



# Placement Empowerment Program Cloud Computing and DevOps Centre

Create a Simple Backup Script

Name: Oviya G Department : IT



# Introduction

In the world of software development , safeguarding your work is crucial. One way to ensure your valuable codebase is always protected is through regular backups. This tutorial will guide you through the process of creating a simple yet effective backup script for your Git repository . The script will be designed to run daily , automatically saving a copy of your repository to a

specified local folder. By following these steps, you can ensure that your code is securely backed up and easily recoverable in case of any unforeseen issues. Let 's dive in and set up a robust backup system to keep your projects safe and sound!

# Objectives

Automate Backup Process: Create a script that automates the backup of specified files or directories to a designated backup location.

Minimize Human Error: Reduce the risk of forgetting to back up important data by automating the process.

Ease of Use: Ensure the script is easy to run, either manually or via a scheduled task (e.g., cron job).

Logging: Implement logging to keep track of backup operations, including successes and failures.

Compression: Optionally compress the backup files to save storage space.

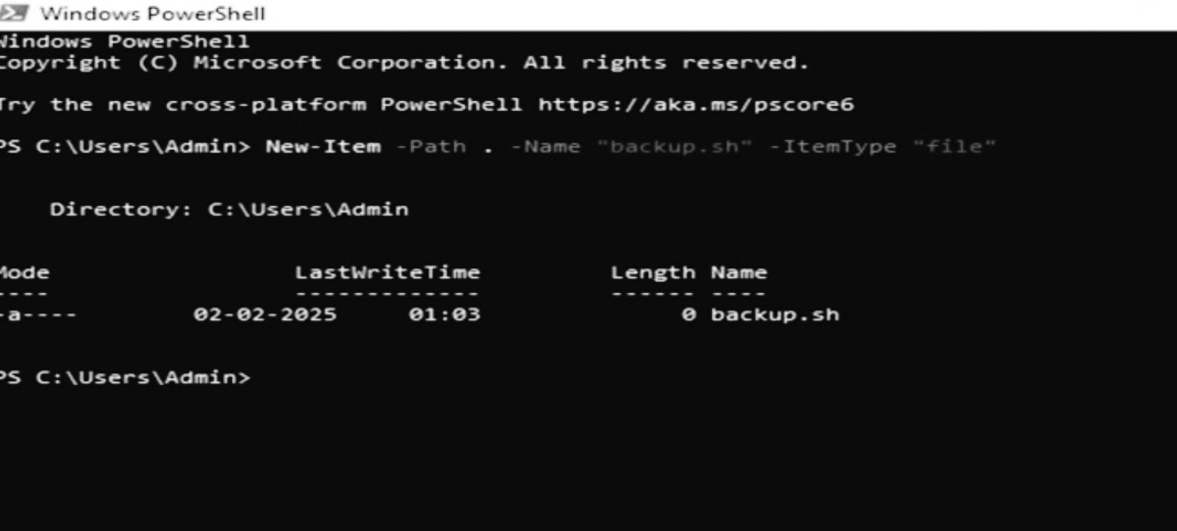
Incremental Backup: Optionally support incremental backups to only back up changes since the last backup.project ,

# Step- by - Step Overview Step1:

Create the Script File: Open your terminal and create a new file

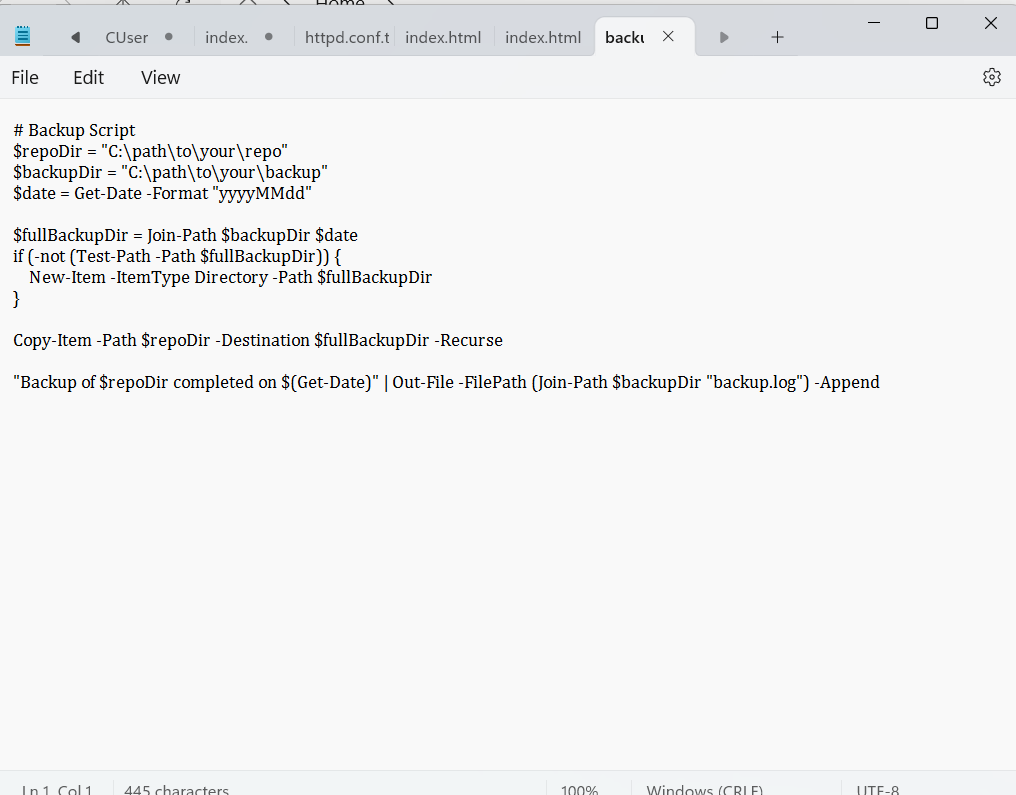
for your backup script . Let 's call it backup.sh.

* New-Item -Path . -Name “backup.sh” -ItemType “file”



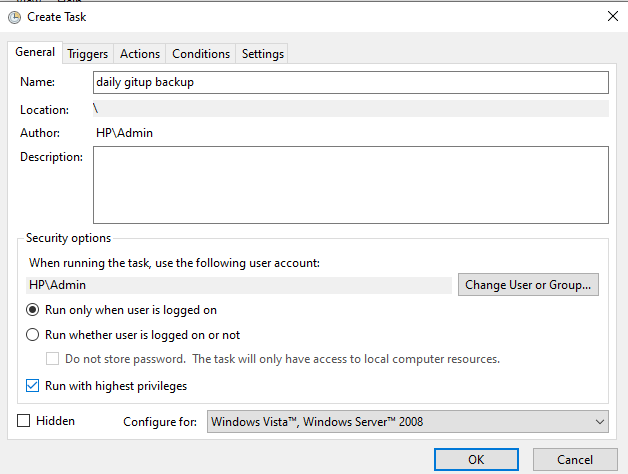
# Step 2

Open the file in your preferred text editor, such as Notepad



# Step 3

Schedule the Script with Task Scheduler.



# Step 4

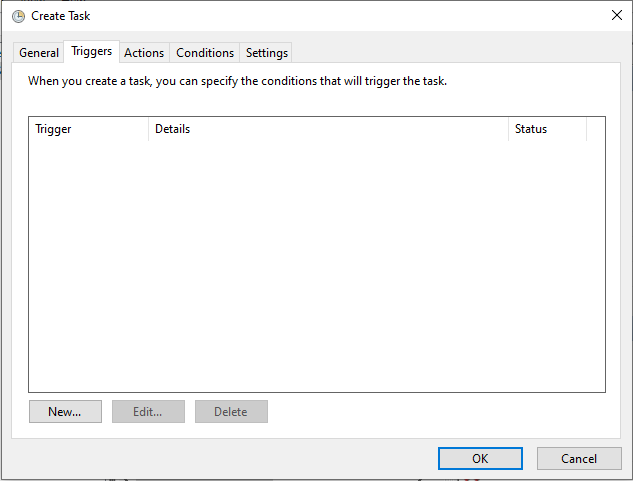
Triggers Tab:

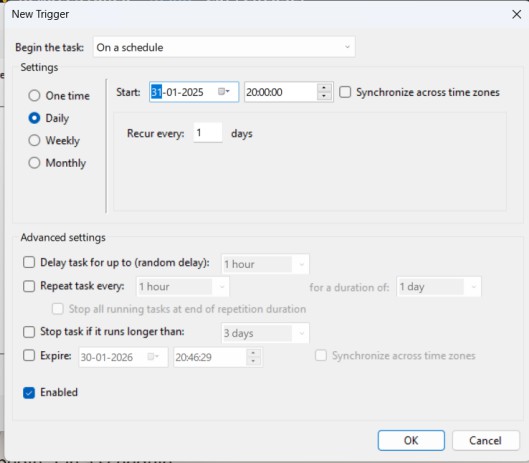
Click on "New...".

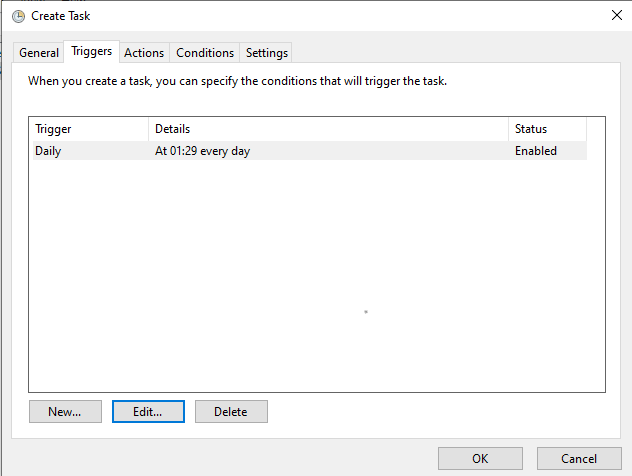
Set the task to begin "On a schedule".

Choose "Daily " and set the time you want the script to run (e.g., 2:00 AM).

Click "OK".

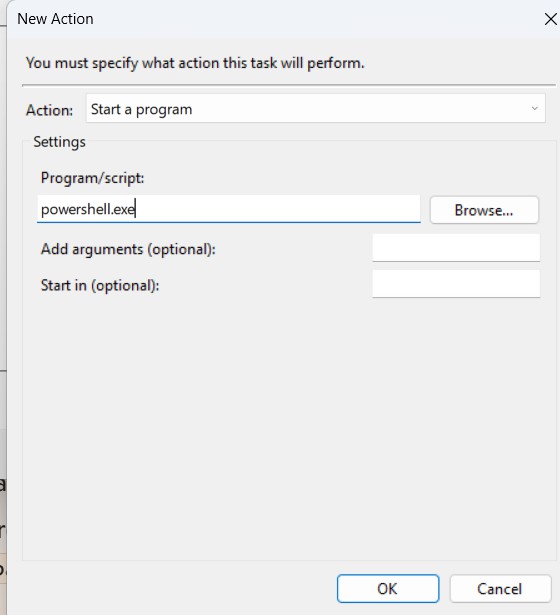






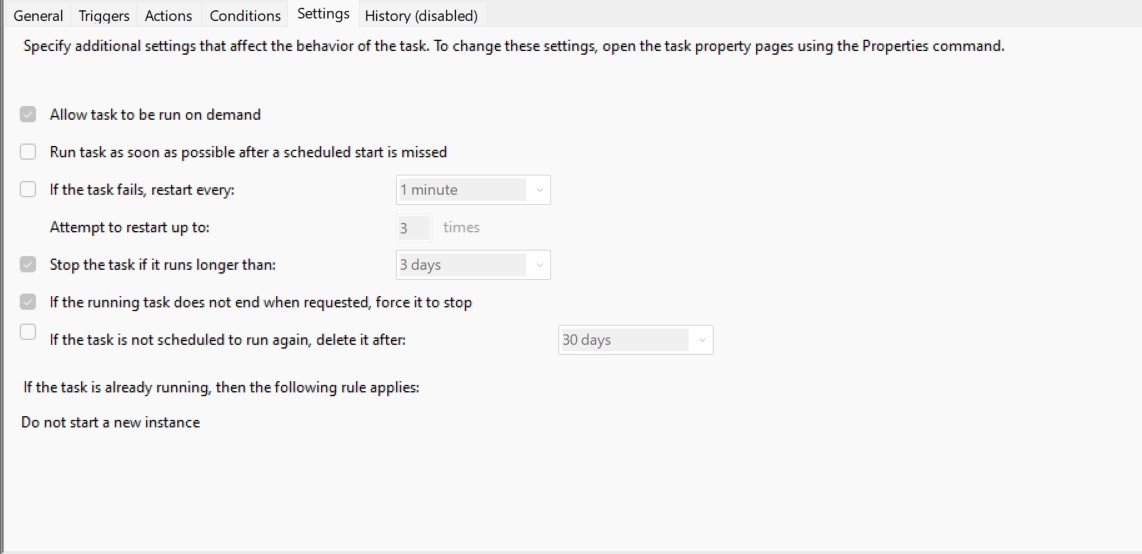
Step 6

In the "Action" dropdown, select "Start a program". In the "Program/script " field, enter powershell.exe.



# Step 5

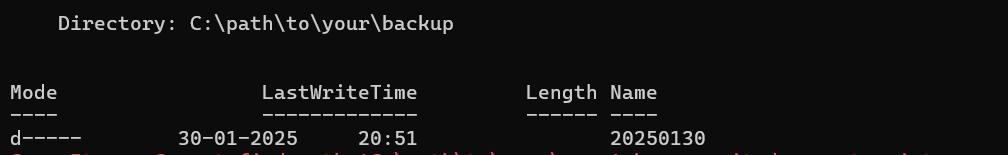
Ensure "Allow task to be run on demand" and "Run task as soon as possible after a scheduled start is missed" are checked.Adjust other settings as needed.Click "OK".



# Step 6

Run the script to ensure it works correctly . Open PowerShell and run:

.\backup.ps1



# Expected Outcome

Reliable Backup: A reliable backup of the specified files or directories is created at the designated location.

Log File: A log file is generated that records the date, time, and status (success/failure) of each backup operation.

Compressed Backup (Optional): If compression is enabled, the backup files are compressed into a single archive (e.g., .tar.gz or .zip).

Incremental Backup (Optional): If incremental backup is enabled, only the files that have changed since the last backup are copied.

Scheduled Execution (Optional): The script can be scheduled to run at regular intervals (e.g., daily, weekly) using a task scheduler like cron (Linux) or Task Scheduler (Windows).