1. Git archieve

The **git archive** command is used to create a tar or zip archive of a specified Git tree. This can be useful for creating snapshots of a project at a specific commit or branch without including the entire Git history.

Here's the basic syntax for **git archive**:

Where:

* **<format>** specifies the format of the archive (e.g., tar, zip).
* **<file>** specifies the name of the output archive file.
* **<branch/commit>** specifies the branch or commit to create the archive from.

For example, to create a tar archive of the **main** branch and save it as **project.tar**, you would run:

Similarly, you can create a zip archive by specifying **--format=zip**.

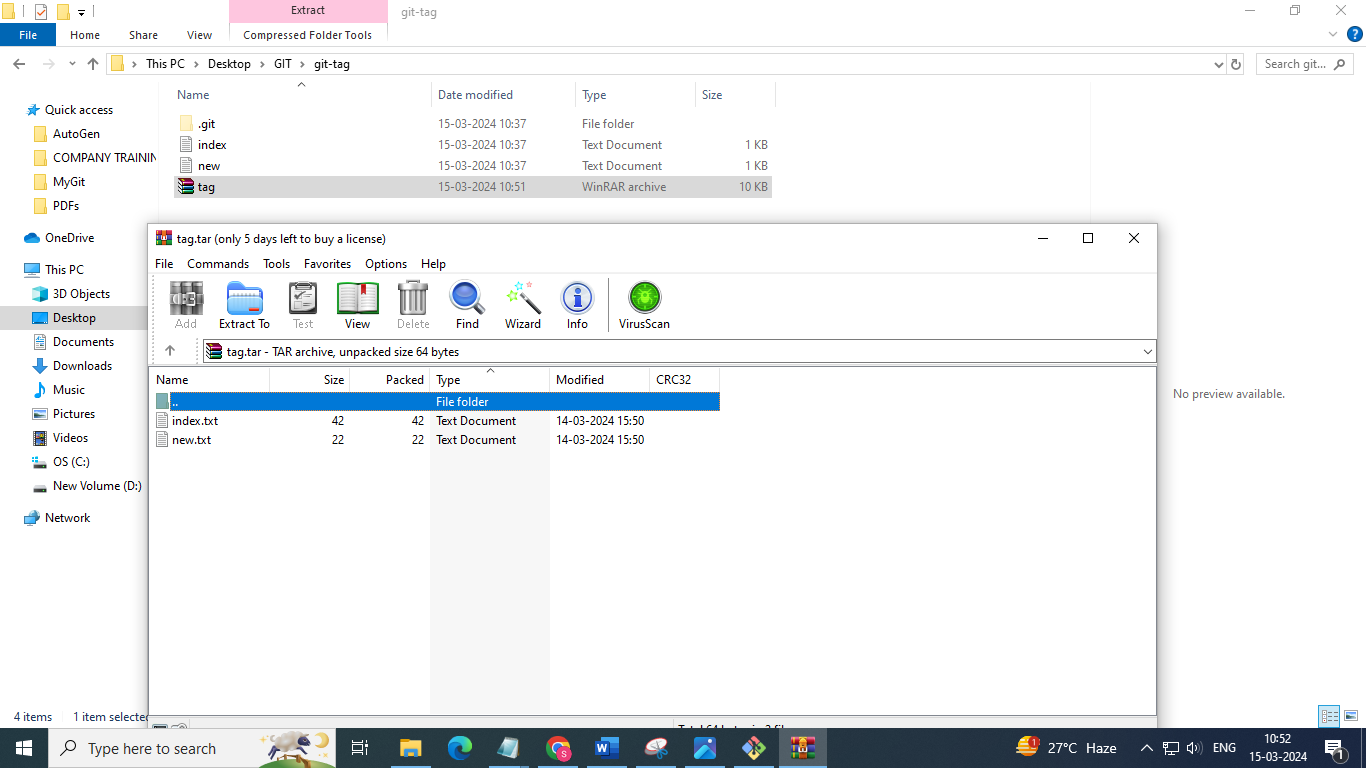
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)

$ git archive --format=tar --output=tag.tar master

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)

$ ls

index.txt new.txt tag.tar



The **git init** command is used to initialize a new Git repository in the current directory or in a specified directory. When you run **git init**, Git creates a new subdirectory named **.git** that contains all of the necessary files for your repository, including the repository's configuration, object database, and references to commits.

Here's how you can use **git init**:

1. **Initialize a New Repository in the Current Directory:** If you want to start tracking changes in an existing project or create a new project, navigate to the project's directory in your terminal or command prompt and run:

git init

1. **Initialize a New Repository in a Specific Directory:** If you want to initialize a Git repository in a directory other than the current one, specify the directory path after **git init**: