Automated Backup Script Explanation - backup.sh

📄 Project Title: Automated Backup Script (backup.sh)  
🧑‍💻 Author: [Souvik Mandal]  
🏢 Organization: ABC International Inc.  
📅 Project Goal:  
To automate the process of backing up encrypted password files modified in the past 24 hours, reducing human error and improving efficiency.

# Script Breakdown and Explanation

#!/bin/bash

Shebang – Specifies the interpreter to execute the script, in this case /bin/bash.

if [[ $# != 2 ]]

Validates that exactly two arguments are passed to the script.

echo "backup.sh target\_directory\_name destination\_directory\_name"

Usage message if incorrect arguments are passed.

if [[ ! -d $1 ]] || [[ ! -d $2 ]]

Validates that both arguments are valid directories.

targetDirectory=$1

Assign the first command-line argument to targetDirectory.

destinationDirectory=$2

Assign the second command-line argument to destinationDirectory.

echo "Target Directory: $targetDirectory"

Display the target directory.

echo "Destination Directory: $destinationDirectory"

Display the destination directory.

currentTS=$(date +%s)

Capture the current timestamp in seconds.

backupFileName="backup-$currentTS.tar.gz"

Set the backup filename using the timestamp.

origAbsPath=$(pwd)

Save the absolute path of the current working directory.

cd "$destinationDirectory" || exit

Change to the destination directory and exit if fails.

destDirAbsPath=$(pwd)

Save the absolute path of the destination directory.

cd "$origAbsPath" || exit

Return to the original directory and exit if fails.

cd "$targetDirectory" || exit

Navigate into the target directory and exit if fails.

yesterdayTS=$((currentTS - 24 \* 60 \* 60))

Calculate the timestamp for 24 hours ago.

declare -a toBackup

Declare an array to store files modified in the last 24 hours.

for file in \*

Loop through all files in the current directory.

if [[ $(date -r "$file" +%s) -gt $yesterdayTS ]]

Check if file was modified within the last 24 hours.

toBackup+=("$file")

Add the file to the toBackup array.

tar -czvf $backupFileName ${toBackup[@]}

Create a compressed archive of the files to backup.

mv $backupFileName $destDirAbsPath

Move the backup file to the destination directory.

# Crontab Examples

To run this script daily:

0 0 \* \* \* /usr/local/bin/backup.sh /path/to/source /path/to/destination

To run every 1 minute for testing:

\*/1 \* \* \* \* /usr/local/bin/backup.sh /path/to/source /path/to/destination