16 Tar Commands to Compress and Extract Files in Linux

April 12, 2018LINUX COMMANDS, LINUX HOWTOUpdated April 12, 2018

Tar is a command line tool used to create and manipulate archive files in Linux and Unix systems. Tar can extract from tar, pax, cpio, zip, jar, ar, and ISO 9660 cdrom images and can create respective archives.

Below is a list of commonly used tar options.

`- j` : Used to filter archive through bzip2.
`- v` : Run the command in verbose mode to show the progress of archive file.
`- f` : Specify archive filename.
`- W` : Used to verify an archive file.
`- z` : Filter an archive through gzip tool.
`- t` : This is used to view the content of archive file.
`-c` : Create a new archive containing the specified items.
`-r` : Used for appending or updating existing archive with files or directories
`-t` : List contents of an archive on stdout.
`-u` : Like -r, but new entries are added only if they have a modification date newer than the corresponding entry in the archive.
`-x` : Extract archive file to disk.

For all tar operations, the utility exits with a **0** on success, and **>0** if an error occurs. It is important to understand this if you're using tar with a script. Let's see tar in action through examples.

Create Archives with tar

1) Create uncompressed archive (.tar) with tar

To create an uncompressed tar, use the command syntax:

```
$ tar -cvf /path/to/foo.tar /path/to/foo/
```

Where */path/to/foo.tar* is the location of resulting tar file, and /path/to/foo is the location of the file or directory to use as input. E.g

```
$ tar -cvf powerlog.tar powerlog

a powerlog

$ ls

powerlog powerlog.tar
```

You can use an absolute path instead of relative shown in the example.

2) Create a .gz archive with tar

The command syntax to create a **.gz** archive with tar is:

```
$ tar -czvf /path/to/foo.tgz /path/to/foo/
```

Where */path/to/foo.tgz* is the destination compressed file, and */path/to/foo/* is source directory or file to be compressed.

3) Create a .bz2 archive

The option **-j** is used to work with .bz2 archives.,

```
$ tar -cjvf /path/to/foo.tgz /path/to/foo/
```

4) Create a .gz archive and exclude all jpg,flv... from the tgz

We use --exclude tar option to exclude a specific extension or list of file extensions. The syntax is similar to below:

```
$ tar -czvf /path/to/foo.tgz --exclude=\*.{jpg,gif,png,wmv,flv,tar.gz,zip} /path/to/foo/.
```

5) Use tar with no absolute path on compress

When using tar to compress files and you provide an absolute path, a compressed file will have a full path in its metadata, So when you extract the content, it will be in directories similar to the absolute path that was passed. To avoid this, add a **.** at the end of the command.

```
$ tar -zcvf /path/to/foo.tgz -C/path/to/foo .
```

6) Create a tar archive of everything in the current directory starting with an “i”

```
# tar -cvf fullbackup.tar i*

install.log

install.log.syslog
```

7) Above command archived only those files, which starts from i.

Append a file or add a new file to existing tar archive.

```
# tar --append --file=backup.tar anaconda-ks.cfg
```

The above command will append an anaconda-ks.cfg file in the backup.tar archive

8) Adding two archive files with concatenate option.

```
# tar --concatenate --file=backup.tar fullbackup.tar
```

The above command adds the content of the fullbackup.tar to backup.tar archive.

Extract Archive with tar

9) Extract an uncompressed archive with tar

To extract an uncompressed archive which often has an extension **.tar**, the command syntax is:

```
$ tar -xvf /path/to/foo.tar
```

Where /path/to/foo.tar is the absolute/relative path to the location of a tar file. E.g

```
$ tar -xvf powerlog.tar

x powerlog/
```

10) Extract a .gz archive

To extract a .gz archive, we use the command syntax:

```
$ tar -xzvf /path/to/foo.tgz
```

Where */path/to/foo.tgz* is the absolute or relative path to the compressed file.

11) Extract a .bz2 archive

The command syntax to extract a .bz2 archive is

```
$ tar -xjvf /path/to/foo.bz2
```

Note that you can pass a .tgz file as long as it was compressed with a **-j** option. tar will auto-detect the format and uncompress it.

12) Extract a .tar in specified Directory

To extract a tar archive to a specific directory, use **-C** option to provide destination directory.

```
$ tar -xvf /path/to/foo.tar -C /path/to/destination/
```

13) Extracting the file from the tar archive

```
# tar --extract -vv --occurrence --file=./backup.tar anaconda-ks.
cfg

-rw-r--r-- root/root 766 2008-04-12 06:52:42 anaconda-ks.cfg
```

The above command extracts the anaconda-ks.cfg file in the backup.tar archive.

List Archive contents with tar

14) List the content of a .gz archive

Suppose you only want to view the contents of a **.tgz** file without extracting them, you'll use the command syntax:

```
$ tar -ztvf /path/to/foo.tgz
```

Just replace /path/to/foo.tgz with correct path of your *.tgz* file

15) List the content of a .bz2 archive

If you would like to just view the content of *.bz2* archive without extracting, use the command:

```
$ tar -jtvf /path/to/foo.tgz
```

16) How to preserve symbolic links using the tar command

Use tar **-cvhf** to preserve my symbolic links when generating a tar archive (Use the option “h” for that).

Tar is one of the commands that are often used in Linux/Unix world, from software installation, backups, file sharing, file encryption/decryption process. Mastering tar will help you a lot when administering Linux/Unix systems.

Linux Commands, tar Command

About Josphat Mutai

Red Hat Certified Engineer with more than 2 years experience managing critical Cloud Infrastructures and Services. Passionate about Technology and loves testing and reviewing gadgets and Open-source tools.

Author Archive Page