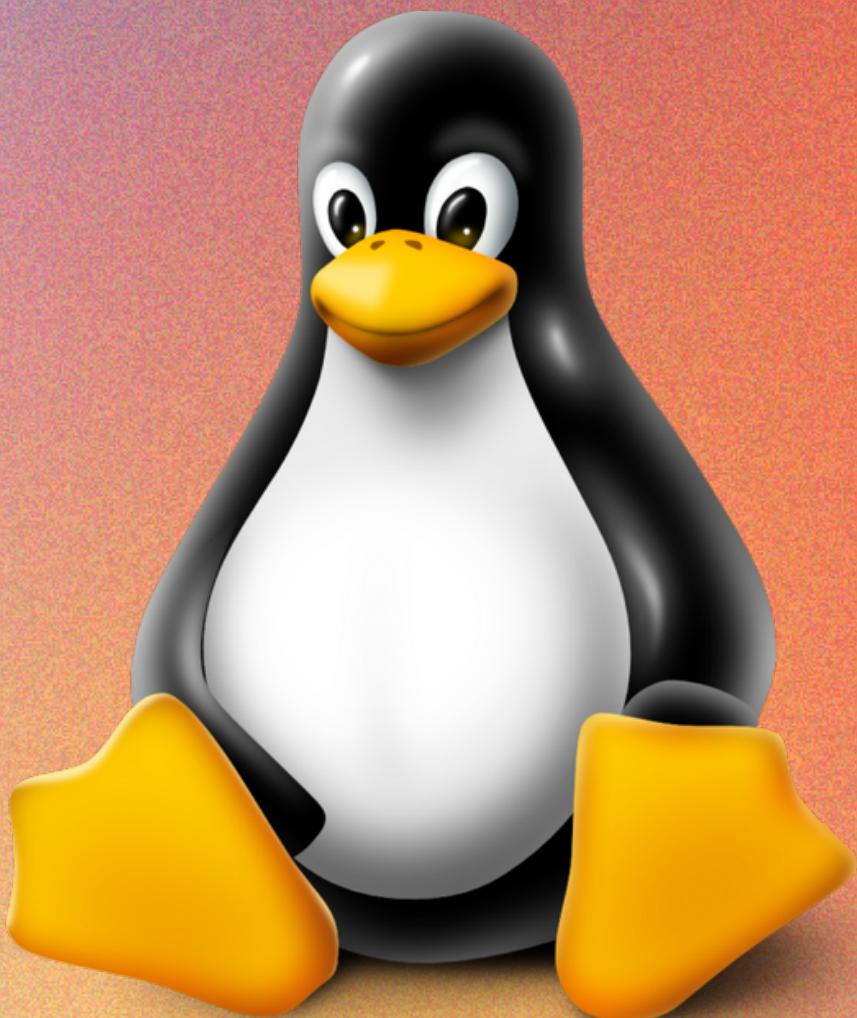


LINUX

ARCHIVE COMMANDS & REMOTE SYSTEMS



GOWTHAM S
gowtham26.work@gmail.com

Linux

Archive commands and Remote systems

Introduction

Linux offers powerful tools to archive files and access remote systems. Archiving helps in compressing, storing, and transferring files efficiently. Remote access tools like `ssh` and `telnet` allow users to securely log in and manage systems over networks.

1. Archiving Files

zip & unzip

1. Purpose

- **zip:** Compresses one or more files/folders into a **.zip** archive.
- **unzip:** Extracts files from a **.zip** archive.

2. Use Case

- Create compressed backups to save storage.

- Group files for easier file transfer (e.g., via email or USB).
- Extract only specific files from large archives when needed.

3. Common Commands

Command	Meaning
<code>zip archive.zip file1 file2</code>	Compress file1 and file 2 into archive.zip
<code>zip -r archive.zip folder/</code>	Recursively compresses the entire folder and its contents into archive.zip.
<code>unzip archive.zip</code>	Extracts all files from archive.zip into the current directory.

<code>unzip -l archive.zip</code>	Lists the contents of <code>archive.zip</code> without extracting them.
-----------------------------------	---

4. Examples

Command	Meaning
<code>zip mydocs.zip report.txt notes.txt</code>	Creates a <code>mydocs.zip</code> archive containing <code>report.txt</code> and <code>notes.txt</code> .
<code>zip -r project_backup.zip my_project/</code>	Compresses the entire <code>my_project</code> folder including subfolders and files.
<code>unzip mydocs.zip</code>	Extracts all contents into the current folder.
<code>unzip -l mydocs.zip</code>	Shows a list of all files in the archive.



1. Purpose

- **tar** (short for tape archive) combines multiple files into a single **.tar** archive.
- Often used with compression tools like **gzip** or **bzip2** to create **.tar.gz** or **.tar.bz2** files.

2. Use Case

- Create **.tar** or **.tar.gz** files for packaging.
- Archive entire directories into a single file.
- View contents of an archive without extracting.

- Exclude specific files from the archive.

3. Common Commands

Command	Meaning
<code>tar -cvf archive.tar file1 file2</code>	Creates <code>archive.tar</code> from <code>file1</code> and <code>file2</code> . <code>-c</code> (create), <code>-v</code> (verbose), <code>-f</code> (filename).
<code>tar -xvf archive.tar</code>	Extracts all files from <code>archive.tar</code> . <code>-x</code> (extract).
<code>tar -tvf archive.tar</code>	Lists files inside <code>archive.tar</code> without extracting.
<code>tar --exclude='*.log' -cvf logs_removed.tar folder/</code>	Archives the folder but excludes all <code>.log</code> files.

4. Examples

Command	Meaning
<code>tar -cvf myfiles.tar file1.txt file2.txt</code>	Combines the files into a <code>myfiles.tar</code> archive.
<code>tar -xvf myfiles.tar</code>	Extracts all files in the current directory.
<code>tar -tvf myfiles.tar</code>	Displays the list of files in the archive.
<code>tar -czvf archive.tar.gz folder/</code>	<code>-z</code> adds gzip compression to reduce file size.
<code>tar -xzvf archive.tar.gz</code>	Extract a <code>.tar.gz</code> file
<code>tar --exclude='*.log' -cvf clean_archive.tar project/</code>	Archives everything from <code>project/</code> except <code>.log</code> files.

5. Tips

- **.tar** is just archiving (not compression).
- **.tar.gz** = **.tar** archive + gzip compression.
- Use **-j** instead of **-z** for **.tar.bz2** (bzip2 compression):

```
tar -cjvf archive.tar.bz2 folder/
```

gzip & gunzip

1. Purpose

- **gzip**: Compresses a single file and replaces it with a **.gz** file.
- **gunzip**: Decompresses **.gz** files and restores the original.

2. Use Case

- Compress large log or text files to save disk space.
- Share smaller file sizes over networks.
- Use with **tar** to create compressed archives (**.tar.gz**).

- Quickly view compressed content without full extraction.

3. Common Commands

Command	Meaning
<code>gzip filename</code>	Compress filename and creates filename.gz
<code>gunzip filename.gz</code>	Decompresses filename.gz back to filename
<code>gzip -r folder/</code>	Recursively compresses all files in the folder.
<code>zcat file.gz</code>	Displays contents of a compressed file without decompressing.

4. Examples

Command	Meaning
<code>gzip logfile.txt</code>	Replaces <code>logfile.txt</code> with <code>logfile.txt.gz</code> .
<code>gunzip logfile.txt.gz</code>	Restores the original <code>logfile.txt</code> .
<code>gzip -r mylogs/</code>	Compresses all files inside <code>mylogs/</code> , leaving folders untouched.
<code>zcat logfile.txt.gz</code>	Prints the content of <code>logfile.txt.gz</code> to the terminal.

5. Tips

- **gzip** works only on individual files, not directories.
- Use **tar** with **gzip** to compress multiple files:

```
tar -czvf archive.tar.gz folder/
```

- You can also use gunzip as:

```
gzip -d file.gz
```

2. Remote System Access

ssh - Secure Shell

1. Purpose

- ssh provides secure, encrypted communication between two systems over a network.
- Commonly used for remote login, administration, and file transfers.

2. Use Case

- Log in to a remote Linux server or system.
- Manage servers from anywhere without physical access.

- Transfer files securely using `scp` or `sftp`.
- Automate login with SSH keys (no password prompt).

3. Common Commands

Command	Meaning
<code>ssh user@host</code>	Connect to a remote system using a password (default method).
<code>ssh -i key.pem user@host</code>	Login using a private key file (<code>key.pem</code>) instead of a password.
<code>scp file user@host:/path</code>	Securely copy a file from local system to remote server (<code>scp = secure copy</code>).

<code>ssh-copy-id user@host</code>	Copies your local SSH public key to the remote server's <code>authorized_keys</code> file for passwordless login.
------------------------------------	---

4. Examples

Command	Meaning
<code>ssh gowtham@192.168.1.1 0</code>	Prompts for the password of user <code>gowtham</code> on the remote system.
<code>ssh -i ~/keys/mykey.pem ubuntu@54.234.10.11</code>	Useful for cloud servers (e.g., AWS EC2).
<code>scp report.txt</code>	Sends <code>report.txt</code> to

<code>user@192.168.1.20:/home/user/docs/</code>	the <code>/home/user/docs/</code> folder on the remote machine.
<code>ssh-copy-id student@192.168.1.25</code>	Copies your public key to the remote machine, so you won't need to enter the password again.

5. Tips

- `host` can be an IP address or domain name.
- Always secure your private key files with correct permissions:

```
chmod 400 key.pem
```
- Use `-p` to specify a port if the remote system uses a custom SSH port:

```
ssh -p 2222 user@host
```

telnet

1. Purpose

- telnet is a text-based protocol used to connect to remote systems over a network.
- Note: It is not secure—data (including passwords) is sent in plain text.

2. Use Case

- Rarely used today for actual remote login (due to lack of encryption).
- Mainly used for testing open ports and troubleshooting network services.

- Useful when checking if a server is listening on a particular port (e.g., web server on port 80).
- Connect to legacy devices or systems that still use telnet.

3. Common Commands

Command	Meaning
telnet host	Connects to a remote system using default port 23.
telnet host 80	Connects to a specific port (like 80 for HTTP, 25 for SMTP, etc.).

4. Examples

Command	Meaning
<code>telnet 192.168.1.10</code>	Attempts a connection to port 23 on the given IP.
<code>telnet example.com 80</code>	If the connection succeeds, the port is open and listening.
<code>telnet mail.server.com 25</code>	You can manually type SMTP commands to test email server responses.