

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; `
    >> [System.Net.ServicePointManager]::SecurityProtocol = `
    >> [Net.Object]::System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1') >> [System.ServicePointManager]::SecurityProtocol -bor 3072; `

Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.6.0.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.6.0 to C:\Users\CAPACI-1\AppData\Local\Temp\chocolatey\chocoinstall\chocolatey.zip
Not using proxy.
Extracting C:\Users\CAPACI-1\AppData\Local\Temp\chocolatey\chocoinstall\chocolatey.zip to C:\Users\CAPACI-1\AppData\Local\Temp\chocolatey\chocoinstall
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib\
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A cache file for the command line goes to 'C:\ProgramData\chocolatey\bin\
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey CLI folders if they do not already exist.

chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrap.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not waiting for completion: Profile file does not exist at
'C:\Users\CAPACI-1\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey CLI (choco.exe) is now ready.
You can call choco from anywhere, command line or PowerShell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart PowerShell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32>
```

I opened Windows PowerShell in Administrator mode to obtain elevated privileges required for system-level operations. I created a new local user using the net user command and verified successful creation by listing system users. This demonstrates user account management within the Windows operating system.

2. File and Permission Management

Nandi Zulu
OPERATING SYSTEMS

```
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\CAPACITI-JHB\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey CLI (choco.exe) is now ready.
You can call choco from anywhere, command line or PowerShell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart PowerShell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32> net user testuser123 Password@123 /add
The command completed successfully.

PS C:\Windows\system32> net user

User accounts for \\DESKTOP-70F0ITP

-----
Administrator      CAPACITI-JHB      DefaultAccount
Guest               postgres          testuser123
WDAGUtilityAccount
The command completed successfully.

PS C:\Windows\system32> 
```

```
PS C:\Windows\system32> mkdir C:\PowerUserTest

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          2026/02/24   08:35                PowerUserTest

PS C:\Windows\system32> 
```

Explanation:

I created a test directory and used the `icacls` command to modify Access Control Lists (ACLs). Full control permissions were granted to the newly created user. This demonstrates practical understanding of file system security and permission structures in Windows.

3. Package Management Using CLI

Nandi Zulu
OPERATING SYSTEMS

```
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\CAPACITI-JHB\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey CLI (choco.exe) is now ready.
You can call choco from anywhere, command line or PowerShell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart PowerShell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32> net user testuser123 Password@123 /add
The command completed successfully.
```

```
PS C:\Windows\system32> net user
```

```
User accounts for \\DESKTOP-70F0ITP
```

```
-----
Administrator      CAPACITI-JHB      DefaultAccount
Guest               postgres          testuser123
WDAGUtilityAccount
The command completed successfully.
```

```
PS C:\Windows\system32> mkdir C:\PowerUserTest
```

```
Directory: C:\
```

```
Mode                LastWriteTime         Length Name
----                -
d-----          2026/02/24    08:35             PowerUserTest
```

```
PS C:\Windows\system32> icacls C:\PowerUserTest /grant testuser123:F
```

```
processed file: C:\PowerUserTest
```

```
Successfully processed 1 files; Failed processing 0 files
```

```
PS C:\Windows\system32> icacls C:\PowerUserTest
```

```
C:\PowerUserTest DESKTOP-70F0ITP\testuser123:(F)
                   BUILTIN\Administrators:(I)(OI)(CI)(F)
                   NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
                   BUILTIN\Users:(I)(OI)(CI)(RX)
                   NT AUTHORITY\Authenticated Users:(I)(M)
                   NT AUTHORITY\Authenticated Users:(I)(OI)(CI)(IO)(M)
```

```
Successfully processed 1 files; Failed processing 0 files
```

```
PS C:\Windows\system32>
```

```
GoogleChrome v145.0.7631.110 [Approved]
GoogleChrome package files install completed. Performing other installation steps.
WARNING: Unable to find the architecture of the installed Google Chrome application
Downloading googlechrome 64 bit
  from 'https://dl.google.com/dl/chrome/install/googlechromestandaloneenterprise64.msi'
Download of googlechromestandaloneenterprise64.msi (-1 B) completed.
Error - hashes do not match. Actual value was 'AE349E9683BC895CB034E3D233C374E8C911516D0AE81AD624E9B67948996315'.
ERROR: Checksum for 'C:\Users\CAPACITI-JHB\AppData\Local\Temp\chocolatey\GoogleChrome\145.0.7631.110\googlechromestandaloneenterprise64.msi' did not meet '47bc0b7e2f6dd3a6970572e6e37ba070a6eacdee45761ca0b609e35d3fd'
Checksum type 'sha256'. Consider passing the actual checksums through with --checksum --checksums since you validate the checksums are appropriate. A less secure option is to pass --ignore-checksums if necessary.
The install of GoogleChrome was NOT successful!
Error while running 'C:\ProgramData\chocolatey\lib\GoogleChrome\tools\chocolateyinstall.ps1'.
See log for details.

Chocolatey installed 2/3 packages. 1 packages failed.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Failures:
  * GoogleChrome (exited -1) - Error while running 'C:\ProgramData\chocolatey\lib\GoogleChrome\tools\chocolateyinstall.ps1'.
See log for details.
```

Explanation:

I initially attempted to install Google Chrome using the Chocolatey package manager via the Choco install command. However, the installation failed due to a checksum validation error, indicating a mismatch between the downloaded file and the expected SHA256 hash. To maintain secure package management practices and avoid bypassing integrity verification, I proceeded to install VLC media player successfully using Chocolatey. This demonstrates command-line software installation, repository usage, and proper package integrity handling within the Windows environment.

```
Administrator: Windows PowerShell
Chocolatey v2.6.0
Installing the following packages:
vlc
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading vlc.install 3.0.23... 100%

vlc.install v3.0.23 [Approved]
vlc.install package files install completed. Performing other installation steps.
Installing 64-bit vlc.install...
vlc.install has been installed.
WARNING: No registry key found based on 'vlc.install'
WARNING: Can't find vlc.install install location
    vlc.install may be able to be automatically uninstalled.
    The install of vlc.install was successful.
    Deployed to 'C:\Program Files\VideoLAN\VLC'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading vlc 3.0.23... 100%

vlc v3.0.23 [Approved]
vlc package files install completed. Performing other installation steps.
The install of vlc was successful.
    Deployed to 'C:\ProgramData\chocolatey\lib\vlc'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

4. Conclusion

Through this practical task, I demonstrated understanding of operating system internals, user and group management, file system permissions, and package management. I used administrative privileges to perform system-level operations and applied CLI tools to manage software installation efficiently.