

Integración con Contenedores

DOCKERFILE/EVIDENCIAS

SSH en el navegador

SUBIR ARCHIVO

DESCARGAR ARCHIVO

Info → In Use

```
[root@rocky-host-lfs lfs-docker]# ls -lh
total 1.3G
-rw-r--r--. 1 root root 55 Dec 7 15:40 Dockerfile
-rw-r--r--. 1 root root 1.3G Dec 7 15:31 lfs-rootfs.tar.gz
[root@rocky-host-lfs lfs-docker]# docker images

```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
lfs-image:latest	8dc7adb20d82	5.55GB	1.36GB	U

```
[root@rocky-host-lfs lfs-docker]# docker start -ai lfs-container
bash-5.3# ls /
bin  dev  home  lib64      media  proc  sbin      srv  tmp  var
boot  etc  lib  lost+found  opt    root  sources  sys  usr
bash-5.3# uname -a
Linux d00de4d47a66 6.12.0-124.13.1+2.1.el10_1_ciq.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Nov 27 15:
24:00 UTC 2025 x86_64 GNU/Linux
bash-5.3# exit
exit
[root@rocky-host-lfs lfs-docker]# cat /root/lfs-docker/Dockerfile
FROM scratch
ADD lfs-rootfs.tar.gz /
CMD ["/bin/bash"]
[root@rocky-host-lfs lfs-docker]#
```

Limitaciones del LFS dentro de un contenedor

El sistema LFS fue construido desde cero y contiene solamente las herramientas básicas necesarias para arrancar un sistema Linux. Cuando se utiliza dentro de un contenedor Docker/Podman, presenta ciertas limitaciones importantes:

1. No incluye servicios del sistema

LFS no trae:

- sysvinit

- networking automático
- manejadores de logs

Por lo tanto, el contenedor solo ejecuta el proceso indicado en el Dockerfile (en este caso, /bin/bash).

2. No incluye herramientas de red ni utilidades típicas

Comandos como:

- ip
- ping
- ifconfig
- ss

no funcionan a menos que el usuario los compile e instale manualmente.

3. No incluye gestor de paquetes

Debido a que LFS se construye desde código fuente, no existe:

- dnf
- apt
- yum
- pacman

Cualquier software adicional requiere compilarlo manualmente dentro del contenedor.

4. No tiene demonios ni servicios permanentes

Docker no puede iniciar servicios “del sistema” porque:

- No existe systemd
- No existe journald
- No existe init

Un contenedor LFS solo sirve como entorno minimalista o educativo, no como un sistema completo para producción.

5. El contenedor comparte el kernel del host

LFS fue diseñado para correr sobre su propio kernel, pero en Docker:

- No usa su kernel compilado
- Usa el kernel del host (virtualización ligera)

Esto demuestra la diferencia principal entre una VM (virtualización completa) y Docker (contenedores con kernel compartido).

Pruebas de servidor, systemd y PID 1

```
C:\Program Files\WindowsAp x + v
root@lfs:/# /srv/http-lfs/start_server.sh
=== Iniciando Servidor HTTP TP LFS ===

Recargando systemd...
Iniciando servicio lfs-http...

=== Estado del Servicio ===
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pre
  set: enabled)
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 1s ago
 Invocation: b8f8943084054b599ebfd1f910586f57
    Main PID: 394 (python3)
      Tasks: 1 (limit: 38449)
     Memory: 8.9M (peak: 8.9M)
        CPU: 81ms
    CGroup: /system.slice/lfs-http.service
            └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

=== Información de Conexión ===
🐦 Accede desde:
   Local: http://localhost:8081/
   Red:   http://<tu-ip>:8081/

📄 Comandos útiles:
Ver logs:    journalctl -u lfs-http -f
Ver estado:  systemctl status lfs-http
Parar:      systemctl stop lfs-http
root@lfs:/#
```

```
C:\Program Files\WindowsAp x + v

2. Estado del servicio lfs-http:
  ✓ ACTIVO
  Main PID: 394 (python3)
  Tasks: 1 (limit: 38449)
  Memory: 8.9M (peak: 8.9M)
  CPU: 85ms
  CGroup: /system.slice/lfs-http.service
          └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

3. Puerto 8081:
  ✓ EN ESCUCHA
  LISTEN 0 0 0.0.0.0:8081 0.0.0.0:* users:(("python3",
  pid=394,fd=3))

4. Prueba de conexión:
  ✓ Servidor responde (via Python)

5. Logs recientes:
dic 11 17:05:12 lfs systemd[1]: lfs-http.service: Deactivated successfully.
dic 11 17:05:12 lfs systemd[1]: Stopped Servidor HTTP TP LFS.
-- Boot d2e970741c4e4d31a532748f46b09d37 --
dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

6. Información de acceso:
  URL: http://:8081/
  URL local: http://localhost:8081/
root@lfs:/# python3 -c "import urllib.request; print(urllib.request.urlopen(
'http://localhost:8081/').read().decode()[:200])"
<html>
<head><title>LFS Systemd TP</title></head>
<body>
<h1>✓ LFS con Systemd funcionando!</h1>
<p><strong>PID 1:</strong> systemd</p>
<p><strong>Servicio:</strong> HTTP en puerto 8081</p>
<p><em>TP</em>
root@lfs:/#
```

```
C:\Program Files\WindowsAp x + v

Ver estado:    systemctl status lfs-http
Parar:        systemctl stop lfs-http
root@lfs:/# /srv/http-lfs/check_server.sh
=== Verificación Servidor HTTP TP LFS ===
Fecha: jue 11 dic 2025 23:09:28 -03

1. PID 1 del sistema:
  systemd
  ✓ Systemd es el PID 1

2. Estado del servicio lfs-http:
  ✓ ACTIVO
  Main PID: 394 (python3)
  Tasks: 1 (limit: 38449)
  Memory: 8.9M (peak: 8.9M)
  CPU: 85ms
  CGroup: /system.slice/lfs-http.service
          └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

3. Puerto 8081:
  ✓ EN ESCUCHA
  LISTEN 0 0 0.0.0.0:8081 0.0.0.0:* users:(("python3",
  pid=394,fd=3))

4. Prueba de conexión:
  ✓ Servidor responde (via Python)

5. Logs recientes:
dic 11 17:05:12 lfs systemd[1]: lfs-http.service: Deactivated successfully.
dic 11 17:05:12 lfs systemd[1]: Stopped Servidor HTTP TP LFS.
-- Boot d2e970741c4e4d31a532748f46b09d37 --
dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

6. Información de acceso:
  URL: http://:8081/
  URL local: http://localhost:8081/
root@lfs:/#
```

```
C:\Program Files\WindowsAp x + v
root@lfs:/# ps -p 1 -o pid,comm,args
PID COMMAND          COMMAND
 1 systemd          /sbin/init
root@lfs:/# systemctl status lfs-http
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pr>
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 2min 15s a>
  Invocation: b8f8943084054b599ebfd1f910586f57
    Main PID: 394 (python3)
      Tasks: 1 (limit: 38449)
     Memory: 8.9M (peak: 8.9M)
        CPU: 104ms
    CGroup: /system.slice/lfs-http.service
            └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.
... skipping ...
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pr>
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 2min 15s a>
  Invocation: b8f8943084054b599ebfd1f910586f57
    Main PID: 394 (python3)
      Tasks: 1 (limit: 38449)
     Memory: 8.9M (peak: 8.9M)
        CPU: 104ms
    CGroup: /system.slice/lfs-http.service
            └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.
~
~
~
~
~
~
~
```

```
root@lfs:/# journalctl -u lfs-http --since "5 minutes ago"
dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.
root@lfs:/# # Generar reporte automático
{
echo "≡ TP LFS - Systemd y Servicio HTTP ≡"
echo "Fecha: $(date)"
echo "Hostname: $(hostname)"
echo "PID 1: $(ps -p 1 -o comm=)"
echo "Estado servicio: $(systemctl is-active lfs-http)"
echo "Puerto: $(ss -tlnp | grep :8081 | awk '{print $4}')"
} > /tmp/tp-evidence.txt
root@lfs:/# cat /tmp/tp-evidence.txt
≡ TP LFS - Systemd y Servicio HTTP ≡
Fecha: jue 11 dic 2025 23:12:33 -03
Hostname: lfs
PID 1: systemd
Estado servicio: active
Puerto: 0.0.0.0:8081
root@lfs:/#
```

El conjunto de imágenes documenta de manera exhaustiva el inicio, la verificación y el funcionamiento de un servicio de servidor HTTP denominado lfs-http dentro de un entorno Linux. El proceso se inicia con la ejecución del script /srv/http-lfs/start_server.sh, lo que lleva a systemd a cargar e iniciar el servicio, el cual pasa inmediatamente al estado activo y en ejecución (active (running)) desde el jueves 11 de diciembre de 2025, a las 23:09:06. El servidor es identificado con el PID 394 y es ejecutado por Python 3 a través del script /srv/http-lfs/mini_server.py. Las comprobaciones posteriores, realizadas mediante un script

de verificación y el comando `systemctl status lfs-http`, confirman que el servicio está EN ESCUCHA en el puerto 8081 (0.0.0.0:8081), haciéndolo accesible tanto localmente como a través de la red, y una prueba de conexión directa utilizando código Python confirma que el Servidor responde correctamente con una página HTML que incluye el mensaje "LFS con Systemd funcionando!". Además, los logs recientes confirman que, si bien el servicio se había detenido previamente a las 17:05:12, su inicio a las 23:09:06 fue exitoso, y la generación de un reporte automático final consolida toda esta información, demostrando que el estado del servicio es active y que el puerto 8081 se encuentra correctamente en uso.