

rsync backup

Backup your data to a NAS

In this example we will cover how to backup (optional: automatically) the most important data folder /userdata, which contains all important (custom) configurations and games of your Batocera system so that in case of emergency you will have all your data in a recoverable state.

Example for a Synology NAS

The following tutorial is based on a [Synology NAS](#) which is running on Synology's DSM version 6.x or later.

This tutorial assumes you have already set up the following:

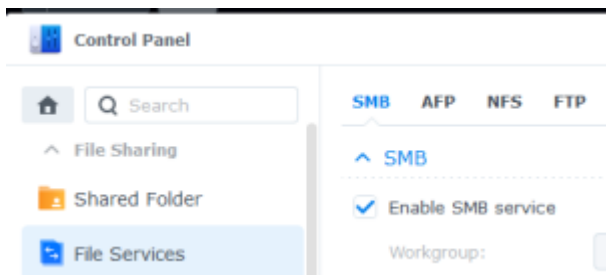


- Static IP address and/or DNS hostname for your Synology NAS (you can do this via your DHCP/DNS server (mostly your router))
- Static IP address and/or DNS hostname for your Batocera system (you can do this via your DHCP/DNS server (mostly your router))
- Optional: If you want to set up an automated backup (which is also covered in this tutorial), your Batocera system has to be [Wake on LAN \(WoL\)](#) capable.

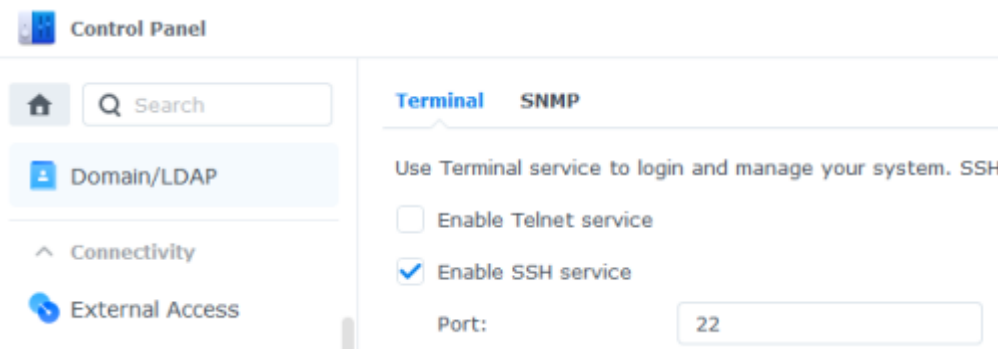
Preapring your Synology NAS

For this tutorial you will need to enable the following services on your Synology NAS via the Synology DSM web GUI as follows:

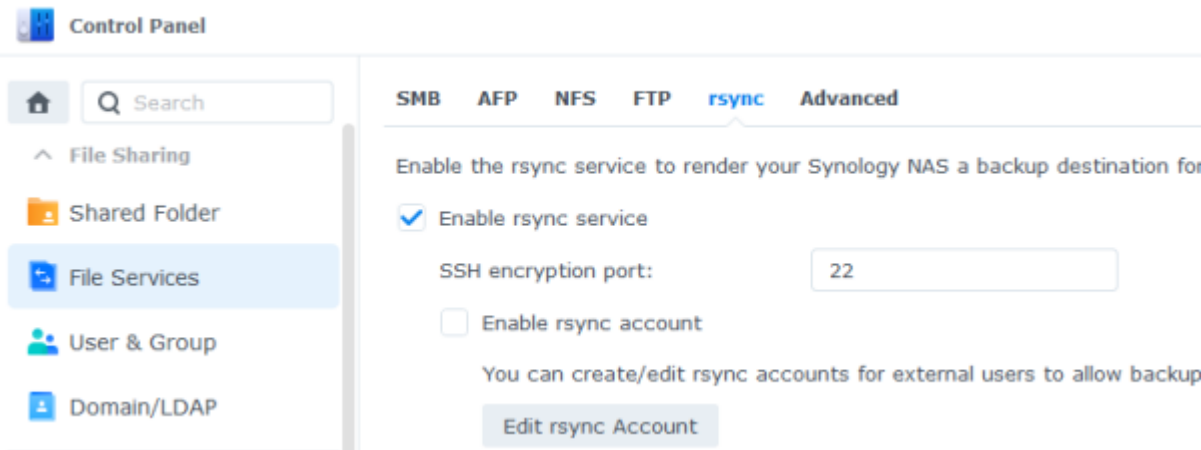
- **SMB** → Navigate to Control Panel → File Services → SMB → Check Enable SMB service:




- **SSH** → Navigate to Control Panel → Terminal & SNMP → Check Enable SSH service and for Port: set the value 22:



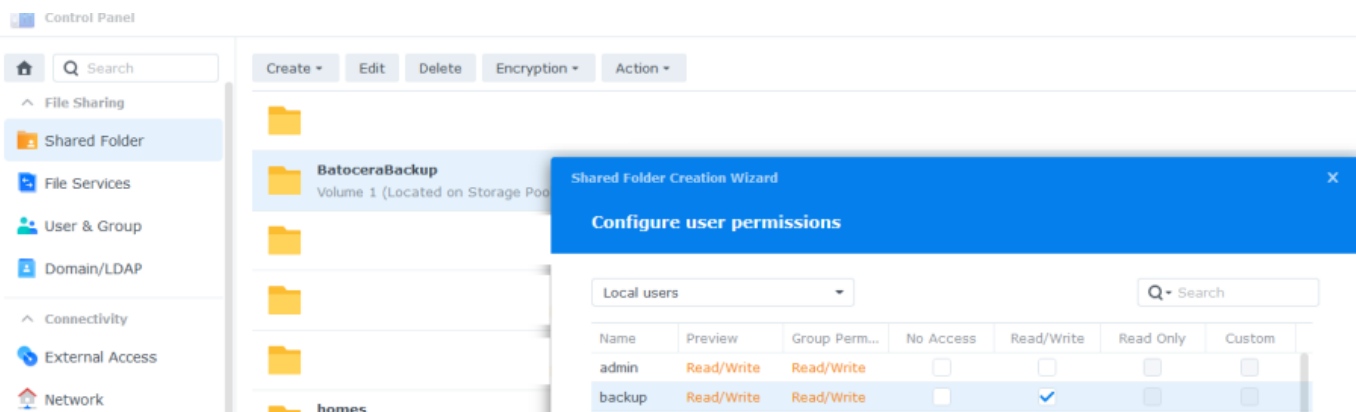
- **rsync** → Navigate to Control Panel → File Services → rsync → Check Enable rsync service and for SSH encryption port: set the value 22:



 You don't have to worry that port 22 is used by SSH *and* rsync at the same time. This is a common setup and the functionality of each service will not be interfered/disrupted in any way!

Next, you will need to [create a user](#) (standard local user without administrative privileges) for backup purposes on your Synology NAS. For this tutorial we will name the user backup.

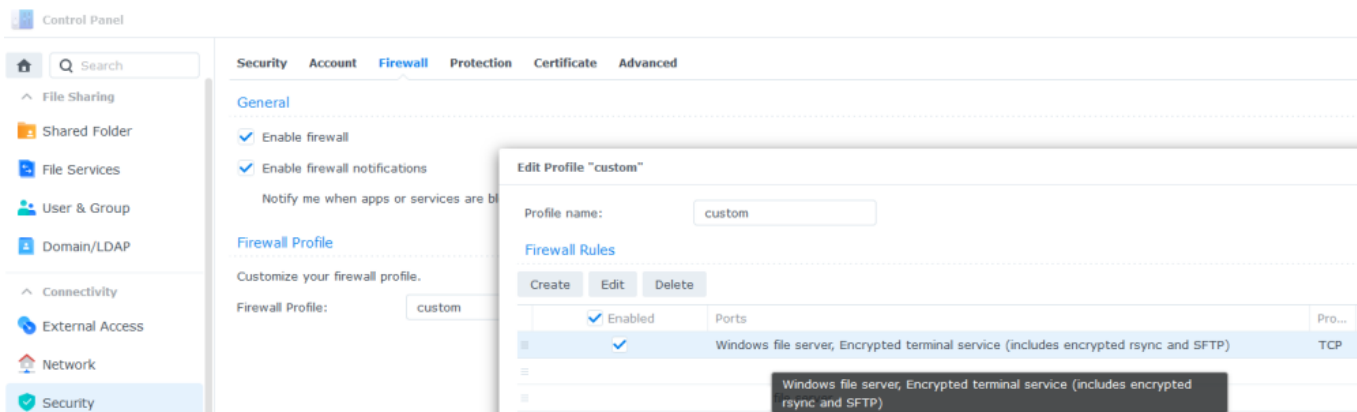
Next, you will need to [create a shared folder](#) where your Batocera backup data can be stored to. For this tutorial we will name the shared folder batoceraBackup. You have to set *read/write* permissions on that shared folder for the specific user you have created before:



Next, you will need to [create a firewall rule](#) on your Synology NAS so that your Batocera system can send its data over the network to the Synology NAS. The rule must allow the following two services:

- Windows file server
- Encrypted terminal service (includes encrypted rsync and SFTP)

In addition it is also strongly(!) recommended to limit the firewall rule to allow only the static IP address of your Batocera system (that's why it is important to set a static IP for your Batocera system!):



Now **SSH** into your Batocera system and execute the following command to create a custom scripts folder:

```
mkdir -p /userdata/scripts
```

Preapring your Batocera system

On the following steps we will cover on how to create three scripts here:

- `romsBackup.sh` → Which will backup your `/userdata/roms` folder.
- `userdataBackup.sh` → Which will backup your `/userdata` folder, excluding the `/userdata/roms` folder as it *may* take a very long time to finish, depending on how many games you have to copy over. This way you can save time in case you want to run the scripts manually for whatever reason and backup the rest of your userdata independent from your games.
- `systemBackup.sh` → Which will only execute the two scripts `romsBackup.sh` and `userdataBackup.sh`.

We are separating the first two scripts instead of putting them in a single script so you can execute them manually whenever you want to, indepented from the automation steps we are covering below. The third script is exactly for those automation steps.



Beware: The rsync command's `--delete` option within the following two scripts `romsBackup.sh` and `userdataBackup.sh` does delete every file on the target side which is not available on the source side. We are using this option due to keep the target side consistent with the source side (for example if you have deleted some games on the source side and you want them being deleted on the target side too). If you don't want this to happen then just remove the `--delete` option on both scripts! Also note that those rsync scripts do synchronize only files that have been changed/added since the last synchronization. This will prevent unnecessary syncing for files that have already been synced in the past.

Now create the following script file:

```
nano /userdata/scripts/romsBackup.sh
```

And paste the following content (replace `<your_password>` with the according `backup` user its user password and `<Synology_NAS_IP_address/DNS_hostname>` with the static IP address/DNS hostname of your Synology NAS):

```
#!/bin/sh
USER=backup
PASS=<your_password>
SHARE=//<Synology_NAS_IP_address/DNS_hostname>/batoceraBackup
MOUNT_POINT=/tmp/tempMountFolder
mkdir -p "$MOUNT_POINT"
mount -t cifs -o username="$USER",password="$PASS" "$SHARE" "$MOUNT_POINT"
rsync -Pav --delete -e ssh /userdata/roms "$MOUNT_POINT"
cd /tmp
umount -f "$MOUNT_POINT"
exit 0
```

Save the file and quit the editor.

Now create the following script file:

```
nano /userdata/scripts/userdataBackup.sh
```

And paste the following content (replace `<your_password>` with the according `backup` user its user password and `<Synology_NAS_IP_address/DNS_hostname>` with the static IP address/DNS hostname of your Synology NAS):

```
#!/bin/sh
USER=backup
PASS=<your_password>
SHARE=//<Synology_NAS_IP_address/DNS_hostname>/batoceraBackup
```

```
MOUNT_POINT=/tmp/tempMountFolder
mkdir -p "$MOUNT_POINT"
mount -t cifs -o username="$USER",password="$PASS" "$SHARE" "$MOUNT_POINT"
cd "$MOUNT_POINT"
tar cfv "$MOUNT_POINT"/batoceraBackup.tar /userdata --exclude=roms --
exclude=saves/flatpak
rm -rf ./batoceraBackup.tar
cd /tmp
umount -f "$MOUNT_POINT"
exit 0
```

Save the file and quit the editor.

Now create the following script file:

```
nano /userdata/scripts/systemBackup.sh
```

And paste the following content:

```
#!/bin/sh
/userdata/scripts/romsBackup.sh
/userdata/scripts/userdataBackup.sh
shutdown -h now
```

Save the file and quit the editor.

Now make all three scripts executable by executing:

```
chmod +x /userdata/scripts/*.sh
```

You can test your scripts by executing them as follows:

Backup userdata:

```
/userdata/scripts/userdataBackup.sh
```

Backup games:

```
/userdata/scripts/romsBackup.sh
```

Backup userdata and games:

```
/userdata/scripts/systemBackup.sh
```

If you are planning to only backup your data manually by executing your scripts manually, you can stop here.

Automation part (optional)

Now let's set up the optional automation part for automatic backups of your userdata *and* your games:

SSH with user `root` into your Synology NAS (Unfortunately you have to use a Synology user which is part of the administrators group, otherwise SSH access will be declined. To avoid any permission issues, we will use `root` here.). Then execute the following commands (replace `<Batocera_IP_address/DNS_hostname>` with the static IP address of your Batocera system) for passwordless logins from your Synology NAS to your Batocera system (if you have already created an SSH key for your `root` user in the past on your Synology NAS, skip the first command!):

```
ssh-keygen
```

```
cat ~/.ssh/id_rsa.pub | ssh root@<Batocera_IP_address/DNS_hostname> 'cat >> /userdata/system/ssh/authorized_keys'
```

Insert the Batocera root user's password (Default password: `linux`).

Now go back to the Synology DSM web GUI and create the according automation task as follows: Navigate to Control Panel → Task Scheduler → Create → Scheduled Task → User-defined script → General: For Task insert a custom task name (e.g. `batoceraBackup`) and for user choose `root` → Schedule: Configure a desired custom schedule → Task Settings: For User-defined script insert the following (Replace `<Batocera_MAC_address>` with the according MAC address of your Batocera system, `<Synology_network_interface>` with the according network interface name through which you want to send the magic packet to (the one which is within the same Layer 2 subnet as your Batocera system (e.g. `eth0`)) and `<Batocera_IP_address/DNS_hostname>` with the static IP address/DNS hostname of your Batocera system.):



If your Synology NAS does have multiple network interfaces, to find out which network interface name you have to take, from Synology DSM web GUI navigate to: Control Panel → Info Center → Network. All of the available network interfaces are listed here with all the unique network/subnet informations. LAN1 is always `eth0`, LAN2 is always `eth1`, LAN 3 is always `eth2`, etc.)



If your Batocera system takes longer than 30 seconds to boot up, feel free to increase the `sleep` variable to whatever value you think does fit the real boot time of your Batocera system.

```
#!/bin/sh
sudo synonet --wake <Batocera_MAC_address> <Synology_network_interface>
sleep 30
ssh root@<Batocera_IP_address/DNS_hostname> sh ../scripts/systemBackup.sh
```

Confirm by clicking on OK.

That's it! Your scheduled backup automation should work now.

You can test it by running the task manually via the Task Scheduler as follows:

Navigate to Control Panel → Task Scheduler → Look out for your specific task, highlight it and click on Run.

From:

<https://www.wiki.batocera.org/> - **Batocera.linux - Wiki**

Permanent link:

https://www.wiki.batocera.org/rsync_backup?rev=1642965713

Last update: **2022/01/23 19:21**

