File Transfers, Synchronisation & Shared Storage

Unison, NFS, rSync, SCP

- Unison
- NFS
- rSync
- FTP Troubleshooting
- vSFTPd

Unison

==========

What is Unison?

Install Unison

Unison will need to be installed on both servers that are sharing files.

RHEL:

```
yum install unison
```

Debian:

```
apt install unison
```

Configuration of unison

Before configuring unison itself, you need to ensure that the hosts have shared keys (since this connection is made via SSH).

```
ssh-keygen -t rsa


ssh-copy-id root@otherserver
```

Once that's sorted, the unison service itself can be configured.

default unison config file:

/root/.unison/default.prf

```
root = /
root = ssh://b4sed-01//

path = var/spool/cron/
path = etc/passwd
path = etc/shadow
path = etc/group
path = etc/motd
path = etc/drbd.conf
path = etc/cluster/cluster.conf
path = etc/php.ini
path = etc/nginx/

ignore = Name access.log*
```

To have unison run automatically, you'll need to configure a cron:

```
vi /usr/local/bin/sync.sh
```

Contents of the file (the SISTER= value needs to be updated).

```
#!/bin/bash

SISTER=ABC-WEBDB-01

[ -f /var/run/file_sync.pid ] && exit 1;

trap "{
        rm /var/run/file_sync.pid;
        exit;
}" EXIT;

trap "{
        echo 'Bailing out!' 1>&2
        ssh -T -p2020 root@$SISTER <<<'killall unison; exit' &>/dev/null
}" KILL ABRT INT TERM HUP SEGV

touch /var/run/file_sync.pid
/usr/bin/unison -sshargs "-p 2020" -batch -terse -silent -owner -group -numericids -prefer /
```

Once added, then the cron needs setting up

```
crontab -e


* * * * * /usr/local/bin/sync.sh > /root/.unison/unison.log
```

You would then need to add a cron on the 2nd server, you'll need to add this with the '-prefer' omitted, as below:

```
#!/bin/bash


SISTER=ABC-WEBDB-01


[ -f /var/run/file_sync.pid ] && exit 1;


trap "{
        rm /var/run/file_sync.pid;
        exit;
}" EXIT;


trap "{
        echo 'Bailing out!' 1>&2
        ssh -T -p2020 root@$SISTER <<<'killall unison; exit' &>/dev/null
}" KILL ABRT INT TERM HUP SEGV

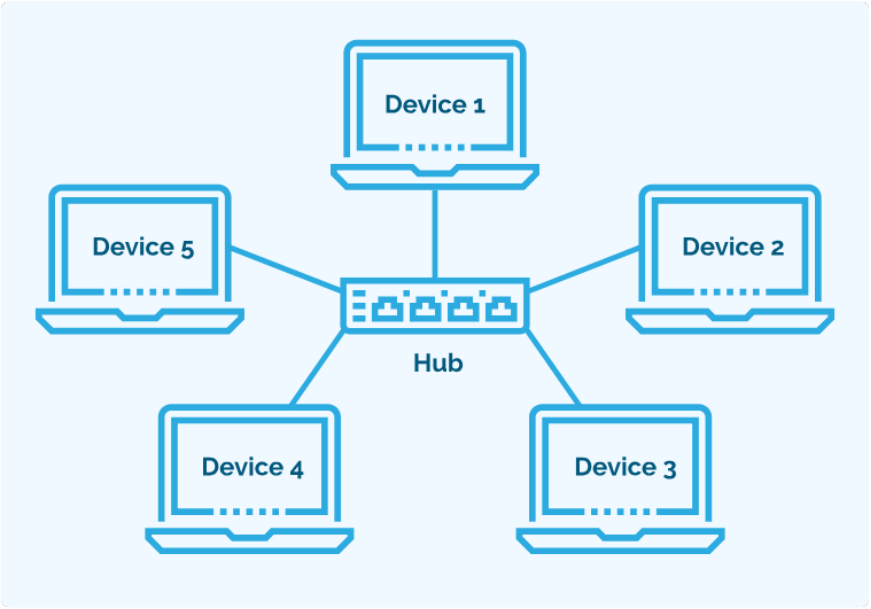

touch /var/run/file_sync.pid
/usr/bin/unison -sshargs "-p 2020" -batch -terse -silent -owner -group -numericids /
```

==========

Multi-server (2+) Unison setup

To preface this; Advanced Unison topologies such as the one I detail on this page aren't ideal, one reason for this is that any change requires that unison be run bidirectionally for each node - meaning that synchronisation is definitely not instant. Depending on the content type, a better option might be to have an NFS Share configured.

There are different methods that can be used to setup Unison when more than 2 servers are installed, for example, ring and star topology. In this example, I've focussed on the star topology:



To set the context for this example and give some explanation;

This setup consists of 6 servers, with 1 of those servers acting as the 'master' - being the middle of the star.

Unison should be configured on only the master node, from here, you'll need to configure the .prf files and alter the Unison cronjob script accordingly.

Configuration

1. Install Unison on the master node

Decide which node is going to be the master, and install Unison on there.

2.

==========

NFS

NFS

(Guidance based on RHEL/CentOS 7, packages and commands may differ depending on OS)

==========

NFSSHARE SERVER CONFIGURATION

1. Install NFS packages on NFS Server

```
yum install nfs-utils rpcbind
```

2. Start on-boot

If this is **not** a cluster then start these at boot time:

```
systemctl enable --now rpcbind nfs-server nfs-lock nfs-idmap
```

3. exports

Once installed, we can look to configure the nfs share within the /etc/exports file:

```
vim /etc/exports
```

Add the below config (updated with IP of the client server that need to be able to access the nfsshare.

```
/nfsshare IP_IP_IP_IP(rw,sync,no_root_squash)
```

To specify multiple client servers, add separate lines, as below:

```
/nfsshare IP_IP_IP_IP(rw,sync,no_root_squash)
/nfsshare IP_IP_IP_IP(rw,sync,no_root_squash)
```

```
/nfsshare IP_IP_IP_IP(rw,sync,no_root_squash)
```

Once added, you can publish the exports changes using the below command:

```
exportfs -a
```

==========

NFS CLIENT CONFIGURATION

1. Install the required NFS packages:

```
yum install nfs-utils
```

2. Mount the NFSSHARE (won't persist reboot until next step):

Firstly, create the file path that you wish for the NFSSHARE to be mounted to:

```
mkdir /nfsshare
```

The below command mounts the NFSSHARE. Ensure to replace the IP_IP_IP_IP:nfsshare with the NFS server IP, and location of the NFSSHARE on the NFS server. Finally, also ensure to update the second /nfsshare on the command to the path you wish for the share to be mounted to on the client server.

```
mount -t nfs IP_IP_IP_IP:/nfsshare /nfsshare
```

3. Test the configuration

Test that mounting the nfsshare has worked using the df -h command. You should now see an entry dedicated to the NFSSHARE, as below:

4. Permanently configure the NFSSHARE

Once you've confirmed that the NFSSHARE is working, you can then look to add the fstab entry that will allow this configuration to persist time & reboots.

```
vim /etc/fstab
```

Add the below link, ensuring the update the syntax as mentioned above

```
IP_IP_IP_IP:/nfsshare /nfsshare nfs rw,relatime,vers=4,_netdev,timeo=100,retrans=4 0 0
```

Once you've added this, ensure that this is now working by forcing all fstab-configured mounts to mount:

```
mount -a
```

rSync

==========

What is rSync?

rSync is a file transfer command that can be used for both local and remote transfers.

> Remember to be careful with rSync - it's a synchronisation command and can overwrite files.
>
> example:
>
> I have 2 directories:
> /:
> directory1:
> |_>file1
> directory2:
> |_> file1
> |_> file2
> If I run `rsync directory1 directory2`, then file2 would be removed, and file1 would be synced.

---

rSync command syntax

rsync [option] [user@host]

Basic local file transfer

```
rsync /sourcefile /destinationfile
```

Basic remote file transfer:

```
rsync /sourcefile user@IP_IP_IP_IP:destination/location/on/remote/server
```

rSync Tunelling (secure rsync over SSH)

```
rsync -e ssh /sourcefile user@IP_IP_IP_IP:destination/location/on/remote/server
```

flags:

-a	archive
-z	gzip compression
-u	skip existing files at destination
-r	recursive
-P	show progress of transfer
--exclude=	specify files to ignore (ie "*.log") (needs to be paired with --include). Appended to command prior to source file specification.
--include=	(only needed if using --exclude) If you're excluding files from an rsync transfer, you'll need to add --include="*". Appended to command prior to source file specification.
--dry-run	Test run of rSync to display the changes that would be made. No file synchronisation is performed with this option enabled. Appended to end of command.

==========
==========

FTP Troubleshooting

==========

FTP Passive Mode

Passive ports are used to allow multiple FTP connections by moving open FTP connections from port 21 to a port specified in the passive port range (40,000-40,100 is the standard ANS passive port range).

If your FTP client is using passive mode, you'll usually see an output similar to the below. We can use this to calculate the port being used for passive mode (which we can then check for any restrictions on).

```
80,244,185,220,156,149
```

First 4 numbers are the server IP,

Multiply 5th number by 256 = x

X+6th number = passive port which is being used

156x256=39936+149=40085

SFTP

SFTP, which stands for Secure File Transfer Protocol, is a network protocol that provides file access, file transfer, and file management functionalities over any reliable data stream. Unlike FTP (File Transfer Protocol), which is often used with FTP over SSL/TLS (FTPS) for security, SFTP inherently provides secure file transfer through the SSH (Secure Shell) protocol.

The primary cPanel/Plesk user can use SFTP if enabled.

==========

FTP Troubleshooting

---

cPanel

Identify which FTP server is running

```
lsof -i:21
```

For Pure-FTPd:

```
/var/cpanel/conf/pureftpd/local
```

For ProFTPD:

```
/var/cpanel/conf/proftpd/local
```

Add this line to set which ports your server should use.

```
PassivePortRange: 40000 40100
```

If your server is behind a firewall and you are seeing unroutable address errors, add the following line, replacing `123.123.123.123` with your server's public IP:

```
ForcePassiveIP: IP_IP_IP_IP
```

Restart `Pure-FTPd` by running:

```
/usr/local/cpanel/scripts/setupftpserver pure-ftpd --force
```

Allow inbound connections on the passive port range.

Plesk

Plesk also uses the ProFTPD server, but the configuration is slightly different.

Plesk Onyx:

Edit/create the file `/etc/proftpd.d/55-passive-ports.conf`

Add the following configuration this file:

```
<Global>
PassivePorts 40000 40100
</Global>
```

Restart the FTP service to pick up the changes:

```
systemctl restart xinetd
```

On your firewall, allow inbound connections on the passive port range.

If your server is behind a firewall and you are seeing unroutable address errors, add the following line, replacing `123.123.123.123` with your server's public IP:

```
MasqueradeAddress IP_IP_IP_IP
```

---

vSFTPd

`vsftpd` (Very Secure FTP Daemon) is a popular FTP server for Linux systems. To use vSFTPd, you'll need to install the `vsftpd` package.

---

vSFTPd Config Options

pasv_enable=YES pasv_min_port=40000 pasv_max_port=40100	Enable passive mode and set port range
pasv_address=your.external.ip.address	Specify FTP listening IP.
anonymous_enable=NO	Disable anonymous access
write_enable=YES	Enable file uploads
local_enable=YES	Enable local users
chroot_local_user=YES	Enable user chroot
chroot_list_enable=YES chroot_list_file=/etc/vsftpd.chroot_list	Configure chroot bypass for users.
ssl_enable=YES	Enable SSL
rsa_cert_file=/etc/ssl/certs/vsftpd.pem rsa_private_key_file=/etc/ssl/private/vsftpd.pem	Specify SSL certificate files for FTPS/ES
force_local_data_ssl=YES force_local_logins_ssl=YES	Force SSL usage
ssl_tlsv1=YES ssl_sslv2=NO ssl_sslv3=NO	SSL Protocol options
ssl_ciphers=HIGH	SSL cipher
force_local_data_ssl=NO force_local_logins_ssl=NO	Enable FTPES

Adding users for FTP usage

1. In order to create an account that can use VSFTPd, you will first need to set up a user on the server that you want to transfer files to and from.

```
useradd guest
```

2. Once created, you'll want to set a password for that user

```
passwd guest
```

3. Also disable shell access for the user

```
usermod -s /sbin/nologin guest
```

Chrooting users

Chrooting a user in `vsftpd` ensures that the user is restricted to their home directory and cannot navigate to other parts of the file system.

1. Add the user

Either alter the existing user's home directory, or add a new user to be used for FTP

```
sudo adduser --home /var/ftp/ftpuser ftpuser
```

2. Set a password

```
passwd ftpuser
```

3. Set home directory permissions

```
sudo chown ftpuser:ftpuser /var/ftp/ftpuser
sudo chmod 755 /var/ftp/ftpuser
```

4. Configure vsftpd

Ensure that the following is present within /etc/vsftpd.conf

```
chroot_local_user=YES
```

FTPS & FTPES

```
# Enable SSL
ssl_enable=YES

# Paths to the SSL certificate and key
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem

# Require SSL for both data and login
force_local_data_ssl=YES
force_local_logins_ssl=YES

# Allow anonymous users to use SSL
allow_anon_ssl=YES

# SSL protocol options
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO

# Strong ciphers
ssl_ciphers=HIGH

# Optional: Require SSL reuse for data connections
require_ssl_reuse=NO

# Enable Explicit SSL (FTPES)
# By default, vsftpd will use implicit FTPS (default port 990)
# If you prefer explicit FTPS (FTPES), enable the following:
force_local_data_ssl=NO
force_local_logins_ssl=NO

# Explicitly request SSL for login
ssl_request_cert=YES
```

==========
==========