



**St. JOSEPH'S**  
**GROUP OF INSTITUTIONS**  
OMR, CHENNAI - 119



## **Placement Empowerment Program**

### ***Cloud Computing and DevOps Centre***

**Create a Simple Backup Script: Create a script that backs up your entire Git repository to a local folder daily**

**Name: ROHITH S**

**Department: INFORMATION TECHNOLOGY**



## Introduction

Backing up your Git repositories is essential for safeguarding your work and maintaining version control integrity. It protects against unexpected data loss caused by accidental deletions, hardware failures, or repository corruption. Automating the backup process not only ensures regular updates but also saves time and minimizes manual effort, providing a reliable and hassle-free solution.

## Overview

This guide walks you through setting up an automated backup system for your Git repository on a Windows machine using a batch script and Task Scheduler. The script fetches the latest updates from the repository daily and saves them in a designated backup folder. Additionally, it compresses the repository into a timestamped archive for better organization. This ensures that your codebase and version history are securely stored in a local directory, minimizing the risk of data loss.

## Key Components

- **Batch Script:** A .bat file executes commands to clone the repository, pull the latest updates, and compress the backup for efficient storage.
- **Task Scheduler:** A built-in Windows utility automates the script, ensuring it runs daily without requiring manual execution.

## Objectives

- 1. Automate Backups:** Develop a script to back up the entire Git repository daily.
- 2. Minimize Data Loss:** Safeguard the repository from accidental deletions or hardware failures.
- 3. Ease of Management:** Create timestamped backups for quick identification and restoration.
- 4. Hands-Free Automation:** Leverage Task Scheduler to eliminate the need for manual execution.

## Importance

- **Disaster Recovery:** In case of repository failures or accidental deletions, you can quickly restore your work from the local backup.
  - **Version History Preservation:** All changes and version history are secured, ensuring no progress is lost.
  - **Efficient Workflow:** Automating the process allows you to focus on development tasks instead of managing backups manually.
  - **Organization:** Timestamped backups provide a clear, structured way to keep track of changes over time.
- 

## Step-by-Step Overview

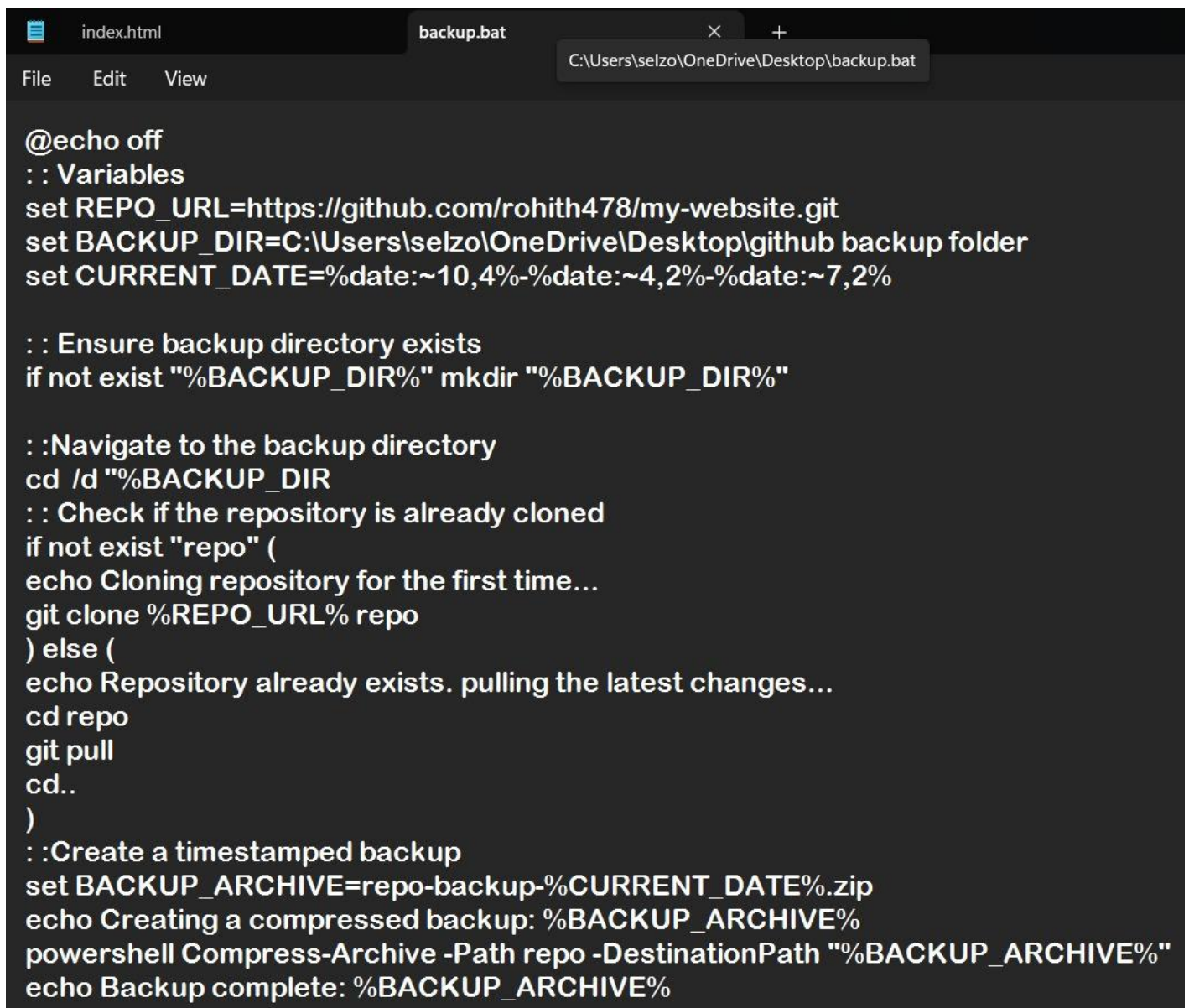
### Step 1:

Create a folder named **GitHub Backup Folder** to store your backup files.



### Step 2:

- Open **Notepad** and type the following script:  
Set REPO\_URL to the URL of the repository you want to back up.  
Set BACK\_DIR to the path of the folder created in Step 1.
- Save the file with a .bat extension (e.g., backup.bat) on your **Desktop**.

A screenshot of a Windows File Explorer window. The address bar shows the path 'C:\Users\selzo\OneDrive\Desktop\backup.bat'. The file name 'backup.bat' is in the top tab. The main area displays the contents of the batch file in a dark-themed font. The script starts with '@echo off' and ': : Variables', followed by setting environment variables for REPO\_URL, BACKUP\_DIR, and CURRENT\_DATE. It then checks if the backup directory exists and creates it if not. Next, it navigates to the backup directory and checks if the repository is already cloned. If not, it clones the repository; otherwise, it pulls the latest changes. Finally, it creates a timestamped backup using PowerShell's Compress-Archive command and echoes the completion message.

```
@echo off
: : Variables
set REPO_URL=https://github.com/rohith478/my-website.git
set BACKUP_DIR=C:\Users\selzo\OneDrive\Desktop\github backup folder
set CURRENT_DATE=%date:~10,4%-~%date:~4,2%-~%date:~7,2%

: : Ensure backup directory exists
if not exist "%BACKUP_DIR%" mkdir "%BACKUP_DIR%"

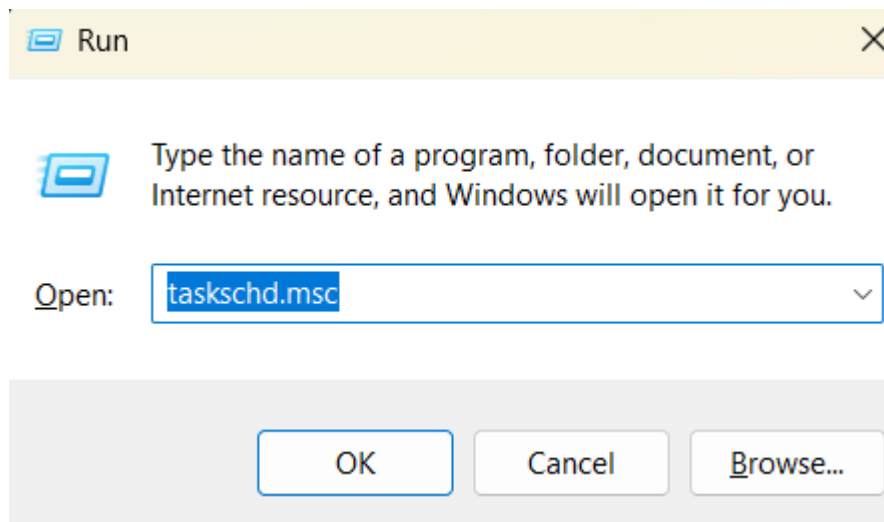
: : Navigate to the backup directory
cd /d "%BACKUP_DIR"

: : Check if the repository is already cloned
if not exist "repo" (
    echo Cloning repository for the first time...
    git clone %REPO_URL% repo
) else (
    echo Repository already exists. pulling the latest changes...
    cd repo
    git pull
    cd..
)

: : Create a timestamped backup
set BACKUP_ARCHIVE=repo-backup-%CURRENT_DATE%.zip
echo Creating a compressed backup: %BACKUP_ARCHIVE%
powershell Compress-Archive -Path repo -DestinationPath "%BACKUP_ARCHIVE%"
echo Backup complete: %BACKUP_ARCHIVE%
```

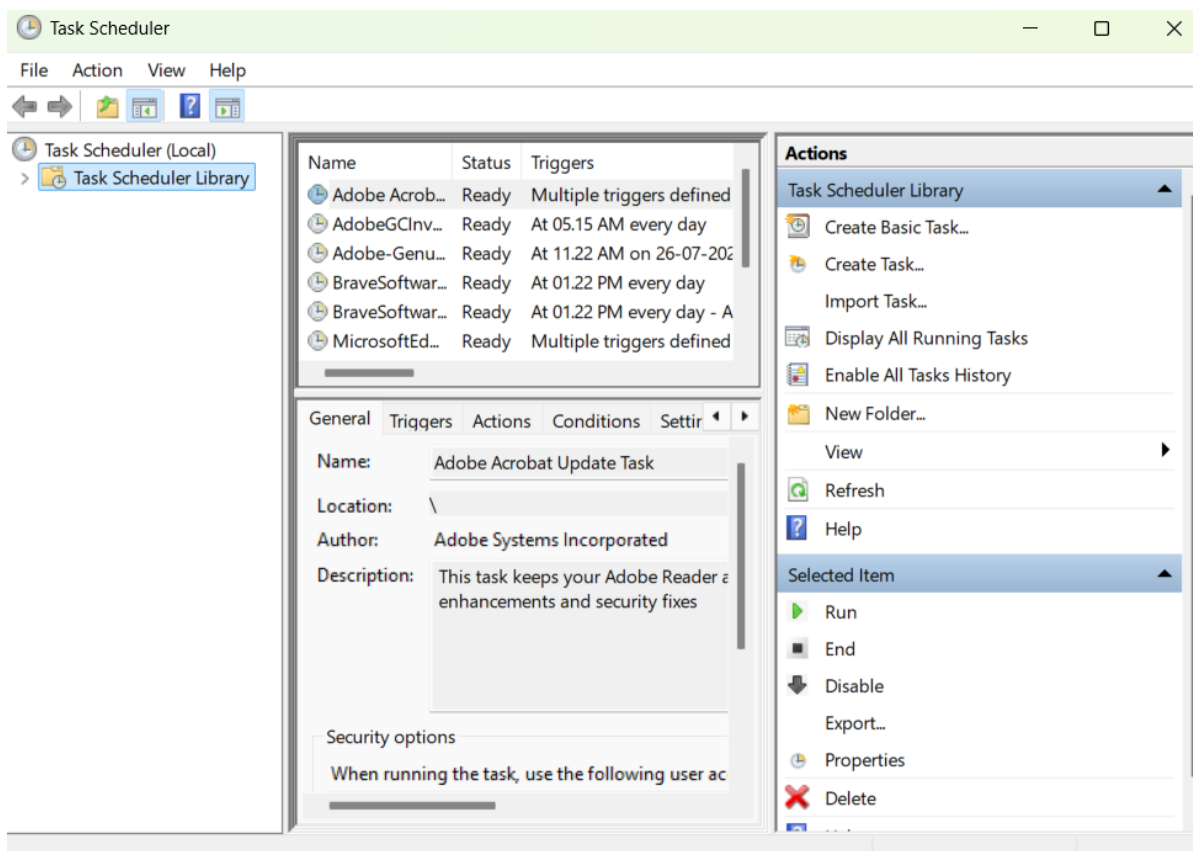
### Step 3:

- Press **Win + R**, type `taskschd.msc`, and press **Enter**.
- The **Task Scheduler** window will open.



#### Step 4:

- Click "**Create Basic Task**" on the right-hand panel.
- A wizard will guide you through the setup.




#### Step 5:

- Enter a **name** (e.g., GitRepoBackup).
- Optionally, add a **description** (e.g., *"Backs up files daily"*).
- Click **Next**.

Create Basic Task Wizard

X



Create a Basic Task

Create a Basic Task

Trigger

Daily

Action

Start a Program

Finish

Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.

Name:

gitrepobackup

Description:

backup files daily

< Back

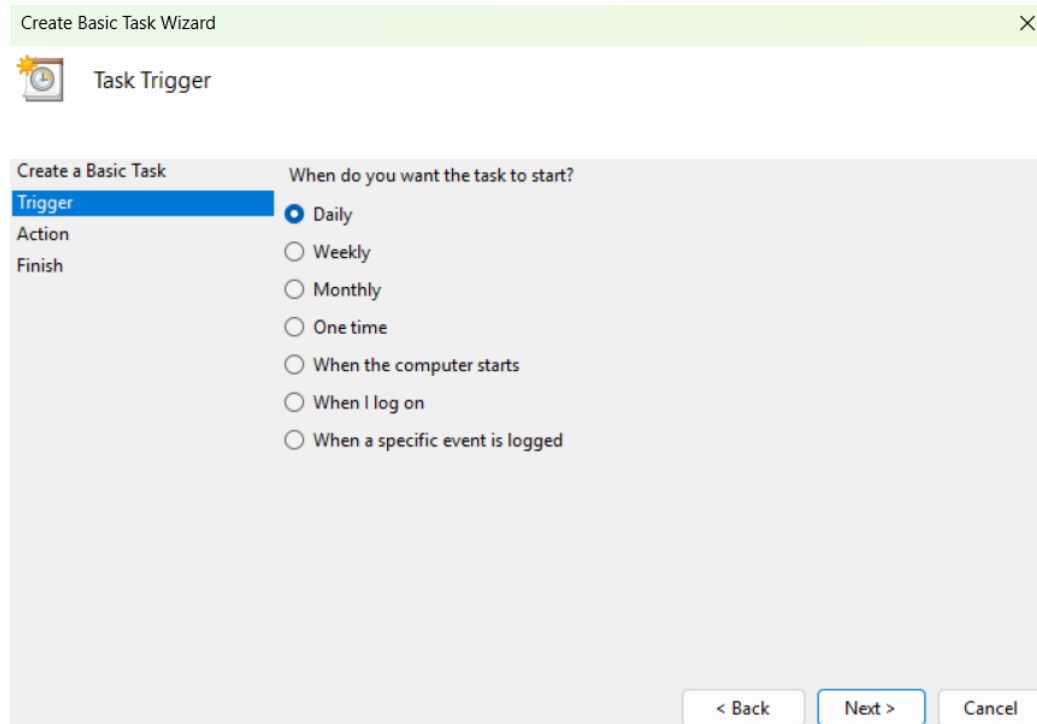
Next >

Cancel

## Step 6:

- Select how often the task should run:
  - **Daily** (recommended)
  - **Weekly**
  - **One-time**

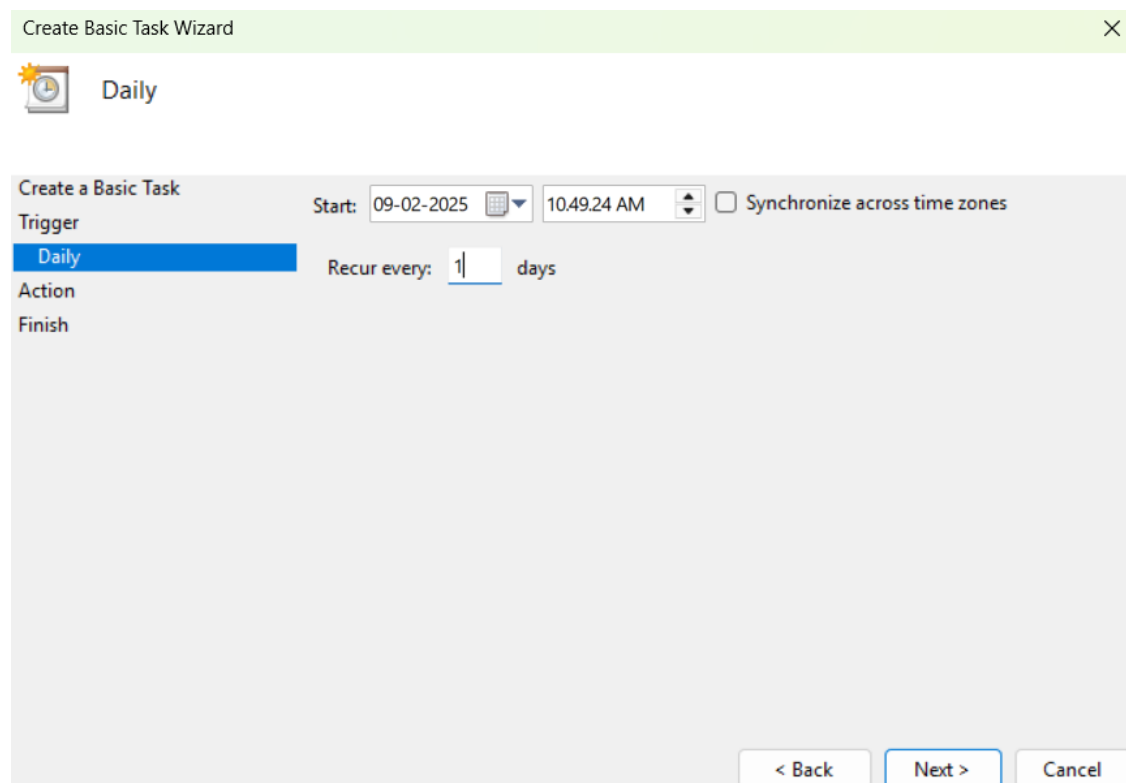
- Click **Next**.



The screenshot shows the 'Create Basic Task Wizard' window. The title bar is green and says 'Create Basic Task Wizard' with a close button. Below the title bar is a tab labeled 'Task Trigger' with a clock icon. The main area is titled 'Create a Basic Task' and has a left sidebar with 'Trigger', 'Action', and 'Finish'. The 'Trigger' section is selected. The right side is titled 'When do you want the task to start?' and lists seven options with radio buttons: 'Daily' (selected), 'Weekly', 'Monthly', 'One time', 'When the computer starts', 'When I log on', and 'When a specific event is logged'. At the bottom right are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

### Step 7:

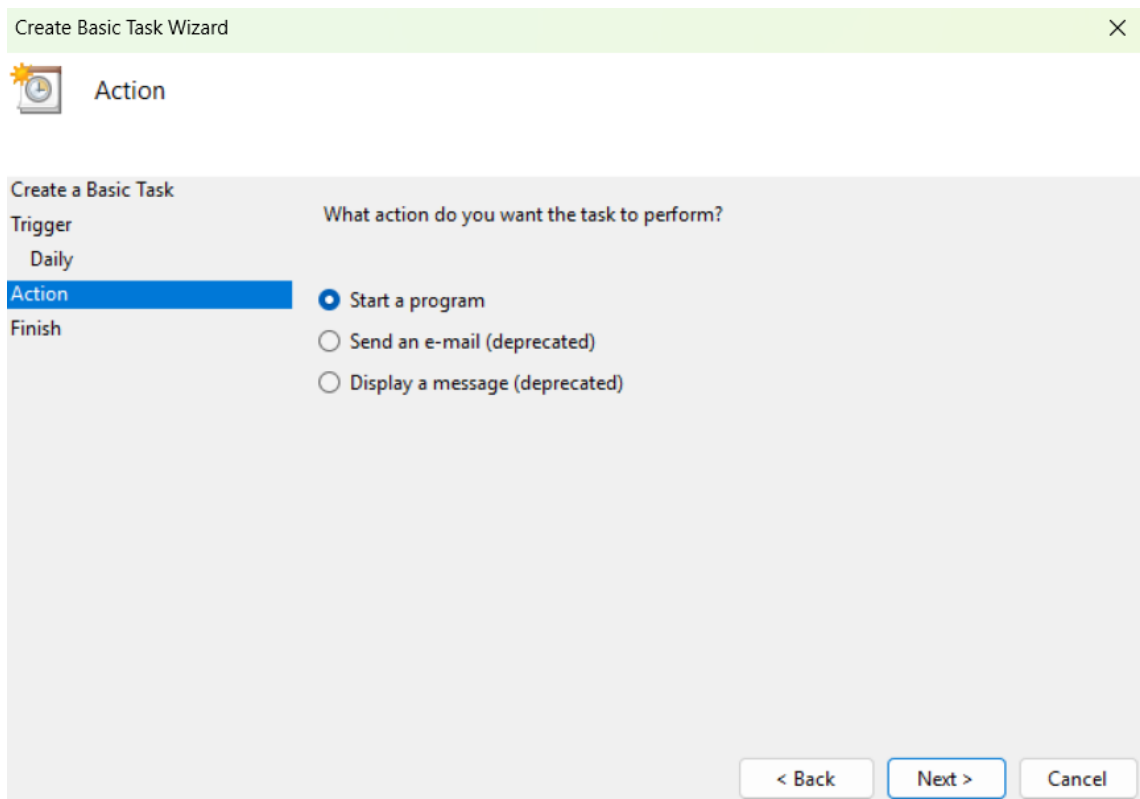
- If you chose **Daily**, specify:
  - The **start date** (defaults to today).
  - The **time** (e.g., **06:20 PM**).
- Click **Next**.



The screenshot shows the 'Create Basic Task Wizard' window at the 'Daily' step. The title bar is green and says 'Create Basic Task Wizard' with a close button. Below the title bar is a tab labeled 'Daily' with a clock icon. The main area is titled 'Create a Basic Task' and has a left sidebar with 'Trigger', 'Action', and 'Finish'. The 'Trigger' section is selected and labeled 'Daily'. The right side shows the configuration for a daily task: 'Start:' with a date picker set to '09-02-2025' and a time spinner set to '10:49:24 AM', and a checkbox for 'Synchronize across time zones' which is unchecked. Below this is 'Recur every: 1 days'. At the bottom right are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

## Step 8:

Select "**Start a Program**" and click **Next**.




## Step 9:

- In the **Program/script** field, click **Browse** and locate your .bat file (e.g., backup.bat on the Desktop).
- Click **Next**.



Create Basic Task Wizard

 Start a Program

Create a Basic Task

Trigger  
Daily

Action  
**Start a Program**

Finish

Program/script:  
C:\Users\selzo\OneDrive\Desktop\backup.bat Browse...

Add arguments (optional):


Start in (optional):

< Back   Next >   Cancel

## Step 10:

Click **Finish** to save and schedule the task.

Create Basic Task Wizard

 Summary

Create a Basic Task

Trigger  
Daily

Action  
Start a Program

**Finish**

Name: gitrepobackup

Description: backup files daily

Trigger: Daily; At 10:53 every day

Action: Start a program; C:\Users\selzo\OneDrive\Desktop\backup.bat

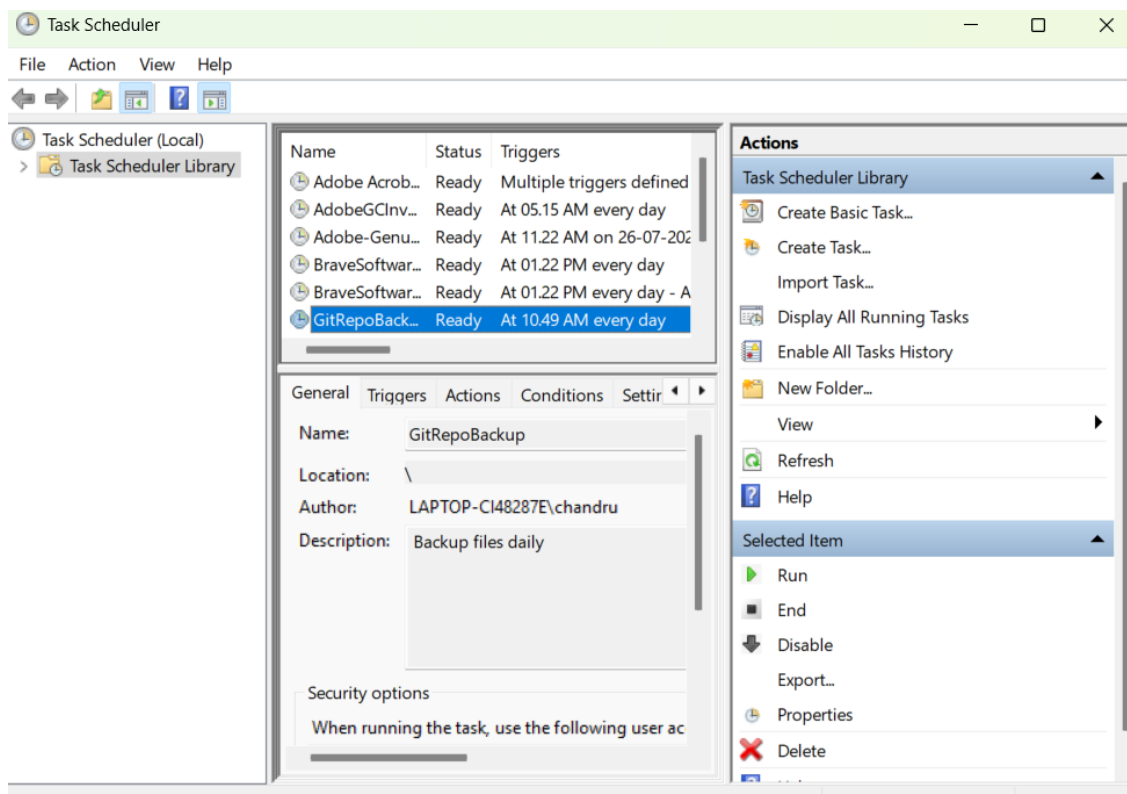
☐ Open the Properties dialog for this task when I click Finish  
When you click Finish, the new task will be created and added to your Windows schedule.

< Back   **Finish**   Cancel

## Step 11:

- In **Task Scheduler**, go to **Task Scheduler Library** (left panel).

- Find your task (GitRepoBackup), right-click it, and select **Run**.



```

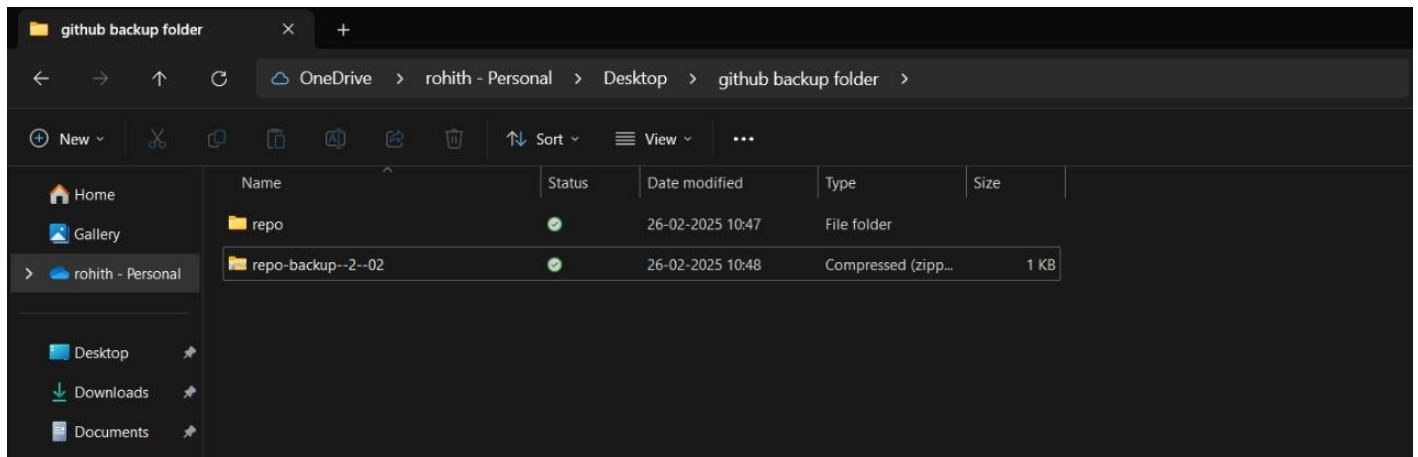
+  v
Cloning repository for the first time...
Cloning into 'repo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
Creating a compressed backup: repo-backup-2025-01-24.zip
Backup complete: repo-backup-2025-01-24.zip

[process exited with code 0 (0x00000000)]
You can now close this terminal with Ctrl+D, or press Enter to restart.

```

## Step 12:

- Open the **GitHub Backup Folder** created in Step 1.
- It should now contain the files from your repository.



## Outcome

By completing this Proof of Concept (PoC) of automating Git repository backups, you will:

- **Successfully implement a backup system for Git repositories:** Automate the process of creating daily backups for your Git repositories, ensuring that all updates and changes are securely stored in a local folder.
- **Master the use of batch scripting for task automation:** Learn to create and execute a .bat script that clones, pulls updates, and compresses a Git repository into timestamped backup archives.
- **Understand Task Scheduler's automation capabilities:** Gain practical experience with Task Scheduler, learning how to set triggers, define actions, and configure conditions to automate repetitive tasks seamlessly on a Windows system.