

---

# Bitcoin DCA

Nov 21, 2020



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Getting started</b>	<b>5</b>
<b>3</b>	<b>Configuration</b>	<b>9</b>
<b>4</b>	<b>Scheduling buy &amp; withdrawing</b>	<b>13</b>
<b>5</b>	<b>Setting up persistent storage for Bitcoin DCA</b>	<b>15</b>
<b>6</b>	<b>Deriving new Bitcoin addresses from your XPUB</b>	<b>17</b>
<b>7</b>	<b>Using tags to keep track of balance</b>	<b>19</b>
<b>8</b>	<b>Frequently Asked Questions</b>	<b>21</b>
<b>9</b>	<b>About this software</b>	<b>23</b>
<b>10</b>	<b>Supported Exchanges</b>	<b>25</b>
<b>11</b>	<b>Telegram / Support</b>	<b>27</b>



**Warning:** Please be aware this is beta software. Test thoroughly with small amounts of money at first. This software is provided “as is” and comes without warranty. See [LICENSE](#).



**Warning:** Please be aware this is beta software. Test thoroughly with small amounts of money at first. This software is provided “as is” and comes without warranty. See [LICENSE](#).

## 1.1 Requirements

- You need to have an account on a supported Exchange;
- You need to have Docker installed: <https://docs.docker.com/get-docker/>;
- You need to have an API key active on a supported Exchange. It needs **read**, **trade** and **withdraw** permission.

---

**Note:** Add your unprivileged user to the correct group to execute Docker commands without root: `$ sudo usermod -aG docker ${USER}`. You might need to log out & log back in for this to take effect.

---

## 1.2 Using Docker Hub (easiest)

### 1.2.1 Installing

Use these commands to download this tool from Docker Hub:

```
$ docker pull jorijn/bitcoin-dca:latest
```

### 1.2.2 Upgrading

Using these commands you can download the newest version from Docker Hub:

```
$ docker image rm jorijn/bitcoin-dca
$ docker pull jorijn/bitcoin-dca:latest
```

### 1.3 Build your own (more control)

If you desire more control, pull this project from [GitHub](#) and build it yourself. To do this, execute these commands:

```
cd ~
git clone https://github.com/Jorijn/bitcoin-dca.git
cd bitcoin-dca
docker build . -t jorijn/bitcoin-dca:latest
```

When an upgrade is available, run `git pull` to fetch the latest changes and build the docker container again.

Next: *[Configuration](#)*

---

**Note:** This guide is meant for people on Linux. You can use it on your VPS or Raspberry Pi. The Getting Started guide assumes you will be setting up Bitcoin DCA using the BL3P exchange. If you need to configure another exchange substitute the exchange specific configuration with the correct ones from *[Configuration](#)*.

---



---

**Note:** See *Installation* on how to download the tool to your server.

---

## 2.1 Configuration

Create a new file somewhere that will contain the configuration needed for the tool to operate. If your account is called bob and your home directory is */home/bob* lets create a new file in */home/bob/.bitcoin-dca*:

---

**Note:** See *Configuration* for all available options.

---

You can test that it work with:

Listing 1: Checking the Exchange balance

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca jorijn/bitcoin-dca:latest_
↵balance
```

If successful, you should see a table containing your balances on the exchange:

## 2.2 Testing

For safety, I recommend buying and withdrawing at least once manually to verify everything works before proceeding with automation.

### 2.2.1 Buying €10,00 of Bitcoin

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca jorijn/bitcoin-dca:latest buy_
↪10
```

### 2.2.2 Withdrawing to your hardware wallet

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca jorijn/bitcoin-dca:latest_
↪withdraw --all
```

**It will ask you:** Ready to withdraw 0.00412087 BTC to Bitcoin Address bc1abcdefghijklmnopqrstuvwxyz123456? A fee of 0.0003 will be taken as withdraw fee [y/N]:

**Warning:** When testing, make sure to verify the displayed Bitcoin address matches the one configured in your `.bitcoin-dca` configuration file. When confirming this question, withdrawal executes immediately.

## 2.3 Automating buying and withdrawing

The `buy` and `withdraw` command both allow skipping the confirmation questions with the `--yes` option. By leveraging the system's cron daemon on Linux, you can create flexible setups. Use the command `crontab -e` to edit periodic tasks for your user:

Since it's best to use absolute paths in crontabs, we'll be using `$(command -v docker)` to have it automatically determined for you.

Listing 2: Finding out where Docker is located

```
$ command -v docker
-> /usr/bin/docker
```

### 2.3.1 Example: Buying €50.00 of Bitcoin and withdrawing every monday. Buy at 3am and withdraw at 3:30am.

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↪bitcoin-dca:latest buy 50 --yes --no-ansi
30 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↪bitcoin-dca:latest withdraw --all --yes --no-ansi
```

### 2.3.2 Example: Buying €50.00 of Bitcoin every week on monday, withdrawing everything on the 1st of every month.

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↪bitcoin-dca:latest buy 50 --yes --no-ansi
0 0 1 * * $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↪bitcoin-dca:latest withdraw --all --yes --no-ansi
```

### 2.3.3 Example: Send out an email when Bitcoin was bought

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/  
↳bitcoin-dca:latest buy 50 --yes --no-ansi 2>&1 |mail -s "You just bought more_  
↳Bitcoin!" youremail@here.com
```

---

**Note:** You can use the great tool at <https://crontab.guru/> to try more combinations.

---

## 2.4 Tips

- You can create and run this tool with different configuration files, e.g. different withdrawal addresses for your spouse, children or other saving purposes.
- On Linux, you can redirect the output to other tools, e.g. email yourself when Bitcoin is bought. Use `--no-ansi` to disable colored output.
- Go nuts on security, use a different API keys for buying and withdrawal. You can even lock your BL3P account to only allow a single Bitcoin address for withdrawal through the API.
- Although a bit technical, in a cron job, use `2>&1` to redirect any error output to the standard output stream. Basically this means that any error messages will show up in your email message too.



**Warning:** Please be aware this is beta software. Test thoroughly with small amounts of money at first. This software is provided “as is” and comes without warranty. See [LICENSE](#).

Bitcoin DCA uses [environment variables](#) to configure the inner workings of the tool. An environment variable looks like this: `SOME_CONFIGURATION_KEY=valuehere`.

### 3.1 Getting Started Template

```

1 #####
2 # GENERIC APPLICATION SETTINGS
3 #####
4
5 # You can either use this or WITHDRAW_XPUB. Choosing this one will make the tool
6   ↳ withdraw to the same Bitcoin address
7 # every time.
8 WITHDRAW_ADDRESS=
9
10 # You can either use this or WITHDRAW_ADDRESS. Choosing this one will make the tool
11   ↳ withdraw to a new receiving address
12 # every time a withdrawal is being made by the tool. It'll start at the first address
13   ↳ at index 0, so make sure to
14 # generate a new account or key when using this method.
15 # WITHDRAW_XPUB=
16
17 # Choose the cryptocurrency exchange this Bitcoin DCA tool will operate on. The
18   ↳ default value is "b13p".
19 # Available options: b13p, bitvavo
20 EXCHANGE=b13p

```

(continues on next page)

(continued from previous page)

```
18 #####
19 # BL3P exchange settings
20 #####
21
22 # This is the identifying part of the API key that you created on the BL3P exchange.
23 ↪ You can find it there under the
24 # name Identifier Key.
25 BL3P_PUBLIC_KEY=
26
27 # This is the private part of your API connection to BL3P. It's an encoded secret
28 ↪ granting access to your BL3P account.
29 BL3P_PRIVATE_KEY=
30
31 #####
32 # Bitvavo exchange settings
33 # > no trading fees up to the first € 1000,- if you use my affiliate link: https://
34 ↪ bitvavo.com/?a=DE4151B112
35 #####
36
37 # This is the identifying part of the API key that you created on the Bitvavo
38 ↪ exchange.
39 # BITVAVO_API_KEY=
40
41 # This is the private part of your API connection to Bitvavo. It's an encoded secret
42 ↪ granting access to your Bitvavo account.
43 # BITVAVO_API_SECRET=
```

## 3.2 Available Configuration

This part of the documentation is split up in generic application settings that decide how the tool should act for Dollar Cost Averaging. The last part is for exchange specific configuration like API keys.

### 3.2.1 DCA Settings

#### WITHDRAW\_ADDRESS

You can either use this or WITHDRAW\_XPUB. Choosing this one will make the tool withdraw to the same Bitcoin address every time.

**Example:** WITHDRAW\_ADDRESS=3AT3tf4cVfGRaQ87HpGQppTYmMrb5kpGQb

#### WITHDRAW\_XPUB

You can either use this or WITHDRAW\_ADDRESS. Choosing this one will make the tool withdraw to a new receiving address every time a withdrawal is being made by the tool. It'll start at the first address at index 0, so make sure to generate a new account or key when using this method.

**Example:** WITHDRAW\_XPUB=ypub6Y4RxNmNrdnwdwxERYnXa9rGd4upqeeJ3ixkJQUCQL8UcwYtXj86eXS5fVGU5xsmu



## 3.3 Feeding configuration into the DCA tool

### 3.3.1 Using a configuration file

When handling multiple environment variables, things can get messy. For easier management you can create a simple configuration file somewhere on your disk and use that to provide the tool with the correct configuration.

For example, creating a new configuration file in your home directory: `nano /home/username/.bitcoin-dca`

---

**Note:** You can find out where your home directory is using `$ echo $HOME`.

---

Listing 1: `/home/username/.bitcoin-dca`

```
BL3P_PUBLIC_KEY=....  
BL3P_PRIVATE_KEY=....  
WITHDRAW_ADDRESS=....
```

Now, when running the tool you can use `--env-file` like this:

Listing 2: Providing configuration with Docker's `-env-file`

```
$ docker run --rm -it --env-file=/home/username/.bitcoin-dca jorijn/bitcoin-  
↪dca:latest balance
```

### 3.3.2 Using inline arguments

For maximum control, you can also feed configuration into the tool like this:

---

**Note:** While this gives you more control, it will also allow other people who have access your machine to see the arguments with which you've started the Docker container, thus revealing your API keys.

---

Listing 3: Providing configuration by specifying each configuration item separately

```
$ docker run --rm -it -e BL3P_PUBLIC_KEY=abcd -e BL3P_PRIVATE_KEY=abcd WITHDRAW_  
↪ADDRESS=abcd jorijn/bitcoin-dca:latest balance
```



---

### Scheduling buy & withdrawing

---

---

**Note:** This guide is meant for people on Linux. You can use it on your VPS or Raspberry Pi.

---

The `buy` and `withdraw` command both allow skipping the confirmation questions with the `--yes` option. By leveraging the system's cron daemon on Linux, you can create flexible setups. Use the command `crontab -e` to edit periodic tasks for your user:

Since it's best to use absolute paths in crontabs, we'll be using `$(command -v docker)` to have it automatically determined for you.

Listing 1: Finding out where Docker is located

```
$ command -v docker
-> /usr/bin/docker
```

### 4.1 Example: Buying €50.00 of Bitcoin and withdrawing every mon-day. Buy at 3am and withdraw at 3:30am.

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↳bitcoin-dca:latest buy 50 --yes --no-ansi
30 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↳bitcoin-dca:latest withdraw --all --yes --no-ansi
```

### 4.2 Example: Buying €50.00 of Bitcoin every week on monday, with-drawing everything on the 1st of every month.

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↳bitcoin-dca:latest buy 50 --yes --no-ansi
0 0 1 * * $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↳bitcoin-dca:latest withdraw --all --yes --no-ansi
```

### 4.3 Example: Send out an email when Bitcoin was bought

```
0 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca jorijn/
↳bitcoin-dca:latest buy 50 --yes --no-ansi 2>&1 |mail -s "You just bought more_
↳Bitcoin!" youremail@here.com
```

---

**Note:** You can use the great tool at <https://crontab.guru/> to try more combinations.

---

#### 4.3.1 Tips

- You can create and run this tool with different configuration files, e.g. different withdrawal addresses for your spouse, children or other saving purposes.
- On Linux, you can redirect the output to other tools, e.g. email yourself when Bitcoin is bought. Use `--no-ansi` to disable colored output.
- Go nuts on security, use a different API keys for buying and withdrawal. You can even lock your BL3P account to only allow a single Bitcoin address for withdrawal through the API.
- Although a bit technical, in a cron job, use `2>&1` to redirect any error output to the standard output stream. Basically this means that any error messages will show up in your email message too.

---

## Setting up persistent storage for Bitcoin DCA

---

### 5.1 What do I need persistent storage for?

- *Using tags to keep track of balance*
- *Deriving new Bitcoin addresses from your XPUB*

Since it's plain buying and withdrawing, Bitcoin DCA doesn't need to remember any state for regular operations. However, when it comes to XPUB's and tagging, you do. In the case of tagging it has to remember how much balance each tag has and for XPUB's it needs to save the active index for address derivation. Since it's starting at 0, not saving the state would cause Bitcoin DCA to always return the same, first, address.

Currently, the internal applications stores the data at the `/app/var/storage` path but since this is an internal Docker container path, you will need to `mount` a new location to this path to have the storage be persistent.

### 5.2 Picking a location

Lets create a new directory somewhere in your home directory. For this example, we'll assume your username is `bob` and your home directory is found located at `/home/bob`.

---

**Note:** You can find out where your home directory is using `$ echo $HOME`.

---

We'll be creating a new directory here where the files will be stored:

```
$ mkdir -p /home/bob/applications/bitcoin-dca
```

## 5.3 Running with a mounted volume

Now, when running this tool you need to mount the new storage directory onto the container using argument `-v /home/bob/applications/bitcoin-dca:/app/var/storage`. A typical command could look like this:

Listing 1: Running withdraw with a persistent storage directory

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca -v /home/bob/applications/  
↪bitcoin-dca:/app/var/storage jorijn/bitcoin-dca:latest withdraw --all
```

---

## Deriving new Bitcoin addresses from your XPUB

---

---

**Note:** You need persistent storage to keep track of which address index the tool should use. See *Setting up persistent storage for Bitcoin DCA*

---

Instead of withdrawing to the same static Bitcoin address every time you make a withdrawal, it's also possible to supply a Master Public Key to Bitcoin DCA.

After configuring, Bitcoin DCA will start at the first address (index #0) it can derive from your XPUB.

### 6.1 Configuring a XPUB

For the sake of demonstration, we'll be using the following XPUB here:

**Warning:** It's **very important** that you verify the configured XPUB to make sure your Bitcoin will be sent to addresses in your possession.

### 6.2 Verifying the configured XPUB

You can verify that Bitcoin DCA will derive the correct addresses using the following command:

You can check that the correct address is being used when attempting to withdraw your Bitcoin:

After successful withdrawal, the tool will increase the inner index and use address #1 the next time a withdrawal is being made.



---

## Using tags to keep track of balance

---

---

**Note:** You need persistent storage to keep track of tagged balances. See *Setting up persistent storage for Bitcoin DCA*

---

### 7.1 What is tagging and what can I use it for?

Tagging is a multi tenant solution for DCA. It enables you to categorize each buy and maintain a balance for each category created. For example, you could use it to save some Bitcoin for your children. It's as easy as supplying.

### 7.2 Example: Bobby

Lets say I have a kid called Bobby, I'm a true believer Bitcoin will someday rule the world and I would like to save some Bitcoin for him separately from my own account. I would then buy Bitcoin the regular way, except now I would provide a new argument: `-t bobby`.

Listing 1: Buying € 10,- of Bitcoin for Bobby

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca jorijn/bitcoin-dca:latest buy_
↵10 -t bobby
```

Then, when time comes to withdraw, you can withdraw his funds to a wallet of his own:

### Listing 2: Withdrawing all Bitcoin specifically for Bobby

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca-bobby jorijn/bitcoin-  
→dca:latest withdraw --all -t bobby
```

---

**Note:** Docker can't handle both `-e` and `--file-file`, it's best to create a separate configuration file containing another xpub or withdrawal address.

---

Of course, other examples are possible, e.g. tagging balance for buying a house or a car.

### 7.2.1 Technical note, tagging works like this:

- Buying 10.000 with tag `mike`, Mike's balance is now 10.000, total balance 10.000;
- Buying 10.000 with tag `bobby`, Bobby's balance is now 10.000, total balance 20.000;
- Buying 15.000 with tag `mike`, Mike's balance is now 25.000, total balance 35.000;
- Withdrawing all for tag `mike`, initiating a withdrawal for 25.000 leaving the balance for Mike at 0 and Bobby 10.000.



---

## Frequently Asked Questions

---

### 8.1 I already have MyNode running, can I use this tool too?

Yes! MyNode is based on Linux and has Docker already installed. You can use all features of Bitcoin DCA.

Things you should keep in mind: The default user, `admin` doesn't have permission to run Docker by default.

---

**Note:** Add your unprivileged user to the correct group to execute Docker commands without root: `$ sudo usermod -aG docker ${USER}`. You might need to log out & log back in for this to take effect.

---

See *Getting started* for more information.



## CHAPTER 9

---

### About this software

---

This DCA (Dollar Cost Averaging) tool is built with flexibility in mind, allowing you to specify your own schedule for buying and withdrawing.

A few examples of possible scenario's:

- Buy each week, never withdraw;
- Buy monthly and withdraw at the same time to reduce exchange risk;
- Buy each week but withdraw only at the end of the month to save on withdrawal fees.



## CHAPTER 10

---

### Supported Exchanges

---

Exchange	URL	Currencies
BL3P	<a href="https://bl3p.eu/">https://bl3p.eu/</a>	EUR
Bitvavo	<a href="https://bitvavo.com/">https://bitvavo.com/</a>	EUR



## CHAPTER 11

---

Telegram / Support

---

You can visit the Bitcoin DCA Support channel on Telegram: <https://t.me/bitcoindca>