CHAPTER-15 ACCESS LINUX FILE SYSTEMS

Identify File Systems and Devices

Storage Management Concepts

Red Hat Enterprise Linux **(RHEL) uses** the **Extents File System (XFS)** as the default local file system. **RHEL supports** the **Extended File System (ext4) file system** for managing local files. Starting with **RHEL 9, the exFAT file system is supported for removable media use**. In an **enterprise server cluster, shared disks use** the **Global File System 2 (GFS2) file system** to manage concurrent multi-node access.

File Systems and Mount Points

Access the contents of a file system by mounting it on an empty directory. This directory is called amount point. When mounted, use the ls command to list the contents of that directory. **Many file systems are automatically mounted** when the system boots.
A mount point is a slightly different concept than a Microsoft Windows drive letter, where each file system is a separate entity. **Mount points allow multiple file system devices** to be available in a single tree structure. This is similar to NTFS mounted folders in Microsoft Windows.

File Systems, Storage, and Block Devices

A block device is a file that provides low-level access to storage devices. A block device must be optionally partitioned, and a file system created before the device can be mounted. **The /dev directory stores block device files**, which RHEL creates automatically for all devices. In RHEL 9, the first detected SATA, SAS, SCSI, or USB hard drive is called the /dev/sda device; the second is the /dev/sdb device; and so on. These names represent the entire hard drive.

Block Device Naming

Type of device	**Device naming pattern**
SATA/SAS/USB-attached storage (SCSI driver)	/dev/sda, /dev/sdb, /dev/sdc, …
virtio-blk paravirtualized storage (VMs)	/dev/vda, /dev/vdb, /dev/vdc,…
virtio-scsi paravirtualized storage (VMs)	/dev/sda, /dev/sdb, /dev/sdc, …
NVMe-attached storage (SSDs)	/dev/nvme0, /dev/nvme1, …
SD/MMC/eMMC storage (SD cards)	/dev/mmcblk0, /dev/mmcblk1, …

Disk Partitions

Usually, the entire storage device is not created into one file system. To create a partition, divide **the storage devices into smaller chunks**.

Partitions are **block devices** in their own right. For example, on **the first SATA-attached** storage, the **first partition is the /dev/sda1 disk**. **The second partition** of the same storage is **the /dev/sda2 disk**. **The third partition on** the **third SATA-attached storage device** is the **/dev/sdc3 disk**, and so on. Paravirtualized storage devices have a similar naming system. For example, **the first partition on the first storage device is the /dev/vda1 disk**. The **second partition** of the second storage device is **the /dev/vdb2 disk, and so on.**

An **NVMe-attached SSD device names its partitions** differently from a SATA-attached device. For NVMe storage devices, **the nvmeX part** of the name refers to the device**, the nY part refers to the namespace**, and **the pZ part refers to the partition**. For example, the **first partition for the first namespace on the first disk is the /dev/nvme0n1p1 partition**. The **third partition** for the **first namespace on the second disk is the /dev/nvme1n1p3 partition**, and so on.

SD or MMC cards can sometimes have **a similar naming system to the SATA devices (/dev/sdN),** but it is not always the case. In some cases, SD or MMC cards might have names such as /dev/ mmcblk0p1, where **the mmcblkX part of the name refers to the storage device** and the pY part of the name refers to the partition number on that device.

Logical Volumes

Another way of **organizing disks and partitions** is with **Logical Volume Management (LVM**). With LVM, it is possible to aggregate block devices into a volume group. Disk space in the volume group is separated into logical volumes, which are the functional equivalent of a partition on a physical disk.

The **LVM system assigns names** to **volume groups and logical volumes** on their creation. **LVM creates a directory in the /dev directory that matches the group name, and creates a symbolic link within that new directory with the same name as the logical volume**. That logical volume file is then available to be mounted.

For example, when a myvg volume group and the **mylv logical volume** are present, the full path to the logical volume is the **/dev/myvg/mylv file.**

Examine File Systems

Use the df command to display an overview of local and remote file-system devices, which includes the total disk space, used disk space, free disk space, and the percentage of the entire disk space.

DESCRIPTION	COMMANDS / OPTIONS
To display an overview of local and remote file-system devices	**Syntax:** df [options][directory/file] **-h** human readable format **-H** human readable in SI format **-i** inode **-a** all **-l** Limits listing to local file systems **-T** Prints file system type
To analyze and report on disk usage within directories and files	**Syntax:** du [options] [directory/file] **-a** all information **-h** human readable format **-H** human readable in SI format
To list the details of a specified block device	**Syntax:** lsblk [OPTIONS] [DEVICE...] **-a** or --all **-b** or --bytes **-i** or --inverse **-l** or --list **-fp** lists the full path of the device

Mount and Unmount File Systems

Mount File Systems Manually

To **access the file system** on a removable storage device, **you must mount the storage** device. With the **mount command**, **the root user can mount a file system manually**. The first argument of the mount command specifies the file system to mount. The second argument specifies the directory as the mount point in the file-system hierarchy.

You can mount the file system in one of the following ways with the mount command:

- With the device file name in the /dev directory.
- With the UUID, a universally unique identifier of the device.

DESCRIPTION	COMMANDS / OPTIONS
To mount a file system.	**Syntax:** mount[options]<device><mountPoint> **L** Lists all the file systems mounted yet. **h** Displays options for command. **V** Displays the version information. **a** Mounts all devices described at /etc/fstab. **T** Type of filesystem device uses. **R** Read-only mode mounted.
To unmount a file system.	**Syntax:** umount [/MOUNTPoint]

Locate Files on the System

Search for Files

A system administrator needs tools to search for files that match specific criteria on the file system. This section discusses two commands to search for files in the file-system hierarchy:

- The locate command searches a pre-generated index for file names or file paths and returns the results instantly.
- The find command searches for files in real time by parsing the file-system hierarchy.

DESCRIPTION	COMMANDS / OPTIONS
To find the files by name using database	**Syntax:** locate [OPTION]... PATTERN... **-b** –basename **-c** –count **-d** –database DBPAT **-e** –existing **-L** –follow **-I** –ignore-case **Example**: locate -i messages **-l** –limit, -n limits the number of returned search results **Example:** locate -n 5 messages **-m** –mmap **-P** –nofollow, -H **-S** –statistics **-0** –null
To searching in real time in the file-system hierarchy	**Syntax:** find [path] [options] [expression] -name -iname -user -group -uid -gid -perm **Example:** find /etc -type f perm 764 find /etc -type f perm u=rwx,g=rw,o=r -size +(size) **Example:** find /etc -size -10k -type -size -(size) **Sub Command with Find** find [path] [options] [expression] To find specific file and rmove it **Example:** find ./GFG -name sample.txt -exec rm -i {} \;

Used for searching and manipulating text patterns within files **GREP**	**Syntax:** grep [options] pattern [files] **-c** This prints only a count of the lines that match a pattern **-h** Display the matched lines, but do not display the filenames. **-i** Ignores, case for matching **Example: grep -i "UNix" geekfile.txt** **-l** Displays list of a filenames only. **-n** Display the matched lines and their line numbers. **Example: grep -n "unix" geekfile.txt** **-v** This prints out all the lines that do not matches the pattern **-e** exp Specifies expression with this option. Can use multiple times. **-f** file Takes patterns from file, one per line. **-E** Treats pattern as an extended regular expression (ERE) **-w** Match whole word **-o** Print only the matched parts of a matching line, with each such part on a separate output line. **-A n** Prints searched line and n lines after the result. **-B n** Prints searched line and n line before the result. **-C n** Prints searched line and n lines after before the result.
To List Open Files in Linux	**Syntax:** lsof [option] **-c** <process_name> **-u** <username> **-l** Show network-related information **-p** <pid> List files for a specific process ID **-t** Display only the process IDs (PIDs) rather than full details.