

Bases de Datos

Laboratorio 1

Andrés Calderón, Ph.D.

February 14, 2025

1 Introducción

La idea de este laboratorio es aprender la instalación y configuración de las herramientas básicas que usaremos a lo largo del curso para poner en práctica los conocimientos impartidos durante las sesiones teóricas. Dadas las condiciones del laboratorio, las limitaciones de instalación de nuevo software y para no entorpecer los recursos ya instalados vamos a trabajar en una máquina virtual GNU/Linux. El aplicativo que usaremos para la creación de la máquina virtual será [VirtualBox](#). Puedes descargar el instalador para el sistema operativo que utilizas en tu equipo de computo o usar la versión instalada en los equipos del laboratorio.

El laboratorio consta de tres secciones:

1. Guía paso a paso para la creación de una máquina virtual en VirtualBox para la instalación de Ubuntu Server 24.04.
2. Guía paso a paso para la instalación de Ubuntu Server 24.04 en la máquina virtual previamente creada.
3. Guía paso a paso para la compilación de PostgreSQL 17.2 desde su código fuente.

A continuación describimos el contenido de dichas secciones.

2 Introducción a VirtualBox

VirtualBox es un software de virtualización de código abierto que permite ejecutar múltiples sistemas operativos en un mismo equipo. Es compatible con Windows, macOS y Linux, y es una herramienta ideal para probar sistemas operativos, entornos de desarrollo y configuraciones sin afectar el sistema anfitrión.

2.1 Controles Básicos en VirtualBox

2.1.1 Creación y gestión de máquinas virtuales

- **Nuevo:** Permite crear una nueva máquina virtual.
- **Configuración:** Modifica los parámetros de la máquina virtual antes de iniciarla.
- **Iniciar:** Enciende una máquina virtual seleccionada.
- **Apagar:** Detiene la ejecución de una máquina virtual.
- **Instantáneas:** Guarda el estado actual de la máquina virtual para restaurarlo en cualquier momento.
- **Exportar/Importar:** Permite compartir máquinas virtuales entre equipos.

2.1.2 Controles dentro de una máquina virtual

- **Modo Pantalla Completa:** Visualiza la máquina en pantalla completa.
- **Modo Escalado:** Ajusta la resolución de la ventana de la máquina virtual.
- **Integración del ratón y teclado:** Permite que el ratón y el teclado interactúen sin restricciones.

2.2 Atajos de Teclado en VirtualBox

Atajo	Función
Host + F	Activar/desactivar pantalla completa
Host + L	Bloquear la sesión de la máquina virtual
Host + C	Habilitar/deshabilitar portapapeles compartido
Host + P	Habilitar/deshabilitar vista previa
Host + G	Ajustar ventana de la VM al tamaño de la pantalla
Host + D	Abrir configuración de la máquina virtual
Host + R	Reiniciar la máquina virtual
Host + H	Apagar la máquina virtual
Ctrl + Alt + Supr	Enviar la combinación a la VM

Table 1: Atajos de teclado más usados en VirtualBox

2.3 Guía Paso a Paso: Creación de una Máquina Virtual para Ubuntu Server 24.04

2.3.1 Descargar Ubuntu Server 24.04

Antes de iniciar, descarga la imagen ISO desde:

<https://ubuntu.com/download/server>

2.3.2 Creación de la Máquina Virtual

Paso 1: Crear una nueva máquina virtual

1. Abrir VirtualBox y hacer clic en "Nueva".
2. Asignar un nombre (ej. UbuntuServer24.04).
3. Tipo: Linux, Versión: Ubuntu (64-bit).
4. Asignar al menos **4096 MB** de RAM.
5. Crear un disco virtual (VDI, Reservado dinámicamente, 10GB mínimo, 20GB recomendado).

Paso 2: Configurar la máquina virtual

1. **Procesadores:** Asignar al menos 2 núcleos.
2. **Red:** Elegir NAT o Adaptador Puente.
3. **Montar la imagen ISO:** Ir a Almacenamiento y agregar la ISO descargada.

2.4 Conclusión

Siguiendo estos pasos, habrás creado una máquina virtual y estarás listo para la instalación y configuración de Ubuntu Server 24.04 en VirtualBox.

3 Instalación y Configuración de Ubuntu Server 24.04

3.1 Descarga de Ubuntu Server 24.04

1. Visita la página oficial de Ubuntu Server: <https://ubuntu.com/download/server>
2. Descarga la imagen ISO de Ubuntu Server 24.04 (64-bit).
3. Verifica la integridad del archivo comparando el hash SHA256 (opcional).

3.2 Creación del medio de instalación (Opcional)

Esta sección es únicamente como referencia en el caso de que quieras instalar Ubuntu en una partición de tu disco duro. Para nuestro caso, vamos a trabajar en una máquina virtual así que puedes saltar a la sección [3.3](#)

3.2.1 Opción 1: Crear un USB Bootable (recomendado)

1. Conectar una memoria USB (mínimo 4 GB).
2. Usar una herramienta para crear el USB de arranque:
 - **Windows:** [Rufus](#)
 - **Linux:** dd o [balenaEtcher](#)
 - **macOS:** dd o balenaEtcher

3. Grabar la imagen ISO en la USB:

```
sudo dd if=ubuntu-24.04-live-server-amd64.iso of=/dev/sdX bs=4M status=progress
```

Reemplaza /dev/sdX con tu unidad USB.

3.2.2 Opción 2: Grabar en un DVD (menos recomendado)

1. Usa **Brasero**, **ImgBurn** o cualquier grabador de discos.
2. Quema la ISO en un DVD.

3.2.3 Configuración del BIOS/UEFI

1. Accede a la BIOS/UEFI (normalmente con F2, F12, DEL o ESC al encender).
2. Habilita el modo UEFI (si es compatible).
3. Deshabilita Secure Boot si tienes problemas al iniciar la instalación.
4. Cambia el orden de arranque para que el USB/DVD sea la primera opción.

3.3 Instalación de Ubuntu Server 24.04

1. Inicia desde el USB/DVD y selecciona **Install Ubuntu Server**.
2. Selecciona el idioma (Español o Inglés recomendado).
3. Configura la distribución del teclado.
4. Configura la red:
 - Si tienes DHCP, la red se configura automáticamente.
 - Para IP estática, selecciona "Configuración manual" e introduce los valores correctos.
5. Elige el tipo de instalación:
 - **Ubuntu Server (sin interfaz gráfica)** es la opción estándar.
 - Puedes elegir una instalación mínima si deseas un sistema más ligero.
6. Configura el almacenamiento:
 - Opción recomendada: **Usar disco entero y configurar LVM** (permite ampliar particiones en el futuro).
 - Si deseas RAID o particionado manual, elige la opción correspondiente.
7. Crea un usuario y contraseña.
8. Opcional: Instala OpenSSH marcando la opción "Install OpenSSH server".
9. Selecciona software adicional (Docker, Kubernetes, etc.).
10. Confirma la instalación y espera a que finalice.

3.4 Primer arranque y configuración post-instalación

1. Reinicia el sistema y retira el medio de instalación.
2. Inicia sesión con tu usuario y contraseña.
3. Actualiza el sistema:

```
sudo apt update && sudo apt upgrade -y
```

4. Configura la zona horaria (si no se configuró correctamente):

```
sudo timedatectl set-timezone America/Bogota
```

5. Configura el firewall (opcional pero recomendado):

```
sudo ufw allow OpenSSH  
sudo ufw enable
```

6. Verifica la dirección IP asignada:

```
ip a
```

3.5 Configuración adicional

- Crear un usuario adicional con permisos sudo. Ten presente el nombre que utilizarás como `nuevo_usuario` y su contraseña porque la necesitaremos más adelante. Nos vamos a referir a el como nuestro usuario linux:

```
sudo adduser nuevo_usuario
sudo usermod -aG sudo nuevo_usuario
```

- Configurar acceso remoto con SSH:

```
ssh usuario@ip_del_servidor
```

- Instalar paquetes esenciales:

```
sudo apt install htop net-tools curl git emacs-nox terminator -y
```

- Configurar un hostname personalizado, puedes utilizar el nombre que prefieras como valor para `mi_servidor`:

```
sudo hostnamectl set-hostname mi_servidor
```

3.6 Conclusión

Con estos pasos, Ubuntu Server 24.04 estará listo para usarse. A partir de aquí, puedes instalar servicios como Apache, Nginx, PostgreSQL, Docker, Kubernetes, etc., según tus necesidades. **¡Éxito en tu instalación!**

4 Compilación de PostgreSQL

4.1 Actualizar el Sistema

Antes de empezar, es recomendable actualizar el sistema para asegurarnos de tener las últimas versiones de los paquetes.

```
sudo apt update && sudo apt upgrade -y
```

4.2 Instalar Dependencias Necesarias

PostgreSQL requiere varias bibliotecas y herramientas para compilarse correctamente. Ten en cuenta que usamos el caracter ‘\’ como indicador de salto de línea. Si lo utilizas en la terminal deberas digitar **Enter** y continuar escribiendo en la siguiente línea de la terminal. Si prefieres puedes obviar el ‘\’ y escribir todo en una sola línea.

```
sudo apt install -y build-essential libreadline-dev zlib1g-dev flex bison libxml2-dev \
libxslt1-dev libssl-dev libperl-dev libpam0g-dev libldap2-dev python3-dev uuid-dev
```

4.3 Descargar la Última Versión de PostgreSQL

Obtenemos la versión más reciente desde el sitio oficial de PostgreSQL.

```
cd /usr/local/src
wget https://ftp.postgresql.org/pub/source/v17.2/postgresql-17.2.tar.gz
```

(Sustituye ‘17.2’ por la última versión disponible si hay una más reciente).

4.4 Extraer el Código Fuente

Descomprimos el archivo descargado.

```
tar -xvzf postgresql-17.2.tar.gz
cd postgresql-17.2
```

4.5 Configurar la Compilación

Ejecutamos el script de configuración con las opciones necesarias.

```
./configure --prefix=/usr/local/pgsql --with-ssl=openssl --with-python --with-perl \
    ICU_CFLAGS='-I/usr/include' ICU_LIBS='-L/usr/lib -licui18n -licuuc -licudata'
```

Opcionalmente, puedes añadir otras opciones como `--with-libxml`, `--with-systemd`, etc. Puedes consultar más opciones en <https://www.postgresql.org/docs/current/install-make.html#CONFIGURE-OPTIONS>.

4.6 Compilar PostgreSQL

Ahora compilamos PostgreSQL utilizando 'make'.

```
make -j$(nproc)
```

4.7 Instalar PostgreSQL

Si la compilación fue exitosa, instalamos PostgreSQL en el sistema.

```
sudo make install
```

4.8 Crear el Superusuario Postgres y Configurar Directorios

Creamos un usuario específico para PostgreSQL y configuramos los directorios. Recuerda memorizar la contraseña del usuario 'postgres' porque la utilizaremos más adelante.

```
sudo useradd postgres
sudo mkdir -p /usr/local/pgsql/data
sudo chown postgres:postgres /usr/local/pgsql/data
```

4.9 Inicializar la Base de Datos

Iniciamos PostgreSQL con el usuario 'postgres'.

```
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start
/usr/local/pgsql/bin/createdb test
/usr/local/pgsql/bin/psql test
```

Creamos un rol de superusuario para nuestro usuario linux (el que configuramos en la sección 3.5):

```
CREATE ROLE nuevo_usuario WITH LOGIN SUPERUSER PASSWORD 'password_nuevo_usuario';
```

Para salir de la sesión de 'psql' usamos:

```
\q
```

5 Accesos directos a los Binarios de PostgreSQL

Si continuamos logeados como el usuario 'postgres' salimos pulsando:

```
exit
```

Después de confirmar que estamos logeados como nuestro usuario linux, editamos el siguiente archivo:

```
nano ~/.bashrc
```

y adicionamos las siguientes líneas al final del archivo:

```
PATH=/usr/local/pgsql/bin:$PATH
export PATH
```

Recargamos el fichero con:

```
source ~/.bashrc
```

y ya no necesitaremos digitar la ruta completa para ejecutar los comandos de PostgreSQL.

5.1 Configurar el Servicio de PostgreSQL

Creamos un archivo de servicio para 'systemd'.

```
sudo nano /etc/systemd/system/postgresql.service
```

Pegamos el siguiente contenido:

```
[Unit]
Description=PostgreSQL database server
Documentation=man:postgres(1)
After=network-online.target
Wants=network-online.target
[Service]
User=postgres
ExecStart=/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGINT
TimeoutSec=infinity

[Install]
WantedBy=multi-user.target
```

Guardamos y salimos ('Ctrl + X', luego 'Y' y 'Enter').

5.2 Iniciar y Habilitar PostgreSQL

Recargamos 'systemd' y habilitamos PostgreSQL para que se inicie automáticamente.

```
sudo systemctl daemon-reload
sudo systemctl enable postgresql
sudo systemctl start postgresql
```

5.3 Verificar el Estado del Servidor

Para confirmar que PostgreSQL está corriendo correctamente, usamos:

```
sudo systemctl status postgresql
```

Si PostgreSQL está funcionando bien, deberíamos ver un mensaje indicando que está activo.

5.4 Acceder a PostgreSQL desde el Usuario Linux

Desde ahora podremos iniciar sesión en PostgreSQL con nuestro usuario linux (`'nuevo_usuario'`). Para listar las bases de datos presentes en el sistema usamos:

```
psql -l
```

5.5 ¡PostgreSQL está listo para usarse!

Ahora puedes configurar usuarios, bases de datos y tunear PostgreSQL según tus necesidades.