

---

# Bitcoin DCA

Apr 12, 2023



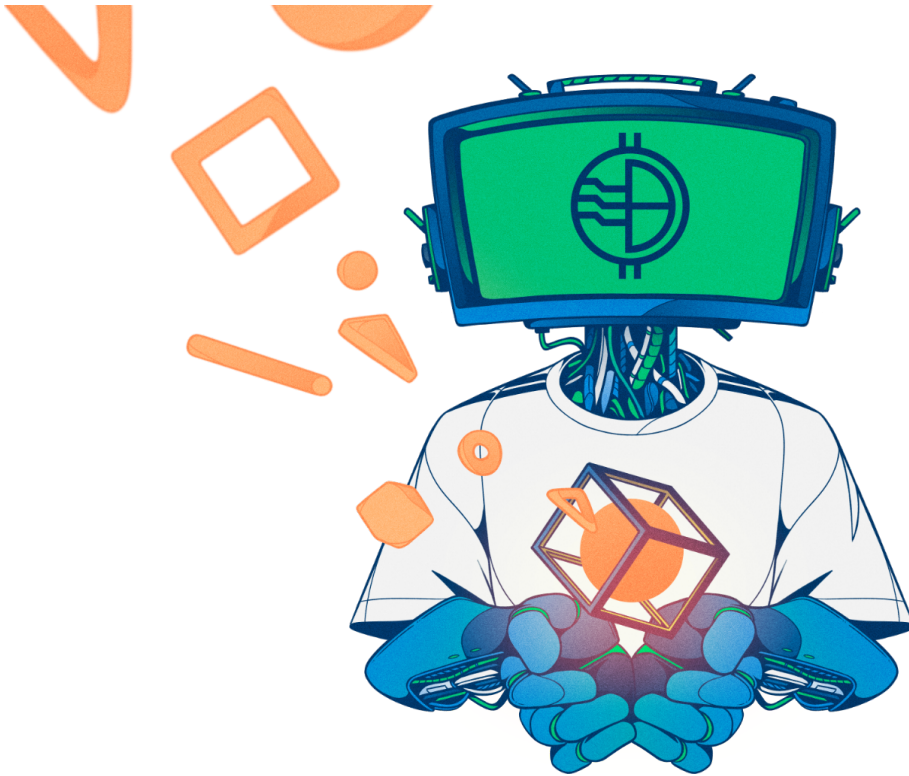
---

## 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>17</b>
<b>5</b>	<b>Setting up persistent storage for Bitcoin DCA</b>	<b>19</b>
<b>6</b>	<b>Deriving new Bitcoin addresses from your XPUB</b>	<b>21</b>
<b>7</b>	<b>Using tags to keep track of balance</b>	<b>23</b>
<b>8</b>	<b>Getting notified when Bitcoin DCA makes a purchase or does a withdrawal</b>	<b>25</b>
<b>9</b>	<b>Frequently Asked Questions</b>	<b>29</b>
<b>10</b>	<b>About this software</b>	<b>31</b>
<b>11</b>	<b>Supported Exchanges</b>	<b>33</b>
<b>12</b>	<b>Telegram / Support</b>	<b>35</b>
<b>13</b>	<b>Contributing</b>	<b>37</b>







### 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 ghcr.io/jorijn/bitcoin-dca:latest
```

#### 1.2.2 Upgrading

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

```
$ docker image rm ghcr.io/jorijn/bitcoin-dca
$ docker pull ghcr.io/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 ghcr.io/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](#).

---



# CHAPTER 2

## Getting started

**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*:

Listing 1: */home/bob/.bitcoin-dca*

```
BL3P_PRIVATE_KEY=bl3p private key here
BL3P_PUBLIC_KEY=bl3p identifier key here
WITHDRAW_ADDRESS=hardware wallet address here
```

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

You can test that it work with:

Listing 2: Checking the Exchange balance

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

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

+	-----+	-----+	-----+
	Currency	Balance	Available
	BTC	0.00000000 BTC	0.00000000 BTC

(continues on next page)

(continued from previous page)

```
| EUR      | 10.00000000 EUR | 10.00000000 EUR |
| BCH      | 0.00000000 BCH  | 0.00000000 BCH  |
| LTC      | 0.00000000 LTC  | 0.00000000 LTC  |
+-----+-----+-----+
```

## 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 ghcr.io/jorijn/bitcoin-
↪dca:latest buy 10
```

### 2.2.2 Withdrawing to your hardware wallet

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

**It will ask you:** Ready to withdraw 0.00412087 BTC to Bitcoin Address bc1abcdeghijklmnopqrstuvwxyz123456? 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 3: 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 ghcr.io/
↳ jorijn/bitcoin-dca:latest buy 50 --yes --no-ansi
30 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca ghcr.io/
↳ 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 ghcr.io/
↳ jorijn/bitcoin-dca:latest buy 50 --yes --no-ansi
0 0 1 * * $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca ghcr.io/
↳ jorijn/bitcoin-dca:latest withdraw --all --yes --no-ansi
```

---

**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.
- 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.



# CHAPTER 3

## Configuration

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 "bl3p".
19 # Available options: bl3p, bitvavo, kraken, binance
20 EXCHANGE=bl3p
21
22 # This setting is for the base currency you're buying with. Options are:
23 # BL3P: EUR
24 # Bitvavo: EUR
25 # Kraken: USD EUR CAD JPY GBP CHF AUD
26 # Binance: USDT BUSD EUR USDC USDT GBP AUD TRY BRL DAI TUSD RUB UAH PAX BIDR NGN
27 ↳ IDRT VAI
```

(continues on next page)

(continued from previous page)

```

23 BASE_CURRENCY=EUR
24
25 # If you want to receive an email when buys or withdrawals are made, activate this_
    ↳ setting. The example here is for the Sendgrid API
26 # Available providers: https://bitcoin-dca.readthedocs.io/en/latest/getting-notified.
    ↳ html#sending-email-from-bitcoin-dca
27 # NOTIFICATION_EMAIL_ENABLED=1
28 # NOTIFICATION_EMAIL_DSN=sendgrid+api://APIKEYHERE@default
29 # NOTIFICATION_EMAIL_TO=you@gmail.com
30 # Optionally you can also configure these extra settings:
31 # NOTIFICATION_EMAIL_FROM=you@gmail.com
32 # NOTIFICATION_EMAIL_PREFIX="Bitcoin DCA"
33
34 # Bitcoin DCA can notify you on Telegram when purchases or withdrawals are made. You_
    ↳ will need to
35 # contact Botfather to create a token: https://t.me/botfather
36 # To find out your Telegram ID you can talk to https://t.me/getmyid_bot
37 # NOTIFICATION_TELEGRAM_ENABLED=1
38 # NOTIFICATION_TELEGRAM_DSN=telegram://BOTFATHERSECRET@default?channel=YOURTELEGRAMID
39
40 # Bitcoin DCA will contact GitHub every time it is executed to let you know if there_
    ↳ is a newer version available.
41 # Newer versions bring important security updates and new features. It transmits no_
    ↳ information about your local environment.
42 # You can audit the code here: https://github.com/Jorijn/bitcoin-dca/blob/master/src/_
    ↳ EventListener/CheckForUpdatesListener.php
43 # You can completely disable remote version checking by uncommenting this setting:
44 # DISABLE_VERSION_CHECK=1
45
46 #####
47 # BL3P exchange settings
48 #####
49
50 # This is the identifying part of the API key that you created on the BL3P exchange._
    ↳ You can find it there under the
51 # name Identifier Key.
52 BL3P_PUBLIC_KEY=
53
54 # This is the private part of your API connection to BL3P. It's an encoded secret_
    ↳ granting access to your BL3P account.
55 BL3P_PRIVATE_KEY=
56
57 #####
58 # Bitvavo exchange settings
59 # > no trading fees up to the first € 1000,- if you use my affiliate link: https://
    ↳ bitvavo.com/?a=DE4151B112
60 #####
61
62 # This is the identifying part of the API key that you created on the Bitvavo_
    ↳ exchange.
63 # BITVAVO_API_KEY=
64
65 # This is the private part of your API connection to Bitvavo. It's an encoded secret_
    ↳ granting access to your Bitvavo account.
66 # BITVAVO_API_SECRET=
67
68

```

(continues on next page)

(continued from previous page)

```

69 #####
70 # Kraken exchange settings
71 #####
72
73 # This is the identifying part of the API key that you created on the Kraken exchange.
74 # KRAKEN_API_KEY=
75
76 # This is the private part of your API connection to Kraken. It's an encoded secret
77 ↪granting access to your Kraken account.
78 # KRAKEN_PRIVATE_KEY=
79
80 # Kraken secured the platform by limiting API usage to pre-whitelisted withdrawal
81 ↪addresses.
82 # This makes it a lot more secure but unfortunately limits the tool to one withdrawal
83 ↪address
84 # thus disabling XPUB generation. On Kraken, go to Funding and create a new Bitcoin
85 ↪withdrawal
86 # address and for description use something without special symbols or spaces.
87 ↪Configure the
88 # value here.
89 #
90 # [!] This option overrides any configured withdrawal address
91 # KRAKEN_WITHDRAW_DESCRIPTION=bitcoin-dca
92
93 # When you request to buy 100 EUR/USD from Kraken they assume you want to buy a
94 ↪minimum of 100
95 # by default. If the fee would be 0.30 that would be added to the 100, resulting in
96 ↪100.30 being
97 # deducted from your EUR/USD balance. If you're transferring a fixed amount of money
98 ↪for a fixed
99 # amount of DCA cycles this would result in a lack of balance for the final complete
100 ↪DCA purchase
101 # of that cycle.
102 #
103 # include (default): deducts the fee estimation from your order, this will ensure you
104 ↪have enough
105 #
106 # balance left for the final DCA cycle.
107 # exclude: Kraken default, the tool will order for 100 and Kraken will pay
108 ↪the fee with
109 #
110 # the remainder of your balance.
111 # KRAKEN_FEE_STRATEGY=include
112 #
113 # ONLY FOR GERMAN RESIDENTS:
114 # If your Kraken account is verified with a German address, you will need to accept a
115 ↪trading agreement
116 # in order to place market and margin orders.
117 #
118 # See https://support.kraken.com/hc/en-us/articles/360036157952
119 #
120 # KRAKEN_TRADING_AGREEMENT=agree
121
122 #####
123 # Binance exchange settings
124 #####
125
126 # This is the identifying part of the API key that you created on the Binance
127 ↪exchange.

```

(continues on next page)

(continued from previous page)

```
113 # BINANCE_API_KEY=  
114  
115 # This is the private part of your API connection to Binance. It's a secret granting_  
    ↳ access to your Binance account.  
116 # BINANCE_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 Application 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`

#### EXCHANGE

This configuration value determines which exchange will be used for buys and withdrawals. The default value is `BL3P`.

Available options: `bl3p`, `bitvavo`, `kraken`, `binance`

**Example:** `EXCHANGE=bl3p`

#### DISABLE\_VERSION\_CHECK

Bitcoin DCA will contact GitHub every time it is executed to let you know if there is a newer version available. Newer versions bring important security updates and new features. It transmits no information about your local environment. You can audit [the code here](#). You can completely disable remote version checking by uncommenting this setting:

**Example:** `DISABLE_VERSION_CHECK=1`

### 3.2.2 Email & Telegram notifications

To provide in-depth information about sending notifications through email and telegram, [please see this article](#).



### 3.2.3 Exchange: BL3P

#### BL3P\_PUBLIC\_KEY

This is the identifying part of the API key that you created on the BL3P exchange. You can find it there under the name **Identifier Key**.

**Example:** BL3P\_PUBLIC\_KEY=0a12345b-01a1-1a1a-012a-a1bc23ef45bg

#### BL3P\_PRIVATE\_KEY

This is the private part of your API connection to BL3P. It's an encoded secret granting access to your BL3P account.

**Example:** BL3P\_PRIVATE\_KEY=aHR0cHM6Ly93d3cueW91dHVizS5jb20vd2F0Y2g/djlkUXc0dzlXZ1hjUQ==

#### BL3P\_API\_URL (optional)

The endpoint where the tool should connect to.

**Example:** BL3P\_API\_URL=https://api.bl3p.eu/1/

### 3.2.4 Exchange: Bitvavo

#### BITVAVO\_API\_KEY

This is the identifying part of the API key that you created on the Bitvavo exchange.

**Example:** BITVAVO\_API\_KEY=1006e89gd84e8f3a5209b2762d1bbef36eds5e6108e7696f6117556830b0e3dy

#### BITVAVO\_API\_SECRET

This is the private part of your API connection to Bitvavo.

**Example:** BITVAVO\_API\_SECRET=aHR0cHM6Ly93d3cueW91dHVizS5jb20vd2F0Y2g/djlkUXc0dzlXZ1hjUQ==

#### BITVAVO\_API\_URL (optional)

The endpoint where the tool should connect to.

**Example:** BITVAVO\_API\_URL=https://api.bitvavo.com/v2/

### 3.2.5 Exchange: Kraken

#### KRAKEN\_API\_KEY

This is the identifying part of the API key that you created on the Kraken exchange.

**Example:** KRAKEN\_API\_KEY=1006e89gd84e8f3a5209b2762d1bbef36eds5e6108e7696f6117556830b0e3dy

### KRAKEN\_PRIVATE\_KEY

This is the private part of your API connection to Kraken. It's an encoded secret granting access to your Kraken account.

**Example:** `KRAKEN_PRIVATE_KEY=aHR0cHM6Ly93d3cueW91dHVhZS5jb20vd2F0Y2g/dj1kUXc0dz1lXZ1hjUQ==`

### KRAKEN\_WITHDRAW\_DESCRIPTION

Kraken secured the platform by limiting API usage to pre-whitelisted withdrawal addresses. This makes it a lot more secure but unfortunately limits the tool to one withdrawal address thus disabling XPUB generation. On Kraken, go to Funding and create a new Bitcoin withdrawal address and for description use something without special symbols or spaces. Configure the value here.

**Example:** `KRAKEN_WITHDRAW_DESCRIPTION=bitcoin-dca`

### KRAKEN\_API\_URL (optional)

The endpoint where the tool should connect to.

**Default:** `KRAKEN_API_URL=https://api.kraken.com/`

### KRAKEN\_FEE\_STRATEGY (optional)

When you request to buy 100 EUR/USD from Kraken they assume you want to buy a minimum of 100 by default. If the fee would be 0.30 that would be added to the 100, resulting in 100.30 being deducted from your EUR/USD balance. If you're transferring a fixed amount of money for a fixed amount of DCA cycles this would result in a lack of balance for the final complete DCA purchase of that cycle.

Option `include` (default): deducts the fee estimation from your order, this will ensure you have enough balance left for the final DCA cycle. Option `exclude`: Kraken default, the tool will order for 100 and Kraken will pay the fee with the remainder of your balance.

**Default:** `KRAKEN_FEE_STRATEGY=include`

### KRAKEN\_TRADING\_AGREEMENT (only for German residents)

If your Kraken account is verified with a German address, you will need to accept a trading agreement in order to place market and margin orders.

See <https://support.kraken.com/hc/en-us/articles/360036157952>

If you agree, fill this value with `agree`, like this: `KRAKEN_TRADING_AGREEMENT=agree`

## 3.2.6 Exchange: Binance

Your Binance API key should hold at least the following permissions:

- Enable Reading
- Enable Spot & Margin Trading
- Enable Withdrawals

You should enable IP access restrictions to use withdrawal through the API. Enter the IP address that matches your outgoing connection. When in doubt, you can check your IP here: <https://nordvpn.com/nl/ip-lookup/>

### BINANCE\_API\_KEY

This is the identifying part of the API key that you created on the Binance exchange.

**Example:** `BINANCE_API_KEY=mkYEtMpzI9q9qrwvYzTe44nB495joEM17bhUDspFEkKHjzLmKwTlexvQYxGcL6db`

### BINANCE\_API\_SECRET

This is the private part of your API connection to Binance. It's a secret granting access to your Binance account.

**Example:** `BINANCE_API_SECRET=xXfw9vEiSdgllWfLs55uGC3ZBS3VyZMy1aGj4mYYlIIhX6hQ98AsGsQHLSKI4uj6`

### BINANCE\_API\_URL (optional)

The endpoint where the tool should connect to.

**Default:** `BINANCE_API_URL=https://api.binance.com/`

## 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 ghcr.io/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 ghcr.io/jorijn/bitcoin-dca:latest balance
```

## CHAPTER 4

---

### 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 ghcr.io/  
→jorijn/bitcoin-dca:latest buy 50 --yes --no-ansi  
30 3 * * mon $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca ghcr.io/  
→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 ghcr.io/  
→jorijn/bitcoin-dca:latest buy 50 --yes --no-ansi  
0 0 1 * * $(command -v docker) run --rm --env-file=/home/bob/.bitcoin-dca ghcr.io/  
→jorijn/bitcoin-dca:latest withdraw --all --yes --no-ansi
```

---

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

---

#### 4.2.1 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.
- 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.

---

## 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 X PUB*

Since it's plain buying and withdrawing, Bitcoin DCA doesn't need to remember any state for regular operations. However, when it comes to X PUB's and tagging, you do. In the case of tagging it has to remember how much balance each tag has and for X PUB'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 ghcr.io/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:

Listing 1: /home/bob/.bitcoin-dca

```
WITHDRAW_  
↪XPUB=zpub6rLtZSoXnXKPXHroRKGCwuRVHjgA5YL6oUkdZnCfbDLdtAKNXb1FX1EmPUYRluYMRBpngvkdJwxqhLvM46trRy5MRk
```

**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:

Listing 2: Verifying the configured XPUB

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca-bobby ghcr.io/jorijn/bitcoin-
↳dca:latest verify-xpub
```

#	Address	Next Withdraw
0	bclqvqatyv2xyntyarej2fcutj6w5yugy0gc9jx2nn	<
1	bclq360p67y3jvards9f2eud5rlu07q8ampfp35vp7	
2	bclqs4k3p9w4ke5np3lr3lgnma9jcaxedau8mpawu	
3	bclqpk48z0s7gvyrupm2wmd7nr0fdzkxa42372ver2	
4	bclq0uam3l30y43q0wjhq0kwf050uyg23mz7p3frr4	
5	bclqef62h9xt937lu9x5ydv204r7lpk3sjdc575kax	
6	bclq2rl0he7zca8a88ax7hf9259c33kd2ux5ffhkhqw	
7	bclqr9ffa3w6tae4g5m4ydnjvphg8tpgarf5yjqgz	
8	bclqr65srxamrmx8zumgv5puljnd93u3sj7lw6cnrg	
9	bclq2ufc8j9uw6x7hwqfsdakungk63etanxtkplel0	

[WARNING] Make sure these addresses match those **in** your client, **do** not use the `↳withdraw` **function if** they **do** not.

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

Listing 3: Here, it takes address #0 (bclqvqatyv2xyntyarej2fcutj6w5yugy0gc9jx2nn) for withdrawal

```
$ docker run --rm -it --env-file=/home/bob/.bitcoin-dca-bobby ghcr.io/jorijn/bitcoin-
↳dca:latestwithdraw --all
Ready to withdraw 0.0013 BTC to Bitcoin Address_
↳bclqvqatyv2xyntyarej2fcutj6w5yugy0gc9jx2nn? A fee of 0.0003 will be taken as_
↳withdraw fee. (yes/no) [no]:
```

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 ghcr.io/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 ghcr.io/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.

---

## Getting notified when Bitcoin DCA makes a purchase or does a withdrawal

---

If you want to get notifications from Bitcoin DCA when purchases and withdrawals are made you can configure the application to do so.

Protocol	Notifications
Email	Buy, Withdraw
Telegram	Buy, Withdraw, New Bitcoin DCA versions

### 8.1 Sending email from Bitcoin DCA

Sending an email is difficult; usually, internet providers limit home connections in connecting outside to prevent infected computers from sending out spam. For this reason, Bitcoin DCA provides multiple providers you can choose from for sending your email through various APIs.

Generally, each provider will ask you to sign up and verify at least your email address. Some provide more control and allow you to verify an entire domain. Low volumes of email are usually free.

Bitcoin DCA supports these email providers. Sending through HTTP is preferred because it has the least risk of getting blocked. However, if you prefer SMTP you can choose that as well.

The configuration parameter `NOTIFICATION_EMAIL_DSN` accepts input on how to send the email by providing a [DSN](#).

Provider	DSN for sending through SMTP	DSN for sending through HTTP
Amazon SES	ses+smtp://USERNAME:PASSWORD@default	ses+https://ACCESS_KEY:SECRET_KEY@default
Google Gmail	gmail+smtp://USERNAME:PASSWORD@default	
Mailchimp Man-drill	mandrill+smtp://USERNAME:PASSWORD@default	mandrill+https://KEY@default
Mailgun	mailgun+smtp://USERNAME:PASSWORD@default	mailgun+https://KEY:DOMAIN@default
Mailjet	mailjet+smtp://ACCESS_KEY:SECRET_KEY@default	
Postmark	postmark+smtp://ID@default	
Sendgrid	sendgrid+smtp://KEY@default	sendgrid+api://KEY@default
Sendinblue	sendinblue+smtp://USERNAME:PASSWORD@default	
<i>Your own SMTP server</i>	smtp://USERNAME:PASSWORD@HOSTNAME:PORT	

**Note:** The easiest service to use is probably Gmail as most people already own a gmail address. If you use 2FA you need to create an [app password](#) first.

Under the hood, Bitcoin DCA uses the Symfony Mailer. You can consult [their documentation](#) for more advanced topics such as alternative ports, high availability and load balancing.

For example, when using Gmail your configuration would look like this:

Listing 1: Example for sending email from Gmail

```
NOTIFICATION_EMAIL_ENABLED=1
NOTIFICATION_EMAIL_TO=satsstacker@gmail.com
NOTIFICATION_EMAIL_DSN=gmail+smtp://USERNAME:PASSWORD@default
```

## 8.2 Using Telegram with Bitcoin DCA

If you want to get notifications on Telegram from Bitcoin DCA you can follow these instructions:

1. Open a new chat with [Botfather](#);
2. Tell it you want to create a new bot with `/newbot`;
3. Pick a name. Be creative;
4. Pick a username for your new bot, it should end in `bot` and should be unique across the Telegram network. For example: `BobsTelegramDCABot`;
5. Note the secret token you just received. It looks like `12345:ABCDetc`;
6. Talk to your new bot by clicking the `t.me/BobsTelegramDCABot` link you received. This will tell Telegram the bot is allowed to talk to you;
7. Open a new chat with [GetMyID bot](#). This bot will tell you the ID of your Telegram account;
8. Note the ID you received.

Make sure you have talked to your new bot. It won't talk back, but this will let Telegram know it's allowed to send messages to you. Now, configure Bitcoin DCA with this information:

Listing 2: Example for connecting Bitcoin DCA to Telegram

```
NOTIFICATION_TELEGRAM_ENABLED=1
NOTIFICATION_TELEGRAM_DSN=telegram://BOTFATHERSECRET@default?
↳channel=YOURTELEGRAMACCOUNTID
# example: NOTIFICATION_TELEGRAM_DSN=telegram://12345:ABCDetc@default?channel=123456
```





---

## Frequently Asked Questions

---

### 9.1 I already have MyNode / Umbrel running, can I use this tool too?

Yes! MyNode and Umbrel are both based on Linux and have Docker pre-installed. You can use all features of Bitcoin DCA.

Things you should keep in mind: The default user doesn't have permission to run Docker by default. MyNode uses user `admin` and Umbrel uses `umbrel`.

---

**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.

### 9.2 How do I make sure the time is set up correctly?

You can check the current system time with this command:

```
$ date
Fri May 28 08:46:37 CEST 2021
```

In some cases, it is possible that the timezone is configured incorrectly. On MyNode, Umbrel or on any other Debian-based system you can update this as following:

```
$ sudo dpkg-reconfigure tzdata
```



## CHAPTER 10

---

### About this software

---

This self-hosted DCA (Dollar Cost Averaging) tool is built with flexibility in mind, allowing you to specify your 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 11

---

### Supported Exchanges

---

Ex-change	URL	XPUB withdraw supported	Currencies
BL3P	<a href="https://bl3p.eu/">https://bl3p.eu/</a>	Yes	EUR
Bit-vavo	<a href="https://bitvavo.com/">https://bitvavo.com/</a>	No *	EUR
Kraken	<a href="https://kraken.com/">https://kraken.com/</a>	No	USD EUR CAD JPY GBP CHF AUD
Bi-nance	<a href="https://binance.com/">https://binance.com/</a>	Yes	USDT BUSD EUR USDC USDT GBP AUD TRY BRL DAI TUSD RUB UAH PAX BIDR NGN IDRT VAI

---

**Note:** Due to regulatory changes in The Netherlands, Bitvavo currently requires you to provide proof of address ownership, thus (temporarily) disabling Bitcoin-DCA's XPUB feature.

---



## CHAPTER 12

---

Telegram / Support

---

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





## CHAPTER 13

---

### Contributing

---

Contributions are highly welcome! Feel free to submit issues and pull requests on <https://github.com/jorijn/bitcoin-dca>.