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

aws

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: Gopinath M

Department: ADS

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

Introduction

In software development, Git repositories serve as essential hubs for code storage, version control, and collaboration. Regular backups of these repositories are vital to ensure the safety and availability of your work, especially in cases of accidental deletion, hardware failures, or other unforeseen events. While Git itself provides tools for version control and remote backups, having a local automated backup system adds an additional layer of security and convenience.

This document outlines the process of creating and automating a script that backs up a Git repository to a local folder daily. By leveraging simple scripting tools and automation methods, developers can establish a robust backup strategy tailored to their needs.

Overview

The backup process involves the following key steps:

1. **Understanding Requirements:**
    - Identify the Git repository to back up. o Choose the target platform (Linux/Unix or Windows) for running the backup script.
2. **Creating the Script:**
    - Use a shell script on Linux systems or a batch/PowerShell script on Windows systems.
    - The script will compress the repository (using tools like tar on Linux) or copy it to the backup folder (using robocopy on Windows).
3. **Automating the Task:**
    - Automate the backup script using:
        - **Cron jobs** for Linux systems.
        - **Task Scheduler** for Windows systems.
    - Configure the automation to run daily.
4. **Testing and Verification:**
    - Test the script manually to confirm it works as intended.
    - Verify that backups are created in the specified folder with the correct structure and naming convention.

Step 1: Understanding the Requirements ·

Target: Backup an entire Git repository.

- **Frequency:** Daily backups.
- **Tools Needed:**
  - **Linux:** Use tar for compression and scheduling with cron.
  - **Windows:** Use robocopy for copying and schedule with Task Scheduler.

Step 2: Backup Script Example For

Linux: Shell Script with tar

1. **Create the Script:**

```
# Define variables
REPO_PATH="/path/to/your/git/repo"
BACKUP_PATH="/path/to/backup/location"
DATE=$(date +%F)
# Create a tarball backup tar -czf "$BACKUP_PATH/backup-
$DATE.tar.gz" -C "$REPO_PATH" .
echo "Backup created at $BACKUP_PATH/backup-$DATE.tar.gz" 2.
```

Make the Script Executable: chmod +x backup.sh

3. **Schedule with cron:**

   - Open the crontab editor:

```
crontab -e
0 0 * * * /path/to/backup.sh
```

For Windows: Batch Script with robocopy

1. **Create the Script:** @echo off set

```
REPO_PATH=C:\path\to\your\git\repo set
BACKUP_PATH=C:\path\to\backup\location
```

```
  set DATE=%date:~-
  4%%date:~4,2%%date:~7,2%

robocopy %REPO_PATH% %BACKUP_PATH%\backup-%DATE% /E echo
Backup created at %BACKUP_PATH%\backup-%DATE%
```

2. **Schedule with Task Scheduler:**
   - Open Task Scheduler.
   - Create a new task and set it to run backup.bat daily.

Step 3: Testing and Verification

1. Run the script manually to test functionality.
2. Verify the backup files are created in the specified location.

Conclusion

The script ensures a daily backup of your Git repository, safeguarding against data loss. Depending on your operating system, follow the Linux or Windows setup.