

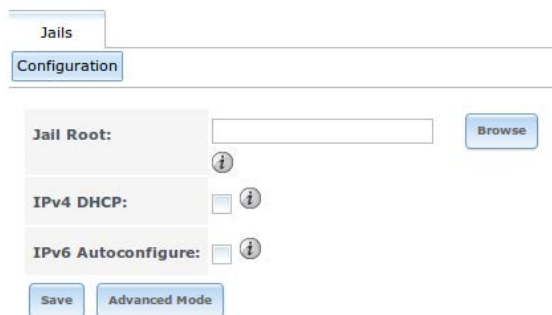
## Using Jails in FreeNAS to set up Backblaze B2

A Jail can be thought of as a virtual machine within the FreeNAS system. It is an implementation of operating system-level virtualization. It allows users who are comfortable using the command line to have more control over software installation and management.

### Step 1 – Create a new dataset on your Storage Volume

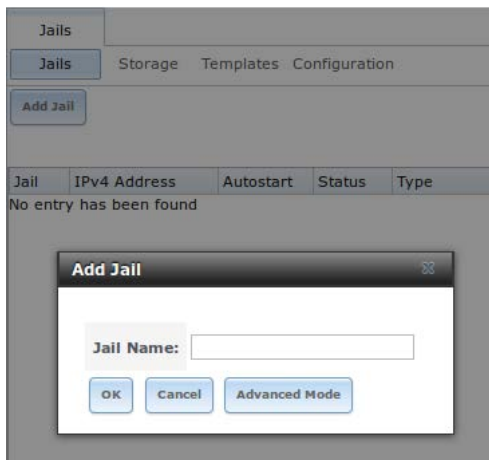
It is good practice to create a 'Jail Root' dataset which all Jails will sit on.

### Step 2 – Jails Configuration



- Jail Root is the new dataset created
- Enable IPv4 DHCP if network uses DHCP server
- If not, click on Advanced mode to set static IP
- Save

### Step 3 - Add Jail

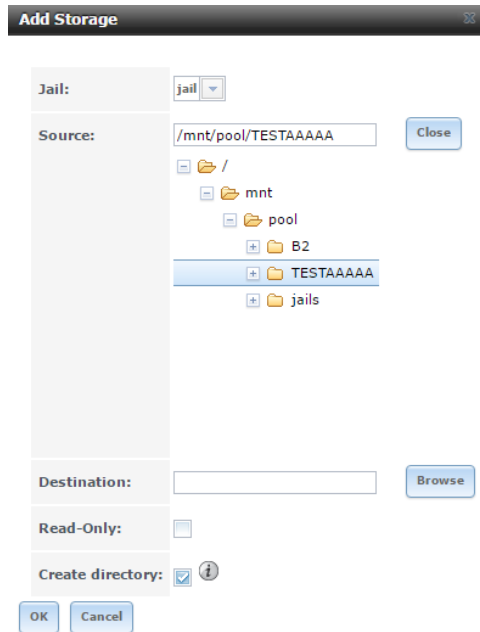


By default, user must only give a name to the Jail. It is recommended to stay away from the Advanced Mode options as manually inputting information can result in simple mistakes.

### Step 4 – Add Storage to the Jail

Click on the Jail, and on the bottom toolbar, click on the black folder (second from left) to add Storage to your Jail. The Source is the directory or dataset on the FreeNAS system you want the Jail to have access to.

Ex: The system shown below has a storage pool called pool, and three datasets on it, the jails dataset we created, one called B2 and the other called TESTAAAAA. If you want to back up data that is on one of those datasets, make it the source storage.



The Destination is a directory within the Jail. Avoid using the /tmp directory as there are sizing issues. We suggest to make the destination in the /mnt directory. For example, in the system above, a directory was created for the destination as /mnt/storage.

This means when you open a shell within the jail, you can change into the /mnt/storage directory, and you'll see all the files from the FreeNAS dataset you set as the storage source.

### Step 5 – Installing Necessary Packages for Rclone from Jail's Shell

Click on the Jail, and on the bottom toolbar, click on the shell button (second from the right) to gain access to the command line. Once there, type ***'pkg install wget'***.

Once this package finished installing, type in ***'wget http://downloads.rclone.org/rclone-v1.35-freebsd-amd64.zip'*** .

When the download is complete, type 'ls' and you should see 'rclone-v1.35-freebsd-amd64.zip' listed. Next, type ***'unzip rclone-v1.35-freebsd-amd64.zip'*** to gain access to the files.

### Step 6 – Configuring rclone

After unzipping the .zip file, type ***'cd rclone-v1.35-freebsd-amd64'***.

Copy rclone to /usr/sbin → ***'cp rclone /usr/sbin'*** (This lets you call rclone from any directory in the command line without needing the full path to the command)

Next, type ***'rclone config'*** to begin the set up.

You will be prompted to answer a few questions, the answers are shown below:

```
root@Jail1:/rclone-v1.35-freebsd-amd64 # ./rclone config
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q>
```

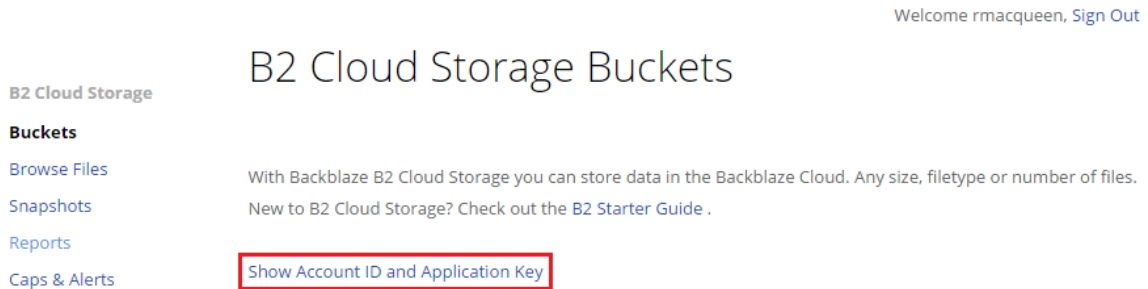
→ Enter 'n' to create a new remote

You will then be prompted to enter a name → for example, we simply chose 'remote'

```
Type of storage to configure.
Choose a number from below, or type in your own value
 1 / Amazon Drive
   \ "amazon cloud drive"
 2 / Amazon S3 (also Dreamhost, Ceph, Minio)
   \ "s3"
 3 / Backblaze B2
   \ "b2"
 4 / Dropbox
   \ "dropbox"
 5 / Encrypt/Decrypt a remote
   \ "crypt"
 6 / Google Cloud Storage (this is not Google Drive)
   \ "google cloud storage"
 7 / Google Drive
   \ "drive"
 8 / Hubic
   \ "hubic"
 9 / Local Disk
   \ "local"
10 / Microsoft OneDrive
   \ "onedrive"
11 / Openstack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
   \ "swift"
12 / Yandex Disk
   \ "yandex"
Storage>
```

→Enter '3' to configure Backblaze B2

Next you will be prompted for you Backblaze B2 Account ID and Application Key. To access this, login to your Backblaze B2 account and click on “Buckets” in the “B2 Cloud Storage” section. Click on the “Show Account ID and Application Key” link shown below.



After entering that, you will be prompted to enter an endpoint. It is recommended to leave this blank and just hit 'enter'.

You will then get to review your information and decide if you're satisfied or need to go back in and edit.

Configuration is now complete, and ready to use.

Sync is used to fully backup a whole directory.

Copy is used to only backup new/changed files.

This remote is called `remote` and can now be used like this

See all buckets

```
rclone lsd remote:
```

Make a new bucket

```
rclone mkdir remote:bucket
```

List the contents of a bucket

```
rclone ls remote:bucket
```

Sync `/home/local/directory` to the remote bucket, deleting any excess files in the bucket.

```
rclone sync /home/local/directory remote:bucket
```

## Step 7 - Installing Necessary Packages for Duplicity from Jail's Shell

Run the following commands to install all necessary packages to run duplicity.

1. `pkg install wget`
2. `cd /usr/ports/sysutils/duplicity`
3. `wget http://launchpad.net/duplicity/0.7-series/0.7.08/+download/duplicity-0.7.08.tar.gz`
4. `tar -xvzf duplicity-0.7.08.tar.gz`
5. `cd duplicity-0.7.08`
6. `pkg install py27-pip`
7. `pkg install py27-lockfile`
8. `pkg install librsync`
9. `setenv CFLAGS -I/usr/local/include`
10. `ln -s /usr/local/bin/python2.7 /usr/local/bin/python2`
11. `python2.7 setup.py install`

Now to actually use duplicity to send files up to B2:

```
duplicity [type (full or incremental)] [--no-encryption (if wanted)] [/directory to share]  
b2://account_id:[application_key]@bucket_name.
```

For example, in testing I ran the following command see screenshot:

```
root@Jail:/stuff # duplicity full --no-encryption /stuff/ b2://21b99f4d4820:0015b4dbc8a19641e5ba43ce1f1c1aedc0a737faa045d-duplicity

Local and Remote metadata are synchronized, no sync needed.
Last full backup left a partial set, restarting.
Last full backup date: Fri Mar 3 05:24:03 2017
RESTART: The first volume failed to upload before termination.
Restart is impossible...starting backup from beginning.
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: none
-----[ Backup Statistics ]-----
StartTime 1488547776.13 (Fri Mar 3 05:29:36 2017)
EndTime 1488548368.21 (Fri Mar 3 05:39:28 2017)
ElapsedTime 592.08 (9 minutes 52.08 seconds)
SourceFiles 3
SourceFileSize 2048000004 (1.91 GB)
NewFiles 3
NewFileSize 2048000004 (1.91 GB)
DeletedFiles 0
ChangedFiles 0
ChangedFileSize 0 (0 bytes)
ChangedDeltaSize 0 (0 bytes)
DeltaEntries 3
RawDeltaSize 2048000000 (1.91 GB)
TotalDestinationSizeChange 2053341932 (1.91 GB)
Errors 0
-----

root@Jail:/stuff # █
```

Paste 132x50

User can login to B2 account and manually create the bucket beforehand, or they can just enter a new bucket name in the command and it will automatically create the bucket and send all files up to it.

## Setting up Cron Jobs with Rclone and Duplicity

Cron Jobs will run in the background is a scheduled time and date, which works out perfect for anybody looking to back up their data to the B2 cloud. Below are the following steps to setup and schedule cron jobs in FreeNAS with B2.

1. Open a shell and download our configuration script -> 'wget images.45drives.com/setup/b2config.sh'
2. Run the configuration script -> 'sh b2config.sh'
3. You will be asked a series of questions based on whether you're using rclone or duplicity.

For rclone, you must specify:

sync or copy

rclone remote name you created in the initial setup

your B2 bucket name

directory you want to back up (on the Jail side)

jail name

overarching storage pool name

For duplicity, you must specify:

full or incremental

your B2 bucket name

directory you want to back up (on the Jail side)

jail name

pool name

as well as your B2 Account ID and Application Key.

**NOTE** – Application key expires once you create a new one, meaning that the old application key will become invalid and you would need to reconfigure the system for both rclone and duplicity.

4. Once you've answered all the questions correctly without the process being interrupted by an error message, you're ready to create a Cron Job. In the FreeNAS WebGUI, click on 'Tasks' in the top toolbar.
5. Create a task like shown below, where the user is root, or the admin account on the system. The example below is for a duplicity full task, where the command is '/conf/base/etc/sync.sh'. See table below for the proper command based on your backup configuration.

Rclone – sync	Rclone – copy	Duplicity – full	Duplicity – incr
/base/conf/etc/sync.sh	/base/conf/etc/copy.sh	/base/conf/etc/full.sh	/base/conf/etc/incr.sh

The command refers to the script that is auto generated when the questions are answered in the b2config.sh setup.

The screenshot shows a cron job configuration interface with the following details:

- User:** root
- Command:** /conf/base/etc/full.sh
- Short description:** Duplicity Full backup to B2
- Minute:** Configured with 'Every N minute' selected and '00' chosen from a grid of 00-53.
- Hour:** Configured with 'Each selected hour' selected and '00' chosen from a grid of 00-23.
- Day of month:** Configured with 'Every N day of month' selected and no specific day chosen.

Scheduling options gives you an option for specifying an exact time to backup, as well as an option for every X hours and Y minutes.

Our thoughts are that people could be an initial rclone sync or duplicity full back up to the cloud. This will put all data in the specified directory up to the B2 cloud in full.

Then, they could create a cron job with a rclone copy of duplicity incr, which only sends up the updated files / new files that were put into the directory. This cron job could be scheduled to be every day of the week at 12:00 AM to ensure all data is back up every day.