Mounting Network shares with fstab

Auto Mounting a Network SMB or CIFS Drive

> As always, backup your working fstab first:
> `sudo cp /etc/fstab /etc/fstab.bak`

The following is a way to automatically mount a network share without the risk of it blocking you computer from booting. This is useful for non-critical drives such as media or backups. Review the params before saving your fstab file, you may need to adjust some!

In this example we will mount the SMB share `photos` to the local directory `/mnt/photos`. Replace with your shared drive and preferred local folder.

1. Create the directory your drive will mount to. This can be owned by any user: EG. root
    1. `sudo mkdir /mnt/photos`
2. Add the mount as a new line to /etc/fstab
    1. `//IP_ADDRESS/photos /mnt/photos cifs vers=3.0,nofail,noauto,x-systemd.automount,username=USERNAME,password=PASSWORD,uid=1000,gid=1000,dir_mode=0777,file_mode=0666 0 0`
    2. Read the explanation below before continuing to save!
3. Make sure you didn't make a mistake. Validate your fstab file changes with:
    1. `sudo findmnt --verify --verbose`
4. Tell the system to re-load the fstab changes with:
    1. `systemctl daemon-reload`
    2. Because the example here sets `noauto`, you cannot mount them with the "mount -a" command
5. Usually the network shares will mount automatically once you try and access the local directories. Sometimes this takes a few minutes the first time. You can speed this up with the following command:
    1. `sudo systemctl restart local-fs.target`

Explanation of line #2:

- **//IP_ADDRESS/photos**
  - IP and name of the network share
- **/mnt/photos**
  - Local folder to mount to
- **cifs**
  - Identifies this as an SMB/CIFS share
- **vers=3.0,nofail,noauto,x-systemd.automount,username=USERNAME,password=PASSWORD,uid=1000,gid=1000,dir_mode=0777,file_mode=0666**
  - **vers=3.0** - Optionally identify the version of SMB used to connect. Usually default to 1.0 if not specified. This will depend on your specific network share
  - **nofail** - Don't fail booting if there's an issue with this drive
    - Not required here but I like it for extra safety
  - **noauto** - Don't mount this drive automatically at boot or when `mount -a` is called
  - **x-systemd.automount** - Automatically mount the drive when you attempt to access it using the local directory
  - **username=,password=** - Your SMB user and password go here. Leave these off if there isn't one.
    - Instead of placing credentials directly here you can instead put them in a file and reference that file here:
      - credentials=/home/[username]/.sharelogin
      - Inside the file place the following two lines with your credentials:
        - `username=[username]`
          `password=[password]`
  - **uid=1000,gid=1000**
    - Access controls: User and Group the drive's files and folder should be mounted as. You can look up the ones you want by running `cat /etc/passwd` and finding your desired local user. The two numbers shows are the uid and gid in that same order.
    - I HIGHLY suggest you use IDs and not names here. If you use names they must match EXACTLY on your local machine and on the network share.
  - **dir_mode=0777,file_mode=0666**
    - Set the folder and file permissions to whatever you need/want them to be. The permissions listed here allow EVERYONE to read and write files, but no one to execute them for safety. You may want to lower these permissions if you don't want other users to read/write, or add execute permissions if you want to live dangerously and trust all the files on the share.
- **0 0** - These last two zeros are additional mount params for dump and pass. For our (and most) purposes these can remain zero. We aren't using dump and pass is used for boot order checks, but we aren't mounting these at boot. 0 for both options disables both these features.

Notes:

You may have a different preference about the **noauto** param preventing mounting a boot time. If you absolutely need your drive to mount first thing and it should hold up booting until it has mounted, you can remove that option. If you want booting to fail entirely if the drive is not available, you can also remove the **nofail** param.

Revision #3
Created 2025-03-10 16:59:31 UTC by William
Updated 2025-03-10 17:43:55 UTC by William