

# Using Rsync for DR Purpose

## Symptoms

You have a SnapHA configured but would like to setup a third node for DR purpose should in case you need to temporarily have access to your data should in case your HA pair is not reachable for whatever reason

## Purpose

The following provides guidance on how to implement our rsync script to meet the need of a DR use case

1. Please enable Authentication with password on your SoftNAS instances, both Primary and Target by performing the following:
  - a. From UI go to Settings --> General System Settings --> Servers --> SSH Server --> Authentication --> and change Allow authentication by password? to "YES" and Allow login by root? to "YES"
  - b. Restart ssh the server

NOTE: Please take note of these changes as you will need to revert them back to their defaults for security reasons.

2. From the SoftNAS DR node let's setup SSH keys to push to the Primary and Target node to get them ready for the rsync script:

- a. Create the RSA Key Pair:  
`# ssh-keygen -t rsa -b 2048`
- b. Use default location `/root/.ssh/id_rsa` and don't setup a passphrase
- c. The public key is now located in `/root/.ssh/id_rsa.pub`
- d. The private key (identification) is now located in `/root/.ssh/id_rsa`
- e. Permissions for the private key should be 0600  
`# chmod 600 /root/.ssh/id_rsa`

3. Copy the public key to both the Source and Target node using the `ssh-copy-id` command:

```
# ssh-copy-id root@Primary_IP
# ssh-copy-id root@Secondary_IP
```

4. Please revert back the changes made in step # 1
5. Please download the rsync script from [here](#) .It has already been optimized but you can tweak the 'bwlimit' as you see fit since rsync is a resource intensive service.
6. Make sure the name is `rsync-pull.sh`, it has the execute bit on, 'chmod +x rsync-pull.sh', it is placed in the directory you are running the line below:

```
#. Usage "rsync-pull.sh -l /path/to/log_file -v -u user -k /path/to/ssh_key -s remote_server_hostname -p /path/on/remote/host /path/to/sync"
```

Example:

```
# sh ~/rsync-pull.sh -l rsysc.log -v -u root -k ~/.ssh/id_rsa -s Target_IP -p /pool-name1 /pool-name2 /
shsh
```

7. Now we can automate the process to allow it to run at a set interval. In this demo I'll set the rsync script to run every hour

- a. Here is the an example of job definition:

```
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
```

- b. Example of creating the cron Job

```
#. crontab -e
```

c. Add this line. It is set to an hour interval in this guide.

```
#. 0 * * * * ~/rsync-pull.sh -l /var/log/rsync.log -u root -k ~/.ssh/id_rsa -s Target_IP -p /pool-name1 /pool-name2 /
```