# Archiving & Compression (Core)

Linux Commands Course · Section 7

#### Goal

Learn how to bundle and compress files efficiently on Linux.

You'll understand how to create archives, unpack them, and choose the right compression tool for each situation.

#### What Is Archiving?

Archiving combines multiple files or folders into one container file. Compression makes that container smaller.

Common reasons to archive:

- Backup and transfer data
- Package projects or logs
- Preserve directory structures

Linux standard tools: tar, gzip, bzip2, xz, zstd, zip.

# **Creating Tar Archives**

tar (tape archive) is the most common archiving utility.

Create an archive from a folder:

tar -cvf backup.tar project/

Options:

c - create
v - verbose (show files)
f - file name

Extract archive:

tar -xvf backup.tar

List contents without extracting:

tar -tvf backup.tar

#### **Compressed Tarballs**

tar can compress directly using gzip, bzip2, xz, or zstd.

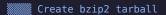
Create compressed archive (gzip)

tar -czvf project.tar.gz project/

tar -xzvf project.tar.gz

You can also use different extensions to choose the compression algorithm automatically.

### bzip2 and xz Examples



tar -cjvf data.tar.bz2 data/

#### Extract it:

tar -xjvf data.tar.bz2

#### Create xz tarball

tar -cJvf data.tar.xz data/

#### Extract it:

tar -xJvf data.tar.xz

Flag	Algorithm	Extension
z	gzip	.gz
j	bzip2	.bz2
J	xz	.xz

#### **Modern Compression — zstd**

zstd (Zstandard) is a fast modern compressor with excellent ratios.

tar -I zstd -cvf project.tar.zst project/

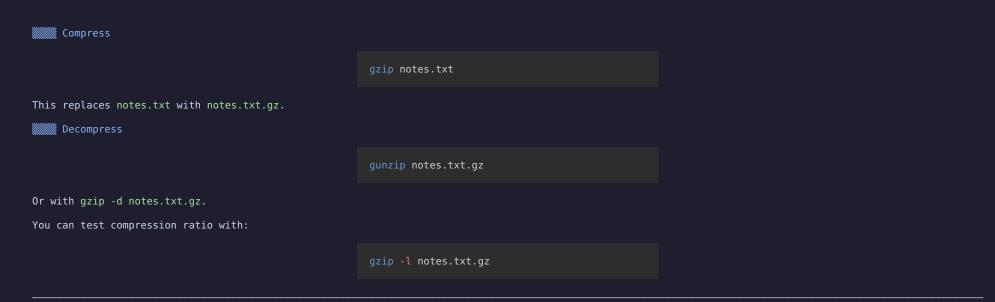
Extract:

tar -I zstd -xvf project.tar.zst

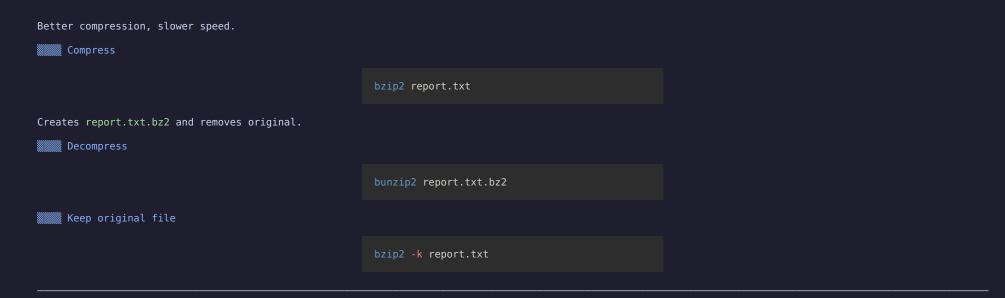
You can also use the standalone tools:

zstd file.txt # creates file.txt.zst # decompresses it

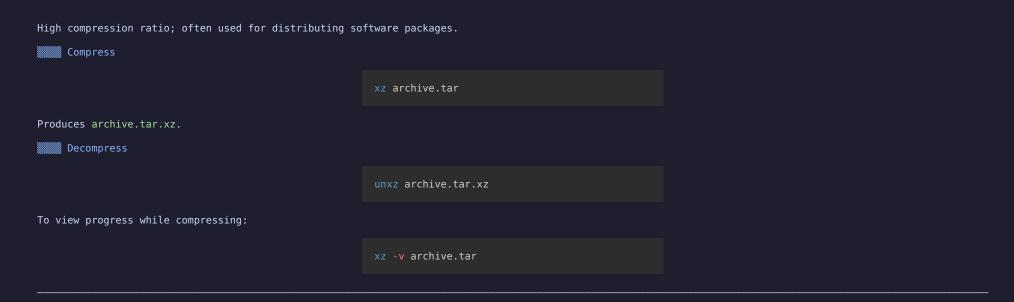
#### gzip and gunzip (Classic Pair)



### bzip2 and bunzip2



#### xz and unxz



# Cross-Platform Archives — zip and unzip

ZIP is widely supported across operating systems.		
Create zip archive		
	<pre>zip -r project.zip project/</pre>	
Extract zip file		
	unzip project.zip	
Extract to specific folder		
	unzip project.zip -d /tmp/project	
List contents:		
	unzip -l project.zip	

# **Choosing the Right Tool**

Tool	Format	Speed	Compression	Portability	Use case
gzip	.gz	Fast	Medium	High	Everyday backups
bzip2	.bz2	Medium	Higher	Medium	Logs, archives
xz	.xz	Slow	Very High	Medium	Software packaging
zstd	.zst	Very Fast	High	Medium	Modern systems
zip	.zip	Fast	Medium	Very High	Cross-platform

# **Inspecting Archive Contents**

List files in an archive without extracting:

tar -tvf archive.tar
unzip -l project.zip

Test integrity (for .zip):

unzip -t project.zip

#### **Combine with Pipelines**

Create and compress on the fly:

tar -czf - project/ | ssh backup@server "cat > /backups/project.tgz"

Or decompress remotely:

ssh backup@server "cat /backups/project.tgz" | tar -xz

This allows archiving without intermediate files.

#### Recap

- tar archive multiple files (-cvf, -xvf)
   gzip / bzip2 / xz / zstd compression algorithms
   zip / unzip cross-platform archives
   Choose based on speed, ratio, and compatibility needs.

#### **Practice**

- Create a tar archive of your home directory.
   Compress it using gzip, bzip2, and xz compare sizes.
   Extract each version and verify integrity.
   Create a .zip archive of your project folder.
   List contents without extracting.
   Try using zstd for a fast modern backup.

### Next Up

**Essential Linux Directories (Core)** — understanding the structure of the filesystem.