 

**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: Nanthini R Department: ADS



**Introduction**

. Backing up your Git repository is crucial to ensuring your code remains safe from accidental deletions, system crashes, or corruption. While Git itself helps track changes, keeping **daily backups** of your repository provides an additional safety net.

This guide will walk you through setting up an **automated backup system on Windows** using **PowerShell and Task Scheduler**. The script will:  
 Create a compressed backup of your repository.  
 Keep only the last **7 days** of backups to save storage.  
 Run automatically **every day** without manual intervention.

By the end of this guide, you will have a fully automated backup system in place for your Git repository.

**Overview**

We will accomplish this task by:

1. **Creating a PowerShell script** to back up the repository.
2. **Configuring Windows Task Scheduler** to run the script automatically.
3. **Ensuring old backups are removed** after a set number of days.

This method is lightweight, efficient, and ensures your work remains safe without extra effort.

**Objectives**

By following this guide, you will:

* Learn how to write a PowerShell script for Git backups.
* Configure Windows Task Scheduler to run the script daily.
* Implement automatic cleanup of old backups.
* Ensure regular, reliable backups of your Git repository.

**Importance of Setting Up a Local Git Repository**

**Track Changes**: Git records all modifications, ensuring a clear history of your project.

**Rollback**: Easily revert to previous versions to recover from mistakes.

**Collaboration**: Prepares your project for team work, enabling smooth integration of changes.

**Step-by-Step Overview**

**Step 1:**

**Create a Backup Directory**

1. Choose a location for your backups, e.g., C:\GitBackups.
2. Create the folder manually or by running this PowerShell command:

powershell

CopyEdit

New-Item -ItemType Directory -Path "C:\GitBackups" -Force

**Step 2:**

Write the PowerShell Backup Script

1. Open Notepad (or any text editor).
2. Copy and paste the following script:

powershell

# Define variables

$repoPath = "C:\path\to\your\git\repository" # Change this to your actual repository path

$backupDir = "C:\GitBackups"

$timestamp = Get-Date -Format "yyyy-MM-dd\_HH-mm-ss"

$backupName = "repo\_backup\_$timestamp.zip"

$backupPath = "$backupDir\$backupName"

$retainDays = 7 # Number of days to keep old backups

# Ensure backup directory exists

If (!(Test-Path -Path $backupDir)) {

New-Item -ItemType Directory -Path $backupDir -Force

}

# Navigate to the repository

Set-Location -Path $repoPath

# Pull latest changes from remote repository

git pull origin main # Change "main" if using a different branch

# Create a compressed backup

Compress-Archive -Path "$repoPath\\*" -DestinationPath $backupPath

# Remove old backups older than specified days

Get-ChildItem $backupDir -Filter "repo\_backup\_\*.zip" | Where-Object { $\_.LastWriteTime -lt (Get-Date).AddDays(-$retainDays) } | Remove-Item -Force

Write-Output "Backup completed: $backupPath"

1. Save the file as:

makefile

C:\GitBackups\backup\_git\_repo.ps1

**Step 3:**

**Enable PowerShell Script Execution**

Windows may block PowerShell scripts from running. To allow execution:

1. Open PowerShell as Administrator.
2. Run the following command:

Set-ExecutionPolicy RemoteSigned

1. Type "Y" and press Enter.

**Step 4:**

**Test the Script Manually**

1. Open PowerShell.
2. Run the script:

C:\GitBackups\backup\_git\_repo.ps1

1. If successful, a .zip file should appear in C:\GitBackups.

**Step 5:**

**Automate the Script with Task Scheduler**

To ensure the script runs daily:

1. Press Win + R, type taskschd.msc, and press Enter.
2. Click Create Basic Task (on the right panel).
3. Name the task "Git Backup", then click Next.
4. Select Daily, then click Next.
5. Set the start time (e.g., 12:00 AM), then click Next.
6. Choose Start a Program, then click Next.
7. In Program/Script, enter:

powershell.exe

1. In Add arguments, enter:

arduino

-ExecutionPolicy Bypass -File "C:\GitBackups\backup\_git\_repo.ps1"

1. Click Finish to schedulethe task.

**Step 6:**

**Verify Automation**

* Wait for the scheduled time or manually right-click the task in Task Scheduler and select Run.
* Check C:\GitBackups to confirm a new backup was created**.**

**Expected Outcome**  
 It keeps only **7 days** of backups.

Everything runs **silently in the background**.

This ensures your repository remains backed up **without manual effort**.