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

Placement Empowerment Program
Cloud Computing and DevOps Centre

Create a Simple Backup Script

Name: Shalini D   Department: IT

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

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!

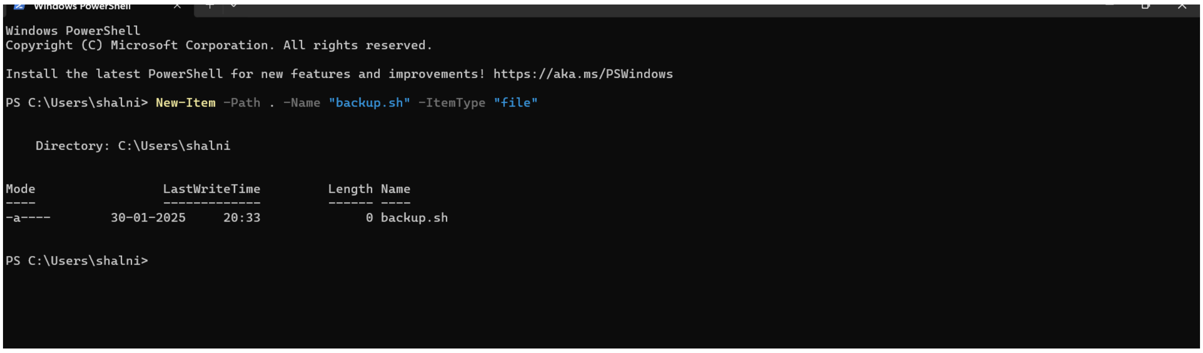
Objectives

The objective of this project is to develop a robust and automated backup solution for a Git repository. The solution involves creating a simple Bash script that will run daily and back up the entire repository to a specified local folder. This process ensures that the codebase is consistently and reliably safeguarded, minimizing the risk of data loss due to accidental deletions, hardware failures, or other unforeseen events. By the end of this project, you will have a fully functional backup system that operates autonomously, providing peace of mind and data security for your development projects.

Step-by-Step Overview

Step 1:

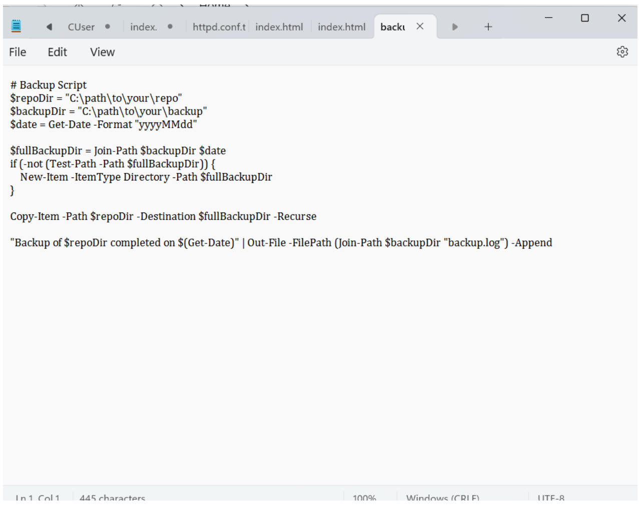
Create the Script File: Open your terminal and create a new file for your backup script. Let's call it backup.sh.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\shalni> New-Item -Path . -Name "backup.sh" -ItemType "file"


    Directory: C:\Users\shalni


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        30-01-2025     20:33              0 backup.sh


PS C:\Users\shalni>
```

Step 2

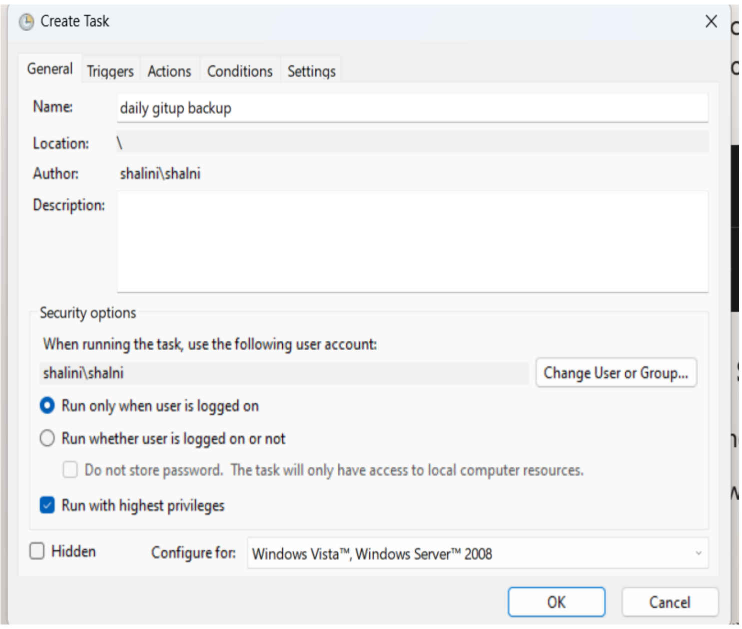
Open the file in your preferred text editor, such as Notepad



```
# Backup Script
$repoDir = "C:\path\to\your\repo"
$backupDir = "C:\path\to\your\backup"
$date = Get-Date -Format "yyyyMMdd"

$fullBackupDir = Join-Path $backupDir $date
if (-not (Test-Path -Path $fullBackupDir)) {
   New-Item -ItemType Directory -Path $fullBackupDir
}

Copy-Item -Path $repoDir -Destination $fullBackupDir -Recurse

"Backup of $repoDir completed on $(Get-Date)" | Out-File -FilePath (Join-Path $backupDir "backup.log") -Append
```

Step 3

Schedule the Script with Task Scheduler.



Name: daily gitup backup

Location: \

Author: shalini\shalni

Description:

Security options

When running the task, use the following user account:

shalini\shalni

● Run only when user is logged on

○ Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

□ Hidden

Configure for: Windows Vista™, Windows Server™ 2008

Step 5

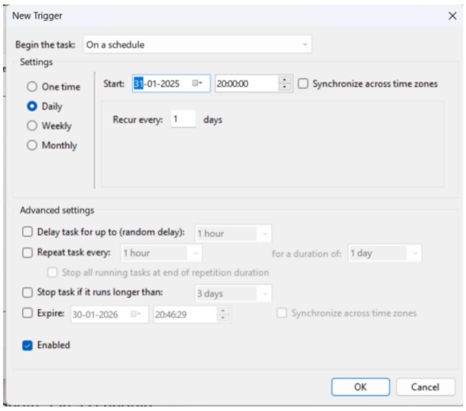
Triggers Tab:

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".



New Trigger

Begin the task: On a schedule

Settings

○ One time
● Daily
○ Weekly
○ Monthly

Start: 31-01-2025 20:00:00 □ Synchronize across time zones

Recur every: 1 days

Advanced settings

□ Delay task for up to (random delay): 1 hour

□ Repeat task every: 1 hour for a duration of: 1 day

□ Stop all running tasks at end of repetition duration

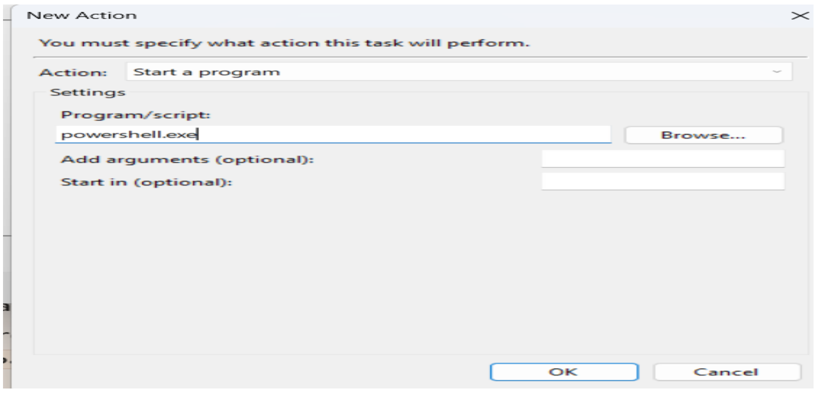
□ Stop task if it runs longer than: 3 days

□ Expire: 30-01-2026 20:46:29 □ Synchronize across time zones

☑ Enabled

Step 6

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



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

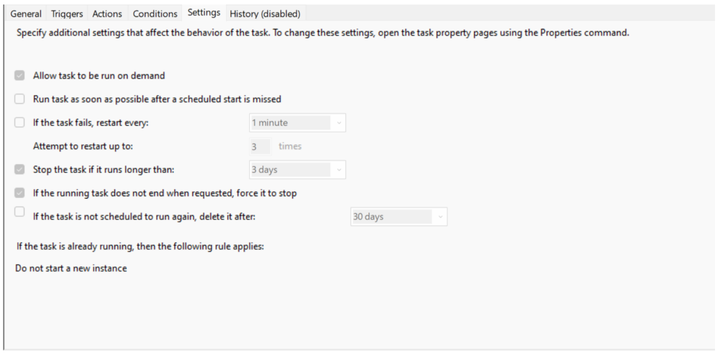
Program/script: powershell.exe

Add arguments (optional):

Start in (optional):

Step 7

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".



Specify additional settings that affect the behavior of the task. To change these settings, open the task property pages using the Properties command.

☑ Allow task to be run on demand

□ Run task as soon as possible after a scheduled start is missed

□ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☑ Stop the task if it runs longer than: 3 days

☑ If the running task does not end when requested, force it to stop

□ If the task is not scheduled to run again, delete it after: 30 days

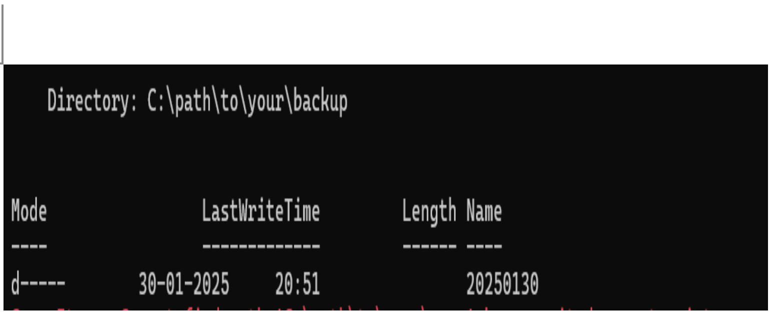
If the task is already running, then the following rule applies:

Do not start a new instance

Step 8

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

```
.\backup.ps1
```



```
    Directory: C:\path\to\your\backup


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        30-01-2025     20:51                20250130
```

Expected Outcome

The expected outcome of implementing this backup script is that a new backup of your Git repository will be created daily in the specified backup directory, each stored in a folder named with the current date. The backup directory will have a structured format, and if logging is enabled, a log entry will be added to the backup.log file each time a backup is completed, noting the date, time, and source repository path. This automated process will run at the scheduled time without manual intervention, ensuring that your Git repository is consistently backed up and protected against data loss. You can manually verify the backups by checking the contents of the backup folders to ensure all necessary files are copied correctly.