**Today Assignment – Date: August 12,2025**

**Name: 34733-Prasang Kumar Singh**

**Gmail:** [**thakurprasang23@gmail.com**](mailto:thakurprasang23@gmail.com)

**Topics**

**Execute process in powershell**

**How to Run a Process in PowerShell**

1. Running Directly from the Shell
   * Move to the folder where the program or script is located using the cd command.
   * Launch it by typing . **\cloudflare\_WARP.exe**or by giving the absolute path, for example:

C:\Folder\Subfolder\YourApp.exe

1. Using Start-Process
   * This command lets you start an application without blocking PowerShell’s execution.  
     Example:

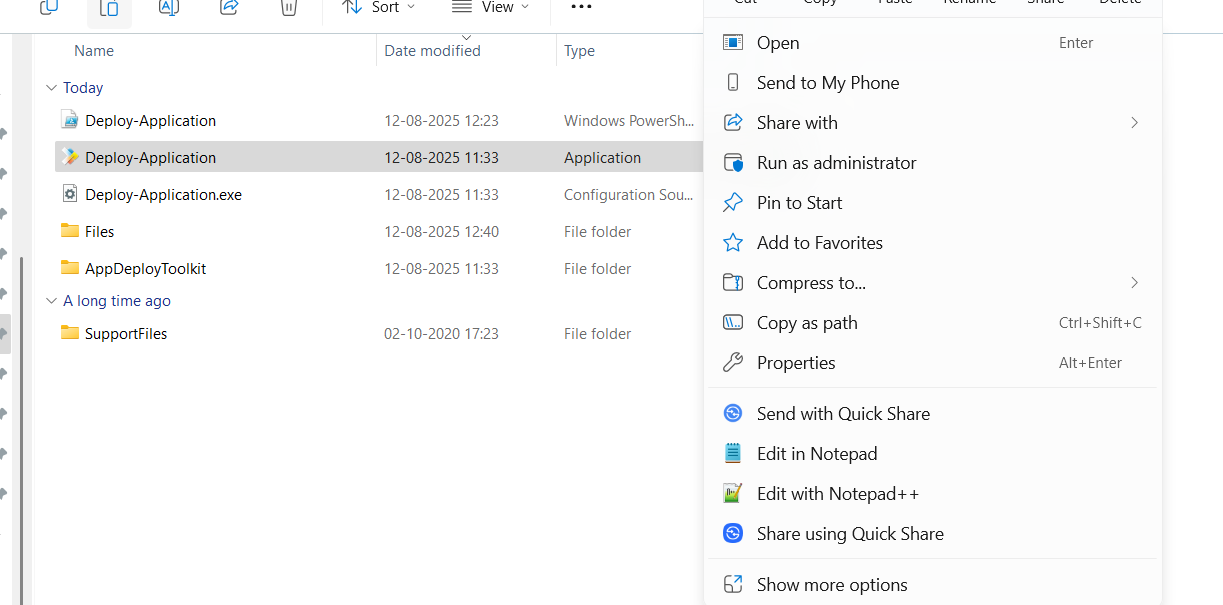
**Start-Process -FilePath "C:\Path\Program.exe"**

* + Optional parameters:
    - -WindowStyle to control how the window appears (Normal, Minimized, Hidden).
    - -ArgumentList to pass command-line arguments.
    - -Credential to run as another user.

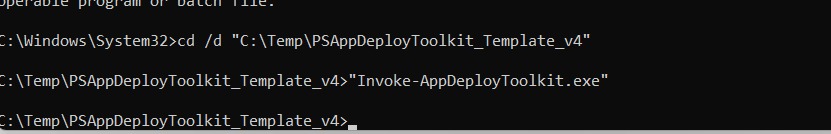
1. Running PowerShell Scripts
   * Switch to the directory containing your .ps1 file.
   * Execute it with

**.\ScriptName.ps1**

* + To run with admin rights, start PowerShell as Administrator before executing.



1. **Executing in the Background**
   * Launch a process without waiting for it to finish by omitting -Wait in Start-Process.
   * Alternatively, append an ampersand & at the end of the command.
2. **Using Invoke-Expression**
   * This cmdlet evaluates and runs command strings dynamically.
   * It’s handy when commands are stored in variables or built on the fly.

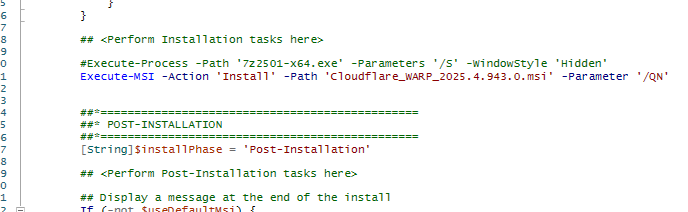


1. Running Under a Specific User Profile
   * With tools like**PSAppDeployToolkit**, you can use Execute-**ProcessAsUser** to launch apps under another user session.

**Adjusting Execution Policy**

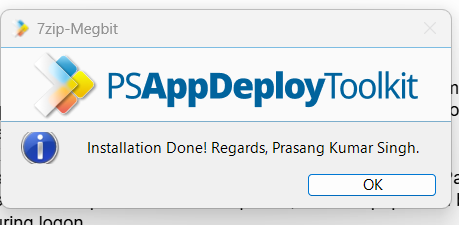
* + If you encounter restrictions when running scripts, use:

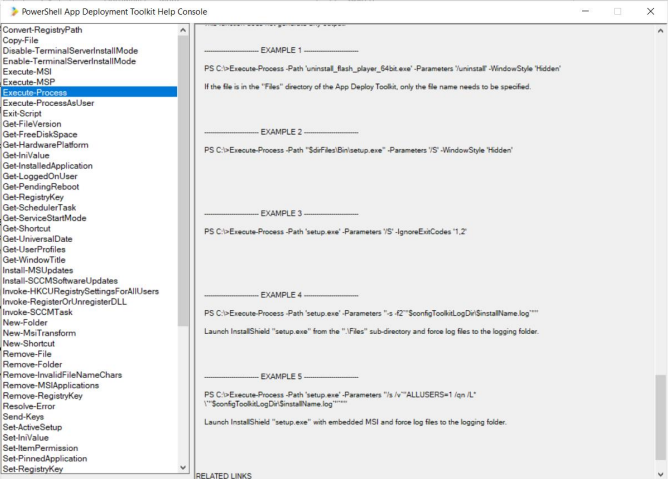
**Set-ExecutionPolicy** to change the policy appropriately.



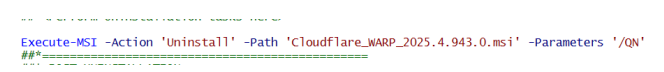
I also use execute -process as well as Execute\_MSI

Execute MSI for install Cloudflare\_Warp\_2025.msi



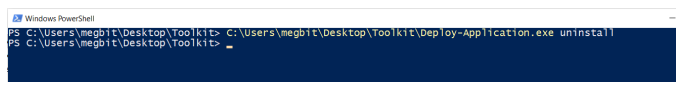


**For Uninstall**

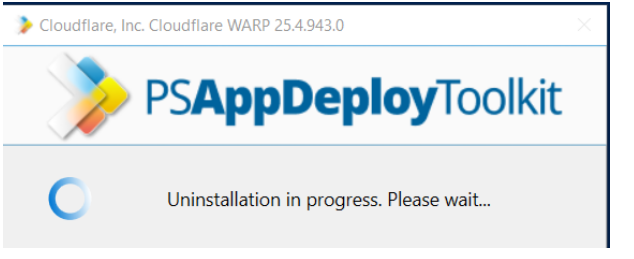
****

**Save the script and run the script (Deploy-Application.ps1)**

**For uninstallation script**

****

**Process is start:**

****

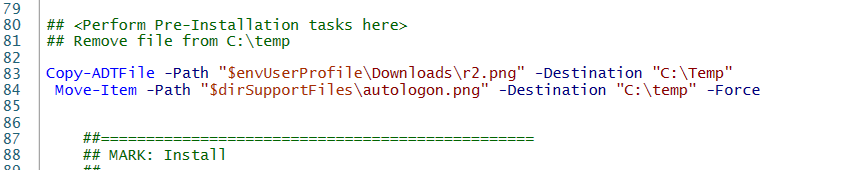
****

Topic 5:Important Commands of Psadt

**1. Copying Files and Folders:**

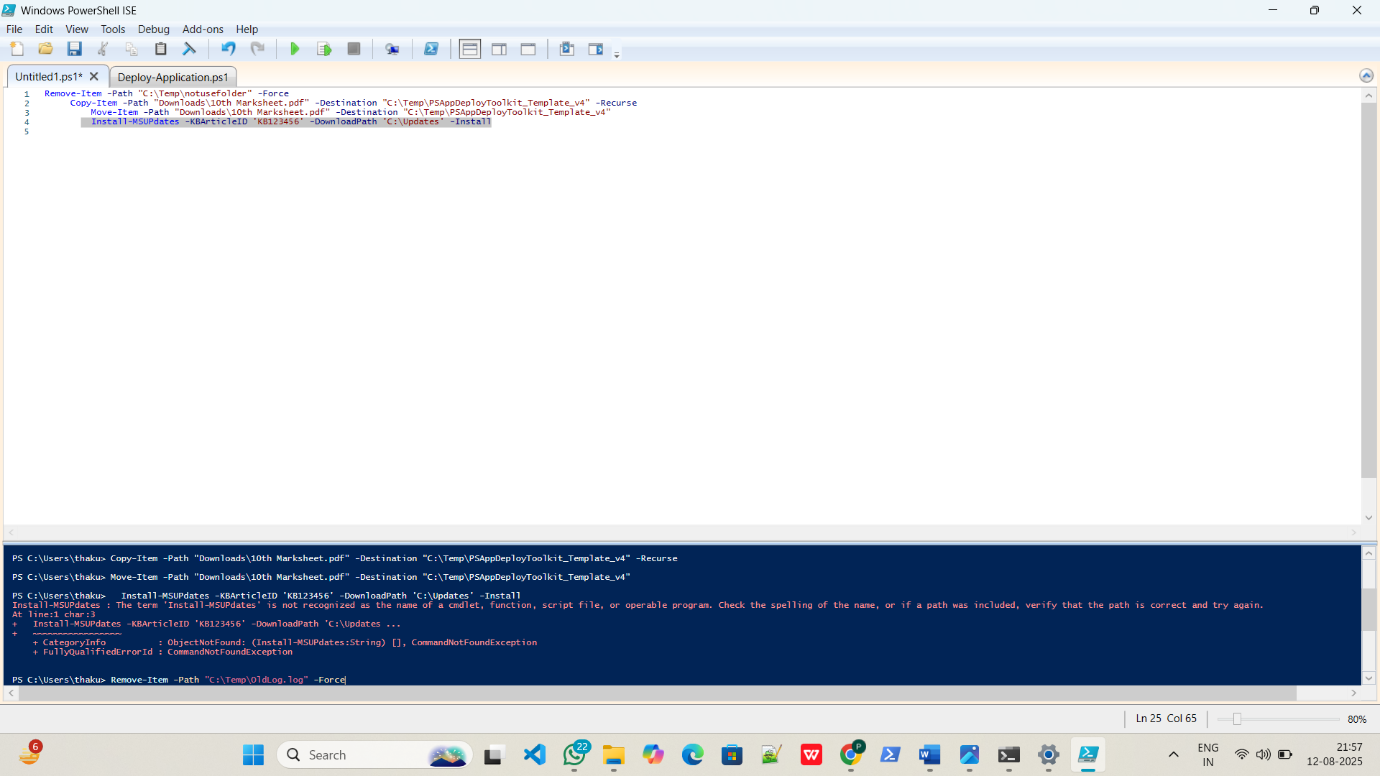
**Copy-ADTFile**  
Copies files during PSADT deployment with error handling and logging.

Copy-ADTFile -Path "$dirSupportFiles\YourFile.txt" -Destination "$envProgramFiles\YourApp"



**2.Copy-Item**  
Native PowerShell cmdlet for basic copy operations.

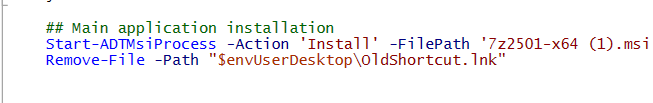
Copy-Item -Path "C:\Source\Folder" -Destination "C:\Destination\Folder" -Recurse



**3. . Deleting Files and Folders:**

Remove-File  
PSADT function to delete specific files.

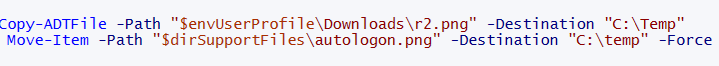
Remove-File -Path "$envUserDesktop\OldShortcut.lnk"



4. **Moving Files and Folders:**

**Move-Item**  
Native PowerShell cmdlet to move files or folders.

Move-Item -Path "C:\Source\File.doc" -Destination "C:\Destination\Documents"



5. **Execute-ADTMSI -Action**

The Execute-ADTMSI function in PSAppDeployToolkit is used to run MSI (Microsoft Installer) packages during software deployment. The -Action parameter specifies what operation to perform on the MSI package.

**Syntax:**

Execute-ADTMSI -Action <Install|Uninstall|Repair> -Path "<PathToMSI>" [-Parameters "<MSI Parameters>"] [-LogName "<LogFileName>"] [-ContinueOnError]

**Parameters:**

* -Action  
  Defines the MSI operation.
  + Install — Installs the MSI package.
  + Uninstall — Uninstalls the MSI package.
  + Repair — Repairs the existing MSI installation.
* -Path  
  Full path to the MSI installer file.
* -Parameters (Optional)  
  Additional command-line parameters to pass to msiexec.exe (e.g., silent install switches like /qn).
* -LogName (Optional)  
  Specifies the name of the installation log file.
* -ContinueOnError (Optional)