Obnam tutorial

2022-01-16 15:48

Contents

With the help of this tutorial, you're going to set up Obnam, make your first backup, and check that you can restore files from it.

In Obnam, **a client** is a computer whose data is being backed up, and **a server** is a computer that holds the backup. A single computer can serve both roles, but don't put your backup onto the same disk as you're backing up; if that disk breaks, the backup won't do you any good. Consider using an USB-attached disk, or better yet, some network-attached storage.

1 Setting up a server

For this, you'll need:

- Git and Ansible installed on your local machine
- a Debian host with plenty of space to keep your backups (this can be the same, local machine)

On your local machine, clone the Obnam repository:

```
$ git clone https://gitlab.com/larswirzenius/obnam.git
$ cd obnam/ansible
```

The next command depends on where your Obnam server is hosted:

- if the server is accessible from the Internet, run the following commands, replacing `obnam.example.com` with the domain name of the host:

  ```
  $ printf '[server]\nobnam.example.com\n' > hosts
  $ ansible-playbook -i hosts obnam-server.yml -e domain=obnam.example.com
  ```

  The above gets a free TLS certificate from Let's Encrypt[1].

- if it's a private server or just the same machine as the Obnam client, run the following:

  ```
  $ printf '[server]\nprivate-vm\n' > hosts
  $ ansible-playbook -i hosts obnam-server.yml
  ```

  This uses a pre-created self-signed certificate from `files/server.key` and `files/server.pem`, and is probably only good for trying out Obnam. You may want to generate your own certificates instead, e.g. using OpenSSL[2] something like this:

  ```
  $ openssl req -x509 -newkey rsa:4096 -passout pass:hunter2 \
    -keyout key.pem -out cert.pem -days 365 -subj /CN=localhost
  ```

  Put the generated keys into `/etc/obnam` (the location can be configured with `tls_key` and `tls_cert` keys in `/etc/obnam/server.yaml`, which we are about to describe).

Check that the server is installed and running:

```
$ sudo systemctl is-active obnam
active
```

Ansible created a directory, `/srv/obnam/chunks`, that will contain the backed-up data. If you want to use a different directory, you have to stop the service, move the existing directory to a new location, and update Obnam's configuration:

```
$ sudo systemctl stop obnam
$ sudo mv /srv/obnam /the/new/location/
$ sudoedit /etc/obnam/server.yaml
```

In the editor, you'll see something like this:

```
address: 0.0.0.0:443
chunks: /srv/obnam/chunks
tls_key: /etc/obnam/server.key
tls_cert: /etc/obnam/server.pem
```

Paths to TLS files might be different if you're using Let's Encrypt. Anyway, you have to edit `chunks` key to point at the new location. Once you're done, save the file and start the server again:

[1] https://letsencrypt.org/

[2] https://www.openssl.org/

```
$ sudo systemctl start obnam
$ sudo systemctl is-active obnam
active
```

Half the job done, another half to go! Let's set up a client now.

2 Setting up a client

There is a Debian package built by CI from every commit. It works on Debian 10 (buster) and later. You can run a script to install it:

```
$ curl -s https://gitlab.com/larswirzenius/obnam/-/raw/main/install-debian.sh | sudo bash
```

If you'd rather not download a script from the Internet and run it as root (kudos!), you can do the same steps manually. Add the following to `/etc/apt/sources.list.d/obnam.list`:

```
deb http://ci-prod-controller.vm.liw.fi/debian unstable-ci main
```

Then save the following PGP public key as `/etc/apt/trusted.gpg.d/obnam.asc`:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFrLO7kBEADdz6mHstYmKU5Dp6OSjxWtWaqTDOX1sJdmmaIK/9EKVIHOMaxp
5kvVO5G6mULLAjv/kLGOMxasHPrq8I2A/y8AqKAGVL8QelwLjQMIFZ3O/VbGQPHS
+T5TZXEnoQtNce1GUhFwJ38ZyjjwHBFV9tSec7rZ2Q3YeM3nNnGPf6DacXGfEOPO
HIN4sXAN2hzNXNjKRzTIvxQseb6nr7afUh/SlZ3yhQOCrIzmYlD7tP9WJe7ofL0p
JY4pDQYw8rT6nC2BE/ioemh84kERCT1vCe+OVFlSRuMlqfEv+ZpKQ+itOmPDQ/lM
jpUm1K2hrW/lWpxT/ZxHKo/w1K36J5WshgMZxfUu5BMCL9LMqMcrXNhNjDMfxDMM
3yBPOvQ4ls6fecOZ/bsFo1p8VzMk/w/eG8vPs5yuNa5XxN95yFMXoOHGb5Xbu8D4
6yiW+Af7OLbiSNpGdmNdneiGB2fY38NxBukPw5u3S5qG8HedSmMr1RvSr5kHoAAe
UbOY+BYaaKsTAT7+1skUW1o3FJSqoRKCHAzTsMWC6zzhR8hRn7jVrrguH1hGbqq5
TZSCFQZExuTJ7uXrTLG0WoBXIjB5wWNcSeXn8myUWYB51nJNF4tJBouZOz9JwWGl
kiAQkrHnBttLQWdW9FyjbIoTZMtpvVx+m6ObGTGdGL1cNlLAvWprMXGc+QARAQAB
tDJJY2sgQVBUIHJlcG9zaXRvcnkgc2lnbmluZyBrZXkgKDIwMTgpIDxsaXdAbGl3
LmZpPokCTgQTAQgAOBYhBKL1uyDoXyxUH3O717Wr+TZVS6PGBQJayzu5AhsDBQsJ
CAcCBhUICQoLAgQWAgMBAh4BAheAAAoJELWr+TZVS6PGB5QQANTcikhRUHwt9N4h
dGc/Hp6CbqdshMoWlwpFskttoVDxQG5OAobuZl5XyzGcmja1lT85RGkZFfbca0IZ
LnXOLLSAu51QBkXNaj4OhjK/0uQ+ITrvL6RQSXNgHiUTR/W2XD1GIUq6nBqe2GSN
31S1baYKKVj5QIMsi7Dq8ls3BBXuPCE+xTSaNmGWjes2t9pPidcRvxsksCLY1qgw
P1GFXBeMkBQ29kBP87SUL15SIk7OiQLlEURCy5iRls5rt/YEsdEpRWIb0Tm5Nrjv
2M3VM+iBhfNXTwj0rJ34mlycF1qQmA7YcTEobT7z587GPY0VWzBpQUnEQj7rQWPM
cDYY0b+I6kQ8VKOaL4wVAtE98d7HzFIrIrwhTKufnrWrVDPYsmLZ+LPC1jiF7JBD
SR6Vftb+SdDR9xoE1yRuXbC6IfoW+5/qQNrdQ2mm9BFw5jOonBqchs18HTTf3441
6SWwP9fY3Vi+IZphPPi0Gf85oMStgnv/Wnw6LacEL32ek39Desero/D8iGLZernK
Q2mC9mua5A/bYGVhsNWyURNFkKdbFa+/wW3NfdKYyZnsSfo+jJ2luNewrhAY7Kod
GWXTer9RxzTGA3EXFGvNr+BBOOxSj0SfWTlOOlo7J5dnxof+jLAUS1VHpceHGHps
GSJSdir7NkZidgwoCPA7BTqsb5LN
```

```
=dXB0
-----END PGP PUBLIC KEY BLOCK-----
```

After that, run the following commands to install Obnam:

```
$ sudo apt update
$ sudo apt install obnam
```

Now verify that everything is installed correctly:

```
$ obnam --version
obnam-backup 0.3.1
```

The version might be different, but at least there should **not** be any errors.

3 Making a backup

To create a backup, client needs to know three things: where the backup server is, where the live data is, and what key to use for encryption. To tell the client about the first two, create a file `~/.config/obnam/obnam.yaml` with contents like this:

```
server_url: https://obnam.example.com:443
roots:
  - /home/joe
  - /home/ann
  - /etc
  - /var/spool
```

Adjust the server address to match what you previously configured on the server. The `roots` key is a list of all the directories that Obnam should back up. Make sure that the roots are accessible to the user who would be doing the backup — the user has to be able to read their contents to back them up.

To generate an encryption key, run `obnam init` and type a passphrase. You don't need to remember it — it's just an additional random input to aid key generation; Obnam will not prompt you for it ever again. The generated key will be saved into `~/.config/obnam/passwords.yaml`, and *that's* the file you should not lose: you can't make or restore backups without it. Consider copying it somewhere separate from your main backup.

With that, you're ready to make your first backup! Run the following command, and watch Obnam go through all the files in your roots:

```
$ obnam backup
elapsed: 7s
files: 3422/0
current: /home/ann/music/Beethoven/1.flac
```

Depending on how much data you have under the roots, this might take a while. But once Obnam is done, it will print out a report like this:

```
status: OK
duration: 85
file-count: 1223
generation-id: 3905a0ad-9971-413c-ac81-ca8587c5f8c2
```

That's how you know you've got a backup! Hold off the celebration, though; the backups are only as good as your ability to use them, so let's check if you can recover the files you just backed up.

4 Restoring a backup

Let's imagine that your disk crapped out. In that case, you probably want to just grab the latest backup. In other cases, you might find that a file you thought useless and deleted long ago is actually important. To restore it, you need to find the backup that still has it.

The first order of business is to restore your `passwords.yaml`. If you already have it on your current machine, great; if not, you'll have to restore it from some *other* backup before you can use Obnam to restore everything else. It's impossible to recover any data without knowing the key, since it's all encrypted.

Got the `passwords.yaml` in place? Good. Let's get a list of all your backups with `obnam list`:

```
$ obnam list
6d35e3dd-3264-4269-a9d3-74fbd354c90e 2021-01-13 02:32:50.482465724 +0300
e4387899-d1dd-4e42-bc57-f56e6097d235 2021-01-14 02:36:00.029561204 +0300
9acde8d9-c167-4ad0-86b6-560c711713e1 2021-01-18 02:45:56.865274252 +0300
708db71e-d863-47e6-92c3-679041e25c8e 2021-01-20 02:49:50.664349817 +0300
0f3a63d0-d992-42ff-ab77-7e2457745a40 2021-01-22 03:00:56.902063598 +0300
028ce888-4a5b-438c-978c-0812646165cf 2021-02-07 16:18:19.008757980 +0300
481bb25f-5377-4e41-b824-4e60fda8f01c 2021-02-08 19:04:44.072710112 +0300
5067e10e-2d4d-4ff4-a9a0-568ed008dd2c 2021-02-11 20:26:06.589610566 +0300
3905a0ad-9971-413c-ac81-ca8587c5f8c2 2021-02-12 22:35:20.431081194 +0300
```

That second-to-last backup, 5067e10e-2d4d-4ff4-a9a0-568ed008dd2c, looks like it's old enough. Let's see what files it contains:

```
$ obnam list-files 5067e10e-2d4d-4ff4-a9a0-568ed008dd2c
```

You might need to `grep` the result to check for specific files. Anyway, suppose this backup it exactly what you need. Let's restore it to a directory called "yesterday":

```
$ obnam restore 5067e10e-2d4d-4ff4-a9a0-568ed008dd2c yesterday
```

Obnam will print out a progress bar and some stats. Once the restoration is done, you can look under `yesterday/` to find the file you needed. Easy!

Now you're prepared for the worst. (Unless *both* your primary and backup disks break. Or your backup server is inaccessible. Or you lose all copies of `passwords.yaml`. Or there is no electrical grid anymore to power your devices. Or the zombies are trying to break in, distracting you from reading this tutorial. Look up "disaster recovery planning"—oh right, no electricity.)

5 Where to go from here

Obnam is still at the alpha stage, so it's likely that the instructions above didn't quite work for you. If so, please open issues[3] and help us improve Obnam!

If you're interested in more details, and especially in how Obnam works internally, take a look at the Subplot file[4]. It not just explains things, but also contains acceptance criteria and tests for them. Great stuff!

[3] `https://gitlab.com/larswirzenius/obnam/-/issues`

[4] `obnam.html`