

How to Compress Files in Linux Using the tar Command

Learn how to compress files in Linux using the tar command with step-by-step instructions, common options, and practical examples for efficient file archiving.

Contents

01	Introduction	3
02	tarCommand Syntax	3
03	Practical Examples of thetarCommand	4
04	Advanced Usage Scenarios	10
05	Conclusion	14

Introduction

The `tar` command in Linux enables archiving and compression of files and directories. It's also known as tape archive (`tar`) and combines multiple files into a single archive file with the `.tar` extension that can be compressed to save disk space to simplify file transfers and backups.

This article explains how to use the `tar` command in Linux to perform archiving and file compression tasks.

`tar` Command Syntax

Below is the basic `tar` command syntax:

CONSOLE

```
$ tar [options] [archive-file] [file/directory]
```

Within the above command, `[options]` modifies the `tar` command behavior, `[archive-file]` specifies the archive filename, and `[file/directory]` specifies the files or directories to include in the archive.

`tar` Command Options

Option Description

- `-c` Create a new archive.
- `-x` Extract files from an archive.
- `-v` Verbose mode. Display progress in the terminal.
- `-f` Filename. Specify the archive filename.
- `-z` Compress the archive using gzip.
- `-j` Compress the archive using bzip2.
- `-J` Compress the archive using xz.
- `-t` List the contents of an archive.

Option Description

- r Append files to an existing archive.
- u Update files in an existing archive.

Practical Examples of the `tar` Command

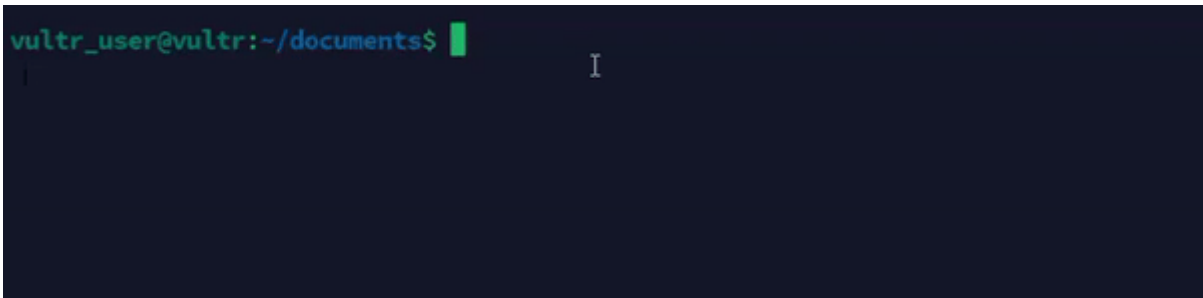
1. Create a new tar archive.

CONSOLE

```
$ tar -cvf archive.tar file1.txt file2.txt directory/
```

The above command creates a new `archive.tar` that contains the file `file1.txt`, `file2.txt`, and all contents in `directory/`.

Output:



```
vultr_user@vultr:~/documents$ tar -cvf archive.tar file1.txt file2.txt directory/
```

2. Create a new compressed `tar` archive using Gzip.

CONSOLE

```
$ tar -czvf archive.tar.gz file1.txt file2.txt directory/
```

The above command creates a new `archive.tar.gz` and enables compression using Gzip.

Output:

```
vultr_user@vultr:~/documents$
```

3. Extract files from a tar archive.

CONSOLE

```
$ tar -xvf archive.tar
```

The above command extracts all files from the `archive.tar` file into your working directory.

Output:

```
vultr_user@vultr:~/documents$
```

4. Extract files from a compressed `tar` archive.

CONSOLE

```
$ tar -xzvf archive.tar.gz
```

The above command extracts all files from the compressed `archive.tar.gz` file to the working directory.

Output:

```
vultr_user@vultr:~/documents$
```

5. List all contents in a tar archive.

CONSOLE

```
$ tar -tvf archive.tar
```

The above command lists all contents in the `archive.tar` file without extraction.

Output:

```
vultr_user@vultr:~/documents$
```

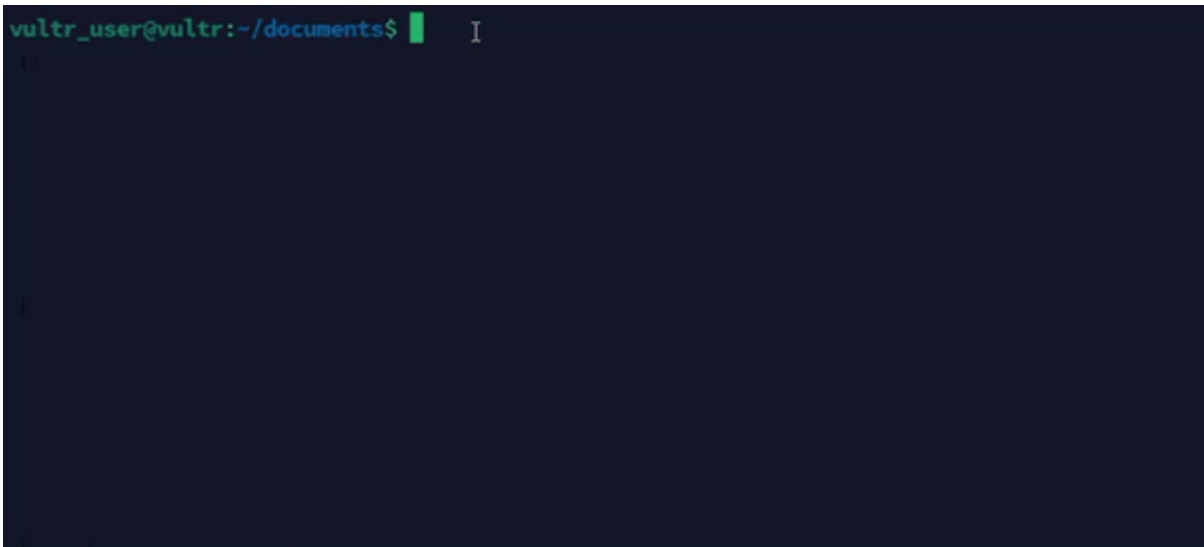
6. Create a compressed `tar` archive using Bzip.

CONSOLE

```
$ tar -cjvf archive.tar.bz2 file1.txt file2.txt directory/
```

The above command creates a new compressed `archive.tar.bz2` file using the `bzip2` utility.

Output:



```
vultr_user@vultr:~/documents$
```

7. Extract files from a Bzip `tar` archive.

CONSOLE

```
$ tar -xjvf archive.tar.bz2
```

The above command extracts all files from the `archive.tar.bz2` to the your working directory.

Output:

```
vultr_user@vultr:~/documents$ █
```

8. Create a new `tar` archive using the `xzipped` compression format.

CONSOLE

```
$ tar -cJvf archive.tar.xz file1.txt file2.txt directory/
```

The above command creates a new compressed `archive.tar.xz` file using XZ.

Output:

```
vultr_user@vultr:~/documents$ █
```

9. Extract files from an `xzipped` archive file.

CONSOLE

```
$ tar -xJvf archive.tar.xz
```

The above command extracts files from the `archive.tar.xz` file to your working directory.

Output:

```
vultr_user@vultr:~/documents$
```

10. Add new files to an existing tar archive.

CONSOLE

```
$ tar -rvf archive.tar newfile
```

The above command appends the file `newfile` to the existing `tar` archive `archive.tar`.

Output:

```
vultr_user@vultr:~/documents$ ls
archive.tar archive.tar.gz archive.tar.xz directory file1.txt file2.txt file3.txt newfile
vultr_user@vultr:~/documents$
```

11. Update files in an existing tar archive.

CONSOLE

```
$ tar -uvf archive.tar updatedfile
```

The above command updates the `updatedfile` file in the `archive.tar` archive if its newer than the existing version.

Output:

```
vultr_user@vultr:~/documents$
```

Advanced Usage Scenarios

1. Exclude files and directories from `tar` archive.

CONSOLE

```
$ tar --exclude='*.log' -cvf archive.tar directory/
```

The above command creates a new `archive.tar` file that includes all files except `.log` files in `directory/`.

Output:

```
vultr_user@vultr:~/dir1$
```

2. Compress and archive files to a remote location.

CONSOLE

```
$ tar -czvf - directory/ | ssh user@remotehost "cat > /path/to/destination/archive.tar.gz"
```

The above command compresses `directory/` and transfers it to `remotehost` using SSH.

Output:

```
vultr_user@vultr:~$ tar -czvf - /home/vultr_user/dir1 | ssh
vultr_user@123.84.123.123 "cat > /home/vultr_user/dir2/archive.tar.gz"
tar: Removing leading `/' from member names
/home/vultr_user/dir1/
/home/vultr_user/dir1/file2.txt
/home/vultr_user/dir1/file1.txt
/home/vultr_user/dir1/file3.txt
The authenticity of host '123.84.123.123 (123.84.123.123)' can't be established.
ED25519 key fingerprint is SHA256:kfJi2t1589s4r3mdDLq32Ziksl1j5f2sXM6aUw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
vultr_user@139.84.176.145's password:
vultr_user@vultr:~$
```

3. Extract a specific file from a tar archive.

CONSOLE

```
$ tar -xvf archive.tar path/to/file
```

The above command extracts `path/to/file` from `archive.tar`.

Output:

```
vultr_user@vultr:~/dir1$
```

```
I
```

4. Verify the integrity of a tar archive.

CONSOLE

```
$ tar -tvf archive.tar > /dev/null
```

The above command lists all contents of `archive.tar` and discards the output to verify the file integrity.

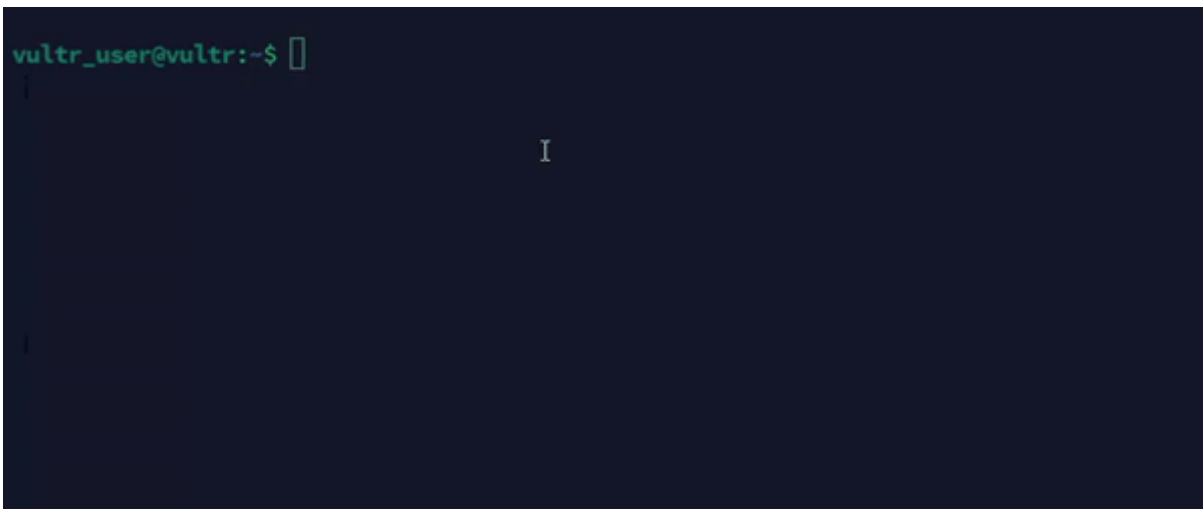
5. Use `tar` with `find` to archive specific files.

CONSOLE

```
$ find . -name "*.txt" -print0 | tar -cvf archive.tar --null -T -
```

The above command finds all `.txt` files and archives them to the `archive.tar` file.

Output:

A terminal window with a dark background. The prompt is 'vultr_user@vultr:~\$'. The command '\$ find . -name "*.txt" -print0 | tar -cvf archive.tar --null -T -' has been entered. The output is a single character 'I' on the next line.

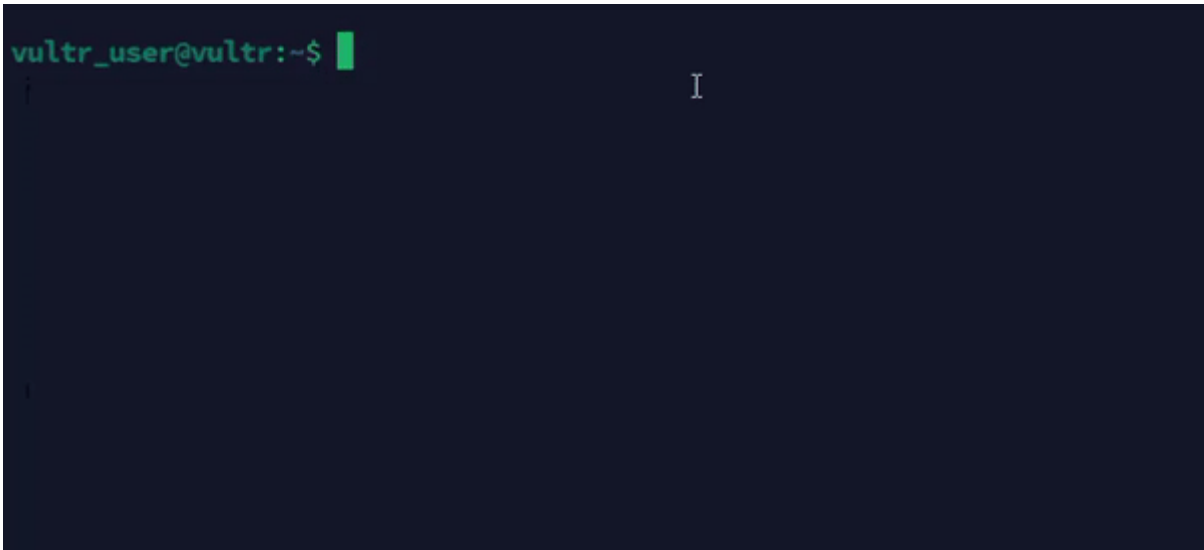
6. Extract files to a specific directory.

CONSOLE

```
$ tar -xvf archive.tar -C /path/to/directory/
```

The above command extracts files from `archive.tar` to `/path/to/directory/`.

Output:



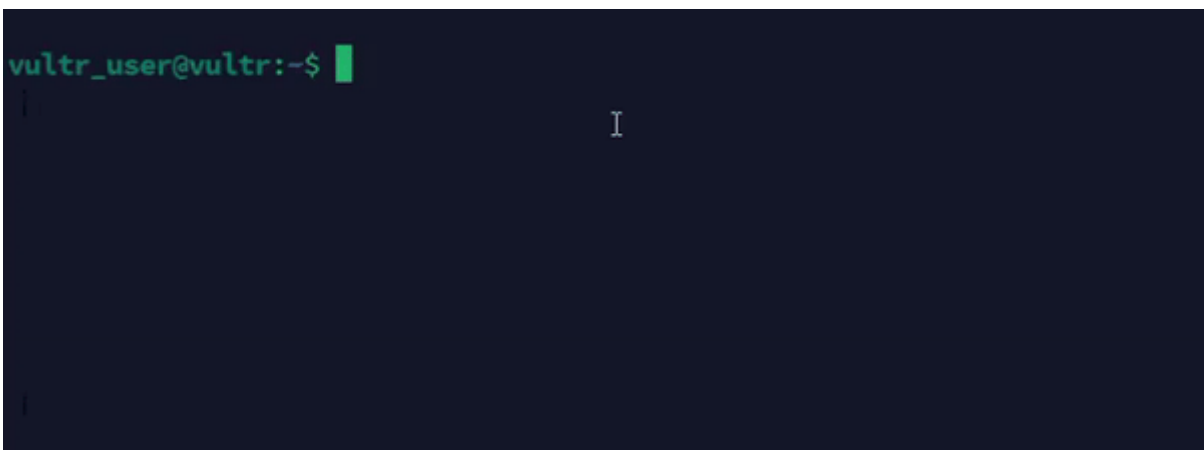
7. Create a new tar archive split into multiple volumes

CONSOLE

```
$ tar -cvf - directory/ | split -b 5M - archive_part_
```

The above command creates a new tar archive using files from `directory/` and splits it into multiple volumes of `5 MB` for each file.

Output:



Conclusion

You have used the `tar` command to archive and compress files, manage backups, and optimize storage on your system. For more command options, run the `man tar` command to view the `tar` manual page.



VULTR

