CHAPTER-13 ARCHIVE AND TRANSFER FILES

Manage Compressed tar Archives

Create Archives from the Command Line

An archive is a single regular file or device file that contains multiple files. The device file could be a tape drive, flash drive, or other removable media. When using a regular file, archiving is analogous to the zip utility and similar variations that are popular on most operating systems.

DESCRIPTION	COMMANDS / OPTIONS
tar Utility	**Syntax:** tar[options]< file.tar><source> **-c** create an archive file **-t** list the contents of an archive **Example:** tar -tf /root/etc.tar ↵ **-x** Extract an archive. **Example:** tar -xf /root/etc.tar ↵ **-v** Show the files being archived extracted during the tar operation. **-f** Follow this option with the archive file name to create or open. **Example:** tar -cf archive.tar file1 file2 file3 ↵ **-p** Preserve the original file permissions when extracting. **--xattrs** Enable extended attribute support **--selinux** Enable SELinux context support **-a** auto-compress Use the archive's suffix to determine the algorithm to use **-z** Use the gzip compression algorithm, resulting in a .tar.gz suffix. **-j** Use the bzip2 compression algorithm, resulting in a .tar.bz2 suffix. **-J** Use the xz compression algorithm

To Create a Compressed Archive	gzip compression is the legacy, fastest method **Syntax:** gzip [options] file.tar.gz **-l** view the uncompressed size of a compressed file **Example:** tar -czf /root/etcbackup.tar.gz /etc **bzip2** compression creates smaller **archives** bzip2 [OPTIONS] filenames **xz** compression is newer, and offers the best compression ratio of the available methods.
To transferring files using secure copy	**Syntax:** scp [options] [[user@]host1:]source_file_or_directory [[user@]host2:]destination **-P** port: Specifies the port to connect on the remote host. **-p** Preserves modification times, access times, and modes from the original file. **-q** Disables the progress meter. **-r** Recursively copy entire directories. **-s** Name of program to use for the encrypted connection. The program must understand ssh(1) options.

Transfer Remote Files with the Secure File Transfer Program

The **OpenSSH** suite is useful for securely running shell commands on remote systems. Use the **Secure File Transfer Program (SFTP)** to interactively upload to or download files from an SSH server. This program is part of the OpenSSH suite. A session with the sftp command uses the secure authentication mechanism and encrypted data transfer to and from the SSH server.

Specify a remote location for the source or destination of the files to copy. For the format of the remote location, **use [user@]host:/path.** The user@ portion of the argument is optional. If this portion is missing, then the sftp command uses your current local username. When you run the sftp command, your terminal provides an sftp> prompt.

DESCRIPTION	COMMANDS / OPTIONS
Secure File Transfer Program (SFTP) to interactively upload to or download files from an SSH server	**Syntax:** sftp [remoteUser]@[remoteHost] **-b** batchfile **-B** buffer_size **-P** port **-r** To copy a whole directory tree recursively **To download the file from the remote host to the current directory on the local system** **get** remote_file [local_path] put local_file [remote_path] **To send file or directory to remote host** sftp> **put** -r directory **To get a remote file with the sftp command on a single command line**, without opening an interactive session, use the following syntax. You cannot use single command line syntax to put files on a remote host. **[user@host ~]$** sftp remoteuser@remotehost:/home/remoteuser/remotefile

Synchronize Files between Systems Securely

Synchronize Remote Files and Directories

The rsync command is another way to copy files from one system to another system securely. The tool uses an algorithm that minimizes the amount of copied data by synchronizing only the changed portions of files. If two files or directories are similar between two servers, then the rsync command copies only the differences between the file systems.

DESCRIPTION	**COMMANDS / OPTIONS**
To copy files from one system to another system securely.	**Syntax:** rsync [options] source [destination] **-r** recursive Synchronize the whole directory tree recursively **-l** links Synchronize symbolic links **-p** perms Preserve permissions **-t** times Preserve time stamps **-g** group Preserve group ownership **-o** owner Preserve the owner of the files **-D** devices Preserve device files