# BackupPC-Clone Documentation

**P.R. Water**

**Oct 05, 2022**

# Contents:

BackupPC-Clone is a tool for cloning, copying, or syncing the data of BackupPC from one filesystem to another filesystem where `rsync` fails.

CHAPTER 1

# Introduction

CHAPTER 2

# Getting Started

In this chapter we discuss all steps for installing, configuring, and running BackupPC-Clone.

In this chapter we assume that the data directory of BackupPC is `/var/lib/BackupPC` and the data directory of BackupPC-Clone is `/var/lib/BackupPC-Clone` and BackupPC is running under user `backuppc`.

The commands in section *Installing BackupPC-Clone* must be executed under `root` and all other commands under user `backuppc`.

## 2.1 Installing BackupPC-Clone

BackupPC-Clone can be installed using `pip`:

```
pip3 install BackupPC-Clone
```

Create the data directory of BackupPC-Clone:

```
mkdir /var/lib/BackupPC-Clone
chown backuppc.backuppc /var/lib/BackupPC-Clone
```

## 2.2 Configuring BackupPC-Clone

BackupPC-Clone requires configuration files in the data directory of BackupPC and in the data directory of BackupPC-Clone.

Create the configuration file of the "original" using the `init-original` command and provide the answers according to the configuration of your BackupPC host:

```
backuppc-clone init-original
```

The output will look like:

```
Creating Original Configuration File
====================================

perl executable [/usr/bin/perl]


Config file [/etc/BackupPC/config.pl]

Writing /var/lib/BackupPC/original.cfg
```

The created configuration file looks like:

```
[BackupPC Clone]
role = original
name = your-host

[Original]
top_dir = /var/lib/BackupPC
conf_dir = /etc/BackupPC
log_dir = /var/log/backuppc
pc_dir = /var/lib/BackupPC/pc
```

Create the configuration file of the "clone" using the `init-clone` command and provide the answers according to the configuration of your BackupPC host:

```
backuppc-clone init-clone
```

```
Initializing Clone
==================

Creating configuration file
---------------------------

top dir of the original [/var/lib/BackupPC]



top dir of the clone /var/lib/BackupPC-Clone



name of clone [your-host-clone]

 Writing /var/lib/BackupPC-Clone/clone.cfg
 Creating directories

Creating metadata database
--------------------------

 Initializing /var/lib/BackupPC-Clone/clone.db
```

The created configuration file looks like:

```
[BackupPC Clone]
role = clone
name = your-host-clone

[Original]
config = /var/lib/BackupPC/original.cfg
name = your-host
```

If you are planning to create multiple clones use distinct names for the clones.

## 2.3 Cloning BackupPC

To create a complete clone of the data directory of BackupPC use the `auto` command:

```
nohup backuppc-clone --ansi auto -v /var/lib/BackupPC-Clone/clone/clone.cfg > auto.
↪log  2>&1 &
```

The first part (the output will be repeated for each backup) of the output will look like (using a slow external hard disk):

```
Inventorying Original Backups
=============================

 Scanning /var/lib/BackupPC/pc

 Found 15 hosts and 257 backups

Cloning Backup nas02/389
========================

Original backup
---------------

 Scanning /var/lib/BackupPC/pc/nas02/389

 2357279/2357279 [============================] 100% 19 mins

 Files found:       3277988
 Directories found: 965820

Clone pool
----------

 Adding files ...

 2460473/2460473 [============================] 100% 24 hrs

 Number of files copied: 2460473
 Total bytes copied    : 345.3GiB (370810283565B)

Clone backup
------------

 Populating ...

 4243808/4243808 [============================] 100% 7 hrs

 Number of files copied       : 3806
 Number of hardlinks created  : 3274182
 Number of directories created: 965820


Inventorying Original Backups
=============================
```

Depending on the size and number of files in data directory of BackupPC and the speed of your hardware this command will take some time to complete. Use the following command to monitor the progress of the `auto` command:

```
tail -f auto.log
```

CHAPTER 3

# Miscellaneous

In this chapter we discuss miscellaneous aspects related to or about BackupPC-Clone.

## 3.1 Ext4 Filesystem and Inodes

The ext4 filesystem is still one of the most used filesystems. One of the properties of the ext4 filesystem is that the number of inodes is fixed and can not be increased later. BackupPC requires relatively a lot of inodes due to fact it is using many hardlinks to the same file.

When creating an ext4 filesystem for a clone make sure that this filesystem has enough inodes. One can compare the number of inodes with the following commands:

```
tune2fs -l `df /var/lib/BackupPC       | awk '{if (NR==2) print $1}'` | grep -i
↪'inode count'
tune2fs -l `df /var/lib/BackupPC-Clone | awk '{if (NR==2) print $1}'` | grep -i
↪'inode count'
```

The `-i`, `-I`, and `-N` option of `mke2fs` impact the number of inodes created.

## 3.2 Clone on Removable Media

When creating a clone on removable media it is recommended to encrypt the filesystem such that in case of loss or theft your data is still safe.

Suppose your external disk is available under `/dev/sdh1` you can encrypt your external disk with the following commands:

```
cryptsetup --verify-passphrase --cipher aes-cbc-essiv:sha256 --key-size 256␣
↪luksFormat /dev/sdh1
cryptsetup luksOpen /dev/sdh1 sdh1-luks
```

The encrypted disk will be available at `/dev/mapper/sdh1-luks`.

We found no noticeable performance impact when using disk encryption

## 3.3 Reserved blocks

When using a dedicated partition (and an ext4 filesystem) for the clone you might set the percentage of the filesystem blocks reserved for the super-user to 0 using the `-m` option of the `mke2fs` command.

## 3.4 Example Creating Filesystem for Clone

In this section we give an example how to create a filesystem for a clone of BackupPC.

```
cryptsetup --verify-passphrase --cipher aes-cbc-essiv:sha256 --key-size 256␣
↪luksFormat /dev/sdh1
cryptsetup luksOpen /dev/sdh1 sdh1-luks

mke2fs -t ext4 -i 16384 -m 0 /dev/mapper/sdh1-luks

mkdir /var/lib/BackupPC-Clone

mount -t ext4 /dev/mapper/sdh1-luks /var/lib/BackupPC-Clone

chown backuppc.backuppc /var/lib/BackupPC-Clone
```

See *Clone on Removable Media* for more information about the *cryptsetup* commands, see *Ext4 Filesystem and Inodes* and *Reserved blocks* for more information on the `-i` and `-m` options of the `mke2fs` command.

## 3.5 Verifying a Clone Backup

A backup is paramount for your company regardless of its size and you should not trust BackupPC-Clone blindly.

You can verify BackupPC-Clone has created a correct clone of a host backup simply with the following command (replace `host` and `num` with the actual hostname and backup number):

```
diff --recursive --brief /var/lib/BackupPC/pc/host/num/ /var/lib/BackupPC-Clone/clone/
↪pc/host/num/
```

You can ignore the following message:

```
Only in /var/lib/BackupPC/pc/host/num/: backuppc-clone.csv
```

CHAPTER 4

# Limitations

In this chapter we discus the limitations of BackupPC-Clone.

- Currently only BackupPC 3.x is supported by BackupPC-Clone. We plan to support BackupPC 4.x when we have migrated our servers to CentOS 8.
- BackupPC-Clone uses the combination of filename and inode number for identifying files in the pool of BackupPC. Hence, if between two consecutive scans of the pool of BackupPC by BackupPC-Clone a pool file is deleted and a new file (different from the deleted pool file) is added to the pool of BackupPC with the same MD5 hash (i.e. a MD5 collision between the deleted and new file) and is associated with the same inode as the deleted file BackupPC-Clone will not detect that the file has been changed in the pool of BackupPC. We regard the chance on such an event very unlikely.

CHAPTER 5

# License

This project is licensed under the MIT license.