File Archiving and Compression Cheat Sheet

The Overall Process

Archiving and compressing files in Linux is a two-step process.

1) Create a Tarball

   First, you will create what is known as a tar file or "tarball". A tarball is a way of bundling together the files that you want to archive.

2) Compress the tarball with a compression algorithm

   Secondly, you will then compress that tarball with one of a variety of compression algorithms; leaving you with a compressed archive.

1. Creating a Tarball

Tarballs are created using the tar command.

Creating a Tar Ball	tar –cvf <name of tarball> <file>…

The –c option: "create". This allows us to create a tarball. [required]

The –v option: "verbose". This makes tar give us feedback on its progress. [optional]

The –f option: Tells tar that the next argument is the name of the tarball. [required]

<name of tarball>: The absolute or relative file path to where you want the tarball to be placed; e.g. ~/Desktop/myarchive.tar. It is recommended that you add .tar to your proposed filename for clarity.

<file>: The absolute or relative file paths to files that you want to insert into the tarball. You can have as many as you like and wildcards are accepted.

1.1 Checking a Tarball's Contents

Once the tarball has been created, you can check what is inside it using the tar command.

Checking the contents of a Tarball	tar –tf <name of tarball>

The –t option: "test-label". This allows us to check the contents of a tarball. [required]

The –f option: Tells tar that the next argument is the name of the tarball. [required]

<name of tarball>: The absolute or relative file path to where you want the tarball to be placed; e.g. ~/Desktop/myarchive.tar

1.2 Extracting From a Tar ball

Let's say that you download a tar file from the internet and you want to extract its contents using the command line. How can you do that?

For this you would again use the tar command

Extracting a Tar ball's Contents	tar –xvf <name of tarball>

The –x option: "extract". This allows us to extract a tarball's contents. [required]

The –v option: "verbose". This makes tar give us feedback on its progress. [optional]

The –f option: Tells tar that the next argument is the name of the tarball. [required]

<name of tarball>: The absolute or relative file path to where the tarball is located; e.g. ~/Desktop/myarchive.tar

Extracting a tarball does **not** empty the tarball. You can extract from a tarball as many times as you want without affecting the tarball's contents.

2. Compressing Tarballs

Tarballs are just containers for files. They don't by themselves do any compression, but the can be compressed using a variety of compression algorithms

The main types of compression algorithms are gzip and bzip2.

The gzip compression algorithm tends to be faster than bzip2 but, as a trade-off, gzip usually offers less compression.

You can find a comparison of various compression algorithms using this excellent blog post.

2.1 Compressing and Decompressing with gzip

Compressing with gzip	gzip <name of tarball>
Decompressing with gzip	gunzip <name of tarball>

When compressing with gzip, the file extension .gz is automatically added to the .tar archive. Therefore, the gzip compressed tar archive would, by convention, have the file extension .tar.gz

2.2 Compressing and Decompressing with bzip2

Compressing with bzip2	bzip2 <name of tarball>
Decompressing with bzip2	bunzip2 <name of tarball>

When compressing with bzip2, the file extension .bz2 is automatically added to the .tar archive. Therefore, the bzip2 compressed tar archive would, by convention, have the file extension .tar.bz2

3. Doing it all in one step

Because compressing tar archives is such a common function, it is possible to create a tar archive and compress it all in one step using the tar command. It is also possible to decompress and extract a compressed archive in one step using the tar command too.

To perform compression/decompression using gzip compression algorithm in the tar command, you provide the z option in addition to the other options required.

Creating a tarball and compressing via gzip	tar –cvzf <name of tarball> <file>…
Decompressing a tarball and extracting via xzip	tar –xvzf <name of tarball>

To perform compression/decompression using bzip2 compression algorithm in the tar command, you provide the j option to the other options required.

Creating a tarball and compressing via bzip2	tar –cvjf <name of tarball> <file>…
Decompressing a tarball and extracting via bzip2	tar –xvjf <name of tarball>

To perform compression/decompression using the xzip compression algorithm in the tar command, you provide the J option to the other options required.

Creating a tarball and compressing via xzip	tar –cvJf <name of tarball> <file>…
Decompressing a tarball and extracting via xzip	tar –xvJf <name of tarball>

4. Creating .zip files

Although .tar.gz and .tar.bz2 archives are the archives of choice on Linux, .zip archives are common on other operating systems such as Windows and Mac OSX.

In order to create such archives, you can use the following commands.

Creating a .zip archive	zip <name of zipfile> <file>…
Extracting a .zip archive	unzip <name of zipfile>

<name of zipfile>: The absolute or relative file path to the .zip file e.g. ~/ myarchive.zip

<file>: The absolute or relative file paths to files that you want to insert into the .zip file. You can have as many as you like and wildcards are accepted.