Automated Incremental Backup for Hyper-V (Daily at 22:00)!!

Created a **scheduled PowerShell script** to perform **incremental backups** of our client Dr. RieXXXX Hyper-V VM backups from **D:\VM** to an **external drive (G:\TM Backup)**.

Step 1: Powershell script writing, testing!

Open Notepad and insert script with this one:

---------- Version 1: ----- ----------

```
$Source = "D:\VM"
$Destination = "G:\TM Backup"
$LogFile = "G:\TM Backup\backup_log.txt"

# Create log folder if not exists
If (!(Test-Path $Destination)) {
    New-Item -ItemType Directory -Path $Destination
}

# Log start time
Add-Content -Path $LogFile -Value "Backup Started: $(Get-Date)"

# Perform incremental backup using Robocopy (First time full, then incremental)
Robocopy $Source $Destination /E /ZB /MT:8 /XO /R:3 /W:10 /LOG+:$LogFile

# Log completion time
Add-Content -Path $LogFile -Value "Backup Completed: $(Get-Date)"
Add-Content -Path $LogFile -Value "----------------------------------"
```

##

Save it as (example):
File Name: HyperV_Backup.ps1 [Write your own choice]
Location: C:\Scripts\HyperV_Backup.ps1 [You can create a folder in your host-server's C:]

Testing: Manually Start the Backup for testing...wait some times and check destination folder!

1. Open PowerShell as Administrator (Win + X → PowerShell (Admin)).
2. Run the script manually: C:\Scripts\HyperV_Backup.ps1
[Adjust name/folder if applicable].

If it works well, then go to step 2!!

Step 2: Schedule the Backup to Run Daily at 22:00 [adjust your own].

Since the backup will be on an external drive, we must make sure the drive is connected when the script runs.

It will make a full backup for the first time then make only incremental backup. Also, if you start it manually and make any interrupt, will resume where he left.

1. Open Task Scheduler (Win + R → Type taskschd.msc → Enter).
2. Click "Create Basic Task" in the right panel.
3. Name: "Hyper-V TM Backup", then click Next.
4. Trigger: Choose "Daily", then click Next.
5. Set Time: 22:00 (10:00 PM), then click Next.
6. Action: Choose "Start a Program", then click Next.
7. Program/Script: powershell.exe
8. Add Arguments (example, my case):

-ExecutionPolicy Bypass -File C:\Scripts\HyperV_Backup.ps1

9. Click Finish.

How This Works!

✔ First time = Full backup from D:\VM to G:\TM Backup.
✔ Future backups = Incremental (only new files copied).
✔ Check the log file (G:\TM Backup\backup_log.txt) → It will show all files copied/success message at the end.
✔ Powershell will close automatically after completion of Backup.
✔ Runs daily at 22:00 automatically.

---------- Version 2: --- ----------

```
$Source = "D:\VM"
$Destination = "G:\TM Backup"
$LogFile = "G:\TM Backup\backup_log.txt"
$ErrorLog = "G:\TM Backup\backup_error_log.txt"
$MaxRetries = 3  # Number of retry attempts
$RetryWait = 120  # Wait time in seconds before retrying (2 minutes)

# Create log folder if not exists
If (!(Test-Path $Destination)) {
   New-Item -ItemType Directory -Path $Destination
}

# Log start time
$StartTime = Get-Date
Add-Content -Path $LogFile -Value "Backup Started: $StartTime"
$Attempts = 0
$Success = $false

# Retry Loop
Do {
   $Attempts++
   $RobocopyLog = "G:\TM Backup\robocopy_log.txt"

   # Run Backup
   Robocopy $Source $Destination /E /ZB /MT:8 /XO /R:3 /W:10 /LOG:$RobocopyLog
   $ExitCode = $LASTEXITCODE  # Get Robocopy exit code

   # Check Exit Code
   If ($ExitCode -eq 0 -or $ExitCode -eq 1) {
      # 0 = No errors, 1 = Some files copied successfully
      $Success = $true
      $EndTime = Get-Date
      $Duration = $EndTime - $StartTime
      Add-Content -Path $LogFile -Value "Backup Completed Successfully on Attempt $Attempts:
$EndTime"
      Add-Content -Path $LogFile -Value "Total Backup Time: $($Duration.Hours)h
$($Duration.Minutes)m $($Duration.Seconds)s"

      # Show success pop-up with duration
      [System.Windows.MessageBox]::Show("Backup completed successfully on attempt
$Attempts!`nTotal Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s",
"Backup Status", 0, 64)
      Break
   } Else {
```

```
        # Capture failure details
        Add-Content -Path $ErrorLog -Value "Backup Attempt $Attempts Failed: $StartTime"
        Add-Content -Path $ErrorLog -Value "Error Code: $ExitCode"
        Add-Content -Path $ErrorLog -Value "Check robocopy_log.txt for details."
        Add-Content -Path $ErrorLog -Value "-----------------------------------"

        # If not the last attempt, wait before retrying
        If ($Attempts -lt $MaxRetries) {
            Add-Content -Path $ErrorLog -Value "Retrying backup in $RetryWait seconds..."
            Start-Sleep -Seconds $RetryWait
        }
    }
} While ($Attempts -lt $MaxRetries -and -not $Success)

# Final Result
If (-not $Success) {
    # If all retries failed
    $EndTime = Get-Date
    $Duration = $EndTime - $StartTime
    [System.Windows.MessageBox]::Show("Backup Failed after $MaxRetries attempts!`nTotal Time:
$($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s`nCheck G:\TM
Backup\backup_error_log.txt", "Backup Error", 0, 16)
    Add-Content -Path $ErrorLog -Value "Backup Failed after $MaxRetries attempts at $EndTime"
    Add-Content -Path $ErrorLog -Value "Total Backup Time: $($Duration.Hours)h
$($Duration.Minutes)m $($Duration.Seconds)s"
}

# Close PowerShell window after 5 seconds
Start-Sleep -Seconds 5
Exit
```

##

How This Works!

Logs Success or Failure in a File:

✔ If backup succeeds, logs Backup Completed Successfully: [timestamp] in backup_log.txt.

✔ If backup fails, logs error in backup_error_log.txt with error code.

✔ Shows total backup time in the pop-up message.

✔ Logs backup time in backup_log.txt.

✔ If failed, logs total failed time in backup_error_log.txt.

✔ Automatically retries if needed.

Exit Code	Meaning
`0`	Success – No files copied (already up to date).
`1`	Success – Files copied successfully.
`2`	Some files failed to copy.

---------- Version 3: --- ----------

```
$Source = "D:\VM"
$Destination = "G:\TM Backup"
$LogFile = "G:\TM Backup\backup_log.txt"
$ErrorLog = "G:\TM Backup\backup_error_log.txt"
$MaxRetries = 3  # Number of retry attempts
$RetryWait = 120  # Wait time in seconds before retrying (2 minutes)

# Gmail notification settings
$SMTPServer = "smtp.gmail.com"
$SMTPPort = "587"
$SMTPUser = "your-email@gmail.com"   # Replace with your Gmail
$SMTPPass = "your-app-password"      # Use an App Password, NOT your normal password
$ToEmail = "your-email@gmail.com"    # Replace with your recipient email

# Create log folder if not exists
If (!(Test-Path $Destination)) {
    New-Item -ItemType Directory -Path $Destination
}

# Log start time
$StartTime = Get-Date
Add-Content -Path $LogFile -Value "Backup Started: $StartTime"
$Attempts = 0
$Success = $false

# Retry Loop
Do {
    $Attempts++
    $RobocopyLog = "G:\TM Backup\robocopy_log.txt"

    # Run Backup
    Robocopy $Source $Destination /E /ZB /MT:8 /XO /R:3 /W:10 /LOG:$RobocopyLog
    $ExitCode = $LASTEXITCODE  # Get Robocopy exit code
```

```
    # Check Exit Code
    If ($ExitCode -eq 0 -or $ExitCode -eq 1) {
        # 0 = No errors, 1 = Some files copied successfully
        $Success = $true
        $EndTime = Get-Date
        $Duration = $EndTime - $StartTime
        Add-Content -Path $LogFile -Value "Backup Completed Successfully on Attempt $Attempts: $EndTime"
        Add-Content -Path $LogFile -Value "Total Backup Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s"

        # Show success pop-up with duration
        [System.Windows.MessageBox]::Show("Backup completed successfully on attempt $Attempts!`nTotal Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s", "Backup Status", 0, 64)

        # Send success email
        $Subject = "Hyper-V Backup Completed Successfully"
        $Body = "Your Hyper-V backup completed successfully at $EndTime.`nTotal Backup Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s.`nLog File: $LogFile"
        Send-MailMessage -To $ToEmail -From $SMTPUser -Subject $Subject -Body $Body -SmtpServer $SMTPServer -Port $SMTPPort -Credential (New-Object System.Management.Automation.PSCredential ($SMTPUser, (ConvertTo-SecureString $SMTPPass -AsPlainText -Force))) -UseSsl
        Break
    } Else {
        # Capture failure details
        Add-Content -Path $ErrorLog -Value "Backup Attempt $Attempts Failed: $StartTime"
        Add-Content -Path $ErrorLog -Value "Error Code: $ExitCode"
        Add-Content -Path $ErrorLog -Value "Check robocopy_log.txt for details."
        Add-Content -Path $ErrorLog -Value "------------------------------------"

        # If not the last attempt, wait before retrying
        If ($Attempts -lt $MaxRetries) {
```

```
        Add-Content -Path $ErrorLog -Value "Retrying backup in $RetryWait seconds..."
        Start-Sleep -Seconds $RetryWait
      }
   }
} While ($Attempts -lt $MaxRetries -and -not $Success)

# Final Result - Send Failure Email
If (-not $Success) {
   $EndTime = Get-Date
   $Duration = $EndTime - $StartTime
   [System.Windows.MessageBox]::Show("Backup Failed after $MaxRetries attempts!`nTotal Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s`nCheck G:\TM Backup\backup_error_log.txt", "Backup Error", 0, 16)
   Add-Content -Path $ErrorLog -Value "Backup Failed after $MaxRetries attempts at $EndTime"
   Add-Content -Path $ErrorLog -Value "Total Backup Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s"

   # Send failure email
   $Subject = "Hyper-V Backup Failed"
   $Body = "Backup failed after $MaxRetries attempts!`nTotal Time: $($Duration.Hours)h $($Duration.Minutes)m $($Duration.Seconds)s.`nCheck $ErrorLog for details."
   Send-MailMessage -To $ToEmail -From $SMTPUser -Subject $Subject -Body $Body -SmtpServer $SMTPServer -Port $SMTPPort -Credential (New-Object System.Management.Automation.PSCredential ($SMTPUser, (ConvertTo-SecureString $SMTPPass -AsPlainText -Force))) -UseSsl
}

# Close PowerShell window after 5 seconds
Start-Sleep -Seconds 5
Exit

################################################################################
```

Setup Gmail to Allow Sending Emails:

Since Gmail blocks scripts from sending emails using passwords, you need to use an App Password.

How to Get Your Gmail App Password

1. Go to Google App Passwords.
2. Select Mail and Windows Computer.
3. Click Generate.
4. Copy the 16-character app password (example: abcd efgh ijkl mnop).
5. Replace $SMTPPass in the script with your generated app password.

Now You Have Gmail Notifications with Backup Time + Auto-Retry!

✔ Success Email + Duration

✔ Failure Email + Error Code

✔ Backup Time Logged

✔ Automatic Retry Up to 3 Times

Restoring!!

Step 1: Copy the Backup to the New Server (If Needed)

If your backup is still on G:\TM Backup, you can either: ✅Run the VM directly from G:\ (not recommended for performance)

✅Copy the backup to D:\VM on the new server (better for stability)

Step 2: Re-Register the VM in Hyper-V

Once the files are in place, follow one of these two restore methods:

Option 1: Restore Using Hyper-V Import (Best Method)

If your backup contains an exported VM structure, you can import the VM directly:

1. Open Hyper-V Manager on the new server.
2. Click "Import Virtual Machine" from the right panel.
3. Browse to D:\VM (or G:\TM Backup, if running from the external drive).
4. Select the VM folder (it should contain Virtual Hard Disks & Virtual Machines).
5. Choose:

✅"Register in-place" (if files are already in D:\VM).

✅"Copy the virtual machine" (if moving to another location).

6. Click Finish and wait for import to complete.

VM is now restored and ready to start!

Option 2: Manually Recreate the VM and Attach Its Disk

If you only backed up the VHDX file and need to manually create the VM:

1. Open Hyper-V Manager.
2. Click "New → Virtual Machine".
3. Set the same VM settings as before (CPU, RAM, and network).
4. When asked for a disk, choose "Use an existing virtual hard disk".
5. Browse to D:\VM\Virtual Hard Disks\YourVM.vhdx (or G:\TM Backup).
6. Click Finish, and then start the VM.

VM is now restored and running!

Step 3: Verify the Restore

✅Check if the VM boots properly.

✅Verify all applications & data inside the VM.

✅Ensure network connectivity is working.

Implemented and documented by:

Mr. Akram
IT-Administrator
Uckert Technology