gzip, xz and zstd Compress files

CONCEPTS

gzip and xz delete the input file by default; zstd keeps it. Add -k to gzip/xz, or --rm to zstd to delete.

These tools compress single streams, not folders. Use tar to bundle a directory first (tar -caf picks the compressor from the suffix).

Rule of thumb: gzip for compatibility, zstd for speed and everyday use, xz for archives that are written once and downloaded often.

GZIP

gzip *<file>*
Compress to <file>.gz, remove original

gzip -k *<file>*
Compress, keep the original

gzip -d *<file.gz>*
Decompress (same as gunzip)

gzip -c *<file>* **>** *<out.gz>*
Write to stdout, keep original

XZ

xz *<file>*
Compress to <file>.xz, remove original

xz -k *<file>*
Compress, keep the original

xz -d *<file.xz>*
Decompress (same as unxz)

xz -T0 *<file>*
Use all CPU cores

ZSTD

zstd *<file>*
Compress to <file>.zst, keep original

zstd -d *<file.zst>*
Decompress (same as unzstd)

zstd -19 *<file>*
High compression, slower

ZSTD (cont.)

zstd -T0 *<file>*
Use all CPU cores

READ AND INSPECT

zcat *<file.gz>*
Print decompressed content

xzcat *<file.xz>*
Print decompressed content

zstdcat *<file.zst>*
Print decompressed content

zgrep *<pat>* *<file.gz>*
Search without unpacking

TEST AND CHECK

gzip -t *<file.gz>*
Test integrity, silent if ok

PIPES AND ARCHIVES

<cmd> **|** **gzip >** *<out.gz>*
Compress a stream

gzip -dc *<file.gz>* **|** *<cmd>*
Decompress into a pipe

tar -caf *<out.tar.zst>* *<dir>*
Archive, compressor from suffix

ZIP AND UNZIP

zip -r *<out.zip>* *<dir>*
Zip a directory

unzip *<file.zip>*
Extract here

WHICH FORMAT

Compared

Tool	Speed	Size	Use for
gzip	fast	medium	compatibility, logs
zstd	very fast	good	everyday, backups
xz	slow	smallest	release tarballs
zip	fast	medium	sharing with Windows