

Logical Volume Management (LVM) Disk Expansion and Persistent Mount Configuration

Overview

This documentation covers the process of adding a new virtual disk to a Linux system, configuring it using Logical Volume Management (LVM), formatting it with an ext4 filesystem, mounting it to a dedicated directory, and ensuring persistence across reboots. The workflow demonstrates standard enterprise practices for scalable storage management and permission control.

Implementation Summary

A new virtual disk was added at the hypervisor level and detected by the operating system.

The disk was initialized as an LVM physical volume, then grouped into a new volume group dedicated to log storage.

A logical volume was created from the volume group and formatted with the ext4 filesystem.

A mount point directory was created, and the logical volume was mounted successfully.

Persistent mounting was configured by adding the filesystem's UUID to `/etc/fstab`.

Mount validation confirmed the filesystem was active and correctly attached.

Ownership and permissions were configured to support application and group-based access, including setting the setgid bit to enforce group inheritance.

Final verification using block device inspection confirmed the correct disk, LVM, and mount layout.

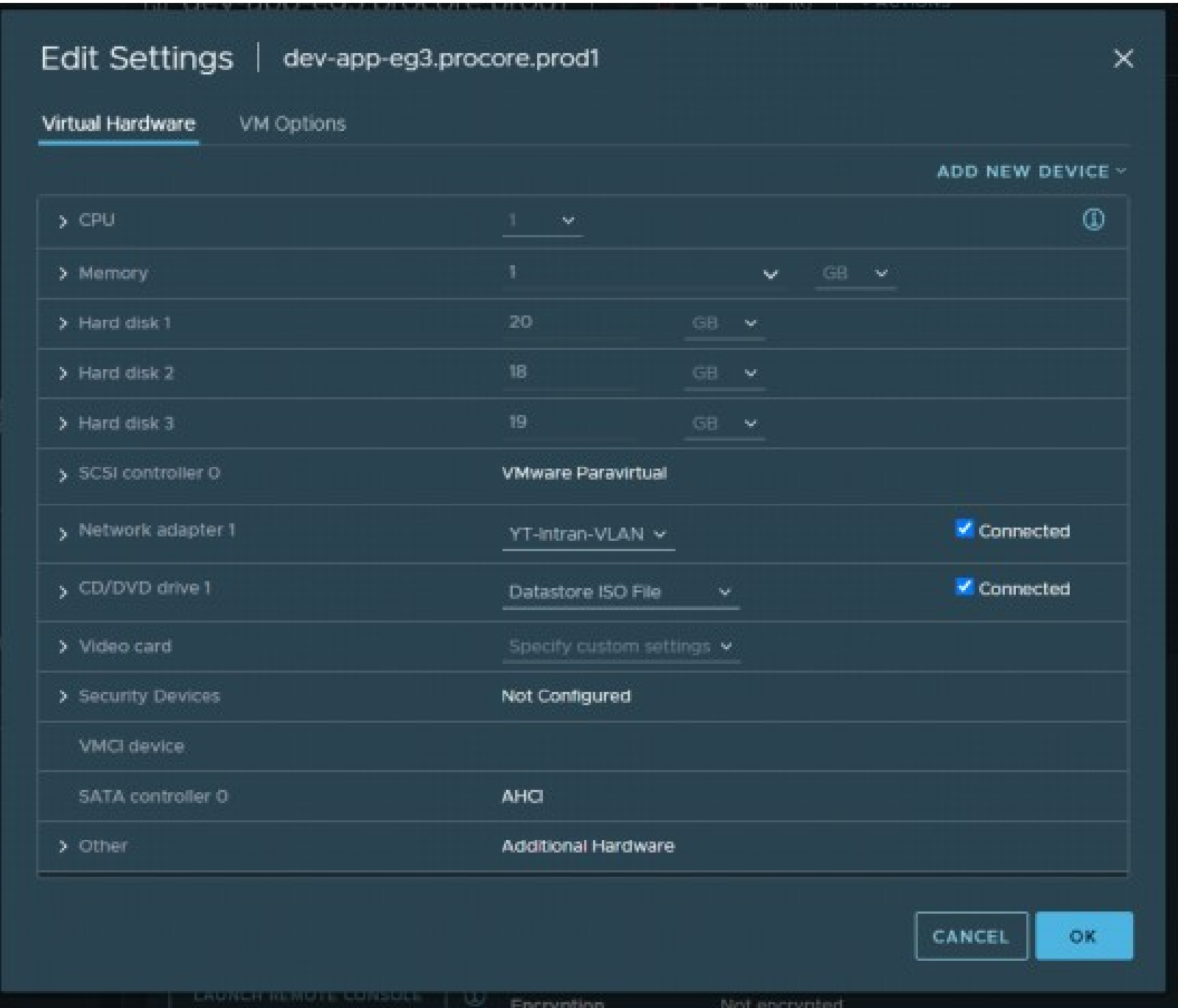
Result

The system now includes a dedicated, LVM-backed filesystem mounted at `/lfjs/logs`, providing isolated and manageable storage for log data. The configuration supports future scalability, consistent access control, and automatic mounting during system startup.

Sanitation Disclosure

All hostnames, usernames, volume names, UUIDs, IP addresses, and environment-specific identifiers shown in these screenshots have been sanitized or altered to protect system security and sensitive infrastructure details. The technical steps and outcomes remain accurate and representative of the real implementation.

The virtual machine's hardware configuration screen is open in VMware vSphere, showing the Edit Settings window for a VM named dev-app-eg3.procore.prod1. The Virtual Hardware tab is selected. The configuration lists 1 CPU, 1 GB of memory, and three virtual hard disks sized 20 GB, 18 GB, and 19 GB, all attached to a VMware Paravirtual SCSI controller. A single network adapter is connected to the YT-Intran-VLAN and is marked as connected. A CD/DVD drive is configured to use a Datastore ISO File and is currently connected. Additional entries include a video card, an unconfigured security device, a VMCI device, and a SATA controller set to AHCI. The window shows Cancel and OK buttons at the bottom, indicating changes can be applied or discarded.



The Edit Settings window for the virtual machine dev-app-eg3.procore.prod1 is open in VMware vSphere with the Virtual Hardware tab selected. The configuration lists 1 CPU, 1 GB of memory, and three existing virtual hard disks sized 20 GB, 18 GB, and 19 GB. A new virtual disk labeled New Hard disk is being added with a size of 1 GB, while the maximum supported size is shown as 6.26 TB.

The disk settings indicate the storage location is set to Store with the virtual machine, the provisioning type is Thin Provision, and sharing is Unspecified. The disk has Normal shares (1000), Unlimited IOPS, and is configured in Dependent disk mode. The virtual device node shows the disk attached to SCSI controller 0 at SCSI(3). The window includes Cancel and OK buttons at the bottom to discard or apply the configuration changes.

Edit Settings | dev-app-eg3.procore.prod1

Virtual Hardware

VM Options

ADD NEW DEVICE

> CPU	1				
> Memory	1		GB		
> Hard disk 1	20		GB		
> Hard disk 2	18		GB		
> Hard disk 3	19		GB		
> New Hard disk *	1		GB		
Maximum Size	6.26 TB				
VM storage policy					
Location	Store with the virtual machine				
Disk Provisioning	Thin Provision				
Sharing	Unspecified				
Shares	Normal	1000			
Limit - IOPs	Unlimited				
Disk Mode	Dependent				
Virtual Device Node	SCSI controller 0	SCSI(0:3) New Hard disk			

CANCEL

OK

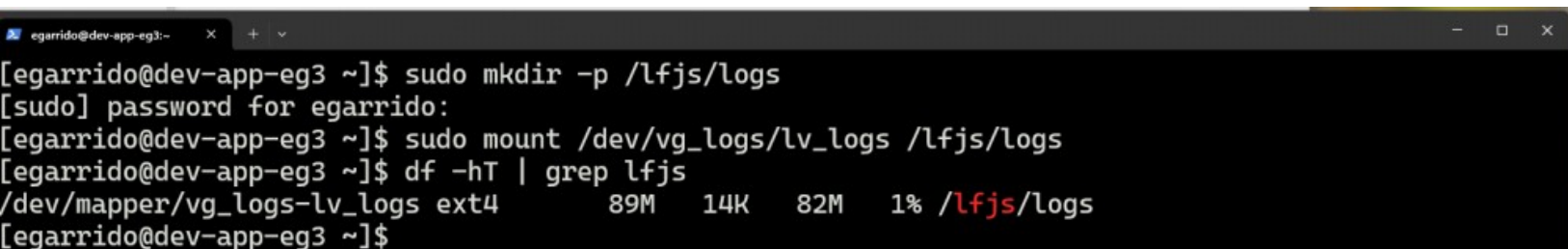
The terminal session shows a successful sequence of LVM setup and filesystem creation commands executed on the host dev-app-eg3 by the user egarrido. A physical volume is created on the disk /dev/sdb, followed by the creation of a volume group named vg_logs using that physical volume. A logical volume called lv_logs is then created with a size of 100 MB inside the volume group. The logical volume is formatted with the ext4 filesystem, during which inode tables, superblocks, and the filesystem journal are created without errors. The command output confirms that all steps completed successfully, returning the user to the shell prompt.

```
egarrido@dev-app-eg3:~$ sudo pvcreate /dev/sdb
[sudo] password for egarrido:
Physical volume "/dev/sdb" successfully created.
egarrido@dev-app-eg3:~$ sudo vgcreate vg_logs /dev/sdb
Volume group "vg_logs" successfully created
egarrido@dev-app-eg3:~$ sudo lvcreate -L 100M -n lv_logs vg_logs
Logical volume "lv_logs" created.
egarrido@dev-app-eg3:~$ sudo mkfs.ext4 /dev/vg_logs/lv_logs
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: bd5ac031-1f5e-40bd-b281-be1cca5747e6
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

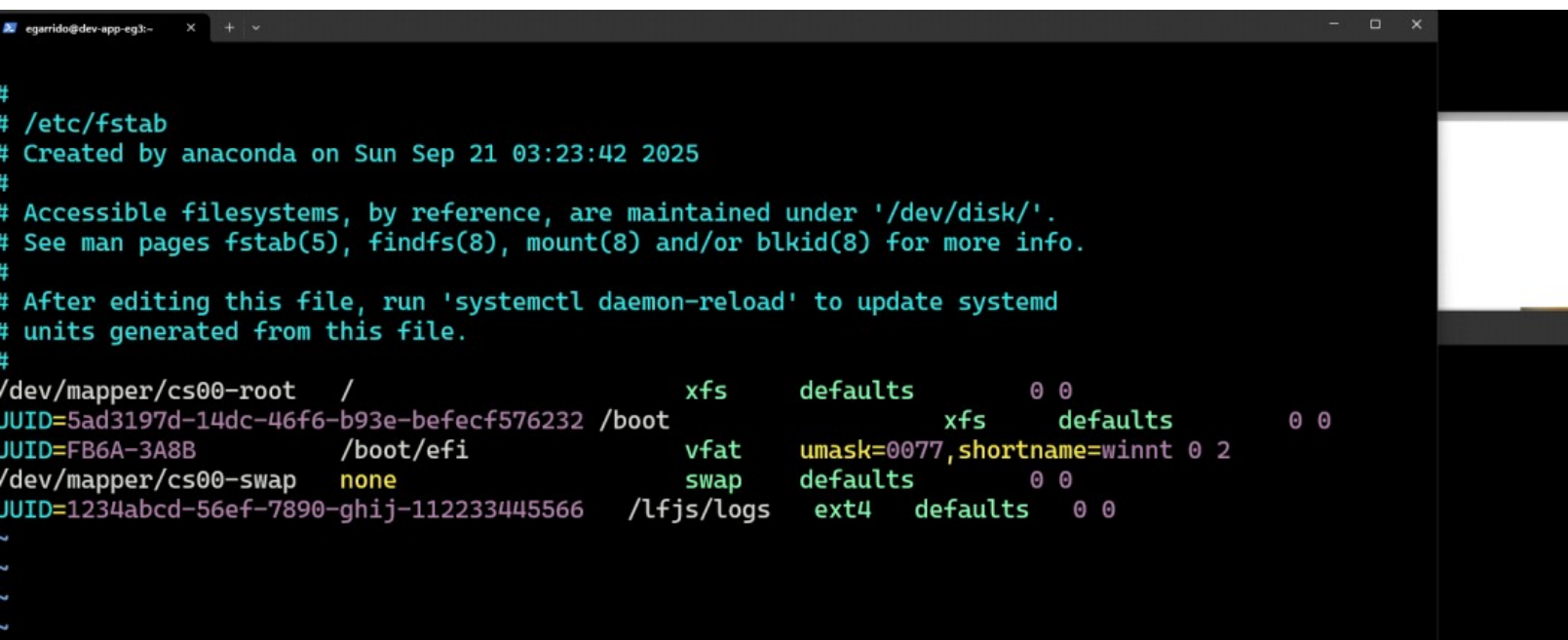
egarrido@dev-app-eg3:~$
```

The terminal output shows the creation and verification of a new mount point for the logical volume on the system dev-app-eg3. A directory named /lfjs/logs is created using a recursive mkdir command. The logical volume /dev/vg_logs/lv_logs is then mounted to this directory. A disk usage check using df -hT filtered for lfjs confirms that the ext4 filesystem is successfully mounted, showing approximately 89 MB in size with minimal usage and the mount point correctly listed as /lfjs/logs.

A terminal window with a dark background and light text. The window title is 'egarrido@dev-app-eg3:~'. The terminal shows a series of commands and their outputs. The first command is 'sudo mkdir -p /lfjs/logs', followed by a password prompt. The second command is 'sudo mount /dev/vg_logs/lv_logs /lfjs/logs'. The third command is 'df -hT | grep lfjs', which produces a table of disk usage information. The final prompt is '[egarrido@dev-app-eg3 ~]\$'.

```
[egarrido@dev-app-eg3 ~]$ sudo mkdir -p /lfjs/logs
[sudo] password for egarrido:
[egarrido@dev-app-eg3 ~]$ sudo mount /dev/vg_logs/lv_logs /lfjs/logs
[egarrido@dev-app-eg3 ~]$ df -hT | grep lfjs
/dev/mapper/vg_logs-lv_logs ext4      89M   14K   82M   1% /lfjs/logs
[egarrido@dev-app-eg3 ~]$
```

The screenshot shows the `/etc/fstab` file open in a text editor on the system `dev-app-eg3`. The file includes standard header comments explaining its purpose, references to filesystem documentation, and instructions to run `systemctl daemon-reload` after making changes. Existing entries define the root filesystem mounted at `/` using XFS, a `/boot` partition mounted with XFS, an EFI partition mounted at `/boot/efi` using `vfat`, and a swap device. A new entry has been added at the bottom specifying an `ext4` filesystem identified by its UUID, configured to mount persistently at `/lfjs/logs` with default mount options.

A terminal window with a dark background and light-colored text. The window title is 'egarrido@dev-app-eg3:~'. The content shows the /etc/fstab file with standard headers and five entries. The last entry is a new addition for an ext4 filesystem. The terminal output is as follows:

```
#  
# /etc/fstab  
# Created by anaconda on Sun Sep 21 03:23:42 2025  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd  
# units generated from this file.  
#  
/dev/mapper/cs00-root    /                xfs      defaults    0 0  
UUID=5ad3197d-14dc-46f6-b93e-befecf576232 /boot            xfs      defaults    0 0  
UUID=FB6A-3A8B          /boot/efi       vfat     umask=0077,shortname=winnt 0 2  
/dev/mapper/cs00-swap    none            swap     defaults    0 0  
UUID=1234abcd-56ef-7890-ghij-112233445566 /lfjs/logs      ext4     defaults    0 0
```

The terminal session captures verification and permission configuration steps for the newly mounted logical volume on dev-app-eg3. A mount -a command reports that the UUID listed in /etc/fstab cannot be found and advises reloading systemd because the file was modified, although a subsequent disk check confirms that /lfjs/logs is already mounted from /dev/mapper/vg_logs-lv_logs as an ext4 filesystem. An attempt to change ownership of the directory initially fails because the lfjs user does not yet exist. A non-privileged attempt to create the user is denied, followed by a successful user creation using sudo. Ownership of /lfjs/logs is then set to lfjs:webmasters, directory permissions are adjusted to 755, and the set-group-ID bit is applied so that new files created in the directory inherit the webmasters group.

```
egarrido@dev-app-eg3:~$ sudo mount -a
mount: /lfjs/logs: can't find UUID=1234abcd-56ef-7890-ghij-112233445566.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
egarrido@dev-app-eg3:~$ df -hT | grep lfjs
/dev/mapper/vg_logs-lv_logs ext4      89M   14K   82M   1% /lfjs/logs
egarrido@dev-app-eg3:~$ sudo chown lfjs:webmasters /lfjs/logs
chown: invalid user: 'lfjs:webmasters'
egarrido@dev-app-eg3:~$ useradd lfjs
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
egarrido@dev-app-eg3:~$ sudo useradd lfjs
egarrido@dev-app-eg3:~$ sudo chown lfjs:webmasters /lfjs/logs
egarrido@dev-app-eg3:~$ sudo chmod 755 /lfjs/logs
egarrido@dev-app-eg3:~$ sudo chmod g+s /lfjs/logs
egarrido@dev-app-eg3:~$
```


The terminal output displays the results of the `lsblk` command on the system `dev-app-eg3`, showing the block device and logical volume layout. The primary disk `sda` is 20 GB and contains multiple partitions used for the operating system, including logical volumes for the root filesystem and swap. A secondary disk `sdb` of 18 GB is dedicated to LVM and contains a logical volume `vg_logs-lv_logs` sized at 100 MB, which is mounted at `/lfjs/logs`. Another disk `sdc` of 19 GB holds the system's boot-related partitions, including `/boot`, `/boot/efi`, and associated logical volumes. An additional small disk `sdd` is present but unused, and a virtual optical device `sr0` is listed as a read-only ROM device. The output confirms that the new LVM-backed filesystem is correctly attached and mounted.

```
[egarrido@dev-app-eg3 ~]$ lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	20G	0	disk	
├─sda1	8:1	0	600M	0	part	
├─sda2	8:2	0	1G	0	part	
├─sda3	8:3	0	18.4G	0	part	
│ └─cs-swap	253:2	0	2G	0	lvm	
│ └─cs-root	253:3	0	16.4G	0	lvm	
sdb	8:16	0	18G	0	disk	
└─vg_logs-lv_logs	253:4	0	100M	0	lvm	/lfjs/logs
sdc	8:32	0	19G	0	disk	
├─sdc1	8:33	0	600M	0	part	/boot/efi
├─sdc2	8:34	0	1G	0	part	/boot
├─sdc3	8:35	0	17.4G	0	part	
│ └─cs00-root	253:0	0	15.5G	0	lvm	/
│ └─cs00-swap	253:1	0	1.9G	0	lvm	[SWAP]
sdd	8:48	0	1G	0	disk	
sr0	11:0	1	11.9G	0	rom	

```
[egarrido@dev-app-eg3 ~]$
```


Summary

A new virtual disk was added to the system and configured using Logical Volume Management to provide dedicated storage for log data. The disk was initialized as a physical volume, grouped into a volume group, and carved into a logical volume that was formatted with the ext4 filesystem. The filesystem was mounted to `/lfjs/logs` and configured for persistence using `/etc/fstab`. Ownership and permissions were applied to support group-based access and consistent file creation. All screenshots have been sanitized to remove or obscure hostnames, UUIDs, and other sensitive environment-specific details while preserving the technical accuracy of the workflow.