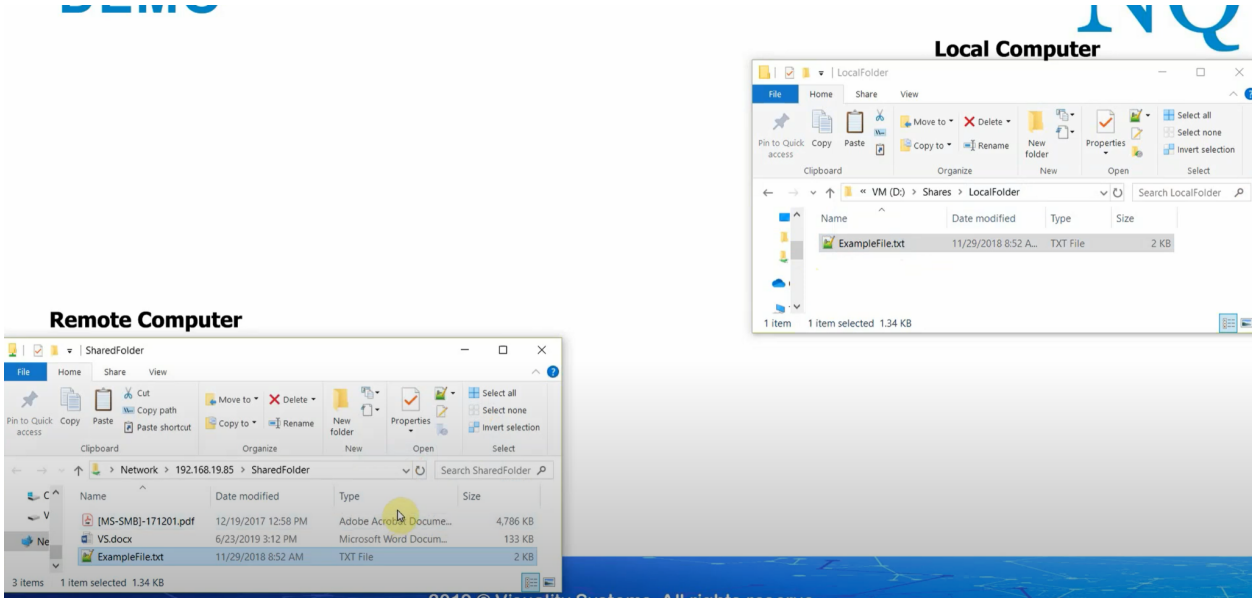
Windows support to File sharing protocols, SMB and NFS

SMB = Server Message Drive

SMB allows the Windows operating system to map a drive to remote file share and have it accessed on windows like a normal hard drive.

smb was renamed to CIFS (common internet file system), but then reverted back to smb, so now cifs is a dialect of smb



Features Described:

- Ability to copy files from local computer to remote computer
- Ability to copy files from remote computer to local computer
- Ability to directly open files on remote computer

What is SMB?

SMB - Server Message Block Protocol - is a client-server communication protocol used for sharing access to files, printers, serial ports and other resources on a network. [source]

Servers make file systems and other resources (printers, named pipes, APIs) available to clients on the network. Client computers may have their own hard disks, but they also want access to the shared file systems and printers on the servers.

The SMB protocol is known as a response-request protocol, meaning that it transmits multiple messages between the client and server to establish a connection. Clients connect to servers using TCP/IP (actually NetBIOS over TCP/IP as specified in RFC1001 and RFC1002), NetBEUI or IPX/SPX.

How does SMB work?

Once they have established a connection, clients can then send commands (SMBs) to the server that allow them to access shares, open files, read and write files, and generally do all the sort of things that you want to do with a file system. However, in the case of SMB, these things are done over the network.

What runs SMB?

Microsoft Windows operating systems since Windows 95 have included client and server SMB protocol support. Samba, an open source server that supports the SMB protocol, was released for Unix systems.

```
# how to access smb folder from linux
https://www.linuxbabe.com/ubuntu/install-samba-server-file-share

# install software for smb
sudo apt install cifs-utils
# create a mount folder
sudo mkdir /mnt/samba-private
#
sudo mount -t cifs -o username=SMB_USERNAME
SERVER_ADDRESS,password=SMB_PASSWORD,domain=SMB_DOMAIN /mnt/samba-private/
# do it from fstab
sudo vim /etc/fstab
# add it at the end of the file
# x-systemd.automount => tells systemd to create an automount unit for the
file system. We use this because it ensures the remote filesystem is
mounted only after there's network access
# _netdev => This specifies that the mount requires network
# credentials => Linux should look for credentials in the
/etc/samba-credential.conf
# uid=1000,gid=1000 => By default the mounted filesystem would be owned by
the root user. We use uid and gid to change the ownership of the
filesystem. Normally you use your own uid and gid
```

```
# x-gvfs-show =>
SERVER_ADDRESS /mnt/samba-private cifs
x-systemd.automount,_netdev,credentials=/etc/samba-credential.conf,uid=100
0,gid=1000 0   0

# add your credentials to /etc/samba-credential.conf file
username=SMB_USERNAME
password=SMB_PASSWORD
domain=WORKGROUP

# this will start the mount
sudo mount -a

# example
mount -t cifs //***.***.com/****** /mnt/dnafiles -o
username=****,password=****,domain=****

# use smbclient to download files
smbclient -L //***.***.com/****** -U domain\\username

smbclient //server/share -c 'cd /path/to/file; put myfile'

smbclient //***.***.com/****** -U doamin\\username%password -c "cd
needed_folder_on_server" -Tc /mnt/dnafiles/file.tar



# run the smb operation from the docker file
ARG LOCALMOUNT=$HOME/dnafiles
RUN mkdir $LOCALMOUNT
ARG REMOTEMOUNT=//***.***.com/******
ARG SMBUSER=domain\\username
ARG SMBPASS=password
RUN echo "//***.***.com/****** /home/spring/dnafiles cifs
user=domain\\username,pass=password 0 0" >> /etc/fstab
# same as above but with variables
RUN echo "$REMOTEMOUNT $LOCALMOUNT cifs user=$SMBUSER,pass=$SMBPASS 0 0"
>> /etc/fstab
```

```
# create kubernetes secret for cifs username/password
apiVersion: v1
kind: Secret
metadata:
  name: cifs-secret
  namespace: ****
type: fstab/cifs
data:
  username: '*****'
  password: '*******'
```

NFS = Network File System
nfs is good for host authentication, and is usually simpler to implement than SMB

When sharing between two windows => use SMB
When sharing between two linux => use NFS

```
# ssh into the server that is going to share the drive
ssh USERNAME@SERVER_ADDRESS
# install NFS software
sudo apt install nfs-kernel-server nfs-common rpcbind
# create folder for NFS
sudo mkdir /var/nfs/public
# give others right to access it (RISKY)
sudo chmod 777 /var/nfs/public
# edit this file
sudo vim /etc/exports
# paste it in the end of that file, allows mentioned ip address that need
to access the shared folder
/var/nfs/public IP_ADDRESS_1(rw,sync,no_subtree_check)
IP_ADDRESS_2(rw,sync,no_subtree_check)
# allow firewall to use NFS
sudo ufw allow nfs
sudo reboot
# show the mount server
showmount -e
```

```
# the client that wants to use the NFS server
# create a folder for the mount and give it all accesses (can restrict it
if needed)
sudo mkdir /mnt/nfs-public
sudo chmod 777 /mnt/nfs-public
# install software needed to access the shared drive
sudo apt install rpcbind nfs-common
# mount the shared folder from server to client
sudo mount SERVER_ADDRESS:/var/nfs/public /mnt/nfs-public
# edit this file, so that we don't need to rewrite the above code on
client each time
sudo vim /etc/fstab
# add it at the end of the file
SERVER_ADDRESS:/var/nfs/public /mnt/nfs-public nfs rw 0 0
```

FSTAB file: https://www.youtube.com/watch?v=A7xH74o6kY0