

S.I.G.T.

Sistemas Operativos III

APPTEC

Rol	Apellido	Nombre	C.I	Email	Tel/Cel.
Coordinador	Pereyra	Emiliano	4.774.396-6	epereyra@apptecuy.com	092324130
Sub- Coordinador	Varela	Michael	4.543.461-8	mvarela@apptecuy.com	099297255
Integrante 1	González	Mauro	5.251.060-7	mgonzalez@apptecuy.com	094866094
Integrante 2	Otero	Gonzalo	5.014.881-8	gotero@apptecuy.com	094762305

Docente: Rodríguez, Carlos

Fecha de culminación

24/07/2023

PRIMERA ENTREGA

I.S.B.O.

3BH

ÍNDICE:

Estudio de los diferentes usuarios del sistema.	3
Root.	3
Administrador del sistema.	3
Instaladores.	3
Usuario común.	3
Sistema operativo a utilizar en el servidor.	4
Justificación.	4
Instalación del servidor.	5
Instalación de docker.	10
Instalación de servidor LAMP.	12
Instalación del servidor web Apache.	12
Instalación MYSQL.	13
Instalación PHP.	15
Primera versión del script.	16
Menú.	16
Alta usuario.	17
Baja usuario.	17
Modificación usuario.	18
Alta grupo.	19

Estudio de los diferentes usuarios del sistema.

En principio tenemos pensado 4 usuarios root, administrador del sistema, instaladores y usuario común.

Root.

El usuario que tendrá todos los privilegios del sistema.

Administrador del sistema.

Estará un escalón por debajo del root, este usuario será para dar el alta de usuarios nuevos, así como para modificarlos y eliminarlos. Este usuario también tendrá permisos para instalar aplicaciones, correr procesos por ejemplo de respaldo, etc.

Instaladores.

Tendrá solo permisos para instalar y desinstalar programas.

Usuario común.

Solo tendrá permisos para visualizar información si es necesario en el sistema, no podrá crear, ni modificar usuarios, tampoco instalar ningún programa.

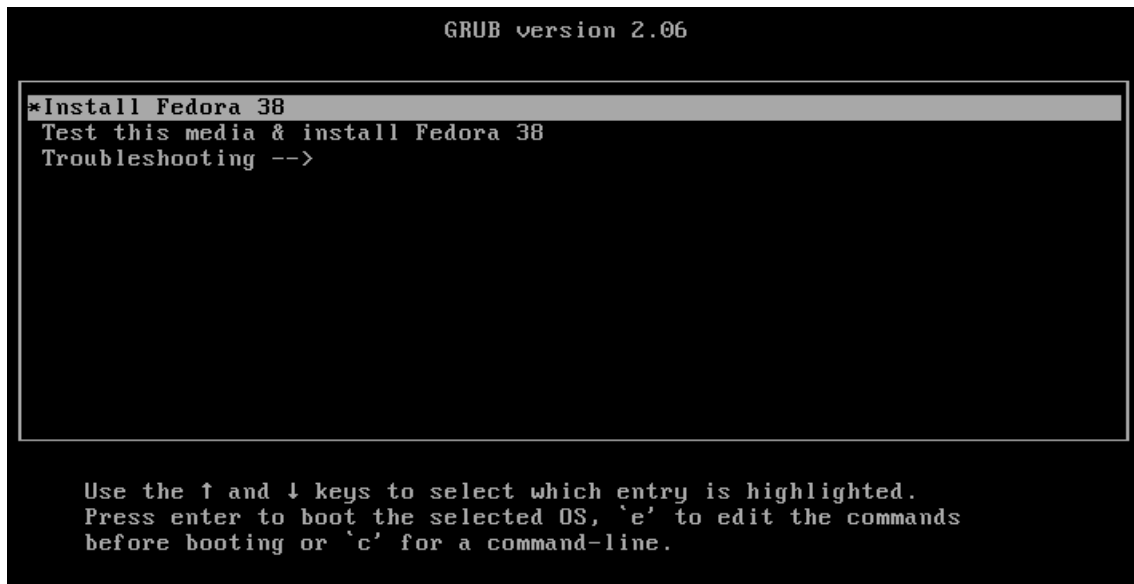
Sistema operativo a utilizar en el servidor.

Optamos por el sistema Fedora Server en su última versión la 38.

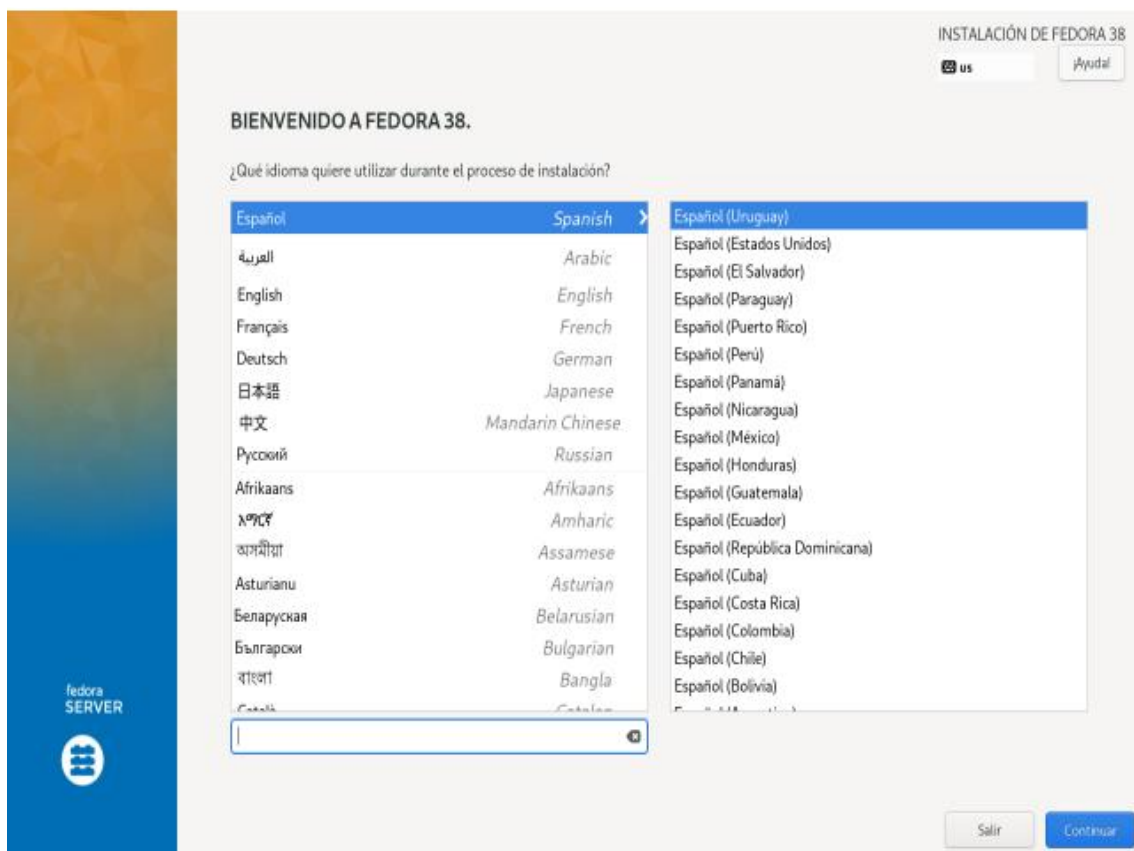
Justificación.

Elegimos este sistema porque Fedora adopta las últimas versiones de software para mantenerse al día, larga actualizaciones frecuentemente y correcciones de seguridad rápidas debido a su ciclo de lanzamiento rápido. Viene habilitado por defecto con SELinux, lo que mejora la seguridad. Utiliza la gestión de paquetes por DNF, lo que facilita la instalación y actualización del software. Ofrece soporte para varias arquitecturas de hardware, además cuenta con una comunidad activa que proporciona soporte y recursos.

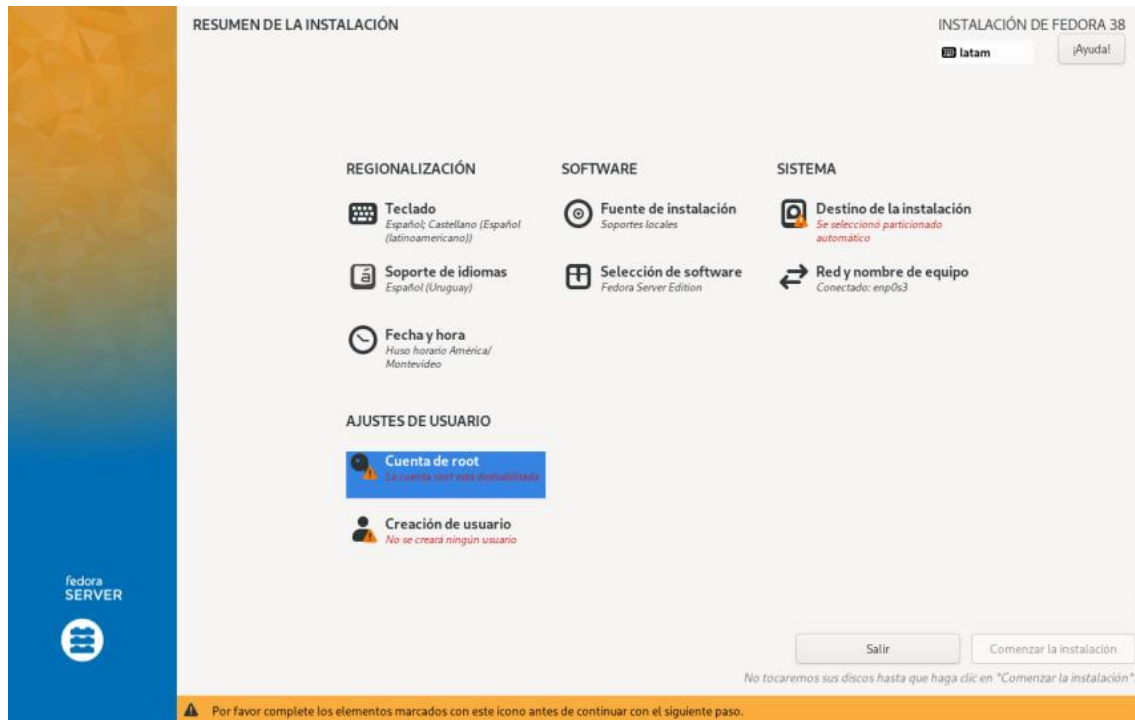
Instalación del servidor.



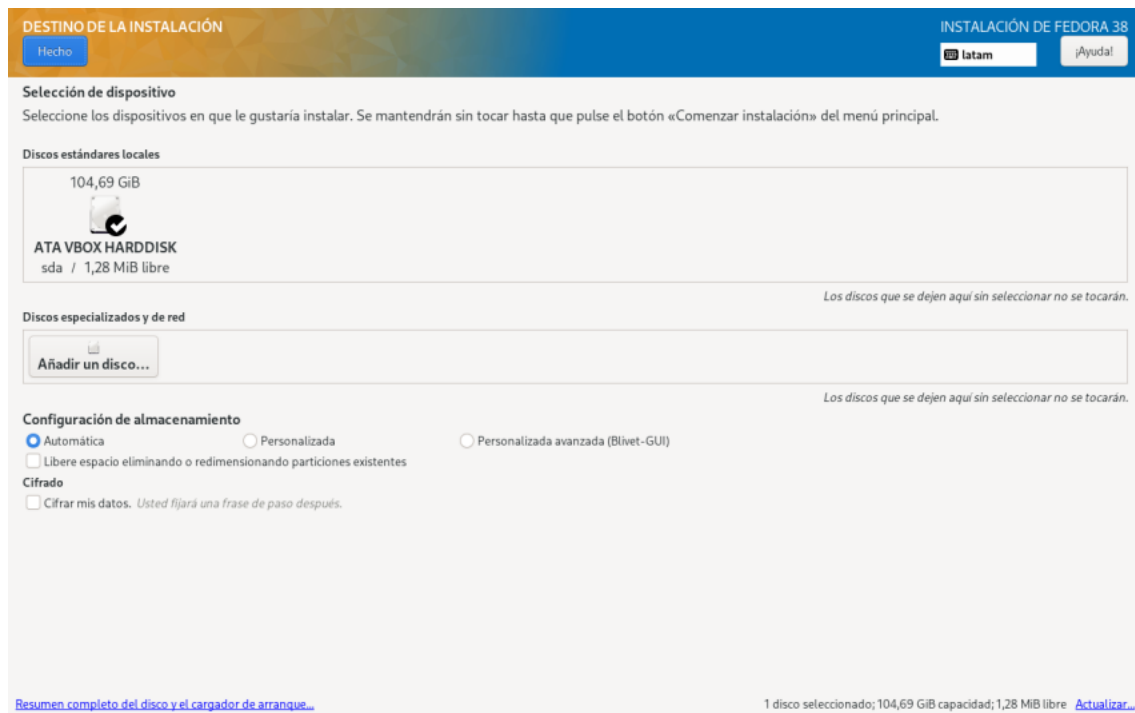
Aquí elegimos el idioma deseado, en este caso elegiremos español.



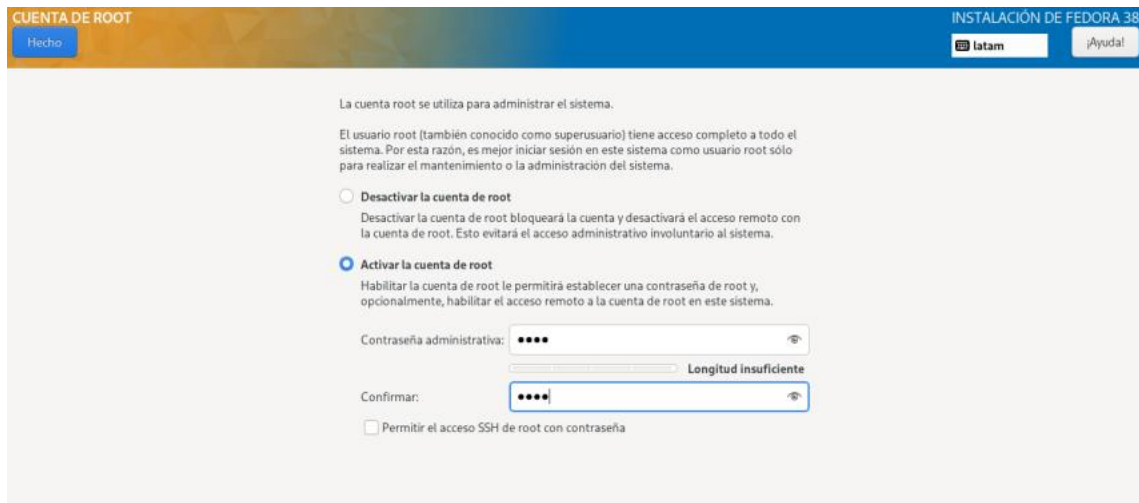
En este paso, deberemos elegir el disco destino de la instalación, elegir si activar la cuenta root o no y decidir si ya crear un usuario.



Aquí elegimos el disco donde se instalara.



Activación de la cuenta de root.



CUENTA DE ROOT INSTALACIÓN DE FEDORA 38

[Hecho](#) [latam](#) [¡Ayuda!](#)

La cuenta root se utiliza para administrar el sistema.

El usuario root (también conocido como superusuario) tiene acceso completo a todo el sistema. Por esta razón, es mejor iniciar sesión en este sistema como usuario root sólo para realizar el mantenimiento o la administración del sistema.

☐ **Desactivar la cuenta de root**
Desactivar la cuenta de root bloqueará la cuenta y desactivará el acceso remoto con la cuenta de root. Esto evitará el acceso administrativo involuntario al sistema.

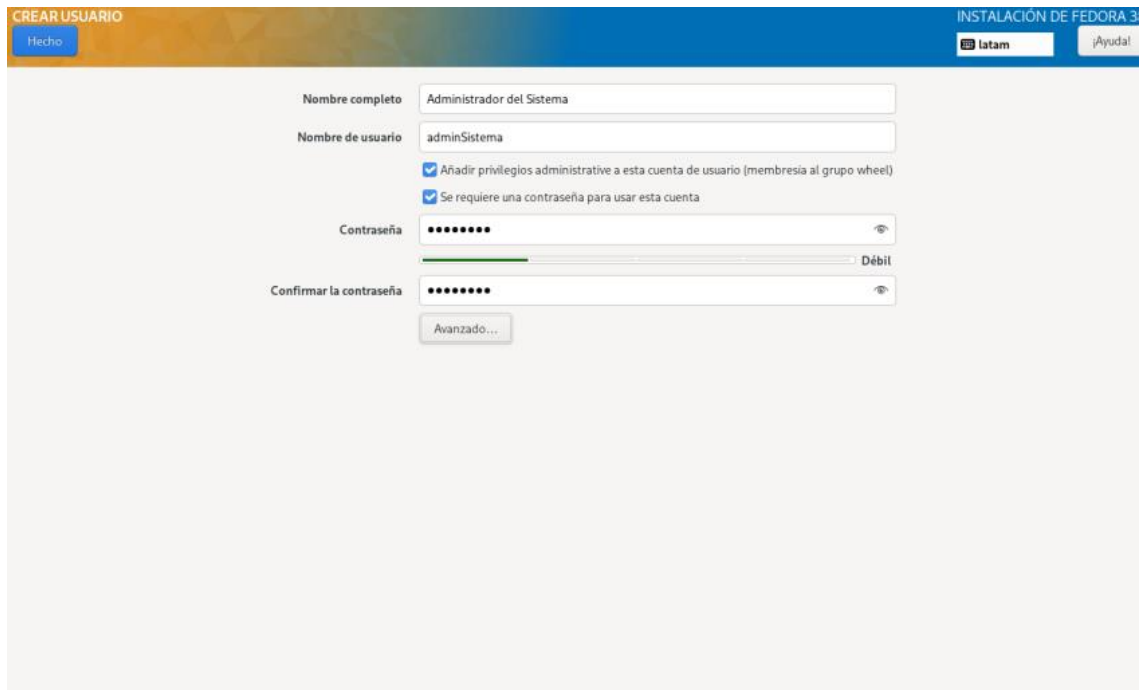
☒ **Activar la cuenta de root**
Habilitar la cuenta de root le permitirá establecer una contraseña de root y, opcionalmente, habilitar el acceso remoto a la cuenta de root en este sistema.

Contraseña administrativa: Longitud insuficiente

Confirmar:

☐ Permitir el acceso SSH de root con contraseña

Creación del usuario que tendrá privilegios para administrar el sistema.



CREAR USUARIO INSTALACIÓN DE FEDORA 38

[Hecho](#) [latam](#) [¡Ayuda!](#)

Nombre completo:

Nombre de usuario:

☒ Añadir privilegios administrativos a esta cuenta de usuario (membresía al grupo wheel)

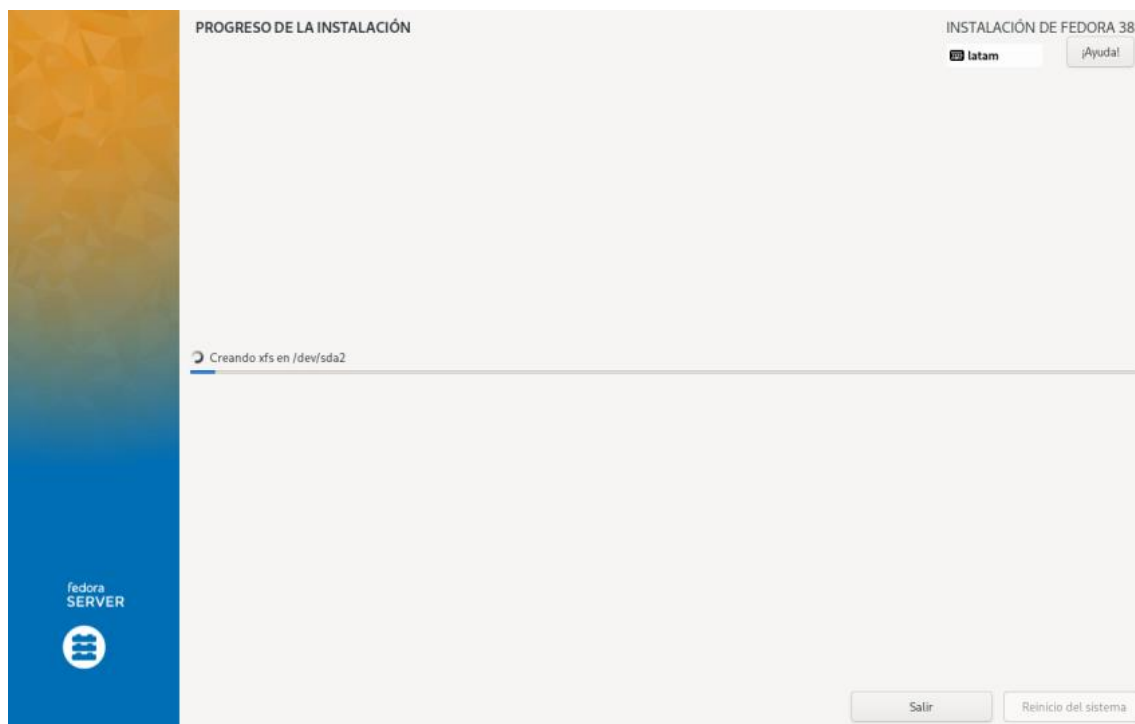
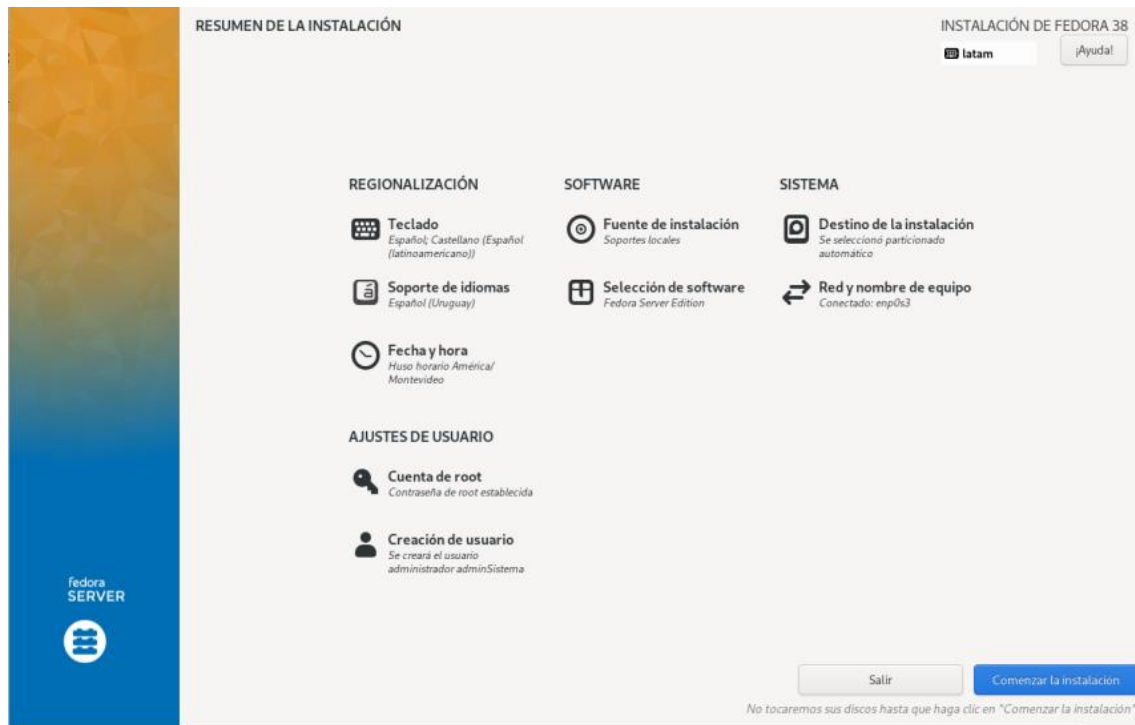
☒ Se requiere una contraseña para usar esta cuenta

Contraseña: Débil

Confirmar la contraseña:

[Avanzado...](#)

Una vez que queden prontos estos pasos le damos a “Comenzar la instalación”.



Una vez que termina le damos a reiniciar.



Instalación de docker.

Instalar el dnf-plugins-core (que proporciona los comandos para administrar sus repositorios DNF) y configure el repositorio.

Comandos: `sudo dnf -y install dnf-plugins-core`
`sudo dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo`

```
[liveuser@localhost-live ~]$ sudo dnf -y install dnf-plugins-core
Fedora 38 - x86_64                               1.1 MB/s | 83 MB      01:14
Fedora 38 openh264 (From Cisco) - x86_64        865 B/s | 2.5 kB     00:02
Fedora Modular 38 - x86_64                      57 kB/s | 2.8 MB     00:49
Fedora 38 - x86_64 - Updates                    468 kB/s | 29 MB     01:02
Fedora Modular 38 - x86_64 - Updates             44 kB/s | 2.1 MB     00:49
00:01 ETA
Last metadata expiration check: 0:00:01 ago on Mon 24 Jul 2023 03:37:00 PM EDT.
Package dnf-plugins-core-4.4.0-1.fc38.noarch is already installed.
Dependencies resolved.
Nothing to do.
[liveuser@localhost-live ~]$ sudo dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
Adding repo from: https://download.docker.com/linux/fedora/docker-ce.repo
[liveuser@localhost-live ~]$
```

Instalar Docker Engine, containerd y Docker Compose:

Comando: `sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`

```
[liveuser@localhost-live ~]$ sudo dnf install docker-ce-cli containerd.io docker-buildx-plugin d
ocker-compose-plugin
Docker CE Stable - x86_64                        14 kB/s | 12 kB      00:00
Dependencies resolved.
=====
Package                                Architecture Version                Repository              Size
=====
Installing:
containerd.io                          x86_64         1.6.21-3.1.fc38         docker-ce-stable        33 M
docker-buildx-plugin                   x86_64         0.11.2-1.fc38           docker-ce-stable        13 M
docker-ce-cli                          x86_64         1:24.0.5-1.fc38         docker-ce-stable        7.1 M
docker-compose-plugin                  x86_64         2.20.2-1.fc38           docker-ce-stable        13 M
Installing dependencies:
container-selinux                       noarch         2:2.209.0-1.fc38        fedora                   51 k
=====
Transaction Summary
=====
Install 5 Packages

Total download size: 65 M
Installed size: 259 M
Is this ok [y/N]:
```

Iniciar Docker.

Comando: `sudo systemctl start docker`

```
[liveuser@localhost-live ~]$ sudo systemctl start docker
[liveuser@localhost-live ~]$
```

Verificar que la instalación de Docker Engine se haya realizado correctamente ejecutando el siguiente comando.

Comando: `sudo docker run hello-world`

```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Instalación de servidor LAMP.

Instalación del servidor web Apache.

Comando: `sudo dnf -y install httpd`

```
[liveuser@localhost-live ~]$ sudo dnf -y install httpd
Last metadata expiration check: 0:07:29 ago on Mon 24 Jul 2023 04:29:53 PM EDT.
Dependencies resolved.
=====
Package                                Architecture Version                Repository              Size
=====
Installing:
httpd                                   x86_64          2.4.57-1.fc38          updates                  51 k
Installing dependencies:
fedora-logos-httpd                    noarch          38.1.0-1.fc38          fedora                   16 k
httpd-core                             x86_64          2.4.57-1.fc38          updates                  1.4 M
httpd-filesystem                       noarch          2.4.57-1.fc38          updates                  13 k
httpd-tools                            x86_64          2.4.57-1.fc38          updates                  80 k
Installing weak dependencies:
julietaula-montserrat-fonts           noarch          1:7.222-4.fc38         fedora                   1.6 M
mod_http2                             x86_64          2.0.11-2.fc38         fedora                   147 k
mod_lua                                x86_64          2.4.57-1.fc38         updates                  59 k
=====
Transaction Summary
=====
Install  8 Packages
Total download size: 3.4 M
```

Comandos básicos para administrar el servidor:

Iniciar:

`sudo systemctl start httpd`

Parar:

`sudo systemctl stop httpd`

Habilitar:

`sudo systemctl enable httpd`

Reiniciar:

`sudo systemctl restart httpd`

Recargar:

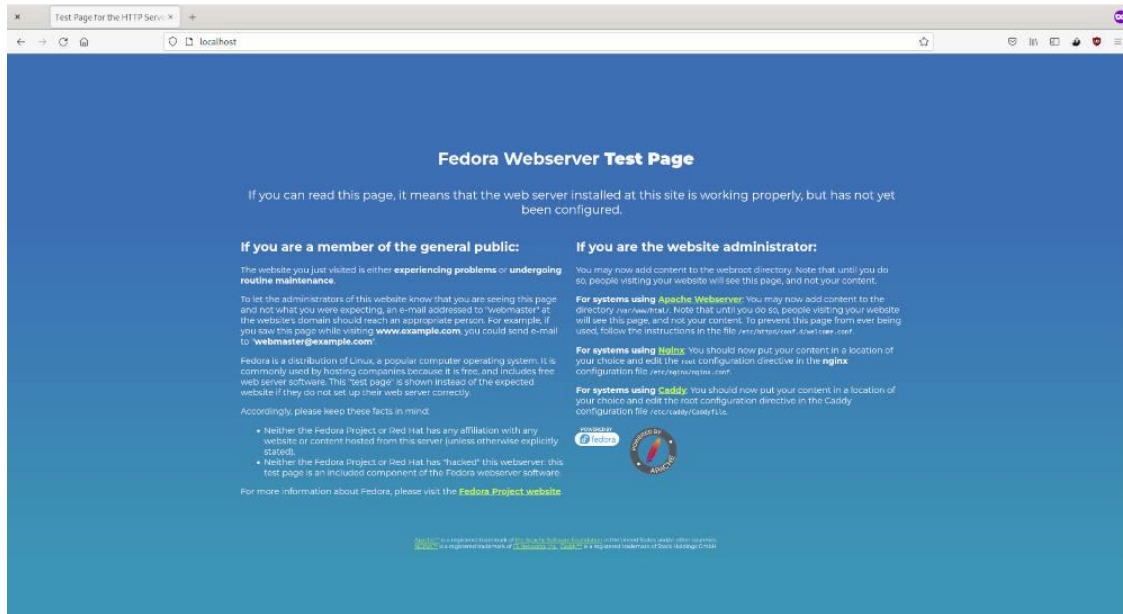
`sudo systemctl reload httpd`

Una vez instalado, ejecutamos los comandos de habilitar e iniciar:

```
[liveuser@localhost-live ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.

[liveuser@localhost-live ~]$ sudo systemctl start httpd
[liveuser@localhost-live ~]$
```

Nos dirigimos a un navegador web y entramos en <http://localhost>



Instalación MYSQL.

Para empezar, ejecutamos: `sudo dnf install mariadb-server`

```
[liveuser@localhost-live ~]$ sudo dnf install mariadb-server
Last metadata expiration check: 0:18:57 ago on Mon 24 Jul 2023 04:29:53 PM EDT.
Package mariadb-server-3:10.5.18-1.fc38.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[liveuser@localhost-live ~]$
```

Los comandos para gestionar el servicio son muy similares a Apache.

Habilitar:

```
sudo systemctl enable mariadb
```

Iniciar:

```
sudo systemctl start mariadb
```

Parar:

```
sudo systemctl stop mariadb
```

Reiniciar:

```
sudo systemctl restart mariadb
```

Como con Apache, ejecutamos los comandos habilitar e iniciar

```
[liveuser@localhost-live ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[liveuser@localhost-live ~]$ sudo systemctl start mariadb
```

Seguridad MySQL

Por defecto, el gestor de bases de datos viene con el usuario root sin contraseña. Vamos a asignar una contraseña ejecutando: `sudo mysql_secure_installation`

```
Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

Nota: Nos hará una serie de preguntas sobre si queremos eliminar usuarios anónimos, borrar las tablas de test, etc. Podemos contestar afirmativamente a todo para poder finalizar

Ahora entraremos en mysql: `sudo mysql -u root`

```
[liveuser@localhost-live ~]$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.5.18-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> s5
```

Vamos a crear un usuario distinto de root que será el que asignemos a las bases de datos que creamos después:

`CREATE USER mi_usuario IDENTIFIED BY 'password';`

```
[liveuser@localhost-live ~]$ CREATE USER mi_usuario IDENTIFIED BY 'password';
```


Creamos una nueva base de datos:

CREATE DATABASE mibase;

```
[liveuser@localhost-live ~]$ CREATE DATABASE mibase;
```

Y asignamos nuestro usuario anterior a la nueva base de datos con todos los privilegios:

GRANT ALL PRIVILEGES ON mibase.* TO mi_usuario

```
[liveuser@localhost-live ~]$ GRANT ALL PRIVILEGES ON mibase*TO mi_usuario
```

Instalación PHP.

Vamos a instalar los siguientes paquetes para PHP:

```
sudo dnf -y install php php-cli php-php-gettext php-mbstring php-mcrypt php-mysqldb  
php-pear php-curl php-gd php-xml php-bcmath php-zip
```

```
php-php-gettext      noarch      1.0.12-14.fc38      fedora      24 k
Installing dependencies:
libmccrypt           x86_64      2.5.8-34.fc38      fedora      108 k
nginx-filesystem     noarch      1:1.24.0-1.fc38    updates     11 k
php-cli              x86_64      8.2.8-1.fc38      updates     5.4 M
php-common           x86_64      8.2.8-1.fc38      updates     853 k
php-mbstring         x86_64      8.2.8-1.fc38      updates     574 k
php-pdo              x86_64      8.2.8-1.fc38      updates     119 k
php-process          x86_64      8.2.8-1.fc38      updates     69 k
php-xml              x86_64      8.2.8-1.fc38      updates     216 k
Installing weak dependencies:
php-fedora-autoloader noarch      1.0.1-10.fc38      fedora      12 k
php-fpm              x86_64      8.2.8-1.fc38      updates     1.9 M
php-opcache           x86_64      8.2.8-1.fc38      updates     618 k
php-sodium            x86_64      8.2.8-1.fc38      updates     62 k
Transaction Summary
-----
Install 20 Packages

Total download size: 11 M
Installed size: 59 M
Downloading Packages:
[1-3/20]: php-fedora-autoloa 0% [ ] --- B/s | 0 B --- ETA
```

Tras la instalación vamos a comprobar la versión de PHP con el comando: `php -v`

```
[liveuser@localhost-live ~]$ php -v
PHP 8.2.8 (cli) (built: Jul 4 2023 14:52:54) (NTS gcc x86_64)
Copyright (c) The PHP Group
Zend Engine v4.2.8, Copyright (c) Zend Technologies
with Zend OPcache v8.2.8, Copyright (c), by Zend Technologies
[liveuser@localhost-live ~]$
```

Primera versión del script.

Menú.

```
apptec@apptec-VirtualBox: ~/Escritorio/APPTec/Menu
Bienvenido al sistema

1)Alta de usuario
2)Baja de usuario
3)Modificacion de usuario
4)Alta de grupo

Ingrese una opcion: 
```

```
#!/bin/bash
clear

echo "Bienvenido al sistema"

echo "
1)Alta de usuario
2)Baja de usuario
3)Modificacion de usuario
4)Alta de grupo
"

read -p "Ingrese una opcion: " op
while [[ $op != 0 ]]
do
case $op in
1) bash Alta.sh|
sleep 2;;
2) bash Baja.sh
sleep 2;;
3) bash Modificacion.sh
sleep 2;;
4) bash AltaG.sh
sleep 2;;
esac
done
```


Alta usuario.

```
apptec@apptec-VirtualBox: ~/Escritorio/APPTec/Menu
Bienvenido al sistema

1)Alta de usuario
2)Baja de usuario
3)Modificacion de usuario
4)Alta de grupo

Ingrese una opcion: 1
Ingrese un usuario: prueba
Ingrese grupo donde alojarlo: prueba
Ingrese contraseña: appte
[sudo] contraseña para apptec:
Creando Usuario

#!/bin/bash
read -p "Ingrese un usuario: " nom
read -p "Ingrese grupo donde alojarlo: " grp
read -p "Ingrese contraseña: " cont
sudo useradd -g ${grp} -s /bin/bash -d /home/${nom} -p ${cont} -m ${nom}
echo "Creando Usuario"
sleep 2
|
```

Baja usuario.

```
apptec@apptec-VirtualBox: ~/Escritorio/APPTec/Menu
Ingrese el nombre del usuario a borrar:prueba
¿Desea borrar el directorio de trabajo y todo su contenido [Y/N]?
Valor por defecto [n]:

#!/bin/bash
clear
echo
echo -n "Ingrese el nombre del usuario a borrar:"
read username
if [ "$username" = root ]; then
echo "No tiene los permisos necesarios "
else
echo
echo "¿Desea borrar el directorio de trabajo y todo su contenido [Y/N]?"
echo -n "Valor por defecto [n]: "
read delete_home
fi
if [ "$delete_home" = "Y" ]; then
sudo userdel -r $username
else
sudo userdel $username
fi
echo
```

Modificación usuario.

```
apptec@apptec-VirtualBox: ~/Escritorio/APPTec/Menu

1)Modificar datos personales
2)Modificar contraseña

Ingrese una opcion: 
```

```
#!/bin/bash
clear
while [[ $op != 0 ]]
do
echo
echo " |-----| "
echo " | 1)Modificar datos personales | "
echo " | 2)Modificar contraseña      | "
echo " |-----| "
echo
read -p "Ingrese una opciónS: " op
case $op in
1)echo
  echo -n "Nombre del usuario al que quiere cambiar su información personal: "
  read username
  echo
  sudo chfn $username
  echo;;
2)echo -n "Ingrese el nombre del usuario que quiere modificar la clave: "
  read username
  echo
  sudo passwd $username
  echo;;
esac
done
```

Alta grupo.

```
apptec@apptec-VirtualBox: ~/Escritorio/APPTec/Menu
Ingrese el grupo a crear: S

#!/bin/bash
clear
read -p "Ingrese el grupo a crear: " grp
sudo groupadd ${grp}
echo "grupo Creado satisfactorio"
sleep 2
```