

# Instalação TypeBot docker

Link: <https://docs.typebot.io/self-hosting/deploy/docker>

em 06/06/2025

The easiest way to get started with Typebot is with [the official managed service in the Cloud](#). You'll have high availability, backups, security, and maintenance all managed for you by me, Baptiste, Typebot's founder. The cloud version can save a substantial amount of developer time and resources. For most sites this ends up being the best value option and the revenue goes to funding the maintenance and further development of Typebot. So you'll be supporting fair source software and getting a great service!

## Requirements

You need a server with Docker installed. If your server doesn't come with Docker pre-installed, you can follow [their docs](#) to install it.

## Installation

### 1. Download the compose file

On your server, download the latest `docker-compose.yml` and the starter `.env` file:

Copy

```
wget https://raw.githubusercontent.com/baptisteArno/typebot.io/latest/docker-compose.yml
wget https://raw.githubusercontent.com/baptisteArno/typebot.io/latest/.env.example -O .env
```

## 2. Add the required configuration

1. You'll first need a random 32-character secret key which will be used to encrypt sensitive data. Here is a simple way to generate one:

Copy

```
openssl rand -base64 24 | tr -d '\n' ; echo
```

2. Fill the `.env` file with your values.
3. Configure at least one authentication provider (Email, Google, GitHub, Facebook or GitLab). More info here: [Configuration](#).

By default the compose file will pull the latest stable Typebot images: `baptistearno/typebot-builder:latest` and `baptistearno/typebot-viewer:latest`. You can decide to replace `latest` with a specific version. You can find all the existing tags [here](#)

## 3. Start the server

Once you've added your configuration to the compose file, you're ready to start up the server:

Copy

```
docker-compose up -d
```

When you run this command, by default, it does the following:

- Create a database
- Run the migrations
- Start the builder on port 8080
- Start the viewer on port 8081
- All Typebot's data is stored in the `.typebot` folder in the current directory

You can now navigate to `http://typebot.domain.com:8080` and see the login screen. Login with the admin email to have access to a Team plan workspace automatically.

Typebot server itself does not perform SSL termination. It only runs on unencrypted HTTP. If you want to run on HTTPS you also need to set up a reverse proxy in front of the server. See below instructions.

# Update Typebot

Typebot is updated regularly, but it is up to you to apply these updates on your server. By virtue of using Docker, these updates are safe and easy to apply.

1. Pull the new images:

Copy

```
docker-compose pull typebot-builder
docker-compose pull typebot-viewer
```

Alternatively, you can pull specific versions:

Copy

```
docker-compose pull typebot-builder:3.4.2
docker-compose pull typebot-viewer:3.4.2
```

2. Stop the server:

Copy

```
docker-compose down
```

3. Start the server (with the new images):

Copy

```
docker-compose up -d
```

The self-hosted version is somewhat of a LTS, only getting the changes (~ once per month) after they have been battle tested on the cloud version. If you want features as soon as they are available, consider becoming a [cloud user](#).

## Optional extras

# Reverse proxy

By default, Typebot runs on unencrypted HTTP on ports 8080 for the builder and 8081 for the viewer. We recommend running it on HTTPS behind a reverse proxy of some sort. You may or may not already be running a reverse proxy on your host, let's look at both options:

## No existing reverse proxy

If your DNS is managed by a service that offers a proxy option with automatic SSL management, feel free to use that. For example, you could use Cloudflare as a reverse proxy in front of Typebot.

Alternatively, you can run your Caddy server as a reverse proxy. This way your SSL certificate will be stored on the host machine and managed by Let's Encrypt. The Caddy server will expose port 443, terminate SSL traffic and proxy the requests to your Typebot server.

Here is an example of a docker-compose file using Caddy as a reverse proxy:

Copy

```
services:
  caddy-gen:
    container_name: caddy-gen
    image: 'wemakeservices/caddy-gen:latest'
    restart: always
    volumes:
      - /var/run/docker.sock:/tmp/docker.sock:ro
      - {$PWD}/.typebot/caddy-certificates:/data/caddy
    ports:
      - '80:80'
      - '443:443'
    depends_on:
      - typebot-builder
      - typebot-viewer

typebot-builder:
  labels:
    virtual.host: 'typebot.domain.com' # change to your domain name
    virtual.port: '3000'
    virtual.tls-email: 'admin@example.com' # change to your email
```

```
typebot-viewer:
  labels:
    virtual.host: 'bot.domain.com' # change to your domain name
    virtual.port: '3000'
    virtual.tls-email: 'admin@example.com' # change to your email
    # Necessary to enable message streaming
    virtual.proxy.directives: |
      flush_interval -1
```

This config requires you to add the following DNS entry:

Copy

```
typebot IN A <server_ip>
bot IN A <server_ip>
```

You can merge this compose file with the first one. Make sure that `NEXTAUTH_URL` is set to `https://typebot.domain.com` and `NEXT_PUBLIC_VIEWER_URL` is set to `https://bot.domain.com`.

When running the compose file, it should automatically enable SSL on your server and you should be able to navigate to:

- `https://typebot.domain.com` for the builder
- `https://bot.domain.com` for the viewer

## Existing reverse proxy

If you're already running a reverse proxy, the most important things to note are:

1. Configure the virtual hosts to match the `NEXTAUTH_URL` and `NEXT_PUBLIC_VIEWER_URL` in your `docker-compose` configuration.
2. Proxy the traffic to `127.0.0.1:8080` or `{ip-address}:8080` and to `127.0.0.1:8081` or `{ip-address}:8081` if running on a remote machine

## SMTP

I highly recommend using an external SMTP service. There are tons of options out there, including [SendInBlue](#), [Mailgun](#) and [SendGrid](#). It will avoid severe headaches ☹️. Then, you will only need to add the required [SMTP configuration variables](#).

If, however, you don't want to, you can instantiate an SMTP server in the docker-compose file.

Copy

```
services:
  mail:
    image: bytemark/smtp
    restart: always
```

And add the following variables to your `.env` file:

Copy

```
SMTP_HOST=mail
NEXT_PUBLIC_SMTP_FROM=notifications@typebot.domain.com
```

You will probably need to make sure that `typebot.domain.com` has a valid SPF record and that your server IP has a rDNS set up.

You can merge this compose file with the main one.

## S3 storage

If you don't already have an S3 storage available, you could include it in your docker-compose file:

Copy

```
services:
  minio:
    image: minio/minio
    command: server /data
    ports:
      - '9000:9000'
    environment:
```

```

    MINIO_ROOT_USER: minio
    MINIO_ROOT_PASSWORD: minio123
volumes:
  - s3-data:/data
# This service just makes sure a bucket with the right policies is created
createbuckets:
  image: minio/mc
  depends_on:
    - minio
  entrypoint: >
    /bin/sh -c "
    sleep 10;
    /usr/bin/mc config host add minio http://minio:9000 minio minio123;
    /usr/bin/mc mb minio/typebot;
    /usr/bin/mc anonymous set public minio/typebot/public;
    exit 0;
    "
volumes:
  s3-data:

```

And add the following variables to your `.env` file:

Copy

```

S3_ACCESS_KEY=minio
S3_SECRET_KEY=minio123
S3_BUCKET=typebot
S3_ENDPOINT=storage.domain.com

```

This config requires you to add the following DNS entry:

Copy

```

storage IN A <server_ip>

```

You can merge this compose file with the main one.

# Config example with all the extras

Here is a config example that spins up Typebot with HTTPS, SMTP and S3 storage.

Copy

```
services:
  caddy-gen:
    image: 'wemakeservices/caddy-gen:latest'
    restart: always
    volumes:
      - /var/run/docker.sock:/tmp/docker.sock:ro
      - {$PWD}/.typebot/caddy-certificates:/data/caddy
    ports:
      - '80:80'
      - '443:443'
    depends_on:
      - typebot-builder
      - typebot-viewer
  typebot-db:
    image: postgres:16
    restart: always
    volumes:
      - db-data:/var/lib/postgresql/data
    environment:
      - POSTGRES_DB=typebot
      - POSTGRES_PASSWORD=typebot
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres"]
      interval: 5s
      timeout: 5s
      retries: 5
  typebot-builder:
    labels:
      virtual.host: 'typebot.domain.com' # change to your domain
      virtual.port: '3000'
      virtual.tls-email: 'admin@example.com' # change to your email
    image: baptistearno/typebot-builder:latest
    depends_on:
```



```
typebot-db:
  condition: service_healthy
restart: always
extra_hosts:
  - 'host.docker.internal:host-gateway'
# See https://docs.typebot.io/self-hosting/configuration for more configuration options
env_file:
  - .env

typebot-viewer:
  labels:
    virtual.host: 'bot.domain.com' # change to your domain
    virtual.port: '3000'
    virtual.tls-email: 'admin@example.com' # change to your email
    # Necessary to enable message streaming
    virtual.proxy.directives: |
      flush_interval -1
  image: baptistearno/typebot-viewer:latest
  depends_on:
    typebot-db:
      condition: service_healthy
  restart: always
  # See https://docs.typebot.io/self-hosting/configuration for more configuration options
  env_file:
    - .env

mail:
  image: bytemark/smtp
  restart: always

minio:
  labels:
    virtual.host: 'storage.domain.com' # change to your domain
    virtual.port: '9000'
    virtual.tls-email: 'admin@example.com' # change to your email
  image: minio/minio
  command: server /data
  ports:
    - '9000:9000'
  environment:
    MINIO_ROOT_USER: minio
    MINIO_ROOT_PASSWORD: minio123
  volumes:
```

```
- s3-data:/data

# This service just make sure a bucket with the right policies is created
createbuckets:
  image: minio/mc
  depends_on:
    - minio
  entrypoint: >
    /bin/sh -c "
    sleep 10;
    /usr/bin/mc config host add minio http://minio:9000 minio minio123;
    /usr/bin/mc mb minio/typebot;
    /usr/bin/mc anonymous set public minio/typebot/public;
    exit 0;
    "

volumes:
  db-data:
  s3-data:
```

# Build your own images

To build your own builder Docker image

Copy

```
docker build -t typebot-builder --build-arg SCOPE=builder .
```

To build your own viewer Docker image

Copy

```
docker build -t typebot-viewer --build-arg SCOPE=viewer .
```

# Troubleshooting

## Avoiding network issues when migrating in Portainer

If you migrate a Typebot stack between Portainer instances, hostname resolution may fail, causing `typebot-viewer` to be unable to connect to `typebot-db`.

Before deploying the stack in Portainer, ensure the network is **explicitly defined as attachable** in `docker-compose.yml`:

Copy

```
networks:
  typebot_network:
    driver: bridge
    attachable: true

services:
  typebot-db:
    ...
    networks:
      - typebot_network
  typebot-builder:
    ...
    networks:
      - typebot_network
  typebot-viewer:
    ...
    networks:
      - typebot_network
```

When deploying the stack, Portainer will automatically create the network with the correct settings. This prevents hostname resolution issues after migration.

If you're self-hosting Typebot, [sponsoring me](#) is a great way to give back to the community and to contribute to the long-term sustainability of the project. It also comes with some perks like priority support and private workshops. ♥

This doc has been inspired by [Plausible docs](#). They have a similar self-hosting solutions, and their documentation is [\[1\]](#).

---

Revision #2

Created 2024-09-13 10:55:47 UTC by Administrador

Updated 2025-06-06 19:41:05 UTC by Administrador