

Team ID	PNT2022TMID29528
Project Name	Customer Care Registry

Import-Module BitsTransfer

\$DownloadedInstaller="\$env:temp\IBM\_Cloud\_CLI.exe"

\$RepoHost="download.clis.cloud.ibm.com"

\$metadataHost="\$RepoHost/ibm-cloud-cli-metadata"

\$binaryDownloadHost="\$RepoHost/ibm-cloud-cli"

# fetch version metadata of CLI

\$infoEndpoint="https://\$metadataHost/info.json"

\$req = [System.Net.WebRequest]::create(\$infoEndpoint)

\$sr = new-object System.IO.StreamReader ((\$req.GetResponse()).GetResponseStream())

try {

    \$result = \$sr.ReadToEnd() | ConvertFrom-Json

}

catch {

    Write-Host 'Download latest CLI metadata failed. Please check your network connection. Quit installation.'

    return

}

finally {

    \$sr.Close()

}

if (\$null -eq \$result -or \$result -eq "") {

    Write-Host 'Download latest CLI metadata failed. Please check your network connection. Quit installation.'

    return

}

# parse latest version from metadata

```
$latestVersion = $result.latestVersion
```

```
if ($null -eq $latestVersion -or $latestVersion -eq "") {
```

```
Write-Host 'Unable to parse latest version number. Quit installation.'
```

```
return
```

```
}
```

```
# Check platform
```

```
$Arch = (Get-Process -Id $PID).StartInfo.EnvironmentVariables['PROCESSOR_ARCHITECTURE'];
```

```
# Download File
```

```
if ($Arch -eq 'x86') {
```

```
    Write-Host 'Current platform is Win32. Downloading corresponding IBM Cloud CLI...';
```

```
    $downloadUrl = "https://$binaryDownloadHost/$latestVersion/IBM_Cloud_CLI_$latestVersion" +  
    "_386.exe"
```

```
}
```

```
elseif ($Arch -eq 'amd64') {
```

```
    Write-Host 'Current platform is Win64. Downloading corresponding IBM Cloud CLI...';
```

```
    $downloadUrl = "https://$binaryDownloadHost/$latestVersion/IBM_Cloud_CLI_$latestVersion" +  
    "_amd64.exe"
```

```
}
```

```
else {
```

```
    Write-Host 'Unknown platform. Quit installation.'
```

```
    return
```

```
}
```

```
$req = [System.Net.WebRequest]::create($downloadUrl)
```

```
$resp = $req.GetResponse()
```

```
Start-BitsTransfer -Source "$($resp.ResponseUri.AbsoluteUri)" -Destination "$DownloadedInstaller"
```

```
Write-Host 'Download complete. Executing installer...'
```

```
# Invoke silent installer
```

```
& $DownloadedInstaller /s /v"REBOOT=ReallySupress /qn"
```

```
Start-Sleep -s 10
```

```
Write-Host 'Install complete. Please restart to make the modification effective.'
```