**How to setup Chocolatey Server for an Offline Repository**

Chocolatey is a Windows Software Package Manager, build on Powershell and Nuget, allowing administrators to remotely install and update all software installed with chocolatey with Powershell commands. This is a guide to show how to setup a Chocolatey Server to host an internal package repository, allowing for software and version control, as well as saving hits on the WAN connection

**Prerequisites**:

1. Windows 8.1/2012 R2+ to act as the Chocolatey Server with access to the internet. Recommended to have a second drive to store packages
2. Windows 7+ to act as the test client for packages and package building
3. Go to <https://github.com/Michael-Lindauer/Chocolatey-Resources> and clone the repository to get all the necessary resources

**Directions:**

1. On the server, install Chocolatey by opening Powershell with Admin Rights and pasting the following: Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('<https://chocolatey.org/install.ps1>'))
2. Once chocolatey is installed, run choco install chocolatey.server
3. Navigate to C:\tools\chocolatey.server\ folder and edit the permissions of the Web.Config. Add Domain Admins (or Administrators if not in an AD environment) and give them Modify rights so we can edit this file. Launch Notepad as an Administrator and then open the Web.Config file.
4. Edit the Web.Config entries as follows:
   1. **<user name="choco" password="rocks" />** - This is the default username and password for basic authentication for IIS and is not part of AD. You can either change these such that it will require the username and password for each software install/update in addition to Domain Admin rights, or simply delete it if you feel Domain Admin rights are good enough
   2. **<requestLimits maxAllowedContentLength="2147482548" />** - This specifies the largest size package that may be uploaded to the chocolatey server. If you need to create packages OVER 2GB, then increase this value
   3. **<add key="apiKey" value="chocolateyrocks" />** - This is a shared password that is used to upload packages to your repository. Change this to a different value. Note that this is sent in cleartext until you apply an SSL cert to the IIS server, and chocolatey will warn you of this each time you upload to it
   4. **<app key="packagePath" value="~/App\_Data/Packages" />** - This is the default path to host packages. Change this to your dedicated drive for hosting Chocolatey packages, such as value="S:\Packages"
   5. **<add key="forceLocalChocolateyPackage" value="false" />** - Change this to true so that chocolatey packages will default to the local repository rather than online
5. Open Powershell with Admin rights and type iisreset to restart the webpage with this new config. Verify that you can get to your repository's webpage by going to: http://YOUR\_REPOSITORY\_NAME/ from the test client. You may need to troubleshoot Windows Firewall to allow access to this website. YOUR\_REPOSITORY\_NAME will be the DNS name of the chocolatey server
6. Add the chocolatey installer package to the chocolatey repository by opening a Powershell prompt with Admin credentials and entering:
   1. choco push C:\ProgramData\chocolatey\lib\chocolatey\chocolatey.nupkg --source "http://YOUR\_REPO/chocolatey" -k="YOUR\_API\_KEY" --force
7. Copy and edit the installChocolateyFromOfflineRepo\_Template.ps1 script to edit in your repository name, and the username and password you created during step 4 (if you didn't delete them). This script will install the chocolatey client from your internal repo, add your internal repo as a source, and then remove the official chocolatey repository, enabling a purely offline source to pull packages from
8. Run your version of the installChocolateyFromOfflineRepo\_Template.ps1 on the **test client** from an Admin Powershell console, and verify that it is installed by running Choco in the powershell client afterwards.
   1. From here, there is a variety of methods to remotely run/install the script to the rest of the clients you'd need the chocolatey client installed on, such as setting up a GPO that runs a task to run your script, a RunOnce Registry entry to run it, or manually running it via PSSession. If you deploy to many clients at once, it's recommended adding a random delay so that your chocolatey server isn't hit all at once
9. From here, Chocolatey server is setup and ready to start serving internal packages. Check the 'How to create offline chocolatey installation packages' instructions