Using Nextcloud with Windows Terminal Server

Nextcloud is a very good tool for sharing files across multiple machines. It is also very good as a backup, since the agent can keep versions and deleted files available if you want. When a user is logged in, any changes to their files can be sync'd to the Nextcloud server in near real time.

Since our primary backup service is Unix based, we can use Nextcloud as a backup, then backup the Nextcloud if we want.

The only issue is Public Documents on a Terminal Server. In this case, you need one user configured to back up the Public Documents, and it will only do that if the user is logged in. Nextcloud provides a command line utility you can set as a task, but it is pretty flaky, and the documentation is not that good.

Procedure

Our procedure is to create a user with Administration privileges and auto-login, configured to back up Public Documents. While this can use additional resources on your Terminal Server, the resources appear to be minimal, and you are getting near real-time backups.

1. Create a Nextcloud user for backing up Public Documents
    1. Note, this is also a good place to create shared calendars and contact lists
2. Create a Windows user on the terminal server, with Admin privileges
3. Set the Windows admin user up to sync Public Documents to the Nextcloud user
    1. Be sure to set Nextcloud to automatically run on login
4. Set the Windows admin user to automatically be logged in on system startup

Warning: If the admin user is ever logged out, backups of Public Documents will stop without warning. Therefor, access to the Windows admin user should be limited.

Auto Login User

Easy Way

Microsoft provides a manual way to automatically log in a user on system startup, but they also have a utility in their sysinternals package that provides a basic GUI to allow you to do it. See https://learn.microsoft.com/en-us/sysinternals/downloads/autologon. An additional benefit is that the Autologon utility encrypts the password. This is the recommended procedure.

Manual Setup

If you want to set up the autologon manually, create a .reg file with the following content, then run it.

autologin.reg

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon]
"DefaultUserName"="USERNAME"
"DefaultPassword"="PASSWORD"
"AutoAdminLogon"="1"
```

Obviously, replace USERNAME and PASSWORD with the correct values for your system. **Warning**: the password is stored in plain text using this approach.

Lock Screen on new session

The only issue with this is that the user is logged in, with an open session when the server boots. One process I have not tried, but is well documented, is to create a startup script for that user which will automatically lock the screen, requiring a password to unlock.

Create a file, autolock.bat with the following contents

autolock.bat

```
@echo off
%windir%\System32\rundll32.exe user32.dll,LockWorkStation
exit
```

Now, place this in the autostart folder for the user. Log in as the user and run the command

```
Shell:Startup
```

Which will open the location for the individual users startup. Copy the file into that location and, whenever the user logs in, their screen will automatically lock.

Links

I found information in the following links which I used in this article

- https://learn.microsoft.com/en-us/sysinternals/downloads/autologon
- https://serverfault.com/questions/840557/auto-login-a-user-at-boot-on-windows-server-2016
- https://jc-lan.org/2022/06/02/autologon-user-at-startup-in-windows-server/
- https://learn.microsoft.com/en-us/troubleshoot/windows-server/user-profiles-and-logon/turn-on-automatic-logon
- https://serverfault.com/questions/840557/auto-login-a-user-at-boot-on-windows-server-2016
- https://4it.com.au/kb/article/how-to-automatically-logon-and-lock-a-windows-server/

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/microsoft_windows/terminalserver/nextcloud

Last update: **2023/11/08 09:11**