

**Universidad Tecnológica  
Nacional Facultad Regional  
Tucumán**



**Cátedra Virtualización: Consolidación de servidores (Electiva)**

Trabajo Final Integrador - Caso de Uso

**Integrantes:**

Pasteris, Anabella Rosa - 52533

**Docentes:**

Carriles, Luis Maria

**Comisión:** 5k2

**Fecha de Presentación:** 18/06/2024

## Trabajo Final Integrador - Caso de Uso

Para el desarrollo de este Trabajo final integrador, se utilizará Proxmox (plataforma de virtualización de código abierto) donde nos permite crear dos contenedores para montar un Blog personal. El objetivo del mismo es configurar el entorno del Blog personal donde incluya las siguientes especificaciones:

- Datos personales
- Imagen personal del alumno
- El blog deberá poder actualizarse en su página principal, agregando un nuevo dato al mismo.

La implementación del mismo consiste, en uno de los contenedores montar solo el FrontEnd y el otro BackEnd. Los mismos son respaldados con herramientas como HTML, CSS que se utilizan juntos para crear y diseñar páginas web. HTML proporciona la estructura y el contenido de la página, mientras que CSS se encarga de la presentación y el diseño visual.

Una base de datos como PostgreSQL, el sistema de gestión de bases de datos para almacenar y gestionar datos de una aplicación web. Además, se utilizará Apache2 para establecer la conexión entre los contenedores y realizar solicitudes a través de una API, permitiendo el acceso a los datos almacenados en la base de datos desde el frontend.

En el contenido del Trabajo se detalla el paso a paso todas las instalaciones necesarias para montar el Blog personal.

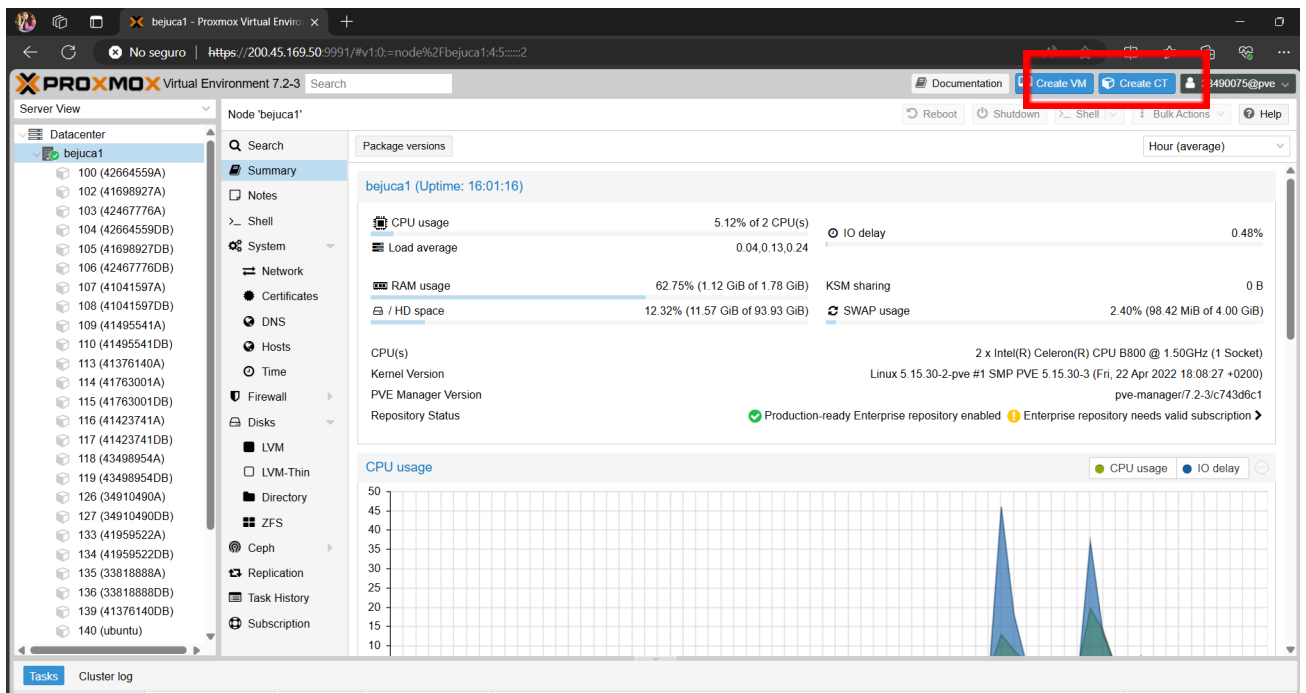
### Creación de los contenedores

Para la creación de los contenedores, tanto para Backend como Frontend, deben cumplir los siguientes requisitos:

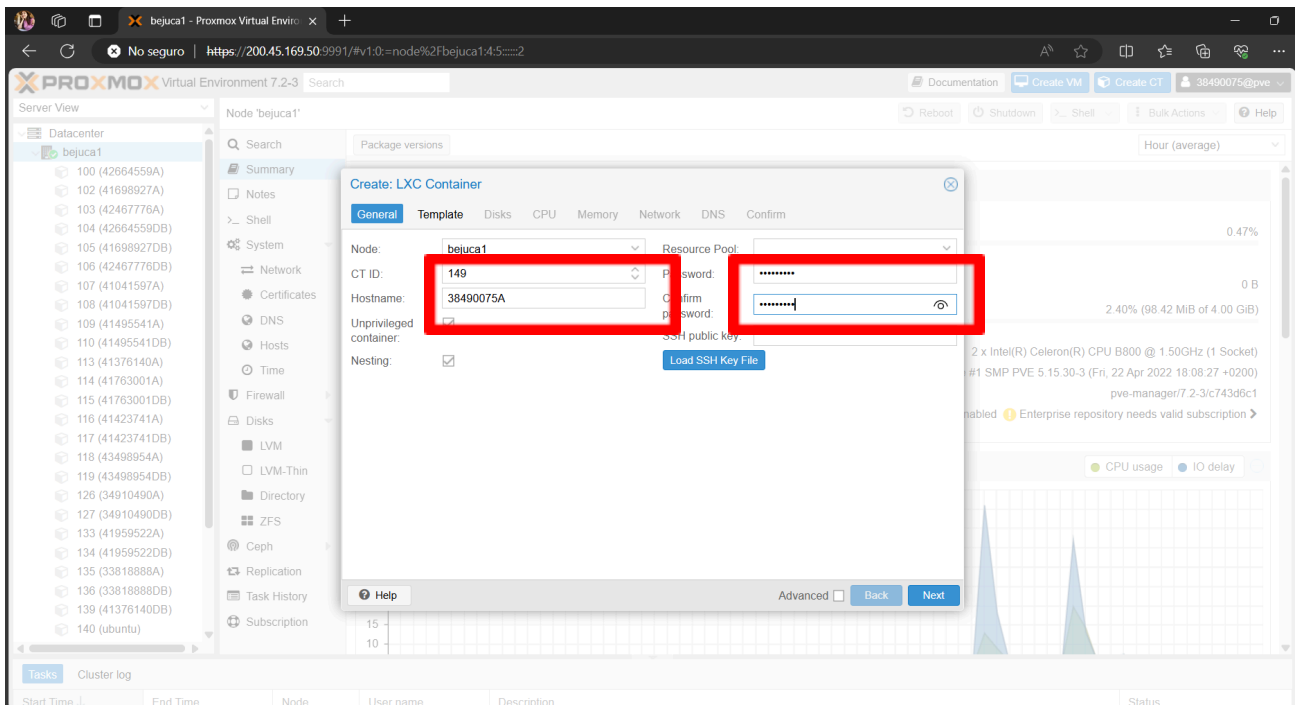
- Nombre del contenedor: "DNI + DB" (Para backend), y "DNI + A" (Para frontend).
- Memoria RAM: 128 MB.
- Almacenamiento: 8GB.
- Procesador: 1 núcleo.
- Red: DHCP.

Los mismos se crean de la siguiente manera:

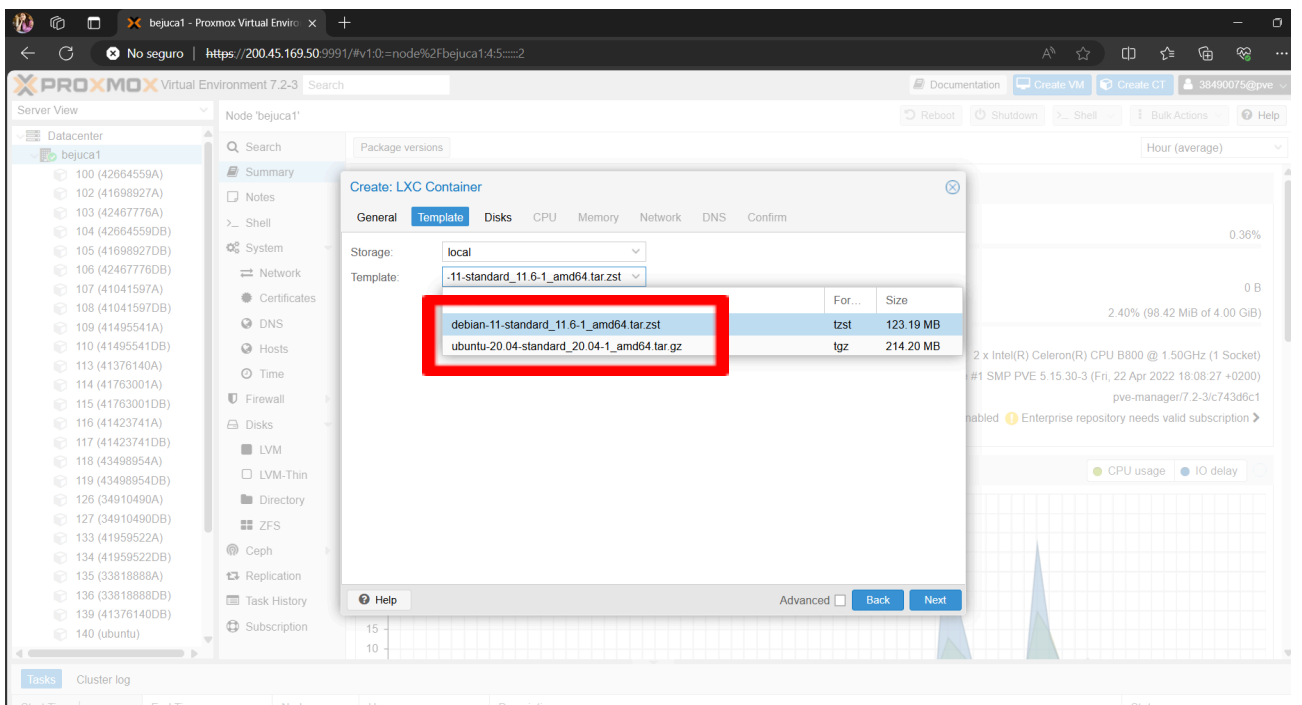
- 1) Una vez que se accedió a Proxmox y se autenticó el usuario, nos dirigimos al botón “Create CT” para crear el contenedor. En este caso elegí el Datacenter “Bejuca1” para la creación. (Estos pasos son para la creación del contenedor FrontEnd pero luego se ve la diferencia con el otro contenedor).



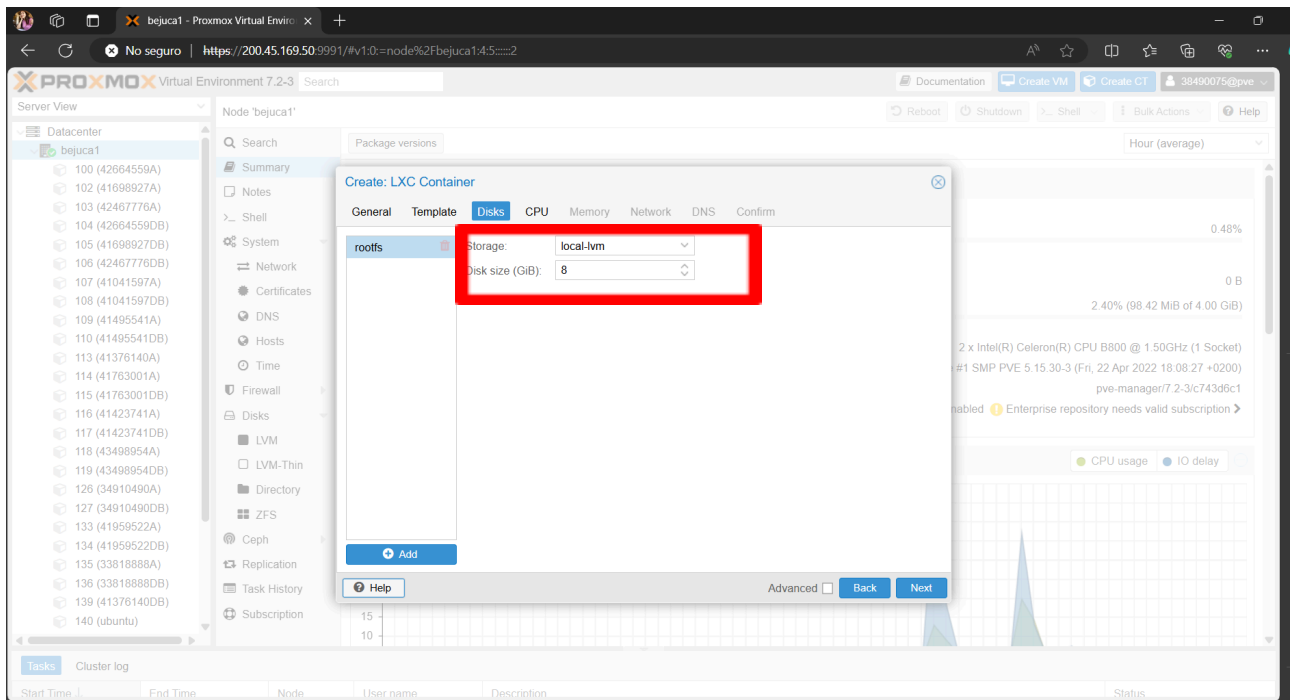
- 2) Luego se rellena los datos correspondientes Hostname (indicados por el enunciado) y la contraseña que deseamos agregar.



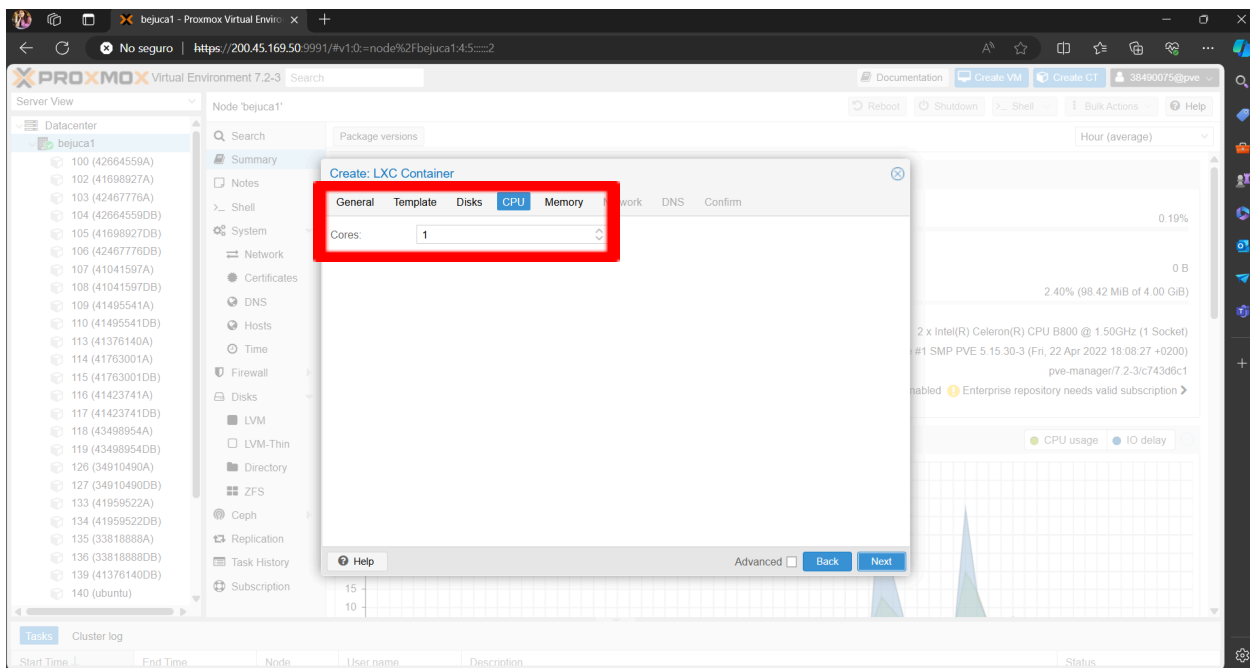
3) Se elige la plantilla con la que se quiere trabajar. En este caso utilizaré debian 11.



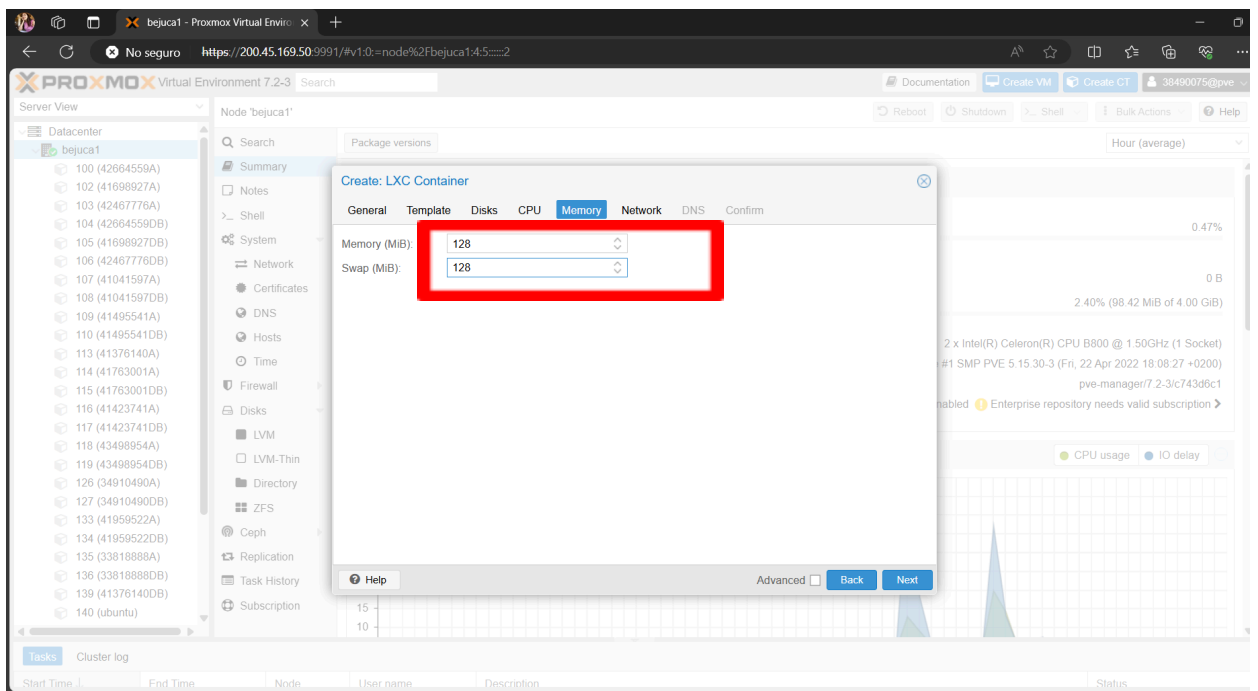
4) Se selecciona el tamaño del disco, en este caso es 8GB (planteado en el enunciado).



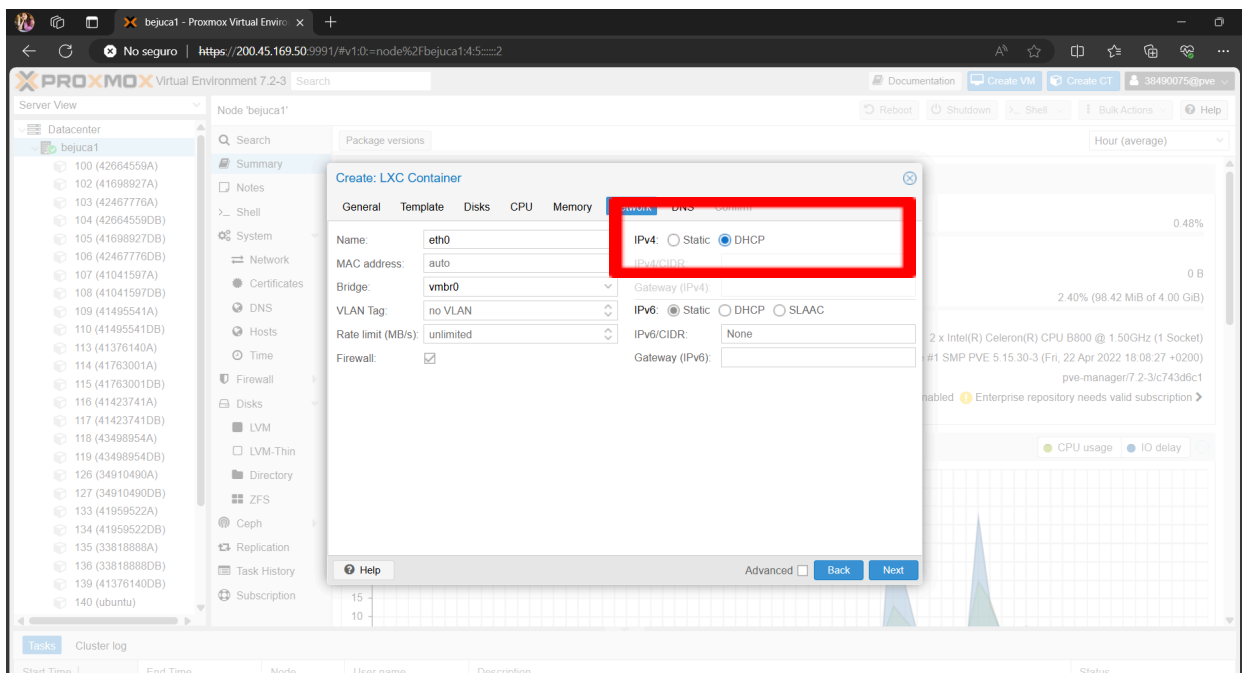
- 5) Se selecciona la cantidad de núcleos del procesador, en este caso es 1 núcleo (planteado en el enunciado).



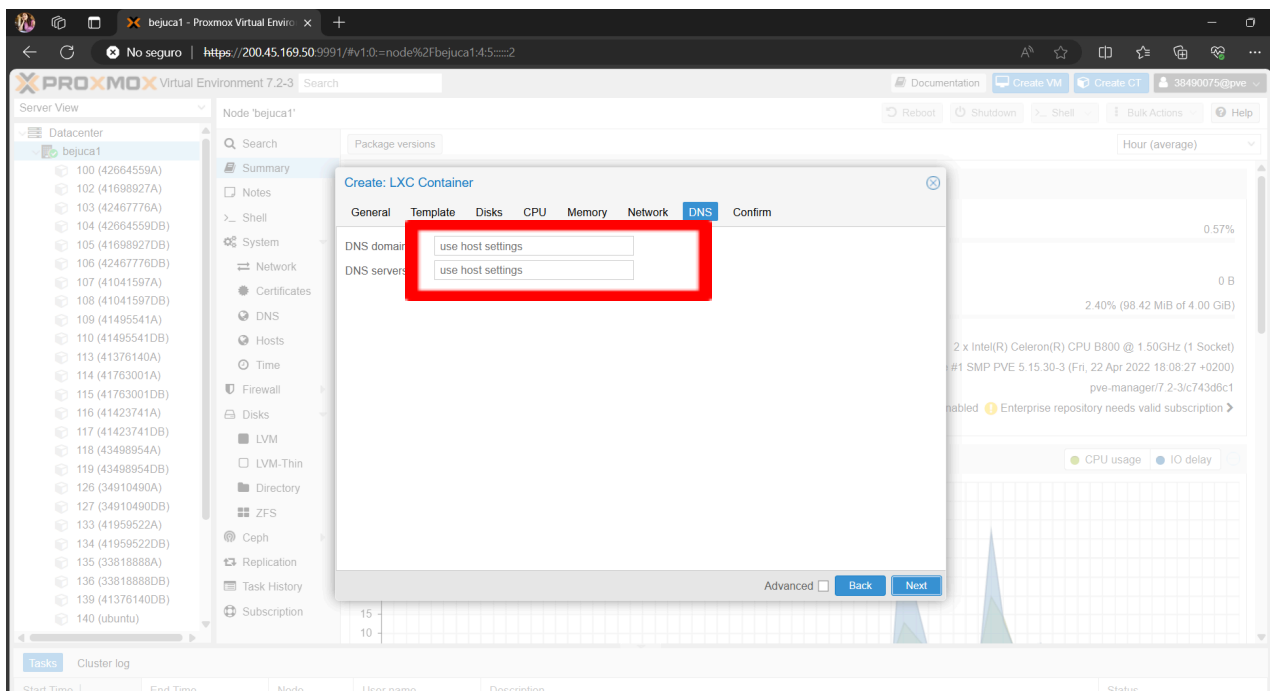
- 6) Se selecciona la cantidad de memoria RAM en este caso es 128 MB (planteado en el enunciado).



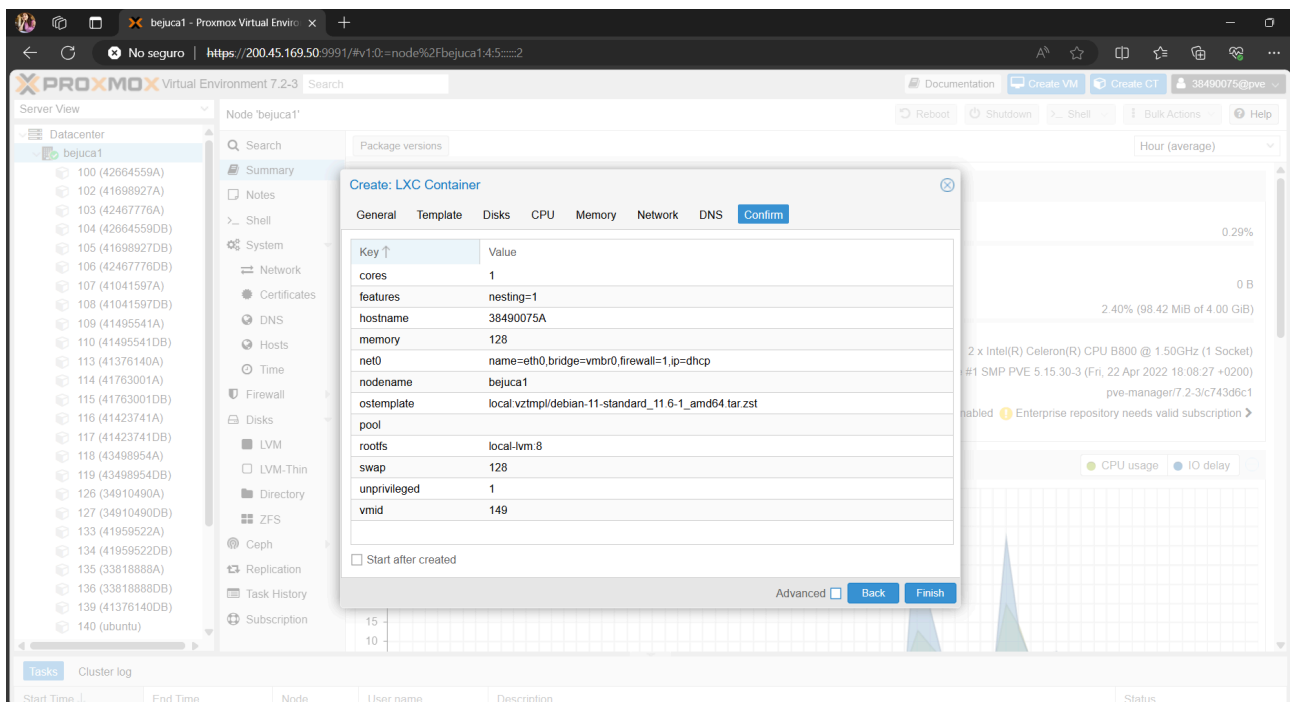
7) Luego se configura la red seleccionando DHCP (planteado en el enunciado).



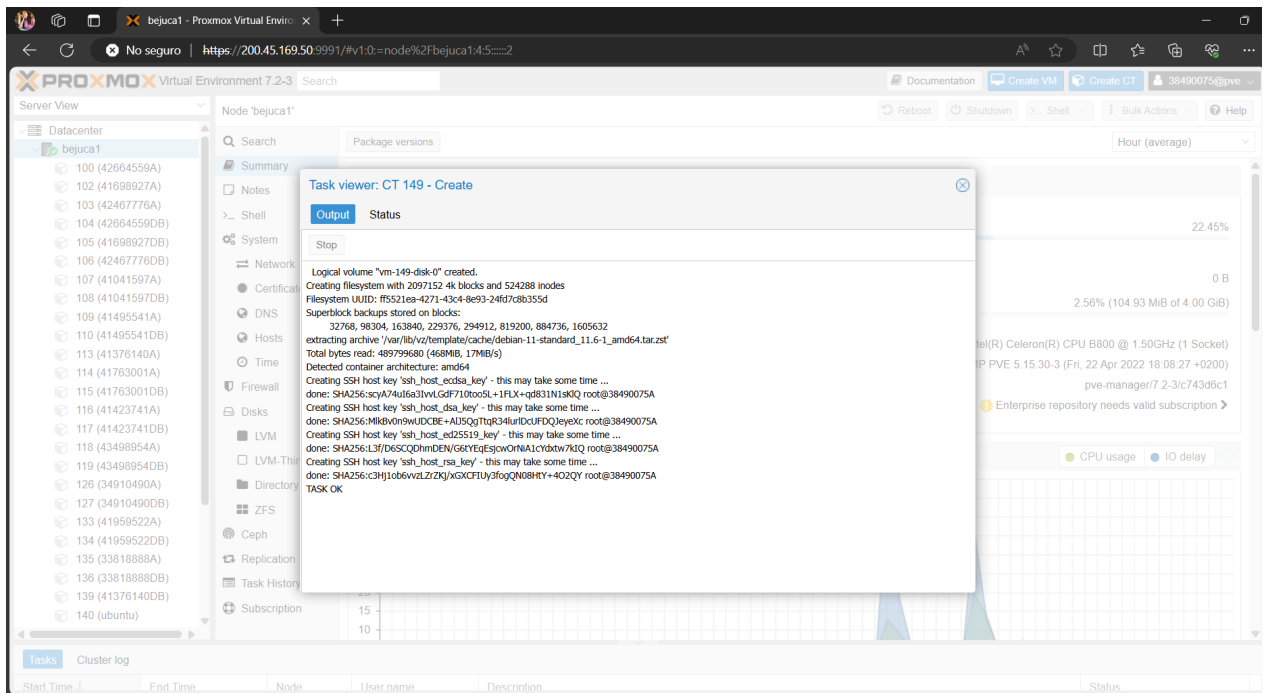
8) Luego se configura el DNS pero en este práctico no es necesario, por lo tanto, se lo deja vacío.



9) Luego podemos observar el resumen de lo configurado.

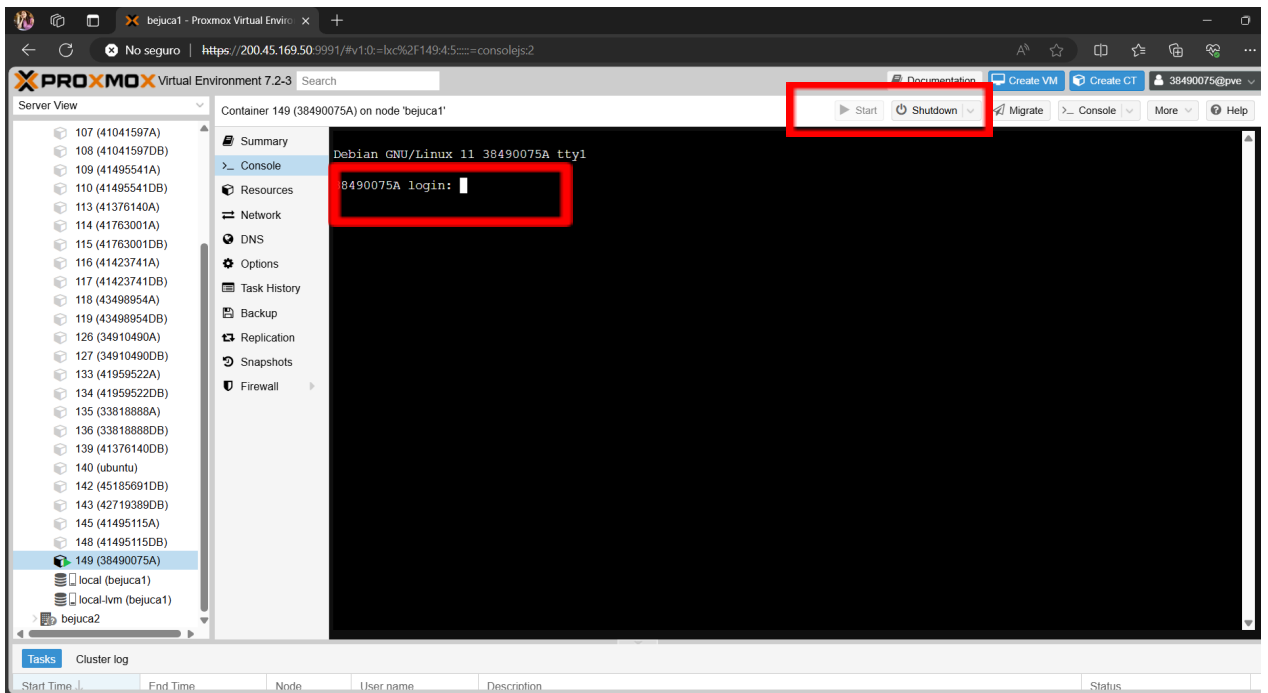


10) Comienza la creación del contenedor configurado y finaliza correctamente.

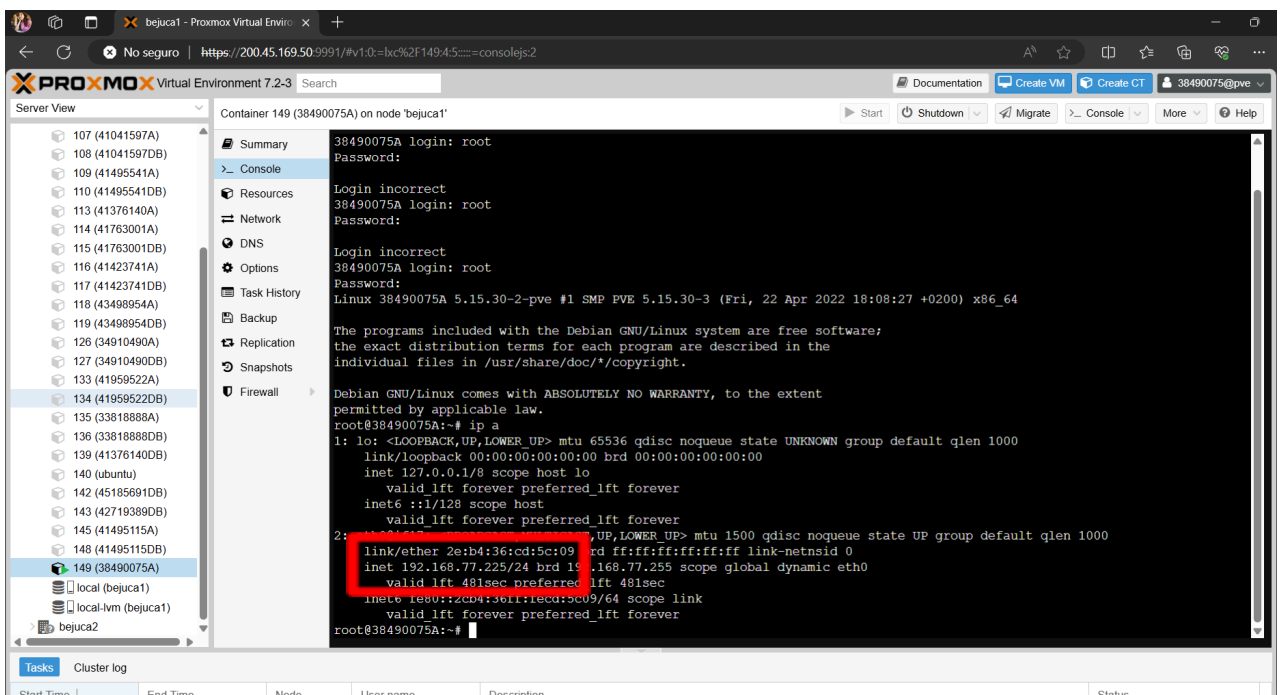


- 11) Una vez creados los contenedores podemos hacerlo correr seleccionando el contenedor deseado y seleccionado "Console" para visualizar la consola y luego apretar el botón "Start" para que comience. Una vez dentro nos pide el login y la contraseña (la cual configuramos en el principio). Utilizamos login: root para disponer de todos los permisos.

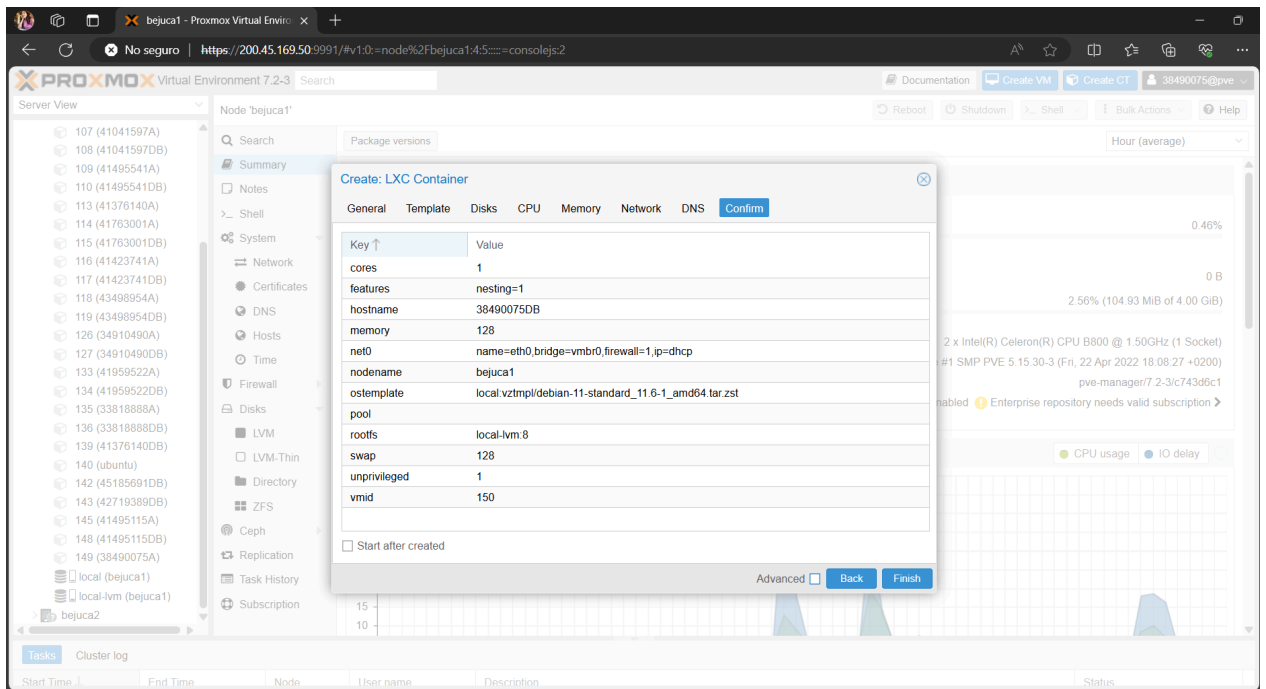




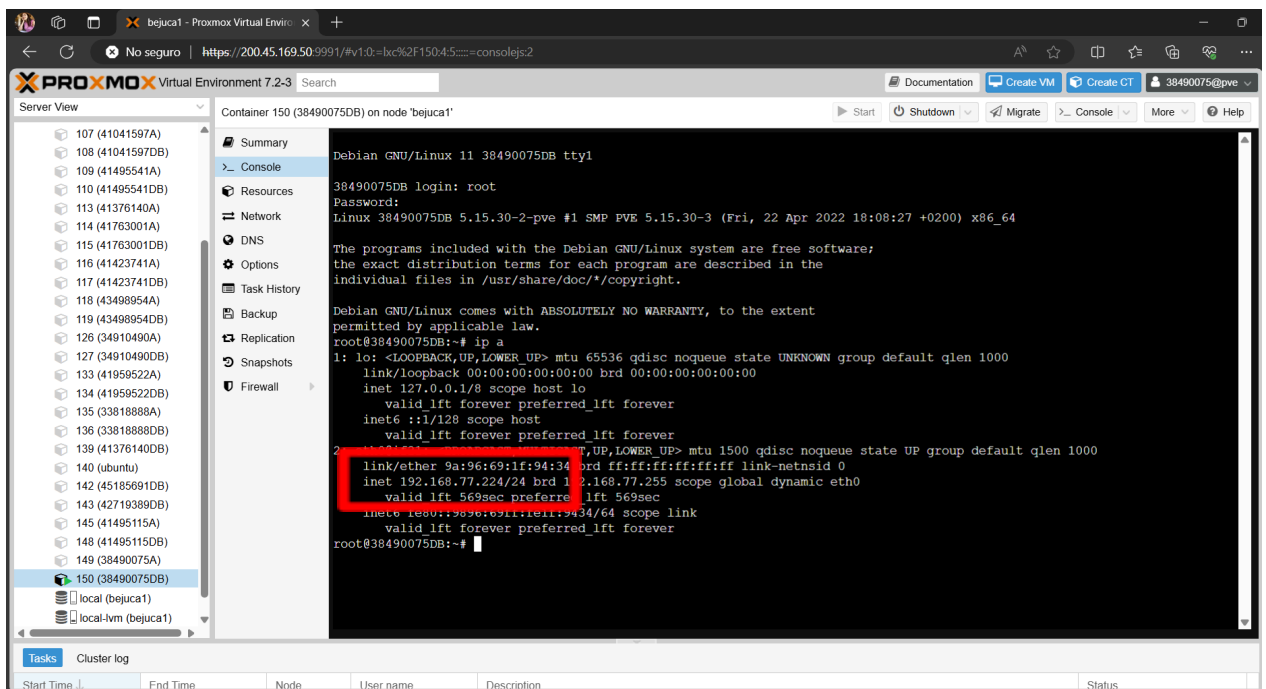
- 12) Luego utilizamos el comando `ip a`, para conocer la dirección ip del contenedor ya que luego la utilizaremos para poder acceder al mismo y asignarle un puerto por el cual se levantará el blog. En este caso la dirección ip es 192.168.77.225 y puerto:80.



- 13) Los mismos pasos detallados anteriormente se realizan para la creación del otro contenedor, que se utilizará para el BackEnd.



- 14) Luego utilizamos el comando `ip a`, para conocer la dirección ip del contenedor ya que luego la utilizaremos para poder acceder al mismo. En este caso la dirección ip es 192.168.77.224.



Configuración de los contenedores

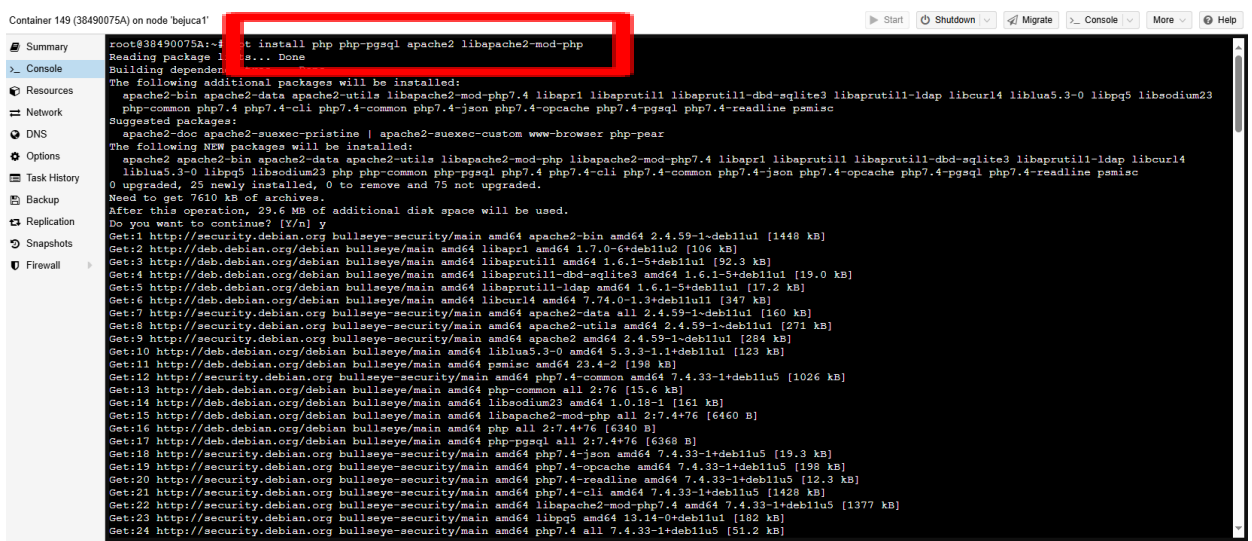
Se debe configurar ambos contenedores para que puedan levantar el blog personal con sus herramientas correspondientes. Cabe destacar que se utiliza Apache y PHP porque son frecuentemente utilizados juntos para desarrollar y servir aplicaciones web dinámicas.

Apache actúa como el servidor web que maneja las solicitudes HTTP y PHP se utiliza para procesar el contenido dinámico. Cuando un navegador solicita una página PHP, Apache pasa la solicitud al motor de PHP, que procesa el script y devuelve el contenido HTML generado al servidor Apache, el cual luego se envía al navegador del usuario.

## Contenedor APP (FrontEnd)

La configuración del mismo se indicará paso a paso, cabe destacar que se utilizarán las herramientas ya mencionadas pero también se instalarán otras dependencias que se necesitarán durante el proceso.

- 1) Se actualizan los repositorios del instalador de paquetes el comando “ apt update” e instalar los siguientes paquetes:
- apt - update: se utiliza para actualizar la lista de paquetes disponibles en los repositorios de software del sistema.
  - php y php-pgsql: Lenguaje PHP y el módulo para la conexión con bases de datos PostgreSQL.
  - apache2 y libapache2-mod-php: Servidor web Apache y el módulo que permite hostear sitios que usan PHP.

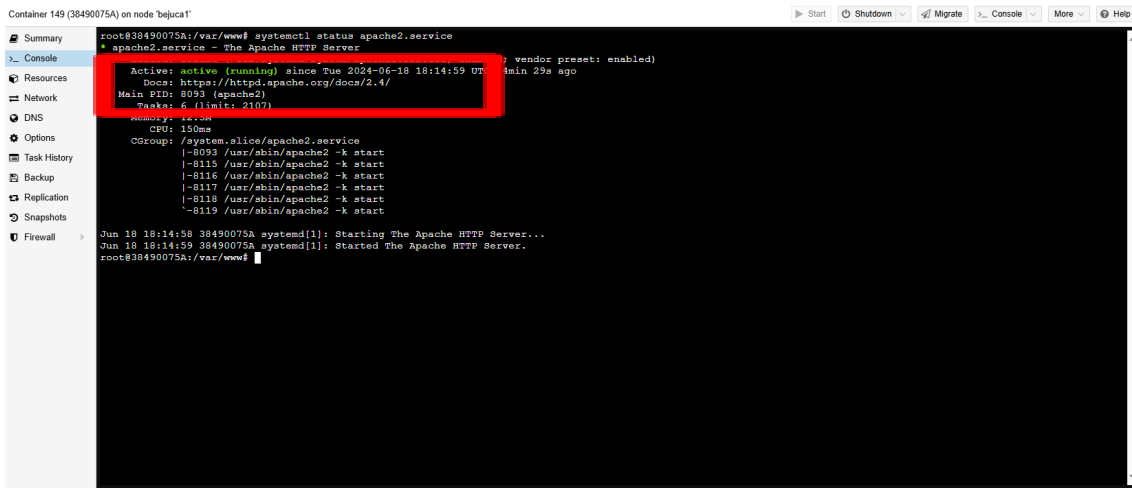


```
Container 149 (38490075A) on node 'bejuca'
```

```
root@38490075A:~# apt install php php-pgsql apache2 libapache2-mod-php
Reading package lists... Done
Building dependency tree
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4 liblua5.3-0 libpq5 libsodium23
  php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-pgsql php7.4-readline psmisc
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser php-pear
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4
  liblua5.3-0 libpq5 libsodium23 php php-common php-pgsql php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-pgsql php7.4-readline psmisc
0 upgraded, 25 newly installed, 0 to remove and 75 not upgraded.
Need to get 7610 kB of archives.
After this operation, 29.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://security.debian.org/debian bullseye-security/main amd64 apache2-bin amd64 2.4.59-1-deb11u1 [1448 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libapr1 amd64 1.7.0-6-deb11u2 [106 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libaprutil1 amd64 1.6.1-5+deb11u1 [92.3 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5+deb11u1 [19.0 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 libaprutil1-ldap amd64 1.6.1-5+deb11u1 [17.2 kB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 libcurl4 amd64 7.74.0-1.3+deb11u1 [347 kB]
Get:7 http://security.debian.org/debian bullseye-security/main amd64 apache2-data all 2.4.59-1-deb11u1 [160 kB]
Get:8 http://security.debian.org/debian bullseye-security/main amd64 apache2-utils amd64 2.4.59-1-deb11u1 [271 kB]
Get:9 http://security.debian.org/debian bullseye-security/main amd64 apache2 amd64 2.4.59-1-deb11u1 [284 kB]
Get:10 http://deb.debian.org/debian bullseye/main amd64 liblua5.3-0 amd64 5.3.3-1.1+deb11u1 [123 kB]
Get:11 http://deb.debian.org/debian bullseye/main amd64 psmisc amd64 23.4-2 [198 kB]
Get:12 http://security.debian.org/debian bullseye-security/main amd64 php7.4-common amd64 7.4.33-1+deb11u5 [1026 kB]
Get:13 http://deb.debian.org/debian bullseye/main amd64 php-common all 2:76 [15.6 kB]
Get:14 http://deb.debian.org/debian bullseye/main amd64 libsodium23 amd64 1.0.18-1 [161 kB]
Get:15 http://deb.debian.org/debian bullseye/main amd64 libapache2-mod-php all 2:7.4+76 [6460 B]
Get:16 http://deb.debian.org/debian bullseye/main amd64 php all 2:7.4+76 [6340 B]
Get:17 http://deb.debian.org/debian bullseye/main amd64 php-pgsql all 2:7.4+76 [6368 B]
Get:18 http://security.debian.org/debian bullseye-security/main amd64 php7.4-json amd64 7.4.33-1+deb11u5 [19.3 kB]
Get:19 http://security.debian.org/debian bullseye-security/main amd64 php7.4-opcache amd64 7.4.33-1+deb11u5 [198 kB]
Get:20 http://security.debian.org/debian bullseye-security/main amd64 php7.4-readline amd64 7.4.33-1+deb11u5 [12.3 kB]
Get:21 http://security.debian.org/debian bullseye-security/main amd64 php7.4-cli amd64 7.4.33-1+deb11u5 [1428 kB]
Get:22 http://security.debian.org/debian bullseye-security/main amd64 libapache2-mod-php7.4 amd64 7.4.33-1+deb11u5 [1377 kB]
Get:23 http://security.debian.org/debian bullseye-security/main amd64 libpq5 amd64 13.14-0+deb11u1 [182 kB]
Get:24 http://security.debian.org/debian bullseye-security/main amd64 php7.4 all 7.4.33-1+deb11u5 [51.2 kB]
```

- 2) Luego se verifica el estado de Apache con el comando “systemctl status apache2”, como se ve en la imagen debe estar activo (running) .

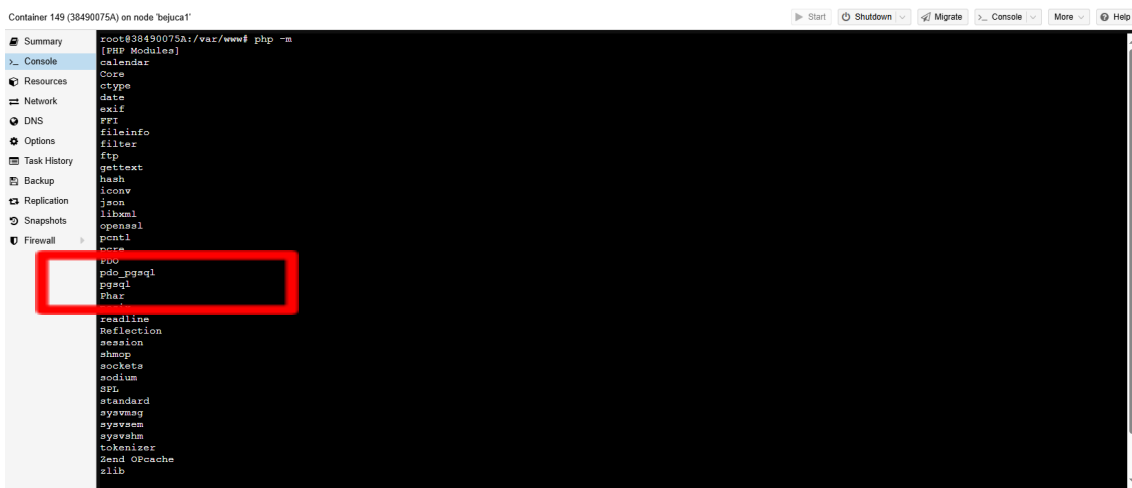
Aclaración: Anteriormente se mencionó que se debe mapear la dirección ip con el puerto: 80, ya que, cuando el software del servidor web Apache2 está instalado en un sistema operativo de forma predeterminada, suele escuchar en el puerto 80 para conexiones HTTP y en el puerto 443 para conexiones HTTPS. ( Para ver si el puerto está escuchando se utiliza el comando “ss –tmp | grep apache”).



```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A: /var/www# systemctl status apache2.service
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; vendor preset: enabled)
   Active: active (running) since Tue 2024-06-18 18:14:59 UTC; 1min 29s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 8093 (apache2)
    Tasks: 6 (limit: 2107)
   Memory: 18.0M
      CPU: 150ms
   CGroup: /system.slice/apache2.service
           └─8093 /usr/sbin/apache2 -k start
             └─8115 /usr/sbin/apache2 -k start
               └─8116 /usr/sbin/apache2 -k start
                 └─8117 /usr/sbin/apache2 -k start
                   └─8118 /usr/sbin/apache2 -k start
                     └─8119 /usr/sbin/apache2 -k start

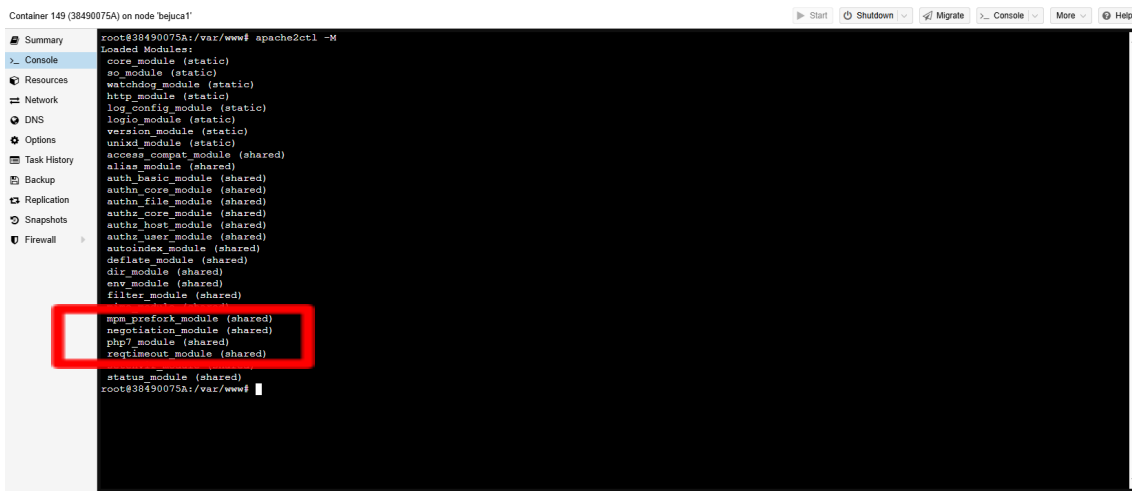
Jun 18 18:14:58 38490075A systemd[1]: Starting The Apache HTTP Server...
Jun 18 18:14:59 38490075A systemd[1]: Started The Apache HTTP Server.
root@38490075A: /var/www#
```

- 3) Habilitamos el módulo pgsql de PHP con el comando “phpenmod pgsql” y verificamos su correcta incorporación con php -m. Esto es necesario para que la aplicación PHP interactúe con una base de datos PostgreSQL (que se utiliza en el contenedor BackEnd).



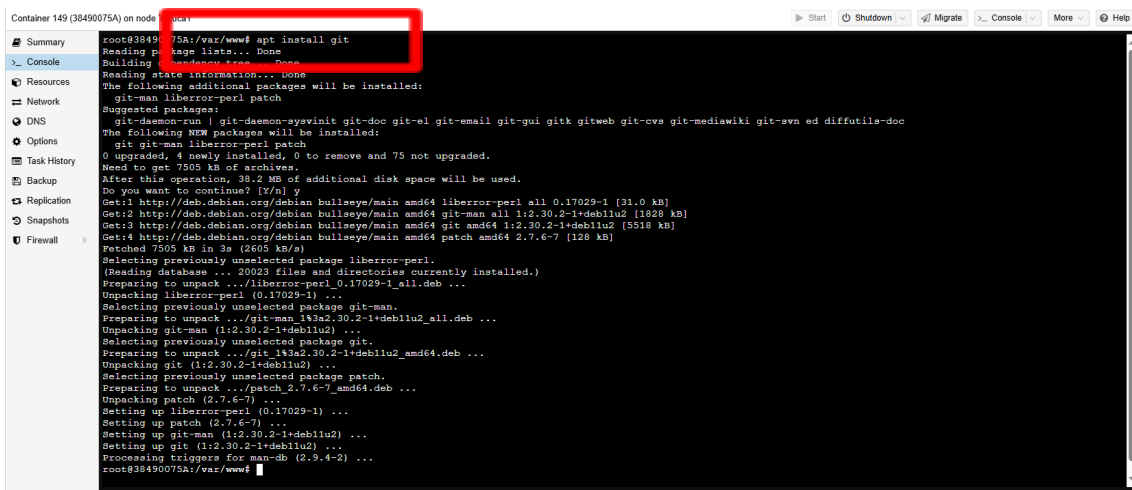
```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A: /var/www# php -m
[PHP Modules]
calendar
Core
ctype
date
date
exif
FFI
fileinfo
filter
ftp
gettext
hash
iconv
json
libxml
openssl
pcntl
pcre
PDO
pdo_pgsql
pgsql
Phar
readline
Reflection
session
shmop
sockets
sodium
SPL
standard
sysvmsg
sysvsem
sysvshm
tokenizer
Zend OPcache
zlib
```

- 4) Verificamos que Apache pueda procesar y servir sitios con PHP mediante `apache2ctl -M`. Para esto determinamos si el módulo `php7_module` fue añadido a la lista.



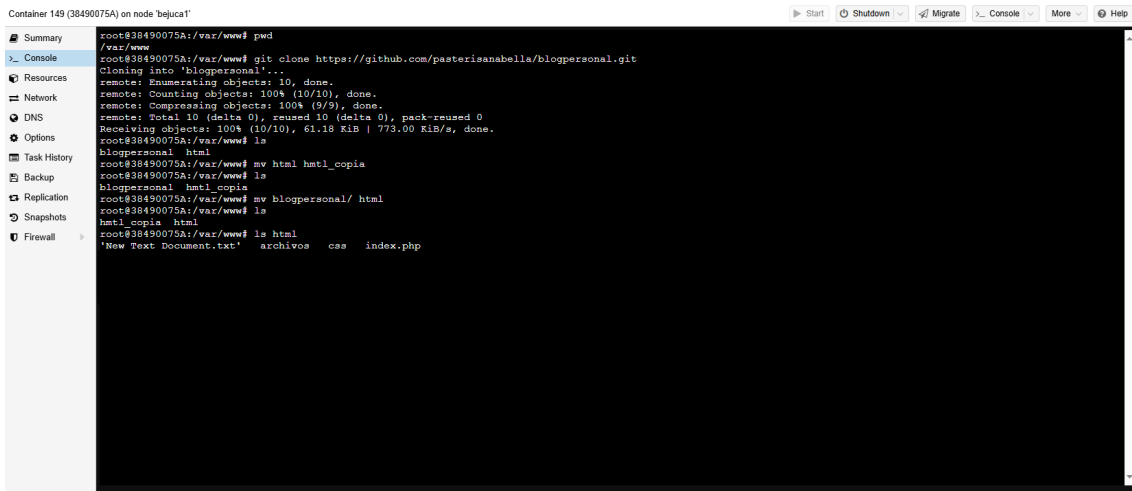
```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A:/var/www# apache2ctl -M
Loaded Modules:
core_module (static)
so_module (static)
watchdog_module (static)
http_module (static)
log_config_module (static)
logio_module (static)
version_module (static)
unixd_module (static)
access_compat_module (shared)
alias_module (shared)
auth_basic_module (shared)
authn_core_module (shared)
authn_file_module (shared)
authz_core_module (shared)
authz_host_module (shared)
authz_user_module (shared)
autoindex_module (shared)
deflate_module (shared)
dir_module (shared)
env_module (shared)
filter_module (shared)
mpm_prefork_module (shared)
negotiation_module (shared)
php7_module (shared)
reqtimeout_module (shared)
status_module (shared)
root@38490075A:/var/www#
```

- 5) Instalamos git en el contenedor por medio del comando “`apt install git`”. Esto nos permitirá descargar el código directamente del repositorio.



```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A:/var/www# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl patch
Suggested packages:
  git-daemon-run | git-daemon-symlink git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn ed diffutils-doc
The following NEW packages will be installed:
  git git-man liberror-perl patch
0 upgraded, 4 newly installed, 0 to remove and 75 not upgraded.
Need to get 7505 kB of archives.
After this operation, 38.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 liberror-perl all 0.17029-1 [31.0 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 git-man all 1:2.30.2-1+deb11u2 [1828 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 git amd64 1:2.30.2-1+deb11u2 [5518 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 patch amd64 2.7.6-7 [128 kB]
Selecting previously unselected package liberror-perl.
(Reading database ... 20023 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1:2.30.2-1+deb11u2_all.deb ...
Unpacking git-man (1:2.30.2-1+deb11u2) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1:2.30.2-1+deb11u2_amd64.deb ...
Unpacking git (1:2.30.2-1+deb11u2) ...
Selecting previously unselected package patch.
Preparing to unpack .../patch_2.7.6-7_amd64.deb ...
Unpacking patch (2.7.6-7) ...
Setting up liberror-perl (0.17029-1) ...
Setting up patch (2.7.6-7) ...
Setting up git-man (1:2.30.2-1+deb11u2) ...
Setting up git (1:2.30.2-1+deb11u2) ...
Processing triggers for man-db (2.9.4-2) ...
root@38490075A:/var/www#
```

- 6) Luego clonamos el repositorio con el blog dentro de `/var/www/`, donde se alojan los sitios accesibles por Apache. Se debe reemplazar la carpeta `/var/www/html` (creada automáticamente durante la instalación) por la carpeta del repositorio clonado (blogpersonal) para que Apache pueda ofrecer acceso al blog.

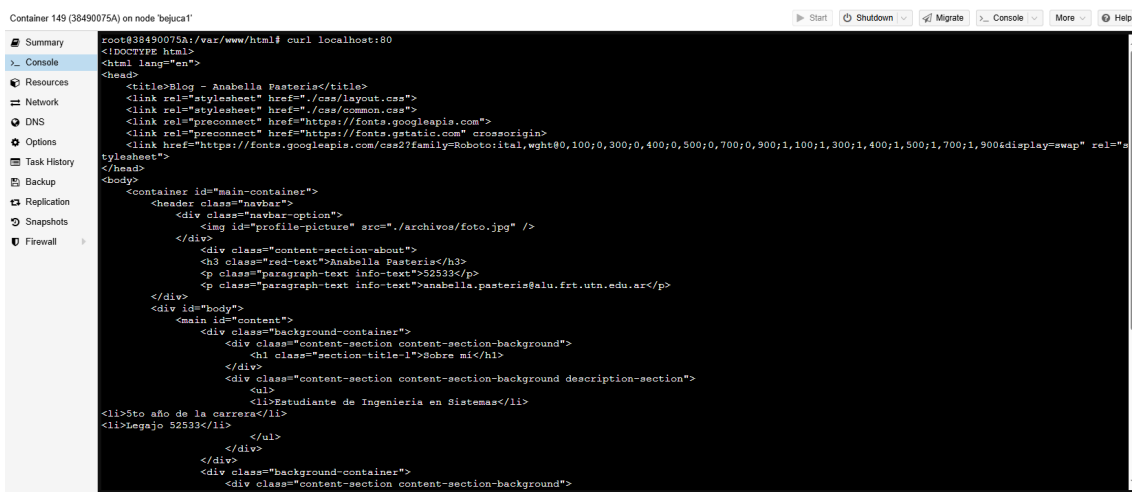


```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A:/var/www# pwd
/var/www
root@38490075A:/var/www# git clone https://github.com/pasterisanabella/blogpersonal.git
Cloning into 'blogpersonal'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 61.18 KiB | 773.00 KiB/s, done.
root@38490075A:/var/www# ls
blogpersonal  html
root@38490075A:/var/www# mv html html_copia
root@38490075A:/var/www# ls
blogpersonal  html_copia
root@38490075A:/var/www# mv blogpersonal/ html
root@38490075A:/var/www# ls
html_copia  html
root@38490075A:/var/www# ls html
'New Text Document.txt'  archivos  css  index.php
```

- 7) Reiniciamos Apache con el comando “`systemctl restart apache2`” e instalamos curl con el comando “`sudo apt-get install curl`” para comprobar si el blog funciona correctamente. El comando curl debe hacerse con `localhost:80`, la IP y puertos configurados en Apache.

Luego se puede visualizar la página web del blog personal.

Aclaración: curl (Client URL) es una herramienta de línea de comandos para transferir datos con URL sintáctico.



```
Container 149 (38490075A) on node 'bejuca1'
root@38490075A:/var/www/html# curl localhost:80
<!DOCTYPE html>
<html lang="en">
<head>
<title>Blog - Anabella Pastoris</title>
<link rel="stylesheet" href="/css/layout.css">
<link rel="stylesheet" href="/css/common.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
</head>
<body>
<container id="main-container">
<header class="navbar">
<div class="navbar-option">

</div>
<div class="content-section-about">
<h3 class="red-text">Anabella Pastoris</h3>
<p class="paragraph-text info-text">52533</p>
<p class="paragraph-text info-text">anabella.pastoris@alu.frt.utn.edu.ar</p>
</div>
<div id="body">
<main id="content">
<div class="background-container">
<div class="content-section content-section-background">
<h1 class="section-title-1">Sobre mí</h1>
</div>
<div class="content-section content-section-background description-section">
<ul>
<li>Estudiante de Ingeniería en Sistemas</li>
</ul>
</div>
</div>
</div>
<div class="background-container">
<div class="content-section content-section-background">
```

## Contenedor DB (BackEnd)

La configuración del mismo se indicará paso a paso, cabe destacar que se utilizarán las herramientas ya mencionadas pero también se instalarán otras dependencias que se necesitarán durante el proceso.

- 1) Actualizar los repositorios del instalador de paquetes e instalar la DB con los siguientes comandos:

- apt update: se utiliza para actualizar la lista de paquetes disponibles en los repositorios de software del sistema.
- apt install postgresql: instala la base de datos PostgreSQL que se utiliza en el contenedor BackEnd.

```
Container 150 (38490075DB) on node 'bejuca1'
```

```
Debian GNU/Linux 11 38490075DB tty1
38490075DB login: root
Password:
Linux 38490075DB 5.15.30-2-pve #1 SMP FVE 5.15.30-3 (Fri, 22 Apr 2022 18:08:27 +0200) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*-copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

last login: Fri Jun 14 15:59:16 UTC 2022 on tty1
root@38490075DB:~# apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:3 http://deb.debian.org/debian bullseye-security InRelease [48.4 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8068 kB]
Get:5 http://security.debian.org/debian-security/main amd64 Packages [275 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages.diff/Index [26.3 kB]
Get:7 http://security.debian.org/debian-security/main Translation-en [177 kB]
Get:8 http://deb.debian.org/debian bullseye/main Translation-en [6236 kB]
Get:9 http://deb.debian.org/debian bullseye/contrib amd64 Packages [50.4 kB]
Get:10 http://deb.debian.org/debian bullseye/contrib Translation-en [46.9 kB]
Get:11 http://deb.debian.org/debian bullseye-updates/main Translation-en [10.9 kB]
Get:12 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2023-12-29-1403.39-F-2023-02-20-2009.57.pdf [6858 B]
Get:12 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2023-12-29-1403.39-F-2023-02-20-2009.57.pdf [6858 B]
Get:13 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [18.8 kB]
Fetched 15.1 MB in 8s (1814 kB/s)
Reading package lists... Done
Building dependency tree... Done
75 packages can be upgraded. Run 'apt list --upgradable' to see them.
N: Repository 'http://deb.debian.org/debian bullseye InRelease' changed its 'Version' value from '11.6' to '11.9'
N: Repository 'http://deb.debian.org/debian bullseye InRelease' changed its 'Suite' value from 'stable' to 'oldstable'
N: Repository 'http://deb.debian.org/debian bullseye-updates InRelease' changed its 'Suite' value from 'stable-updates' to 'oldstable-updates'
N: Repository 'http://security.debian.org/debian-security InRelease' changed its 'Suite' value from 'stable-security' to 'oldstable-security'
root@38490075DB:~#
```

```
Container 150 (38490075DB) on node 'bejuca1'
```

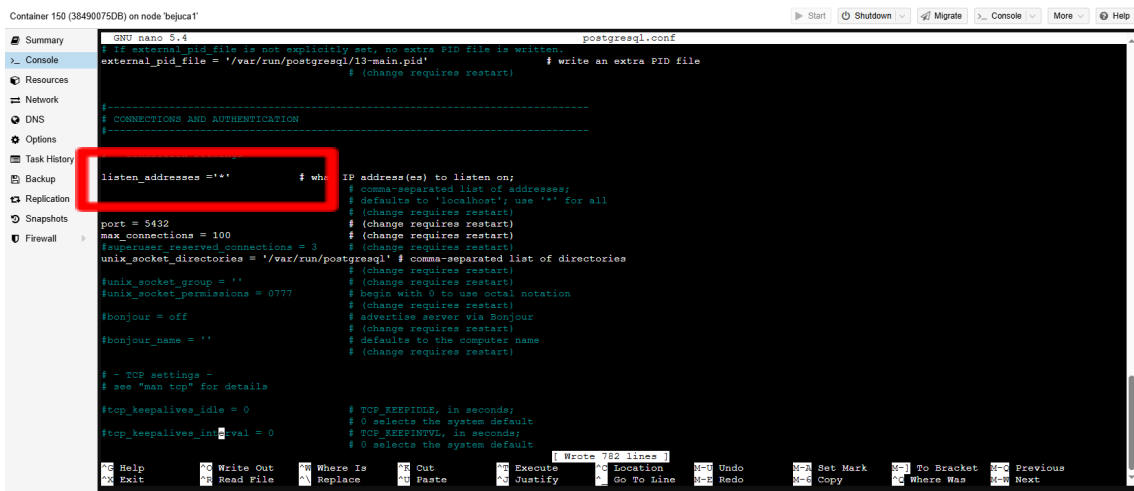
```
root@38490075DB:~# apt install postgresql
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
  libl1vm1 libpq5 libsenors-config libsenors5 libxslt1.1 libz3-4 postgresql-13 postgresql-client-13 postgresql-client-common postgresql-common sysstat
Suggested packages:
  lm-sensors postgresql-doc postgresql-doc-13 libjson-perl isag
The following NEW packages will be installed:
  libl1vm1 libpq5 libsenors-config libsenors5 libxslt1.1 libz3-4 postgresql postgresql-13 postgresql-client-13 postgresql-client-common postgresql-common sysstat
0 upgraded, 12 newly installed, 0 to remove and 75 not upgraded.
Need to get 49.0 MB of archives.
After this operation, 160 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libz3-4 amd64 4.8.10-1 [6949 kB]
Get:2 http://security.debian.org/debian-security/main amd64 libpq5 amd64 13.14-0+deb11u1 [182 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libl1vm1 amd64 1:11.0-1-2 [17.9 MB]
Get:4 http://security.debian.org/debian-security/main amd64 postgresql-client-13 amd64 13.14-0+deb11u1 [1513 kB]
Get:5 http://security.debian.org/debian-security/main amd64 postgresql-13 amd64 13.14-0+deb11u1 [15.2 MB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 libsenors-config all 1:3.6.0-7 [32.3 kB]
Get:7 http://deb.debian.org/debian bullseye/main amd64 libsenors5 amd64 1:3.6.0-7 [52.3 kB]
Get:8 http://deb.debian.org/debian bullseye/main amd64 libxslt1.1 amd64 1.1.34-4+deb11u1 [240 kB]
Get:9 http://deb.debian.org/debian bullseye/main amd64 postgresql-client-common all 225+deb11u1 [89.4 kB]
Get:10 http://deb.debian.org/debian bullseye/main amd64 postgresql-common all 225+deb11u1 [238 kB]
Get:11 http://deb.debian.org/debian bullseye/main amd64 postgresql all 13+225+deb11u1 [64.9 kB]
Get:12 http://deb.debian.org/debian bullseye/main amd64 sysstat amd64 12.5.2-2 [603 kB]
Fetched 49.0 MB in 8s (5355 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libz3-4:amd64.
(Reading database ... 19030 files and directories currently installed.)
Preparing to unpack .../00-libz3-4_4.8.10-1_amd64.deb ...
Unpacking libz3-4:amd64 (4.8.10-1) ...
Selecting previously unselected package libl1vm1:amd64.
Preparing to unpack .../01-libl1vm1_1:11.0-1-2_amd64.deb ...
Unpacking libl1vm1:amd64 (1:11.0-1-2) ...
Selecting previously unselected package libpq5:amd64.
Preparing to unpack .../02-libpq5_13.14-0+deb11u1_amd64.deb ...
Unpacking libpq5:amd64 (13.14-0+deb11u1) ...
Selecting previously unselected package libsenors-config.
Preparing to unpack .../03-libsenors-config_1:3.6.0-7_all.deb ...
```

- 2) Para habilitar las conexiones a la DB se debe editar dos archivos, que se encuentran dentro de los directorios cuando se descarga la base de datos.

*/var/postgresql/13/main/postgresql.conf*

Luego se procede a cambiar la siguiente línea, lo que lleva a aceptar conexiones de cualquier punto, no solo desde el mismo contenedor.

*listen\_address='localhost' a listen\_address='\*'*



```
Container 150 (38490075DB) on node 'bejuca1'
GNU nano 5.4 postgresql.conf
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/13-main.pid' # write an extra PID file
# (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

listen_addresses = '*' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost', use '*' for all
# (change requires restart)

port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
# (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour = off # advertise server via Bonjour
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)

# - TCP settings -
# see "man tcp" for details
#tcp_keepalives_idle = 0 # TCP_KEEPIDL, in seconds;
# 0 selects the system default
#tcp_keepalives_interval = 0 # TCP_KEEPIVL, in seconds;
# 0 selects the system default

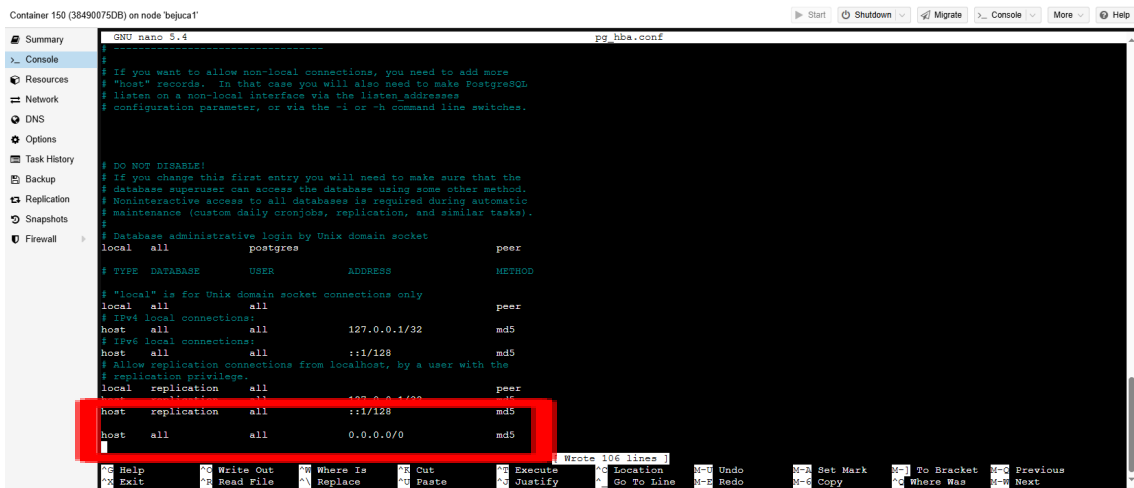
[ Wrote 782 lines ]
Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous
Exit Read File Replace Paste Justify Go To Line Redo Redo Copy Where Was Next
```

- 3) Lo mismo del punto anterior, se realizan cambios en el siguiente directorio:

*/var/postgresql/13/main/postgresql.conf*

Se agregar la siguiente linea, para que las conexiones puedan hacerse de cualquier IP con contraseña segura md5.

*host all all 0.0.0.0/0 md5*



```
Container 150 (38490075DB) on node 'bejuca1'
GNU nano 5.4 pg_hba.conf
#-----
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.
#-----

# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#-----
# Database administrative login by Unix domain socket
local all postgres peer

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all all peer
host replication all ::1/128 md5
host all all 0.0.0.0/0 md5

[ Wrote 106 lines ]
Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous
Exit Read File Replace Paste Justify Go To Line Redo Redo Copy Where Was Next
```

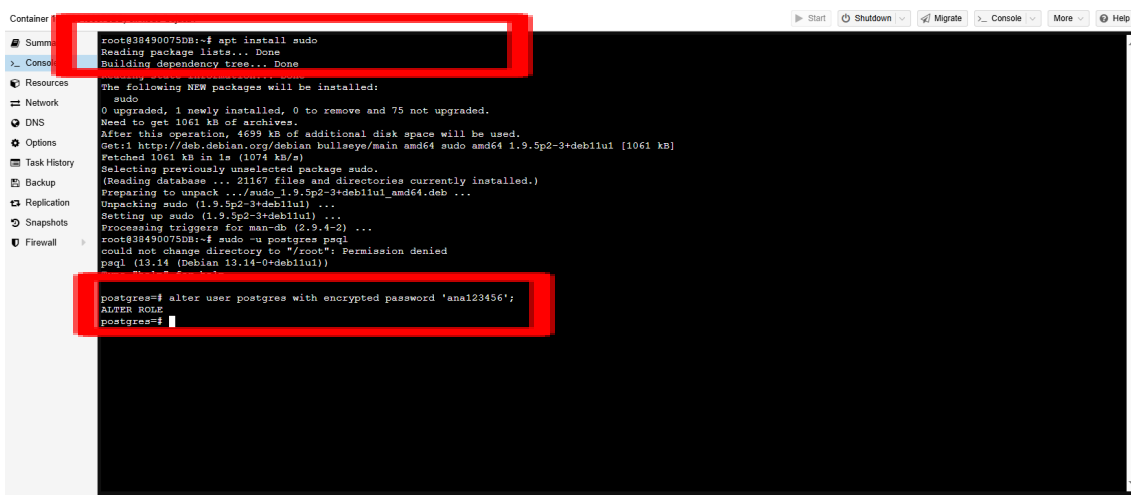


- 4) Para abrir la línea de comandos de la base de datos para realizar las transacciones, primero se instala el comando “ apt install sudo” para poder ejecutar los comandos necesarios. Luego se procede a utilizar el siguiente comando:

- sudo -u postgres psql : permite abrir la línea de comandos de la base de datos para hacer transacciones

Después se le adjudica una contraseña segura al admin de la base de datos PostgreSQL :

- alter user postgres with encrypted password 'ana123456';



```
root@88490075db:~# apt install sudo
Reading package lists... Done
Building dependency tree... Done

The following NEW packages will be installed:
sudo
0 upgraded, 1 newly installed, 0 to remove and 75 not upgraded.
Need to get 1061 kB of archives.
After this operation, 4699 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 sudo amd64 1.9.5p2-3+deb11u1 [1061 kB]
Fetched 1061 kB in 1s (1074 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 21167 files and directories currently installed.)
Preparing to unpack .../sudo_1.9.5p2-3+deb11u1_amd64.deb ...
Unpacking sudo (1.9.5p2-3+deb11u1) ...
Setting up sudo (1.9.5p2-3+deb11u1) ...
Processing triggers for man-db (2.9.4-2) ...
root@88490075db:~# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (13.14 (Debian 13.14-0+deb11u1))
Type "help" for help.

postgres=#
```

- 5) Se procede a crear la base de datos del blog “blog\_db” y luego, nos conectamos a ella. (todo esto se realiza en la misma línea de comando de la base de datos)

- create database blog\_db



```
postgres=# create database blog_db;
CREATE DATABASE
postgres=# \l

   Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
 blog_db | postgres | SQL_ASCII | C       | C     | 
 postgres | postgres | SQL_ASCII | C       | C     | 
 template0 | postgres | SQL_ASCII | C       | C     | =c/postgres+
 template1 | postgres | SQL_ASCII | C       | C     | postgres=CTc/postgres+
(4 rows)

postgres=#
```

- 6) Luego nos conectamos a la base de datos creada, y procedemos a crear las tablas requeridas y luego poblamos la base de datos con los insert correspondientes:



```
postgres=# \c blog_db
You are now connected to database "blog_db" as user "postgres".
blog_db=# CREATE TABLE "Alumnos" (
  "id" integer GENERATED BY DEFAULT AS IDENTITY,
  "Legajo" integer NOT NULL,
  "Nombre" text NOT NULL,
  "Correo" text NOT NULL,
  CONSTRAINT "PK_Alumnos" PRIMARY KEY ("id")
);
CREATE TABLE
```

```

blog_db=# INSERT INTO "Alumnos" ("Legajo", "Nombre", "Correo")
VALUES (52533, 'Anabella Pasteris', 'anabella.pasteris@alu.frt.utn.edu.ar');
blog_db=# INSERT INTO "Datos" ("Description", "AlumnoId")
VALUES ('Estudiante de Ingenieria en Sistemas', 1);

INSERT INTO "Datos" ("Description", "AlumnoId")
VALUES ('5to año de la carrera', 1);

INSERT INTO "Datos" ("Description", "AlumnoId")
VALUES ('Legajo 52533', 1);
INSERT 0 1
INSERT 0 1
INSERT 0 1

```

- 7) Para reflejar los cambios, cerramos psql y reiniciamos la base de datos con el siguiente comando:

systemctl restart postgresql

Aclaración: La estructura de la Base de datos es la siguiente:

