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: DHARSHINI P DEPARTMENT: IT

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

Backup is an essential part of software development, ensuring that your data remains safe in case of accidental deletion, corruption, or hardware failure. Automating backups for your Git repositories can help maintain data integrity and prevent loss. This guide provides a simple script that automatically backs up a Git repository to a local folder daily.

Overview

This script will:

- Create a backup of a specified Git repository.
- Store the backup in a designated local folder.
- Run daily using a task scheduler (e.g., Windows Task Scheduler or cron job in Linux/macOS)

Objective

The primary objectives of this POC are:

The goal is to automate the backup process so that your Git repository is securely stored in a separate location daily, reducing the risk of data loss

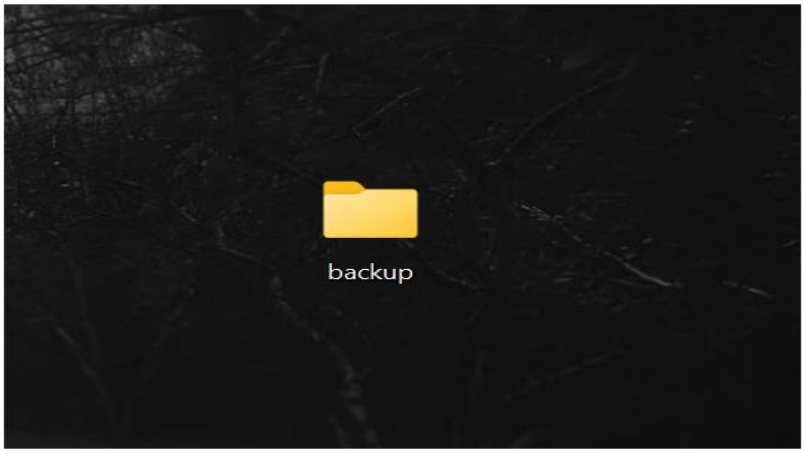
Importance

- Prevents accidental loss of data.
- Allows rollback to previous versions.
- Ensures code availability in case of system failures.
- Facilitates smooth project recovery

Step-by-Step Overview

Step 1:

Create a folder named GitHub Backup Folder to store your Backup files



Step 2:

Create a script:

Open Notepad and type this script

2.html | from1.txt | copyscript.b | from1.txt | from2.txt | from2.txt | feature.txt | #!bink

File Edit View

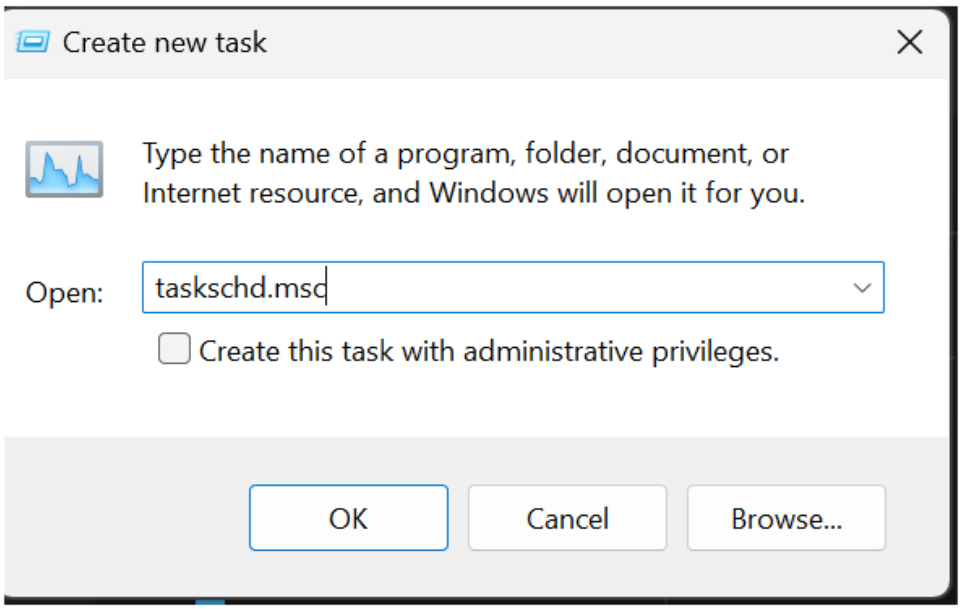
```
#!/bin/bash

# Configuration
REPO_PATH=https://github.com/Dharshini-punniyamoorthi/my-static-website  # Change this to your Git repo path
BACKUP_DIR=C:\Users\Dell\Desktop\backup  # Change this to your backup directory
TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")
BACKUP_NAME="repo_backup_$TIMESTAMP.tar.gz"

# Create backup directory if not exists
mkdir -p "$BACKUP_DIR"

# Navigate to the repository
cd "$REPO_PATH" || { echo "Repository not found!"; exit 1; }

# Ensure the repository is up to date
git fetch --all
git pull

# Create a compressed archive of the repository
tar -czf "$BACKUP_DIR/$BACKUP_NAME" .

echo "Backup completed: $BACKUP_DIR/$BACKUP_NAME"
```

- **REPO_URL**: URL of the repository you want to backup
- **BACK_DIR:** give the file path of the folder which you created
- save it as **.bat format** (eg:backup.bat) in Desktop

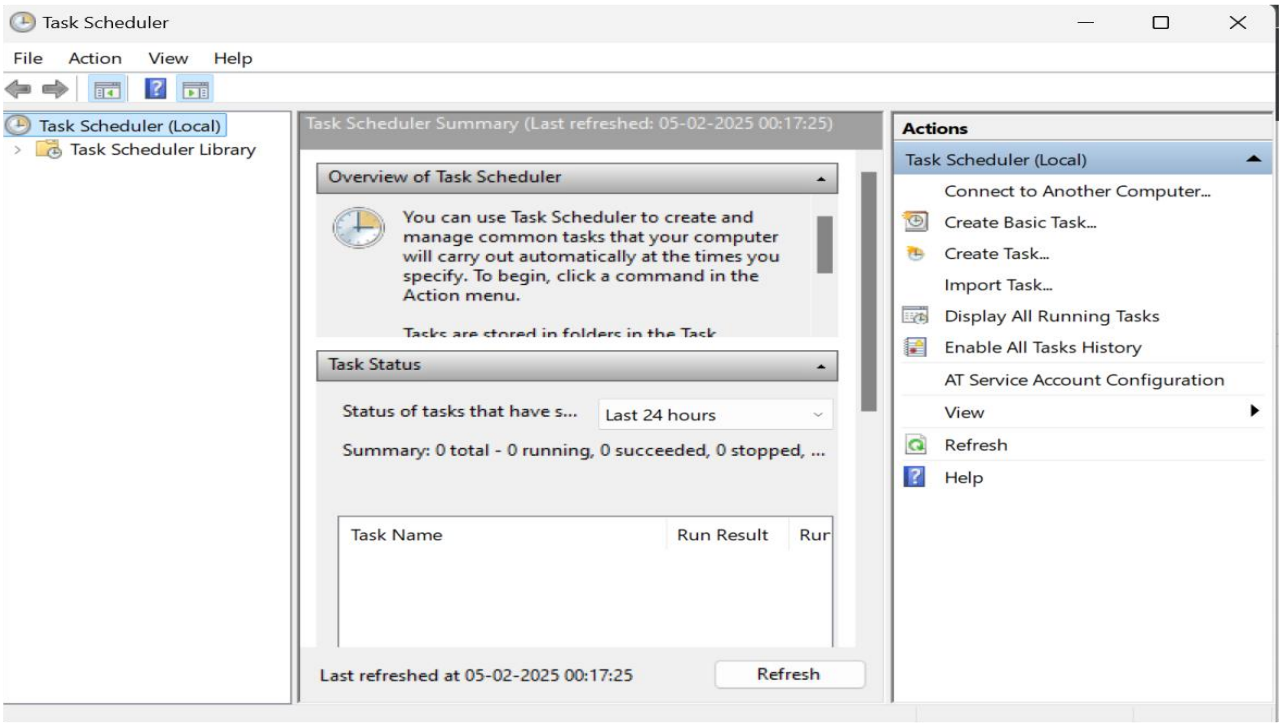
Step 3:

Open task manager

- Type taskschd.msc (without quotes) in the Run box. Press Enter or click OK.This will open the Task Scheduler window

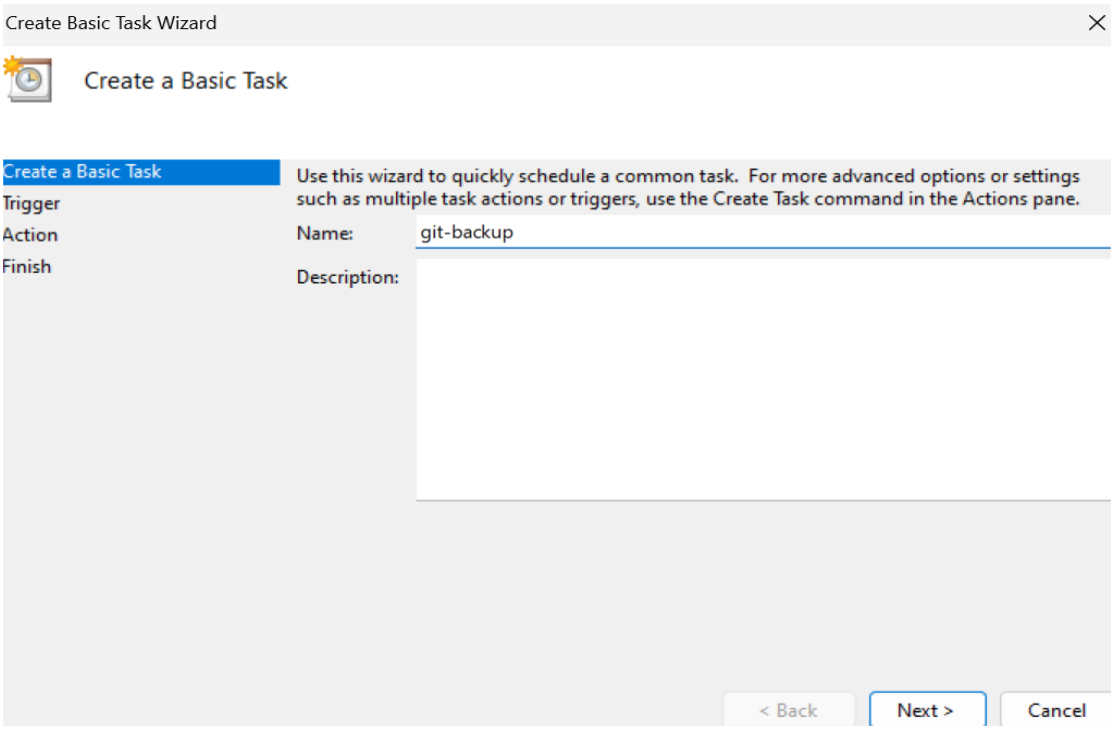


- In the Task Scheduler window, look to the right-hand side for a button called "Create Basic Task".
  Click it



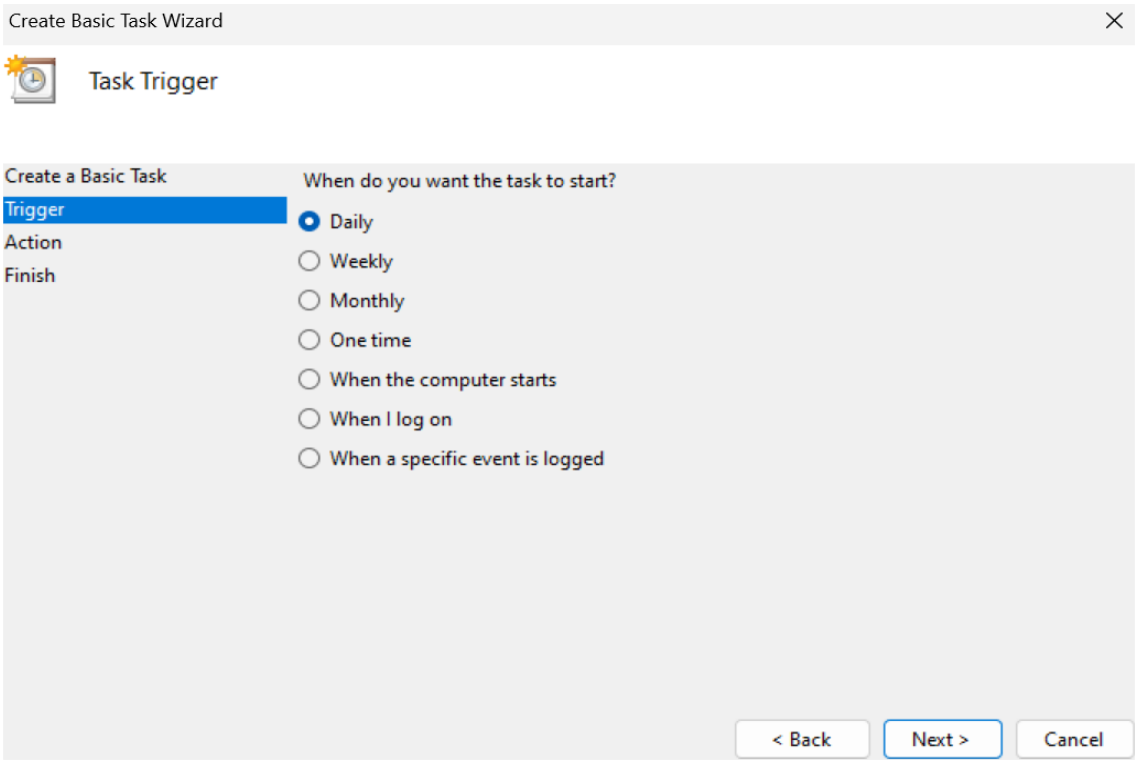
- Enter a Name for the Task:

  Click Next to continue

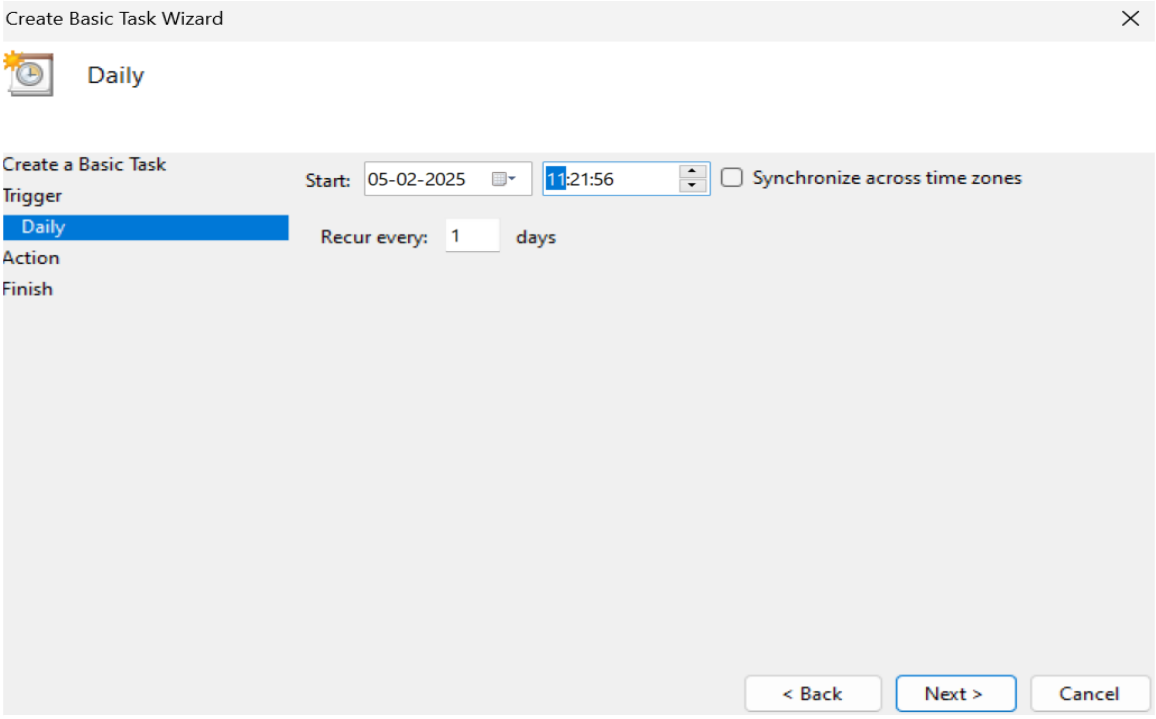


- Choose a Schedule:

Choose what works for you (e.g., Daily) and click Next.

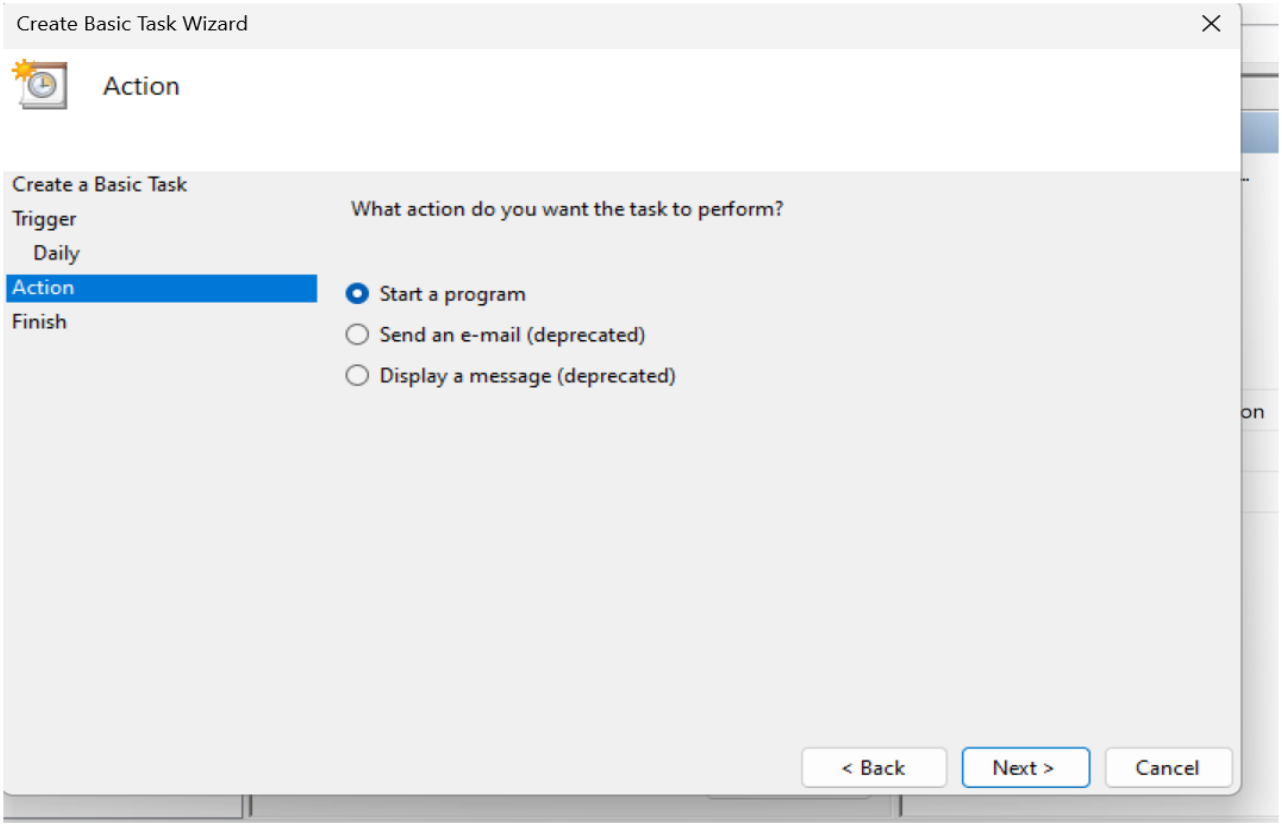


- Set the Time and Frequency:



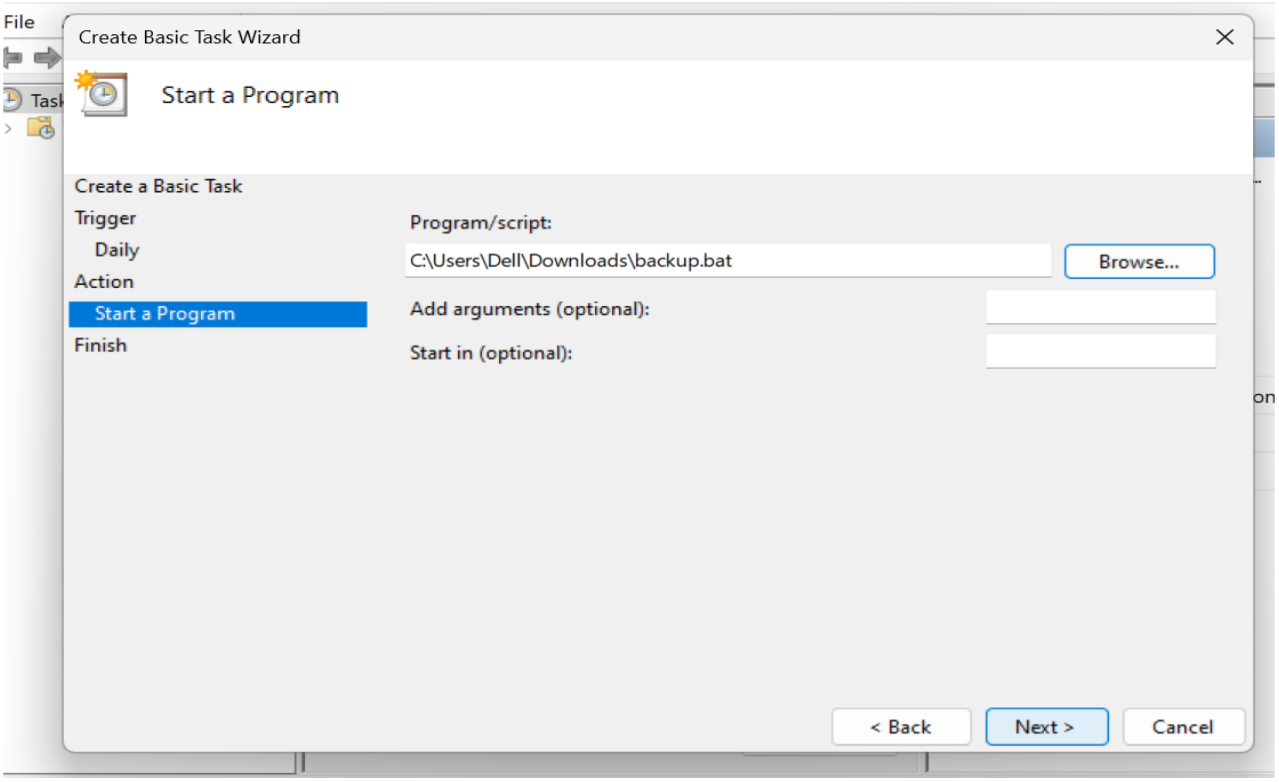
- Set the Action Now:

  Select **"Start a Program"**.



- Point to the Program or Script

  Browse and navigate to the location of **your .bat file.**

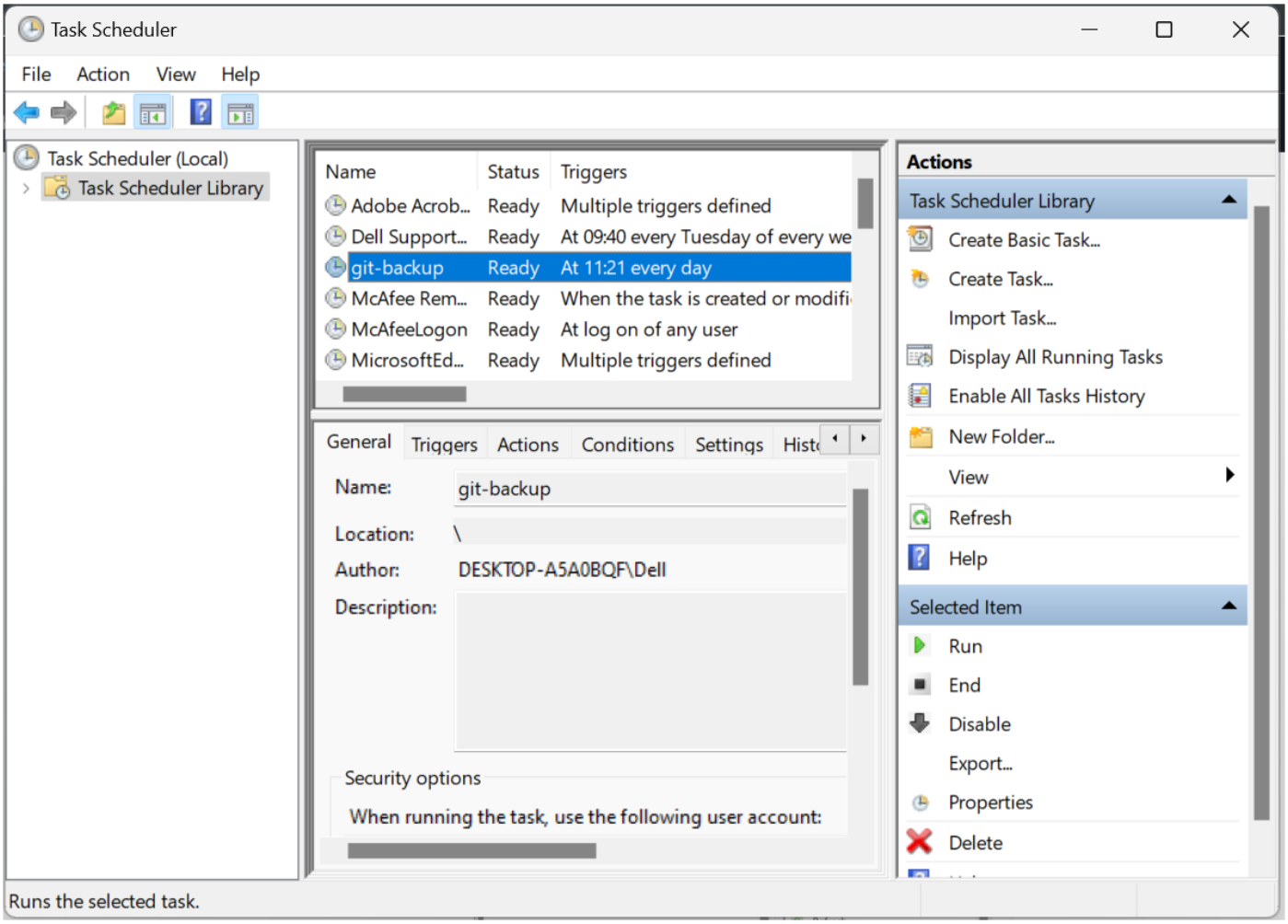


- Review and Finish Click Finish to save and schedule the task.

Step 4:

manually trigger the task immediately

- go to the Task Scheduler Library (on the left hand side).
- Find your task (it should have the name you gave it, e.g., "GitRepoBackup").
- Right-click the task and select Run. This will manually trigger the task immediately



Expected Outcome:

By the end of this process, you will have:

- The script will run daily, creating a timestamped backup of the Git repository.
- The backups will be stored in the specified directory.
- In case of data loss, you can restore the latest backup to recover your repository.

By implementing this script, you ensure a consistent and automated backup system, protecting your work from unexpected failures.