is a list of containers. Container '128 (40114215A)' is selected and highlighted in blue. The main panel on the right shows the 'Console' view for this container. The terminal output shows a successful ping command executed from the container's root shell. The command is 'ping 192.168.77.210'. The output shows 10 successful pings, each receiving 64 bytes of data from 192.168.77.210 with various TTL and time values. At the end, the 'ping statistics' are displayed, showing 10 packets transmitted, 10 received, 0% packet loss, and a total time of 9200ms. Red arrows point to the command and the final statistics line.

```
root@40114215A:/# ping 192.168.77.210
PING 192.168.77.210 (192.168.77.210) 56(84) bytes of data.
64 bytes from 192.168.77.210: icmp_seq=1 ttl=64 time=0.106 ms
64 bytes from 192.168.77.210: icmp_seq=2 ttl=64 time=0.095 ms
64 bytes from 192.168.77.210: icmp_seq=3 ttl=64 time=0.090 ms
64 bytes from 192.168.77.210: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 192.168.77.210: icmp_seq=5 ttl=64 time=0.096 ms
64 bytes from 192.168.77.210: icmp_seq=6 ttl=64 time=0.092 ms
64 bytes from 192.168.77.210: icmp_seq=7 ttl=64 time=0.100 ms
64 bytes from 192.168.77.210: icmp_seq=8 ttl=64 time=0.098 ms
64 bytes from 192.168.77.210: icmp_seq=9 ttl=64 time=0.114 ms
64 bytes from 192.168.77.210: icmp_seq=10 ttl=64 time=0.119 ms
^C
--- 192.168.77.210 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9200ms
rtt min/avg/max/mdev = 0.074/0.098/0.119/0.012 ms
root@40114215A:/#
```

Resultado: Todos los paquetes enviados hacia al contenedor A fueron recibidos, no se produjeron pérdidas (nótese que ambos contenedores se están ejecutando).