

**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

Name: MADHUMETHRA AP

Department: IT



Introduction

Maintaining regular backups of critical data is essential for ensuring data integrity, protecting against accidental deletions, and mitigating risks posed by system failures. This is particularly important for Git repositories, which often store valuable source code, configuration files, and project history. By implementing a simple backup script, developers can automate the process of creating daily backups of their Git repositories, ensuring they always have an up-to-date and secure copy of their work.

Overview

The goal of this script is to create a simple, automated solution to back up a Git repository to a designated local folder daily. The script will compress or copy the repository and organize backups with a clear naming convention based on the current date. The solution will use platform-specific tools—tar for Linux and robocopy for Windows—and can be automated using scheduling tools like cron or Task Scheduler. This ensures seamless operation without manual intervention.

Objectives

Automate Repository Backups:

Develop a script that eliminates the need for manual backups, saving time and reducing the risk of human error.

Ensure Backup Integrity:

The script will validate the success of the backup process, providing logs or notifications to indicate whether the operation was successful.

Daily Backup Organization:

Use date-based naming conventions for backup files or folders, making it easy to track changes and locate backups from specific days.

Platform Compatibility:

Create platform-specific solutions:

Linux: Utilize tar for compressing and archiving the repository.

Windows: Leverage robocopy for efficient file copying.

Automate Scheduling:

Enable daily execution of the backup

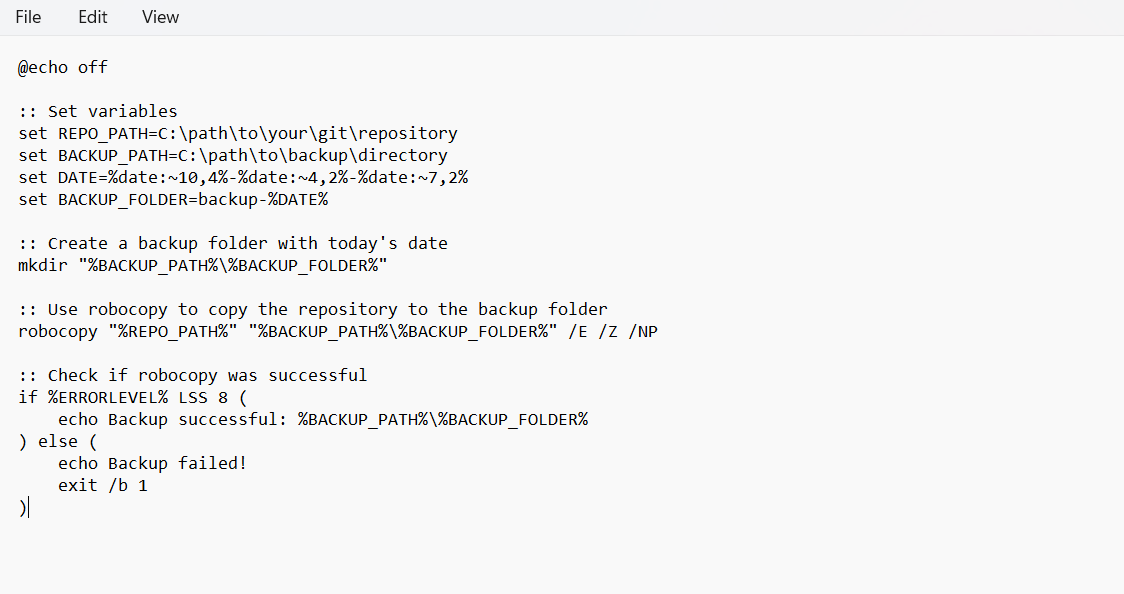
**Step-by-Step Overview**

Step 1

### Set variables:

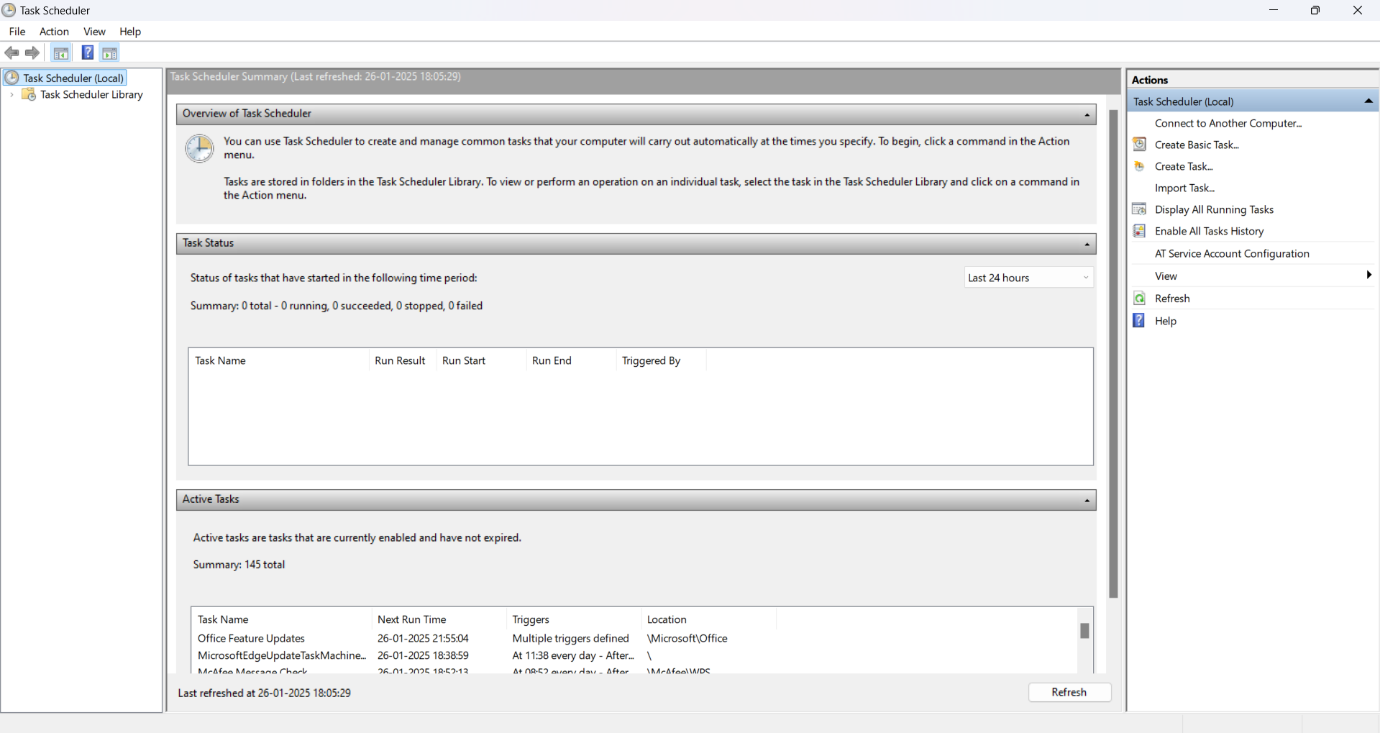
### \* REPOSITORY\_PATH: Replace with the actual path to your Git repository.

### \* BACKUP\_DIR: Replace with the desired path for the backup directory.



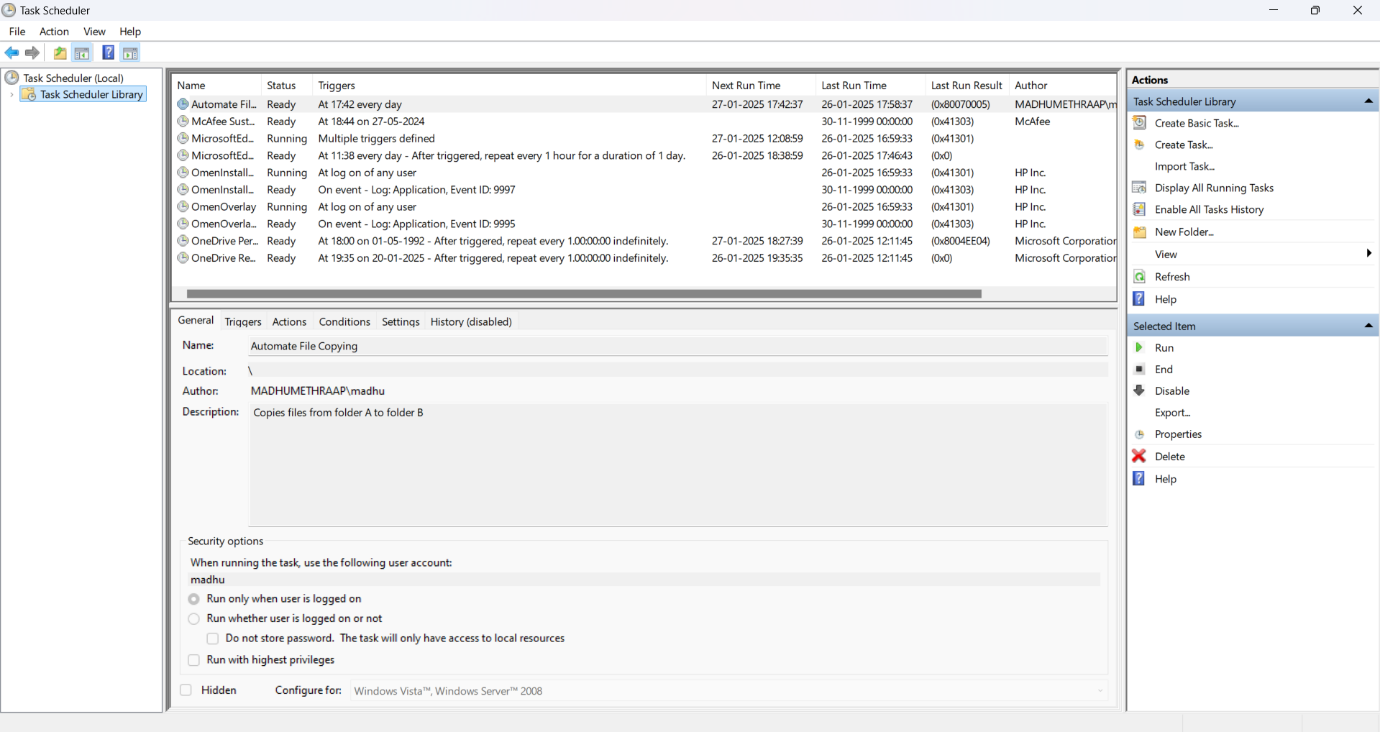
Step 2

Set a task scheduler and work with the task scheduler



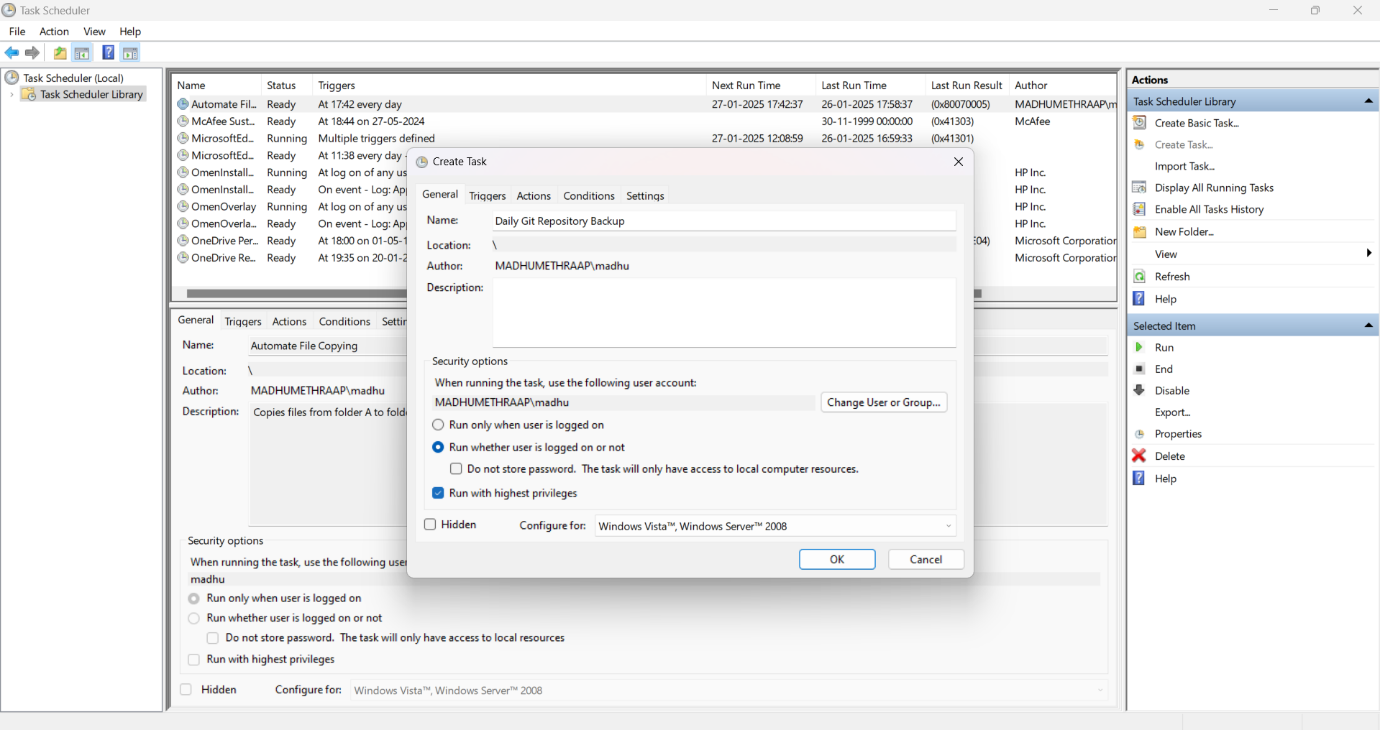
Step 3

Go to the task scheduler library for further task



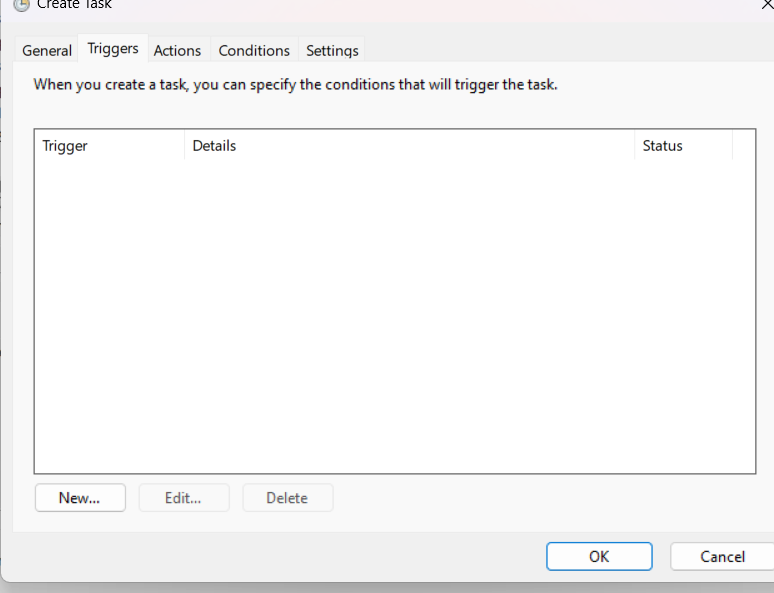
Step 4

Make sure the general settings are enabled properly



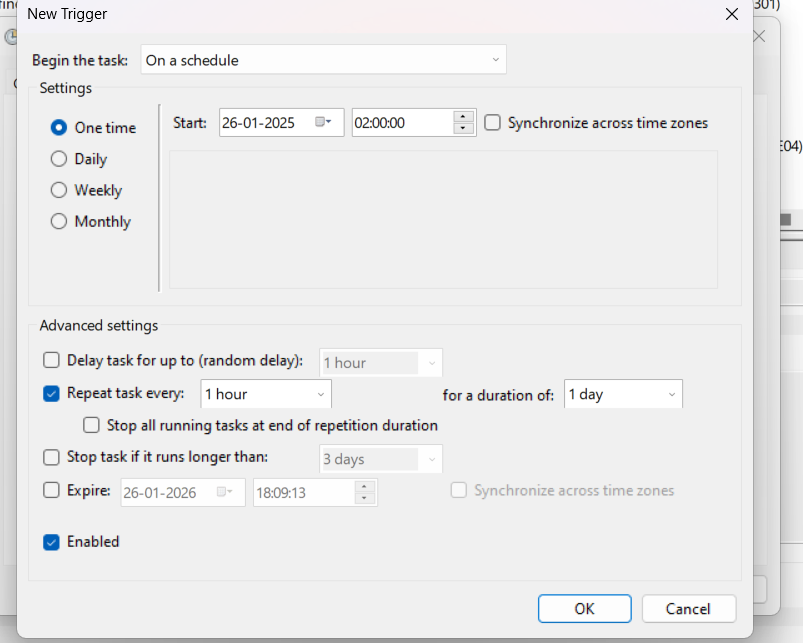
Step 5

Select the trigger and click new and go with the ok button clicking it



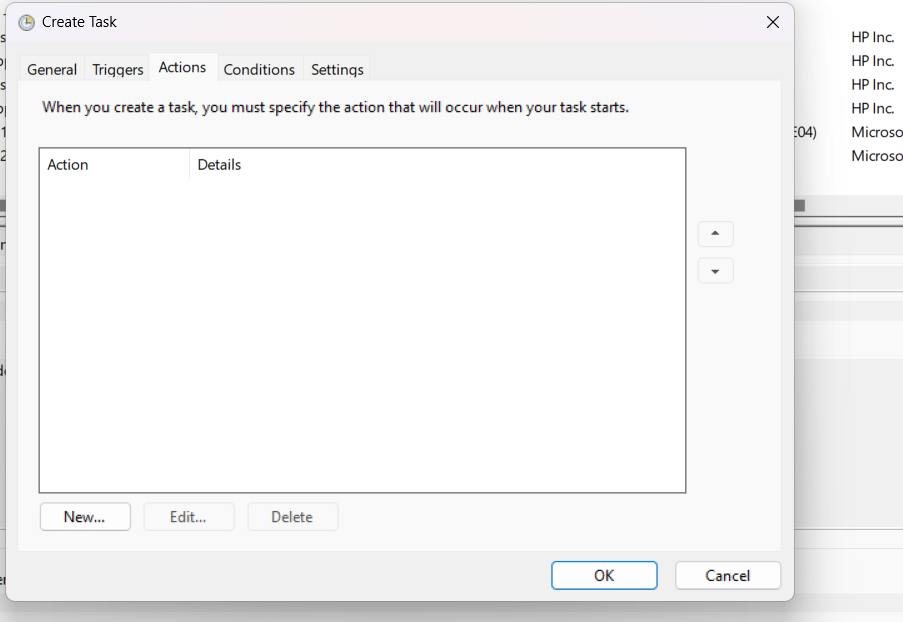
Step 6

Schedule the task to begin the task by choosing the choice of one time,daily,weekly,monthly



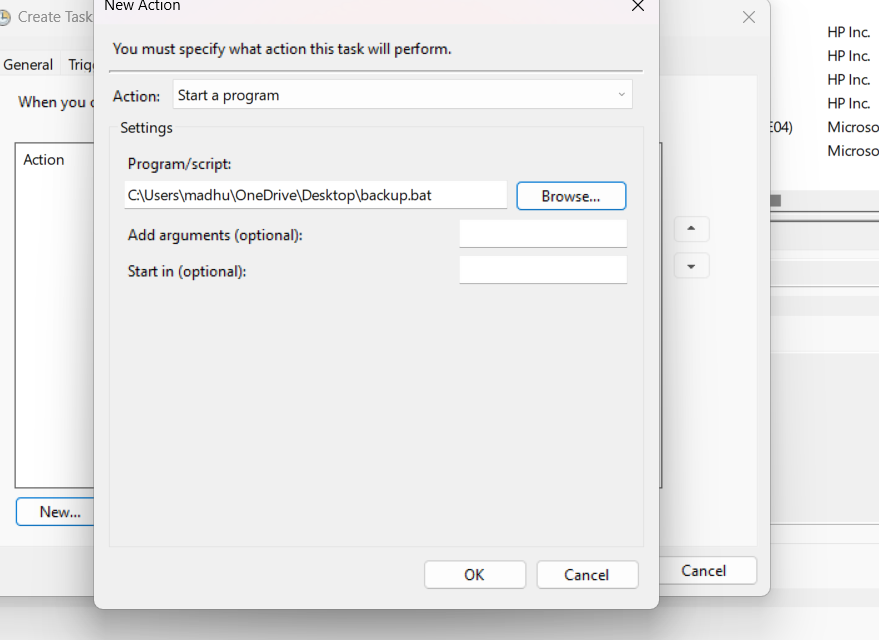
Step 7

Click the action



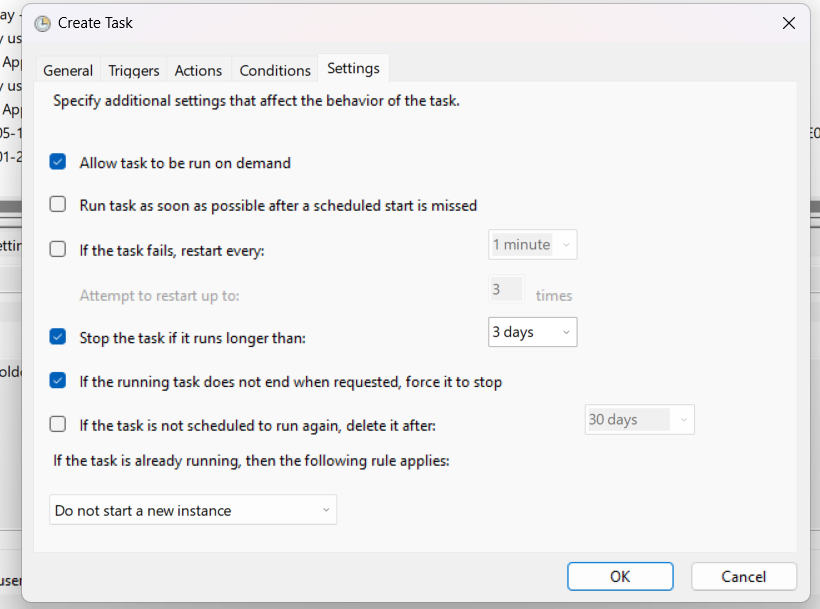
Step 8

In action give program script



Step 9

Give the settings correctly and then click ok



Output

Verify the backup file and bundle exist in the destination directory.

- Check the backup file size and modification date to ensure it matches your expectations.

- Use git bundle verify to verify the integrity of the backup bundle.