BackupPC

This guide is written when installing BackupPC V4.3.2. It covers installation, migration of V3 backup data, and host setup. Since Debian 10 includes BackupPC V3.3, this guide covers installation from tarball rather than by using the packet manager. Find the BackupPC Documentation. I'm using the term *host* interchangeably with *client* to refer to the host to be backed up, and *server* to refer to the host (or server) where BackupPC is running on.

Install V4.3.2

Debian 10 has BackupPC 3.3 in it's repositories. To install version 4 we need to manually install it.

- Install wget, ssh, rsync, and hping3:

```
apt update
apt install wget ssh rsync hping3
```

- Make sure the last line of the following script is commented out, then run it

```
#!/bin/bash
set -e
bpcver=4.3.2
bpcxsver=0.59
rsyncbpcver=3.1.2.1

# Needed only when installing
apt-get update
apt-get install -q -y apache2 apache2-utils libapache2-mod-perl2
glusterfs-client par2 perl smbclient rsync tar sendmail gcc zlib1g
zlib1g-dev libapache2-mod-scgi rrdtool git make perl-doc libarchive-
zip-perl libfile-listing-perl libxml-rss-perl libcgi-session-perl
libacl1-dev
echo -n "Give password or leave empty to generate one: "
read -s PASSWORD
echo
if [[ $PASSWORD == "" ]]; then
  apt-get -qq -y install pwgen
  PASSWORD=`pwgen -s -1 32`
  echo "Generated password: $PASSWORD"
else
  echo "Password given is: $PASSWORD"
fi
echo "$PASSWORD" > /root/password
chmod 600 /root/password
mkdir /srv/backuppc
ln -s /srv/backuppc/ /var/lib/backuppc
adduser --system --home /var/lib/backuppc --group --disabled-password -
-shell /bin/false backuppc
```

```
echo "backuppc:$PASSWORD" | sudo chpasswd backuppc
mkdir -p /var/lib/backuppc/.ssh
chmod 700 /var/lib/backuppc/.ssh
echo -e "BatchMode yes\nStrictHostKeyChecking no" >
/var/lib/backuppc/.ssh/config
ssh-keygen -q -t rsa -b 4096 -N '' -C "BackupPC key" -f
/var/lib/backuppc/.ssh/id_rsa
chmod 600 /var/lib/backuppc/.ssh/id_rsa
chmod 644 /var/lib/backuppc/.ssh/id_rsa.pub
chown -R backuppc:backuppc /var/lib/backuppc/.ssh

# Fetch and install latest stable releases
wget
https://github.com/backuppc/backuppc-xs/releases/download/$bpcxsver/Bac
kupPC-XS-$bpcxsver.tar.gz
wget
https://github.com/backuppc/rsync-bpc/releases/download/$rsyncbpcver/rs
ync-bpc-$rsyncbpcver.tar.gz
wget
https://github.com/backuppc/backuppc/releases/download/$bpcver/BackupPC
-$bpcver.tar.gz
tar -zxf BackupPC-XS-$bpcxsver.tar.gz
tar -zxf rsync-bpc-$rsyncbpcver.tar.gz
tar -zxf BackupPC-$bpcver.tar.gz
cd BackupPC-XS-$bpcxsver
perl Makefile.PL
make
make test
make install
cd ../rsync-bpc-$rsyncbpcver
./configure
make
make install
cd ../BackupPC-$bpcver

# When installing, use this
./configure.pl --batch --cgi-dir /var/www/cgi-bin/BackupPC --data-dir
/var/lib/backuppc --hostname backuppc --html-dir /var/www/html/BackupPC
--html-dir-url /BackupPC --install-dir /usr/local/BackupPC

# When upgrading, use this instead:
# ./configure.pl --batch --config-path /etc/BackupPC/config.pl

# The following is good also when upgrading, unless you have modified
the files yourself
cp httpd/BackupPC.conf /etc/apache2/conf-available/backuppc.conf
sed -i "/deny\ from\ all/d" /etc/apache2/conf-available/backuppc.conf
sed -i "/deny\,allow/d" /etc/apache2/conf-available/backuppc.conf
sed -i "/allow\ from/d" /etc/apache2/conf-available/backuppc.conf

# Note that changing the apache user and group (next two commands)
```

```
could cause other services
# provided by apache to fail. There are alternatives if you don't want
to change the apache
# user: use SCGI or a setuid BackupPC_Admin script - see the docs.
sed -i "s/export APACHE_RUN_USER=www-data/export
APACHE_RUN_USER=backuppc/" /etc/apache2/envvars
sed -i "s/export APACHE_RUN_GROUP=www-data/export
APACHE_RUN_GROUP=backuppc/" /etc/apache2/envvars
echo '<html><head><meta http-equiv="refresh" content="0;
url=/BackupPC_Admin"></head></html>' > /var/www/html/index.html
a2enconf backuppc
a2enmod cgid
service apache2 restart
cp systemd/init.d/debian-backuppc /etc/init.d/backuppc
chmod 755 /etc/init.d/backuppc
update-rc.d backuppc defaults
chmod u-s /var/www/cgi-bin/BackupPC/BackupPC_Admin
touch /etc/BackupPC/BackupPC.users
sed -i "s/$Conf{CgiAdminUserGroup}.*/$Conf{CgiAdminUserGroup} =
'backuppc';/" /etc/BackupPC/config.pl
sed -i "s/$Conf{CgiAdminUsers}.*/$Conf{CgiAdminUsers} = 'backuppc';/"
/etc/BackupPC/config.pl
chown -R backuppc:backuppc /etc/BackupPC

# Needed only when installing
echo $PASSWORD | htpasswd -i /etc/BackupPC/BackupPC.users backuppc

#service backuppc start
```

- Modify the following lines in /etc/BackupPC/config.pl

```
$Conf{PoolV3Enabled} = 1;
$Conf{BlackoutPeriods} = [];
$Conf{XferMethod} = 'rsync';
```

- Copy or setup the host configuration files, as explained in Host Setup
- Link to the data directory

```
ln -s /backuppc /var/lib/backuppc
```

- Start the daemon

```
service backuppc start
```

Changing Passwords after Installation

- Change webserver password

```
htpasswd /etc/BackupPC/BackupPC.users <user>
```

- Change user password

```
sudo passwd <user>
```

To disable authentication, comment the auth instructions in /etc/backuppc/apache.conf and restart backuppc and apache.

Links

- Installing BackupPC 4 from tarball or git on Debian 10
- Build Your Own Packages
- BackupPC 4.3.2 release
- Install and Configure BackupPC on Debian 10

Apache Setup

- If you run a separate server and want to make the web interface available, you need to setup a VirtualHost proxy on your main apache server. Check my guide for Debian 9 or Debian 11 how this is done.
- Add a redirect in BackupPC's apache configuration file /etc/apache2/conf-available/backuppc.conf to allow access from the local network:

```
RedirectMatch /backuppc /BackupPC_Admin
```

- Add an empty index.html to /var/www/html/BackupPC

Migrate data from V3 to V4

- Stop the daemon

```
service backuppc stop
```

- Make sure you have at least 60% free inodes and 10% free disk space

```
df -i
df -h
```

- Run the migration script

```
/usr/local/BackupPC/bin/BackupPC_migrateV3toV4
```

- V3 pool must be enabled for this to work.
- Start the daemon

```
service backuppc start
```

- Run BackupPC_nightly

```
/usr/local/BackupPC/bin/BackupPC_nightly 0 255
```

- Before you turn off the V3 pool by setting $Conf{PoolV3Enabled} to 0, make sure BackupPC_nightly has run enough times (specifically, $Conf{PoolSizeNightlyUpdatePeriod} times) so that the V3 pool can be emptied. You could do this manually, but only if you are very careful to check that the remaining files only have one link.
- Check that hardlinks have been removed from the pool with

```
du -csh /backuppc/cpool/?/?/?
```

- Find hardlinked files in machine backups

```
cd /backuppc/pc
find <machine> -xdev -type f -links +1 -printf "%i %n %p\n" >
machine.txt
```

- If you find hardlinked files in a machine backup, check whether it is in a valid backup folder. If it's an invalid folder, delete that folder and run BackupPC_nightly.

Links

- BackupPC_migrateV3toV4
- BackupPC - migrate V3 to V4
- How to find which backups reference a particular pool file V4
- Troubleshoot full pool for 4.0
- Migrating BackupPC v3 pools to v4
- Is there a way to manually run BackupPC-Nightly ?
- How to find and delete all hard links to a file

Migrate V3 data from a non-encrypted drive to V4 on an encrypted LVM drive

You can follow the steps in this paragraph to migrate an old V3 installation from an obsolete backuppc machine to a new V4 installation on a Debian 10 backuppc machine. You may need to omit steps if they do not apply to your situation.

Move data to a temporary mount point (on a different drive)

You can omit this step if you are setting up a new data drive and still have access to the drive containing V3 data.

- Umount the data drive, and the spare drive to use for temporary data storage
- Copy the entire V3 data to a spare harddrive (example)

```
dd if=/dev/mapper/vg_backup-lv_backup of=/dev/sde1 bs=4M
status=progress
```

- Check progress with the following command, if you forgot to add the status parameter (pid is

dd's process id, output will show on the machine where dd is run)

```
kill -USR1 <pid>
```

- Mount spare and point BackupPC's top directory

```
mount /dev/sde1 /mnt/backup
ln -s /mnt/backup/backuppc /var/lib/backuppc
```

- You need to check and update permissions if you migrate from a different (Debian) installation

```
chown -R backuppc:backuppc /mnt/backup
```

Install BackupPC

- Install V4.3.2

Migrate Data

- Migrate data from V3 to V4

Setup encrypted LVM partition

- Data drive encryption
- Encrypted partitions/folders with auto-mount

Move V4 data to encrypted partition

- Stop the daemon

```
service backuppc stop
service backuppc status
```

- Copy all V4 data between mount points. This also works if you move the data from a non-encrypted partition to an encrypted partition. rsync is instructed to preserve hardlinks, but you should consider this only if your total hardlink file volume is small, since rsync is not very efficient with hardlinks. Check the permissions of the target are set correctly.

```
rsync -az -H --delete --numeric-ids --info=progress2 /path/to/source/
/path/to/dest/
```

- Link to the new data directory

```
ln -s /backuppc /var/lib/backuppc
```

- Start the daemon

```
service backuppc start
```

Links

- How to copy directories with preserving hardlinks?
- Backing up/restoring a LUKS encrypted partition with clonezilla
- Cloning Encrypted SSD to larger SSD

Setup boot configuration

- Remove service backuppc from auto-start at boot to make sure the encrypted volume is mounted before the backuppc service is started

```
update-rc.d backuppc disable
```

- Debian 10 add rc.local

Maintenance

- Delete a backup. If you delete several backups, delete non-filled backups which were taken after a filled backup first.

```
/usr/local/BackupPC/bin/BackupPC_backupDelete -h host -n num
```

  Other Command Line Utilities and BackupPC_backupDelete

Host Setup

All hosts are setup with rsync through ssh. For Windows 10 hosts I use the Windows Subsystem for Linux which allows to setup a Debian layer to access the host. To backup the localhost we need a small tweak which is explained below.

Main Configuration on BackupPC server

- Modify the following lines in /etc/BackupPC/config.pl (after migration of all hosts from previous versions to V4) and leave all other options on their default settings (reference V4.3.2)

```
$Conf{PoolV3Enabled} = 0;
$Conf{BlackoutPeriods} = [{
  hourBegin => 10,
  hourEnd => 3,
  weekDays => [0, 1, 2, 3, 4, 5, 6],
}];
$Conf{XferMethod} = 'rsync';
```

- Add *target="blank"* to the additional navigation bar links to open in a new browser tab

```
$Conf{CgiNavBarLinks} = [
```

```
{
  link  => "?action=view&type=docs\" target=\"blank",
  lname => "Documentation",
}
{
  link  => "https://github.com/backuppc/backuppc/wiki\"
target=\"blank",
  lname => "Wiki",
}
{
  link  => "https://backuppc.github.io/backuppc\" target=\"blank",
  lname => "Homepage",
}
];
```

Check whether a host is accessible

- Make sure the host is added to the /etc/BackupPC/hosts file. All hosts should have the DHCP flag set to 0, even if they obtain their IP addresses per DHCP.
- First DNS is used to lookup the IP address given the host's name using perl's gethostbyname() function. This should succeed for machines that have dynamic or fixed IP addresses that are known via DNS. You can manually see whether a given host have a DNS entry according to perl's gethostbyname function with this command:

  ```
  perl -e 'print(gethostbyname("myhost") ? "ok\n" : "not found\n");'
  ```

- If gethostbyname() fails, BackupPC then attempts a NetBios multicast to find the host. Provided your host is configured properly, it should respond to this NetBios multicast request. Specifically, BackupPC runs a command of this form:

  ```
  nmblookup myhost
  ```

  If this fails you will see output like:

  ```
  name_query failed to find name myhost
  ```

  If it is successful you will see output like:

  ```
  local.domain.subnet.ip myhost<00>
  ```

- Depending on your netmask you might need to specify the -B option to nmblookup. For example:

  ```
  nmblookup -B local.domain.subnet.255 myhost
  ```

  If necessary, experiment with the nmblookup command which will return the IP address of the host given its name. Then update $Conf{NmbLookupFindHostCmd} with any necessary options to nmblookup.

- You need to login to the host as backuppc user via SSH at least once manually to accept the

host key:

```
ssh <host>
```

Configuration on Server for Localhost

- Host access configuration:

```
$Conf{RsyncClientPath} = '/usr/bin/rsync';
$Conf{RsyncSshArgs} = ['-e', '/usr/bin/sudo -p'];
```

- Host share configuration:

```
$Conf{BackupFilesOnly} = ['/etc', '/root', '/home', '/usr', '/var',
'/boot', '/sys'];
```

- Install ssh and rsync:

```
sudo apt install ssh rsync
```

- localhost does not need ssh, this is achieved with a trick, see BackupPC v4 and rsync for localhost
- Allow user backuppc to run rsync as root, add the following line after %sudo:

```
sudo visudo
%sudo ALL=(ALL:ALL) ALL
backuppc ALL=(root:root) NOPASSWD: /usr/bin/rsync
```

Configuration on Server for local hosts

- Host access configuration:

```
$Conf{RsyncClientPath} = 'sudo /usr/bin/rsync';
$Conf{RsyncSshArgs} = ['-e', '$sshPath'];
```

- Host share configuration:

```
$Conf{RsyncShareName} = ['/'];
$Conf{BackupFilesExclude} = ['/proc', '*.vdi'];
```

Configuration on Server for remote hosts

- Host access configuration:

```
$Conf{RsyncClientPath} = 'sudo /usr/bin/rsync';
$Conf{RsyncSshArgs} = ['-e', '$sshPath -p <port>'];
$Conf{PingCmd} = '/usr/bin/sudo /usr/sbin/hping3 $host -p <port> -c 1 -
S';
```

```
$Conf{PingMaxMsec} = 300;
$Conf{ClientNameAlias} = 'subdomain.domain.tld';
```

- Host share configuration:

```
$Conf{RsyncShareName} = ['/'];
$Conf{BackupFilesExclude} = ['/proc', '*.vdi'];
```

- To make the host pingable from the backuppc sever you need to replace “ping” with “hping3”

```
apt update
apt install hping3
```

- Read more: How to ping BackupPC hosts behind a NAT router or firewall

Configuration on Server for remote hosts accessible through relais

- Modify ping command in host access configuration (note: $sshPath is not resolved for PingCmd):

```
$Conf{PingCmd} = '/bin/ssh -p <port> -o ConnectTimeout=2 $host echo "1
packets transmitted, 1 received, 0% packet loss, time 0ms"';
```

Configuration on Hosts

- Windows 10: Install Windows Subsystem for Linux on Windows 10 hosts, which installs a *Debian* command line layer on top of your Windows 10 installation
- Install ssh and rsync:

```
sudo apt install ssh rsync
```

- Harden SSH and restart the service.
- Create user backuppc with a restricted shell, not able to do local but only remote logins (do not expire the user):

```
sudo adduser --shell /bin/rbash --disabled-password backuppc
```

- Copy the Server's backuppc id_rsa.pub file to the Host's backuppc authorized_keys file. The authorized_keys file should contain further restrictions to prevent e.g. port forwarding, preceed the ssh-rsa string with 'restrict,from=“local.domain.subnet.ip”':

```
restrict,from="local.domain.subnet.ip" ssh-rsa <BASE64-PUBKEY-
REPRESENTATION> backuppc
```

It should further be owned by root and being read- but not writable by the backuppc user, to prevent removal of SSH restrictions. In older SSH versions you must enter the following instead of restrict:

```
no-pty,no-port-forwarding,no-X11-forwarding,no-agent-
forwarding,from="local.domain.subnet.ip" ssh-rsa <BASE64-PUBKEY-
REPRESENTATION> backuppc
```

- Allow user backuppc to run rsync as root, add the following line after %sudo:

```
sudo visudo
%sudo ALL=(ALL:ALL) ALL
backuppc ALL=(root:root) NOPASSWD: /usr/bin/rsync
```

- Allowing automatic command execution as root on Linux using SSH
- authorized_keys

Configuration on Synology DSM6 Hosts

- Synology DSM6 contains a modified rsync client, with which backup of a DSM host is not possible. The easiest way is to install an additional rsync client, which is compiled as static and therefore does not depend on any libraries. You can either follow the following steps, or download the rsync client from this website (rename rsync.gz to rsync). It was compiled on a Debian 10.3 Linux host (amd64) and should run on all Intel based Synology NAS. I confirmed it on DS218+, DS716+ II, and DS718+. For other platforms you would need to cross-compile.
- Harden SSH and reboot.
- Create a new user in the DSM web interface. Add him to user groups *users* and *administrators*. Give him permissions for *homes*, but not any other shared folder.
- On another host running linux on amd64 hardware, download sources and compile rsync:

```
cd /home/<user>/temp
wget https://download.samba.org/pub/rsync/rsync-3.1.3.tar.gz
tar xzvf rsync-3.1.3.tar.gz
cd rsync-3.1.3
export CFLAGS=--static
./configure
make
```

- Copy the *rsync* binary to the host's folder /var/services/homes/backuppc/bin
- Copy the Server's backuppc id_rsa.pub file to the Host's backuppc authorized_keys file in folder ~/.ssh. The authorized_keys file should contain further restrictions to prevent e.g. port forwarding, preceed the ssh-rsa string with 'restrict,from="local.domain.subnet.ip"':

```
restrict,from="local.domain.subnet.ip" ssh-rsa <BASE64-PUBKEY-
REPRESENTATION> backuppc
```

  It should further be owned by root and being read- but not writable by the backuppc user, to prevent removal of SSH restrictions.

- Log in to the DiskStation with SSH, sudo to root, and create the file *backuppc* in folder /etc/sudoers.d/. As there is no *visudo* on DSM you must be extra careful to copy the following lines exactly, otherwise you may render sudo unable to elevate to root:

```
cd /etc/sudoers.d
sudo vim
# Allow backuppc
backuppc ALL=(root:root) NOPASSWD:
/var/services/homes/backuppc/bin/rsync
```

- Rsync over ssh: “ERROR: module is read only” suddenly appeared
- Allowing automatic command execution as root on Linux using SSH
- authorized_keys

Special host configurations

- Linux host share configuration for hosts with more than one file system, for example /home residing on an encrypted share:

```
$Conf{RsyncShareName} = ['/', '/home'];
```

- Windows 10 host share configuration:

```
$Conf{RsyncShareName} = ['/', '/mnt/c', '/mnt/d'];
$Conf{BackupFilesExclude} = [
      '/proc',
      '*.vdi',
      'Windows10Upgrade',
      'System Volume Information',
      '$RECYCLE.BIN',
      '*.mkv',
];
```

- Synology DSM6 host share configuration:

```
$Conf{RsyncShareName} = ['/', '/volume1'];
$Conf{BackupFilesExclude} = [
      '/proc',
      '@eaDir',
      'media',
      '*.mkv',
      '*.vdi',
];
```

- Synology DSM6 host access configuration:

```
$Conf{RsyncClientPath} = 'sudo /var/services/homes/backuppc/bin/rsync';
```

Ping commands

- Regular ping (built in)

```
$ /bin/ping <host> -c 1
```

- Host behind firewall

```
$ sudo /usr/sbin/hping3 <host> -p <port> -c 1 -S
```

- Host behind relais host and firewall

```
$ ssh -p <port> -o ConnectTimeout=2 <host> echo "1 packets transmitted,
1 received, 0% packet loss, time 0ms"
```

- No ping (not recommended)

```
$ /bin/echo
```

Alternative setups for Windows hosts

I'm not using any of the methods described in below links. I have collected those during my search for the best way to integrate Windows hosts into BackupPC. If you need to backup a host which does not run Windows 10, or does not provide the Microsoft Windows Subsystem for Linux, then those links might be helpful.

- Cygwin
- Backup Windows System with BackupPC Using Rsyncd
- cygwin-rsyncd: Rsyncd for Cygwin
- Setting up a Windows Backuppc Client using Rsync/Cygwin
- Installing SFTP/SSH Server on Windows using OpenSSH
- How to Transfer Files with Rsync over SSH
- How to setup the rsync daemon on Linux
- Linux rsync command