ARCHIVING FILES

Objectives:

1. Archive files and directories into a compressed file using **tar** and extract the contents of an existing tar archive.

   .

1. MANAGING COMPRESSED TAR ARCHIVES

The **tar** Command

Archiving and compressing files are useful when creating backups and transferring data across a network. One of the oldest and most common commands for creating and working with backup archives is the **tar** command.

With **tar**, users can gather large sets of files into a single file (archive). A **tar** archive is a structured sequence of file data mixed in with metadata about each file and an index so that individual files can be extracted. The archive can be compressed using *gzip, bzip2, or xz* compression.

The **tar** command can list the contents of archives or extract their files to the current system.

Selected **tar** Options

tar command options are divided into operations (the action you want to take): general options and compression options. The table below shows common options, long version of options, and their description:

Overview of tar operations

Option	Description
-c, --create	Create a new archive
-x, --extract	Extract from an existing archive
-t, --list	List the table of contents of an archive

Selected tar General Options

Option	Description
-v, --verbose	Vebose. Shows which files get archived or extracted
-f, --file	File name. This option must be followed by the file name of the archive to use or create
-p,--preserve-permissions	Preserve the permissions of files and directories when extracting an archive, without subtracting the umask.

Overview of tar Compression Options

Option	Description
-z, --gzip	Use gzip compression (**.tar.gz)**
-j, --bzip2	Use bzip2 compression (**.tar.bz2). bzip2** typically achieves a better compression ratio than **gzip**
-J, --xz	Use xz compression (**.tar.xz).** The **xz** compression typically achieves a better compression ration than **bzip2**

Listing Options of the **tar** command

The tar command expects one of the three following options:

- Use the -c or --create option to create an archive.
- Use the -t or --list option to list the contents of an archive.
- Use the -x or --extract option to extract an archive.

Other commonly used options are:

- Use the -f or --file= option with a file name as an argument of the archive to operate.
- Use the -v or --verbose option for verbosity; useful to see which files get added to or extracted from the archive.

Archiving Files and Directories

The first option to use when creating a new archive is the **c** option, followed by the **f** option, then a single space, then the file name of the archive to be created, and finally the list of files and directories that should get added to the archive. The archive is created in the current directory unless specified otherwise.

The following command creates an archive named archive.tar with the contents of file1, file2, and file3 in the user's home directory.

[lokesh@localhost ~]$ tar -cf archive.tar file1 file2 file3

[lokesh@localhost ~]$ ls archieve.tar

archieve.tar

The above tar command can also be executed using the long version options.

[lokesh@localhost ~]$ tar --file=archive.tar --create file1 file2 file3

For tar to be able to archive the selected files, it is mandatory that the user executing the **tar** command can read the files.

For example, creating a new archive of the **/etc** folder and all of its content requires root privileges, because only the root user is allowed to read all of the files present in the **/etc** directory. An unprivileged user can create an archive of the **/etc** directory, but the archive omits files which do not include read permission for the user, and it omits directories which do not include both read and execute permission for the user.

To create the tar archive named, /root/etc.tar, with the /etc directory as content as user root:

[root@localhost ~]# tar -cf /root/etc.tar /etc

tar: Removing leading `/' from member names

[root@localhost ~]#

Archiving Files and Directories

The **t** option directs **tar** to list the contents (table of contents, hence **t**) of the archive. Use the **f** option with the name of the archive to be queried.

For example:

[root@localhost ~]# tar -tf /root/etc.tar

etc/

etc/fstab

etc/crypttab

etc/mtab

...output omitted...

EXTRACTING FILES FROM AN ARCHIVE

A tar archive should usually be extracted in an empty directory to ensure it does not overwrite any existing files. When root extracts an archive, the tar command preserves the original user and group ownership of the files. If a regular user extracts files using tar, the file ownership belongs to the user extracting the files from the archive.

To restore files from the /root/etc.tar archive to the /root/etcbackup directory, run:

```
[root@localhost ~]# mkdir /root/etcbackup
[root@localhost ~]# cd /root/etcbackup
[root@localhost etcbackup]# tar -tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
etc/mtab
...output omitted...
[root@localhost etcbackup]# tar -xf /root/etc.tar
```

By default, when files get extracted from an archive, the umask is subtracted from the permissions of archive content. To preserve the permissions of an archived file, the p option when extracting an archive.

In this example, an archive named, /root/myscripts.tar, is extracted in the /root/scripts directory while preserving the permissions of the extracted files:

```
[root@localhost ~]# mkdir /root/scripts
[root@localhost ~]# cd /root/scripts
[root@localhost scripts]# tar -xpf /root/myscripts.tar
```

CREATING A COMPRESSED ARCHIEVE

The **tar** command supports three compression methods. There are three different compression methods supported by the **tar** command. The **gzip** compression is the fastest and oldest one and is most widely available across distributions and even across platforms. **bzip2** compression creates smaller archive files compared to gzip but is less widely available than **gzip**, while the **xz** compression method is relatively new, but usually offers the best compression ratio of the methods available.

It is good practice to use a single top-level directory, which can contain other directories and files, to simplify the extraction of the files in an organized way.

Use one of the following options to create a compressed tar archive:

- -z or --gzip for gzip compression (filename.tar.gz or filename.tgz)
- -j or --bzip2 for bzip2 compression (filename.tar.bz2)
- -J or -xz for xz compression (filename.tar.xz)

To create a gzip compressed archive named /root/etcbackup.tar.gz, with the contents from the /etc directory on host:

```
[root@localhost ~]# tar -czf /root/etcbackup.tar.gz /etc
    tar: Removing leading `/' from member names
```

To create a bzip2 compressed archive named /root/logbackup.tar.bz2, with the contents from the /var/log directory on host:

```
[root@localhost ~]$ tar -cjf /root/logbackup.tar.bz2 /var/log
tar: Removing leading `/' from member names
```

To create a xz compressed archive named, /root/sshconfig.tar.xz, with the contents from the /etc/ssh directory on host:

```
[root@localhost ~]$ tar -cJf /root/sshconfig.tar.xz /etc/ssh
tar: Removing leading `/' from member names
```

After creating an archive, verify the content of an archive using the tf options. It is not mandatory to use the option for compression agent when listing the content of a compressed archive file. For example, to list the content archived in the /root/etcbackup.tar.gz file, which uses the gzip compression, use the following command:

```
[root@localhost ~]# tar -tf /root/etcbackup.tar.gz /etc
    etc/
    etc/fstab
    etc/crypttab
    etc/mtab
    ...output omitted...
```

EXTRACTING A COMPRESSED ARCHIVE

The first step when extracting a compressed tar archive is to determine where the archived files should be extracted to, then create and change to the target directory. The tar command determines which compression was used and it is usually not necessary to use the same compression option used when creating the archive. It is valid to add the decompression method to the tar command. If one chooses to do so, the correct decompression type option must be used; otherwise tar yields an error about the decompression type specified in the options not matching the file's decompression type.

To extract the contents of a gzip compressed archive named /root/etcbackup.tar.gz in the**/tmp/etcbackup** directory:

```
[root@localhost ~]# mkdir /tmp/etcbackup
[root@localhost ~]# cd /tmp/etcbackup
[root@localhost etcbackup]# tar -tf /root/etcbackup.tar.gz
etc/
etc/fstab
etc/crypttab
etc/mtab
...output omitted...
[root@localhost etcbackup]# tar -xzf /root/etcbackup.tar.gz
```

To extract the contents of a bzip2 compressed archive named /root/logbackup.tar.bz2 in the /tmp/logbackup directory:

```
[root@localhost ~]# mkdir /tmp/logbackup
[root@localhost ~]# cd /tmp/logbackup
[root@localhost logbackup]# tar -tf /root/logbackup.tar.bz2
var/log/
var/log/lastlog
var/log/README
var/log/private/
var/log/wtmp
var/log/btmp
...output omitted...
[root@localhost logbackup]# tar -xjf /root/logbackup.tar.bz2
```

To extract the contents of a xz compressed archive named /root/sshbackup.tar.xz in the /tmp/sshbackup directory:

```
[root@localhost ~]$ mkdir /tmp/sshbackup
[root@localhost ~]# cd /tmp/sshbackup
[root@localhost logbackup]# tar -tf /root/sshbackup.tar.xz
etc/ssh/
etc/ssh/moduli
etc/ssh/ssh_config
etc/ssh/ssh_config.d/
etc/ssh/ssh_config.d/05-redhat.conf
etc/ssh/sshd_config
...output omitted...
[root@localhost sshbackup]# tar -xJf /root/sshbackup.tar.xz
```

Listing a compressed tar archive works in the same way as listing an uncompressed tar archive.