Chocolatey

*Chocolatey is a package manager that will install NuGet file extension packages on your computer. It is a software management application that will promote automation. This is harness the power of PowerShell in a Windows environment.*

**Chocolatey Download and Install**

You **MUST** download and install Chocolatey onto your computer facing the internet. Once you do this Chocolately will then internalize (Download all files needed to install offline) an offline version of Chocolately and all it dependencies onto your computer. Lastly, you then bring these files over to an offline computer and create your Chocolatey environment.

1. Chocolatey built PowerShell script.
   1. Copy and paste Chocolatey pre-made script to download and install Chocolatey onto your public internet facing computer.
      1. You can find that at the bottom of Exercise 0.
      2. https://chocolatey.org/docs/how-to-setup-offline-installation#exercise-1-optional-set-up-chocolatey-installation-on-a-machine-without-network-access
2. Admin PowerShell.
   1. Open PowerShell ISE as admin.
      1. Paste the copied Chocolatey script into your IDE. This will be considered our “Chocolately\_Internet\_Install” script.
   2. Edit lines within script (Optional; you may skip this part).
      1. FIPS compliance.
         1. You may uncomment or comment (#) out this line to follow your company best security practices.
      2. Internalized Packages (bottom of script)
         1. In my case, I removed “chocolatey.server” from this line because this guide will be built around the Nexus repo manager.
            1. “Chocolatey.server” is a default method of setting up your software repo location for your clients to grab software. This is in a Windows system file format and is the most basic non-rich feature to help you automate deployment.
   3. Chocolatey License File
      1. Run your “Chocolately\_Internet\_Install” script!
         1. It will error out, but this action will create a folder structure in the root of your C drive.
      2. Before running the script for the second time!
         1. Place your Chocolatey license file into your “C:\choco-setup\files”
            1. ***Notice:*** You see a “ChocolateyLocalInstall.ps1” script that is a local Chocolatey install script once you get your NuGet Chocolate package. More on that later!
      3. Run “Chocolately\_Internet\_Install” script again!
         1. The script will now continue to run, download and internalize your Chocolatey offline package. This will take some time.
3. Chocolatey Offline Package Internalized!
   1. Full Package, but not yet!
      1. You now have a full offline Chocolatey package. It will be in the root of “C:\choco-setup”. However, you need to copy one more script from Chocolatey website.
         1. Within Exercise 1
            1. <https://chocolatey.org/docs/how-to-setup-offline-installation>
            2. Copy the PowerShell script. We will refer to this script as “Chocolatey\_Config.ps1”. This script will do a few things. But all you need to do is place it in your “C:\choco-setup\files” for now.
4. Package your Chocolatey License!
   1. Exercise 4 – Copy script.
      1. We will refer this to “choco\_LicensePack.ps1”
      2. <https://chocolatey.org/docs/how-to-setup-offline-installation>
   2. Make a copy of your license before hand!
   3. Run the “Choco\_LicensePack” script.
5. Add More Chocolatey Packages! (You has access to the internet)
   1. Nexus-Repository Chocolatey package.
      1. After Chocolatey is install, go to your Windows PowerShell and type in the following:
         1. choco download nexus-repository --internalize --output-directory="$env:SystemDrive\choco-setup\packages" --source="'https://chocolatey.org/api/v2/'"
         2. This will download and internalize Nexus-repo manager for your MDT server. It’s free and will help you manage your NuGet packages (aka Chocolatey packages) offline during MDT deployment.

**You’re Offline Server**

From this point, set-up a domain, DNS, DCHP, MDT, etc in your environment. Next, spin up a member server that is within the domain and this will have Chocolatey installed and your Nexus Repo manager. So from this point on I will be referring to the “Nexus” member server. Lastly, place your offline choco folder into the root of C.

1. Run “Chocolatey\_Config.ps1” script. (As PowerShell Admin)
   1. This script will remove/add the following:
      1. Remove – Source location on the Internet to get Chocolatey packages.
      2. Add- Source location (Named: “Local”) and a local machine path on your offline computer.
         1. This line is telling Chocolatey where to look to get it Chocolatey packages.
      3. Add your Chocolatey license. Same one as before.
      4. Remove – Chocolatey source location for “Chocolatey” blessed packages.
         1. Chocolatey has a market community store on the internet that will provided “Chocolatey Licensed” products. We are also removing that too.
      5. Add/Install – Chocolatey extension.
   2. Chocolatey is installed in your C drive under “Chocolatey” folder.
   3. Is Chocolatey install?
      1. PowerShell 🡪 Type “choco”.
2. Nexus Repo Manager.
   1. It’s simple!
      1. PowerShell (Admin) 🡪 Type “choco install nexus-repository”
   2. After Nexus Repo Manager is installed.
      1. Information will be presented to you in the PowerShell shell that you need to keep in mind.
      2. A Nexus service will be created on your member server.
   3. Here are the following information on what it will tell you in the PowerShell Shell.
      1. Intranet URL: <http://localhost:8081>
         1. Takes about a minute or two.
      2. Account: admin
      3. Password: Located in “C:\ProgramData\sonatype-work\nexus3\admin.password”
3. Nexus Repo Manager Login.
   1. Intranet URL
      1. Go to you web url to access nexus and log in using “admin” and your password.
      2. Go through wizard and add in your new password for the admin account.
      3. Set up “Anonymous Access” on what you want. Depends on you!
   2. Set up your Test repo and your production repo.
      1. Click on gear up top and click repository.
      2. Create new repository (nuget hosted) for both test and production.
         1. Deployment policy – Disable redeploy.
            1. This will only push out the package once and not again.
      3. Once created, go back to your repo’s and copy/note your Nexus repo URL’s.
   3. Chocolatey Source location.
      1. PowerShell (admin) 🡪 “choco source add –n <*CreateName*> -s <*NexusRepoURL*>”
         1. ***Pro Tip!:*** PowerShell 🡪 “choco list –source <*NexusRepoName*>”
            1. This will check if there are any Nuget/Choco packages in your repo.
4. Push Local Chocolatey/NuGet packages to Nexus Repo Manager
   1. Single Nuget package.
      1. PowerShell 🡪 “choco push <*packageName*.nupkg> --source <*NexusRepoURL*> --force”
   2. Multiple
      1. Go to the folder where you have multiple .nupkg (NuGet packages).
         1. PowerShell 🡪 “Get-ChildItem \*.nupkg | ForEach-Object { choco push $\_.Name} –source <*NexusRepoURL*> --force”
            1. This can be done anywhere on domain, but you need to have Chocolatey installed first. Hence the “choco” commandlet. Also, for source in above command it is double dash. Same goes for the force command.
            2. If you don’t include “--force” chocolatey will error out. Chocolatey is security first, that error is because it wants an SSL. But, “--force” will by past it.

**You’re Clients**

Now that you have some tools installed and an infrastructure. You can now prep your clients to look at your Nexus server to get its Chocolatey/Nuget packages. Nexus can host Nuget, but also regular files too in a **raw repo**. We will get a script from Chocolatey website and input our Nexus intranet url. We will then feed it to our clients and they will know where to look for our packages.

1. Nexus Repo Manager.
   1. Copy your specific repo you want your clients to look at, by clicking the repo and getting its internal URL.
   2. Chocolatey website.
      1. https://chocolatey.org/install#organization
      2. Go to the download/install page for Chocolatey and click on the “Organization” tab.
      3. Input your Nexus intranet URL into number 3.
         1. This will automatically create a streamline script for clients on where to look on your network for its packages.
      4. ***IMPORTANT!:*** This script will is the script that will install Chocolatey onto your clients. We will refer this as “chocolatey\_ClientInstall.ps1”
   3. Copy the script into your newly created **raw repo** within Nexus.
   4. Afterwards, make a new PowerShell script with this in it as seen below. We will refer this to as “choco\_ClientConfig.ps1”. Tell your clients to run “Choco\_clientConfig.ps1” to install choco and config.

