ESSENTIALS OF RED HAT LINUX

Session 8

Storage

Objectives

- Explain the different storage considerations
- Explain the concept of volumes

Storage Considerations [1-3]

- RHEL supports a large number of file systems.
- Some of the key file systems are:

Second Extended File System (ext2)
Third Extended File System (ext3)
Fourth Extended File System (ext4)
XFS
Swap
New Technology File System (NTFS)
Virtual File Allocation Table (VFAT)
Global File System (GFS)
Global File System 2 (GFS2)

Storage Considerations [2-3]

- The supports for a variety of file system are:

  - Backward compatibility
  - Interoperability with other operating systems
  - Expandability of RHEL operating system
  - Large file systems
  - Large supported file size

- Each type of file system is capable of storing files and folders, known as directories.
  - The root directory is the master directory that is the root of the file system.
  - All other directories are known as the subdirectories.

Storage Considerations [3-3]

- Some of the key file systems that it supports are listed in the table:

Feature	Ext2	Ext3	Ext4	XFS
Max Supported File Size	2 TB	2 TB	16 TB	16 TB
Max Supported Size	8 TB	16 TB	16 TB	100 TB
Maximum Number of Subdirectories	32,000	32,000	65,000	65,000
Maximum Depth of Symbolic Links	8	8	8	8

Supported File Systems

- RHEL supports some other file systems, such as:

MS-DOS - Used for floppy disks.

ISO 9660 - Used for CDs.

GFS and GFS2 - Used for SANs.

File System Structure [1-2]

- File system structure: Responsible for storing files and folders.
- An operating system utilizes these files and folders to provide relevant information to users and applications.
- It depends on the file system to decide how a user or an application interacts with the files and folders.
- A common structure is responsible for defining the permissions, names, and locations for files and folders.
- File system can divide files into different categories:

Shareable

Unshareable

Static

Variable

File System Structure [2-2]

- A common structure of a file system allows a controlled access to the files and folders.
- Controlled access is defined at the operating system level.
- Permissions defined on the files and folders are tightly integrated with the operating system.
- The methods that an operating system uses are:

Files are stored in the file system.

Permissions are defined by the operating system.

Permissions are assigned to a user or an application.

User accesses the files and folders based on the permissions defined.

Overview of File System Hierarchy Standard [1-3]

- The default file system structure for RHEL is FHS.
- Defines guidelines for the structure of files and directories placement.
- Defines the standardization of files and directories across all UNIX-like operating systems.

The standardization guidelines define certain parameters for the files and directories that are:

- Names
- Locations
- Permissions

The standardization guidelines help greater system interoperability, such as:

- Application designers
- System integrators

Overview of File System Hierarchy Standard [2-3]

The FHS compliance has two key elements:

- The operating system should be compatible with other FHS-compliant operating systems.
- The operating system should have the capability to mount /usr/ read-only partition.

The FHS-compliant file systems have a few key directories:

- /**boot**/: Contains the critical files that are required to boot the system.
- /**dev**/: Contains the device nodes that hold information on the connected devices and virtual devices.
- /**etc**/: Contains the configuration files.

Overview of File System Hierarchy Standard [3-3]

- Other than the mentioned directories, a few more key directories are:

/lib/
/media/
/mnt/
/opt/
/proc/
/sbin/
/srv/
/sys/
/usr/
/var/

Creating an ext3 File System [1-3]

- RHEL 6.0 uses ext4 as the default file system.
- In RHEL 5.0, ext3 was the default file system.
- Define the file system during the installation of an operating system.
- When a new disk has been added or free space of an existing disk is to be used:

Create a partition.

Format it with the ext3 file system.

- Users use a partition that is formatted with the ext2 file system.
- To upgrade to ext3 file system, use the `tune2fs` command.

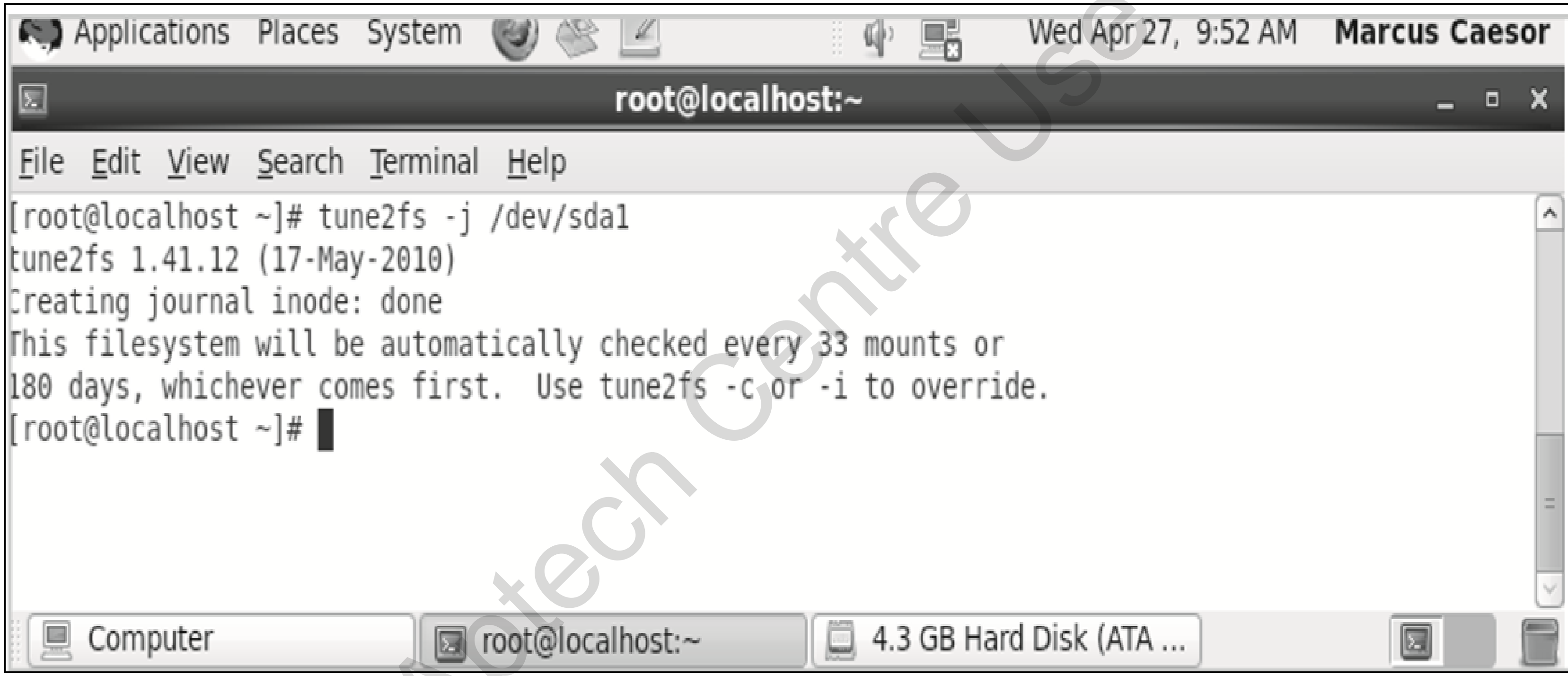
Creating an ext3 File System [2-3]

- The `tune2fs` command has a number of parameters:

`-l`	`-c max-mount-counts`	`-e errors-behavior`	`-f`
`-i interval-between-checks`	`-j`	`-J journal-options`	`-m reserved-blocks-percentage`
`-o [^]mount-options[,...] [ -r reserved-blocks-count]`	`-s sparse-super-flag`	`-u user`	`-g group`
`-C mount-count`	`-L volume-name`	`-M last-mounted-directory`	`-O [^]feature[,...]`
`-T time-last-checked`	`-U UUI`		

Creating an ext3 File System [3-3]

- After a user executes the `tune2fs` command, the user gets the output as shown in the following figure:

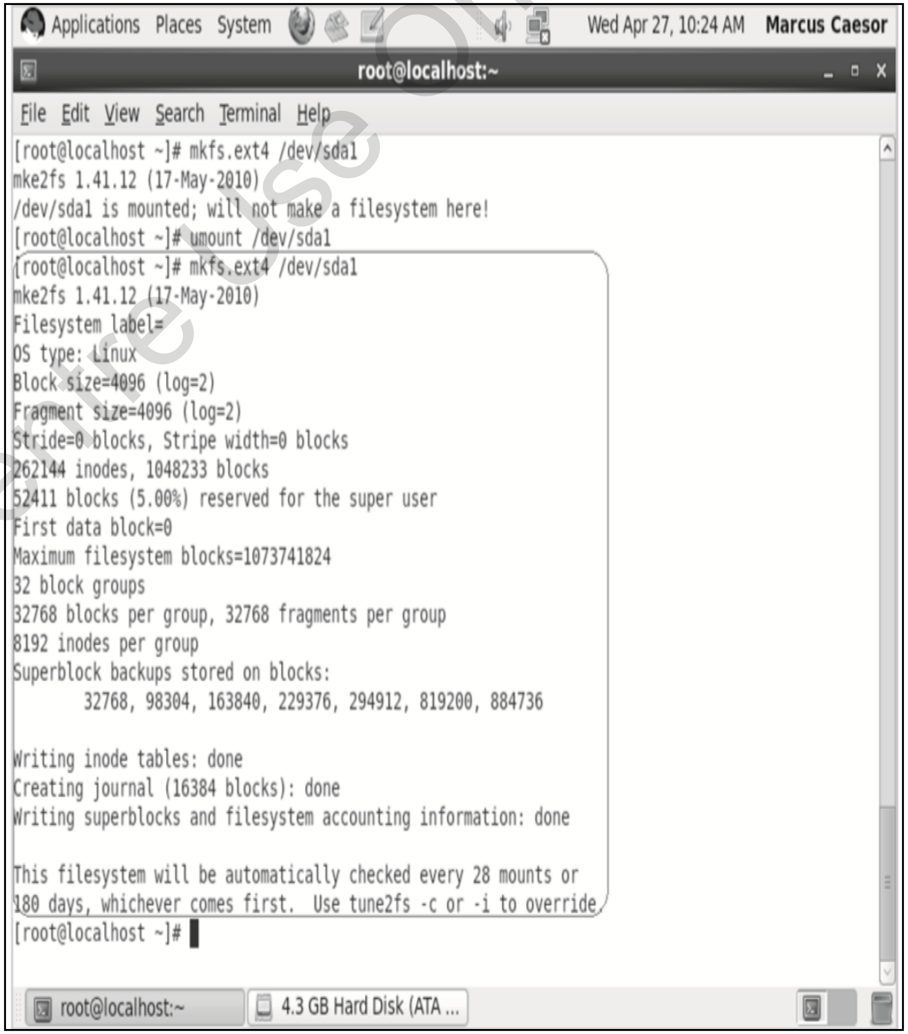


Output of `tune2fs` Command

- After the output is shown, the users can go back to the /etc/fstab file and change the file system from ext2 to ext3.

Creating an ext4 File System

- Default file system for RHEL 6.0.
- Two key advantages are:
  - Increased file system size.
  - Support for larger file size up to 16 TB.
- Uses extents instead of the block mapping scheme.
- Help to reduce metadata overhead when dealing with large files.
- To convert an existing ext3 file system to ext4 file system, execute the command `mkfs.ext4 /dev/device`



Output of `msfs.ext4` Command

Volumes

- Volumes: Defined on the physical hard disks that hold the RHEL operating system.
- Divided into partitions that can be of different.
- Created on a single physical hard disk or multiple hard disks and can be combined to create a single volume.
- A user can configure a disk either as a basic disk or system disk.

When the user install RHEL on an x86 or x64 computer, three partitions are created by the default partitioning layout:

- **root partition(/)**: The basic disk, can be defined on the primary or logical partitions.
- **Swap partition**: Works in coordination with the memory available to the operating system.
- **/boot partition**: Referred for necessary files.

Logical Volumes [1-5]

- Logical volumes are based on physical volumes.

Allow users to use a single or multiple physical storage devices using an abstraction layer.

  - This layer hides the physical storage
  - Allows the users and applications to view only the logical volumes

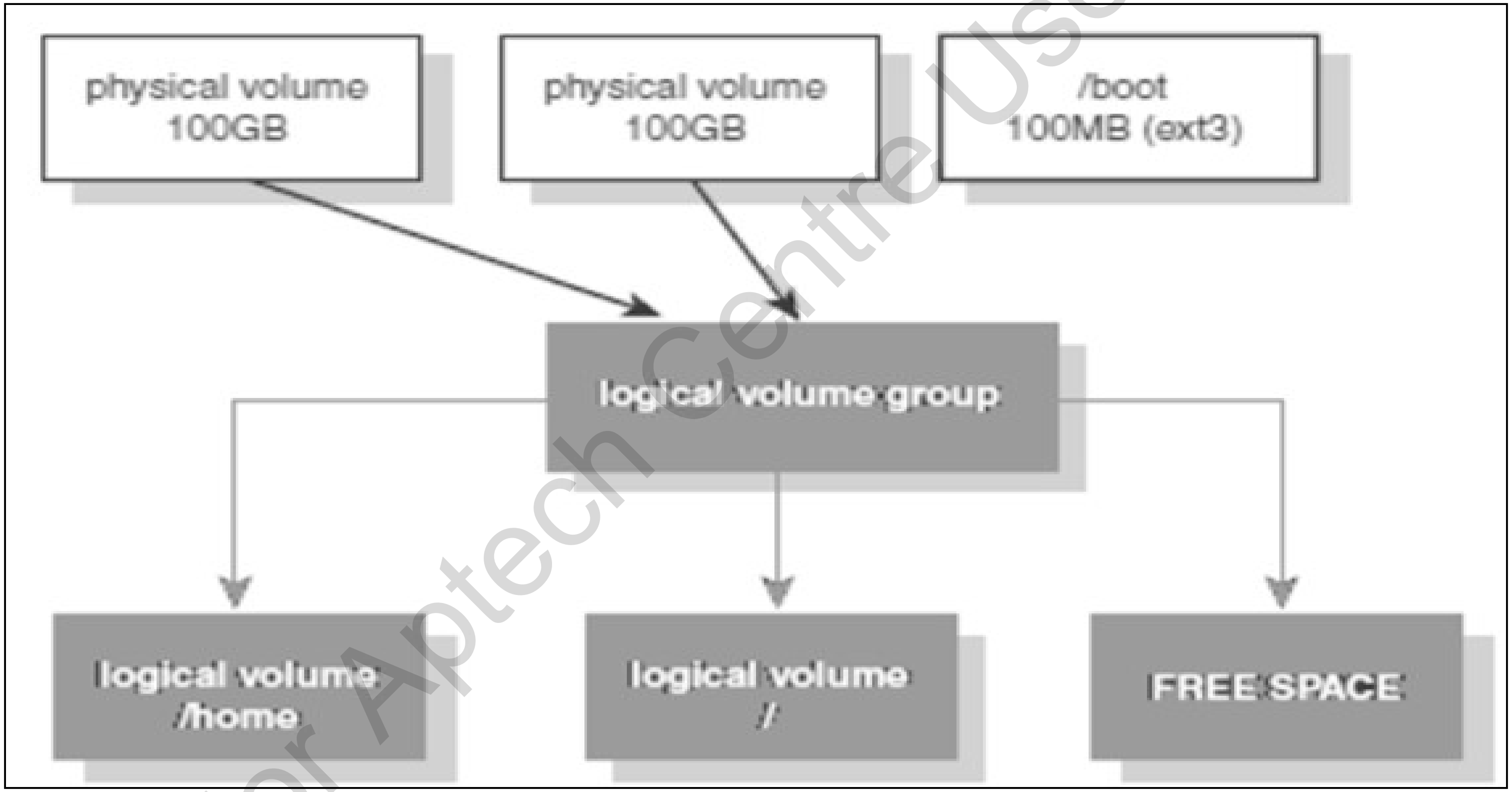
- Create a logical volume group that combines the space and applications or see combined space from all two disks.
- Provides flexibility to the administrator to manage a single logical volume group.

The logical volume group can be split into multiple smaller logical volumes for ease of use.

  - Logical volumes can be used to create file system and mount points.

Logical Volumes [2-5]

- Following figure displays a logical volume group that consists of physical and logical volumes:



Logical Volume Group

Logical Volumes [3-5]

The logical volumes have a number of benefits

- Extension Across Disks - Logical volumes can span across multiple physical disks, that otherwise can be used individually.
- Resizing - A user can resize the logical volumes without formatting the disks, allows flexibility in terms of quickly resizing the logical volumes with simple commands.
- Backups - A user can use volume snapshots for backups, helps in quick efficient backups without taking off the volume for backups.
- RAID - A user can create disk striping or mirrored volumes to increase throughput or redundancy.

Use LVM to manage logical volume groups

- Used during the installation of RHEL to handle all partitions and mount points, except for **/boot** partition.
- The **/boot** partition is on a separate partition that cannot be a logical volume.

Logical Volumes [4-5]

- LVM tools to perform management of physical and logical volumes:

  - pvcreate
  - vgcreate
  - vgextend
  - vgreduce
  - lvcreate
  - lvextend
  - lvremove
  - vgdisplay
  - lvdisplay
  - pvscan

Logical Volumes [5-5]

- Using these tools, LVM can allow a user to perform a number of tasks that are:

Create physical and logical volumes

Add or remove physical volume

Create, remove, or extend a logical volume

Show properties of logical or physical volumes

Manage File System Attributes and Swap Space [1-2]

- The file system attributes define the type of:
  - Partition
  - The layout
  - The storage process
- Using the file system attributes, a partition is defined in a disk drive.
- When a partition is created, its dimensions on the disk drive are defined.

There are three key attributes that define a partition on a file system:

- Partition Geometry
- Partition Type
- Partition Type Field

Manage File System Attributes and Swap Space [2-2]

Partition Geometry

- Defines the physical placement of a disk drive.
- Partition geometry is defined by four key factors:
  - Starting cylinders
  - Ending cylinders
  - Heads
  - Sectors

Partition Type

- A disk can primarily be formatted with three different kinds of partitions:
  - Primary - Maximum of four primary partitions on a disk.
  - Extended - Created when a user requires more than four primary partitions.
  - Logical - An extended partition is usually divided into logical partitions.

Partition Type Field

- Defines how the data is stored.

Swap Space [1-2]

- Serves as an additional memory.
- The key role is to manage resources when the physical memory or RAM is full.
- In this case, serves as an alternate to RAM and handles all inactive pages that were blocking space in RAM.
- When a user uses automatic partition during the installation of RHEL, a swap partition is created.
- Before creating a swap space:

Create a swap file. → Convert the swap file to swap space.

- Swap file: A file that is created on the file system to act as additional memory for the operating system.

Swap Space [2-2]

- Defined as double the space to the RAM available in the system.
- Double amount of swap space is applicable if a system has less than 2 GB of RAM.
- If a system has more than 2 GB of RAM, define the same amount of swap space equal to the RAM available in the system.
- In RHEL, the user must follow a number of steps:

1
- Create swap space, which is usually a swap partition.

2
- Use mkswap to write special signature.

3
- Add entries to the **/etc/fstab** file.

4
- Activate the swap partition using the `swapon` -a command.

Summary

- A file system structure is responsible for storing files and folders. Users require super user or root permissions to create a file system. ext4 is the default file system based on File System Hierarchy Standard (FHS).
- Two key advantages of ext4 file systems are increased file system size and support for larger file size up to 16 TB.
- An existing ext3 file system can be converted to ext4 file system.
- Swap space is defined as double the space to the RAM available in the system.
- It is an alternate mechanism of storing cache that cannot be handled by RAM.