Automated Backup Script

Sharjeel Faiq

June 12, 2023

1 Introduction

This document presents a PowerShell script that automates the backup of specific files or directories.

2 PowerShell Script

```
# Define the source directory or file to be backed up
$sourcePath = "C:\Path\to\source"

# Define the destination directory where the backup will be stored
$destinationPath = "C:\Path\to\destination"

# Create a timestamp for the backup file name
$timestamp = Get-Date -Format "yyyyMMdd_HHmmss"

# Create a backup file name using the timestamp
$backupFileName = "backup_$timestamp.zip"

# Create a full path for the backup file
$backupFilePath = Join-Path -Path $destinationPath -ChildPath $backupFileName

# Compress and archive the source directory or file
Compress-Archive -Path $sourcePath -DestinationPath $backupFilePath

# Output the backup file path
Write-Host "Backup created: $backupFilePath"
```

3 Explanation

The provided PowerShell script automates the backup process as follows:

1. The source directory or file to be backed up is specified using the `$sourcePath` variable.

2. The destination directory where the backup will be stored is specified using the `$destinationPath` variable.

3. A timestamp is generated using the `Get-Date` cmdlet with the desired format.

4. A backup file name is created by combining the prefix ”backup_” with the timestamp.

5. The full path for the backup file is constructed using the `Join-Path` cmdlet, which joins the destination path and backup file name.

6. The `Compress-Archive` cmdlet is used to compress and archive the source directory or file into a zip file at the specified backup file path.

7. The backup file path is displayed using the `Write-Host` cmdlet.

This script can be scheduled to run automatically or triggered manually to perform the backup of specific files or directories.