

# File Archiving and Backup with tar

# What is tar?

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). 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. Examples of how to use the **tar** command are included in this section.

## Operate the tar command

To use the **tar** command, one of the three following actions is required:

- **c** (create an archive)
- **t** (list the contents of an archive)
- **x** (extract an archive)

Commonly used options are:

- **f *file name*** (file name of the archive to operate on)
- **v** (verbosity; useful to see which files get added to or extracted from the archive)

```
[user@host ~]# tar cf archive.tar file1 file2 file3
[user@host ~]# ls archive.tar
archive.tar
```

```
[root@host ~]# tar tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
etc/mtab
...
```

```
[root@host ~]# mkdir /root/etcbackup
[root@host ~]# cd /root/etcbackup
[root@host etcbackup]# tar xf /root/etc.tar
```

To create a compressed tar archive, one of the following **tar** options can be specified:

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

Create (c option) a gzip-compressed (z option) tar archive **/root/etcbackup.tar.gz** of the **/etc** directory on serverX:

```
[root@serverX ~]$ tar czf /root/etcbackup.tar.gz /etc
```

```
[student@desktopX ~]$ scp /etc/yum.conf /etc/hosts serverX:/home/student
student@serverX's password: student
yum.conf          100% 813    0.8KB/s  00:00
hosts            100% 227    0.2KB/s  00:00
```

A user can copy a file from a remote account on a remote machine to the local file system with **scp**. In this example, copy the file **/etc/hostname** from the account **student** on the **serverX** machine to the local directory **/home/student/**.

```
[student@desktopX ~]$ scp serverX:/etc/hostname /home/student/
student@serverX's password: student
hostname          100%  22    0.0KB/s  00:00
```

```
[student@desktopX ~]$ sftp serverX
student@serverX's password: student
Connected to serverX.
sftp>
```

```
sftp> mkdir hostbackup
sftp> cd hostbackup
sftp> put /etc/hosts
Uploading /etc/hosts to /home/student/hostbackup/hosts
/etc/hosts                100% 227    0.2KB/s   00:00
sftp>
```

```
sftp> get /etc/yum.conf
Fetching /etc/yum.conf to yum.conf
/etc/yum.conf              100% 813    0.8KB/s   00:00
sftp> exit
[student@desktopX ~]$
```

The two most common options when synchronizing files and folders with **rsync** are the **-a** and **-v** options. While the **-v** option adds verbosity to the output as the synchronization proceeds, the **-a** option stands for "archive mode" and enables the following options all in one:

- **-r**, synchronize recursively the whole directory tree
- **-l**, synchronize symbolic links
- **-p**, preserve permissions
- **-t**, preserve time stamps
- **-g**, preserve group ownership
- **-o**, preserve the owner of the files
- **-D**, synchronize device files

```
[root@desktopX ~]# rsync -av root@serverX:/var/log /serverlogs
```

```
...
```