Calcom

Sistema de agendamentos e colaboração

Instalação e configurações Calcom

Procedimentos técnicos

Instalação do Calcom docker

Link: https://github.com/calcom/docker/tree/main git clone https://github.com/calcom/docker.git

Cal.com (formerly Calendso)

The open-source Calendly alternative. (Docker Edition)

Learn more »

Slack · Website · Core Cal.com related Issues · Docker specific Issues · Roadmap

Docker

This image can be found on DockerHub at https://hub.docker.com/r/calcom/cal.com

The Docker configuration for Cal.com is an effort powered by people within the community. Cal.com, Inc. does not yet provide official support for Docker, but we will accept fixes and documentation at this time. Use at your own risk.

Important Notes

This Docker Image is managed by the Cal.com Community. Join the team here. Support for this image can be found via the repository, located at https://github.com/calcom/docker

Currently, this image is intended for local development/evaluation use only, as there are specific requirements for providing environmental variables at build-time in order to specify a non-localhost BASE_URL. (this is due to the nature of the static site compilation, which embeds the variable values). The ability to update these variables at runtime is in-progress and will be available in the future.

For Production, for the time being, please checkout the repository and build/push your own image privately.

Requirements

Make sure you have `docker` & `docker compose` installed on the server / system. Both are installed by most docker utilities, including Docker Desktop and Rancher Desktop.

Note: `docker compose` without the hyphen is now the primary method of using docker-compose, per the Docker documentation.

(Most users) Running Cal.com with Docker Compose

If you are evaluating Cal.com or running with minimal to no modifications, this option is for you.

1. Clone calcom/docker

   ```
   git clone https://github.com/calcom/docker.git
   ```

2. Change into the directory

   ```
   cd docker
   ```

3. Prepare your configuration: Rename `.env.example` to `.env` and then update `.env`

   ```
   cp .env.example .env
   ```

   Most configurations can be left as-is, but for configuration options see Important Run-time variables below.
   Update the appropriate values in your .env file, then proceed.
4. (optional) Pre-Pull the images by running the following command:

   ```
   docker compose pull
   ```

   This will use the default image locations as specified by `image:` in the docker-compose.yaml file.
   Note: To aid with support, by default Scarf.sh is used as registry proxy for download metrics.
5. Start Cal.com via docker compose
   (Most basic users, and for First Run) To run the complete stack, which includes a local Postgres database, Cal.com web app, and Prisma Studio:

   ```
   docker compose up -d
   ```

   To run Cal.com web app and Prisma Studio against a remote database, ensure that DATABASE_URL is configured for an available database and run:

   ```
   docker compose up -d calcom studio
   ```

   To run only the Cal.com web app, ensure that DATABASE_URL is configured for an available database and run:

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

   Note: to run in attached mode for debugging, remove `-d` from your desired run command.

6. Open a browser to http://localhost:3000, or your defined NEXT_PUBLIC_WEBAPP_URL. The first time you run Cal.com, a setup wizard will initialize. Define your first user, and you're ready to go!

Updating Cal.com

1. Stop the Cal.com stack

```
docker compose down
```

2. Pull the latest changes

```
docker compose pull
```

3. Update env vars as necessary.
4. Re-start the Cal.com stack

```
docker compose up -d
```

(Advanced users) Build and Run Cal.com

1. Clone calcom/docker

```
git clone https://github.com/calcom/docker.git calcom-docker
```

2. Change into the directory

```
cd calcom-docker
```

3. Update the calcom submodule.

```
git submodule update --remote --init
```

Note: DO NOT use recursive submodule update, otherwise you will receive a git authentication error.

4. Rename `.env.example` to `.env` and then update `.env`

For configuration options see Build-time variables below. Update the appropriate values in your .env file, then proceed.

5. Build the Cal.com docker image:
Note: Due to application configuration requirements, an available database is currently required during the build process.
a) If hosting elsewhere, configure the `DATABASE_URL` in the .env file, and skip the next step
b) If a local or temporary database is required, start a local database via docker compose.

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

6. Build Cal.com via docker compose (DOCKER_BUILDKIT=0 must be provided to allow a network bridge to be used at build time. This requirement will be removed in the future)

```
DOCKER_BUILDKIT=0 docker compose build calcom
```

7. Start Cal.com via docker compose
   (Most basic users, and for First Run) To run the complete stack, which includes a local Postgres database, Cal.com web app, and Prisma Studio:

   ```
   docker compose up -d
   ```

   To run Cal.com web app and Prisma Studio against a remote database, ensure that DATABASE_URL is configured for an available database and run:

   ```
   docker compose up -d calcom studio
   ```

   To run only the Cal.com web app, ensure that DATABASE_URL is configured for an available database and run:

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

   Note: to run in attached mode for debugging, remove `-d` from your desired run command.
8. Open a browser to http://localhost:3000, or your defined NEXT_PUBLIC_WEBAPP_URL. The first time you run Cal.com, a setup wizard will initialize. Define your first user, and you're ready to go!

Configuration

Important Run-time variables

These variables must also be provided at runtime

Variable	Description	Required	Default
CALCOM_LICENSE_KEY	Enterprise License Key	optional	
NEXT_PUBLIC_WEBAPP_URL	Base URL of the site. NOTE: if this value differs from the value used at build-time, there will be a slight delay during container start (to update the statically built files).	optional	`http://localhost:3000`
NEXTAUTH_URL	Location of the auth server. By default, this is the Cal.com docker instance itself.	optional	`{NEXT_PUBLIC_WEBAPP_URL}/api/auth`
NEXTAUTH_SECRET	must match build variable	required	`secret`
CALENDSO_ENCRYPTION_KEY	must match build variable	required	`secret`

Variable	Description	Required	Default
DATABASE_URL	database url with credentials - if using a connection pooler, this setting should point there	required	`postgresql://unicorn_user:magical_password@database:5432/calendso`
DATABASE_DIRECT_URL	direct database url with credentials if using a connection pooler (e.g. PgBouncer, Prisma Accelerate, etc.)	optional	

Build-time variables

If building the image yourself, these variables must be provided at the time of the docker build, and can be provided by updating the .env file. Currently, if you require changes to these variables, you must follow the instructions to build and publish your own image.

Updating these variables is not required for evaluation, but is required for running in production.

Instructions for generating variables can be found in the cal.com instructions

Variable	Description	Required	Default
NEXT_PUBLIC_WEBAPP_URL	Base URL injected into static files	optional	`http://localhost:3000`
NEXT_PUBLIC_LICENSE_CONSENT	license consent - true/false		
CALCOM_TELEMETRY_DISABLED	Allow cal.com to collect anonymous usage data (set to `1` to disable)		
DATABASE_URL	database url with credentials - if using a connection pooler, this setting should point there	required	`postgresql://unicorn_user:magical_password@database:5432/calendso`
DATABASE_DIRECT_URL	direct database url with credentials if using a connection pooler (e.g. PgBouncer, Prisma Accelerate, etc.)	optional	
NEXTAUTH_SECRET	Cookie encryption key	required	`secret`
CALENDSO_ENCRYPTION_KEY	Authentication encryption key	required	`secret`

Git Submodules

This repository uses a git submodule.

For users building their own images, to update the calcom submodule, use the following command:

```
git submodule update --remote --init
```

For more advanced usage, please refer to the git documentation: https://git-scm.com/book/en/v2/Git-Tools-Submodules

Troubleshooting

SSL edge termination

If running behind a load balancer which handles SSL certificates, you will need to add the environmental variable `NODE_TLS_REJECT_UNAUTHORIZED=0` to prevent requests from being rejected. Only do this if you know what you are doing and trust the services/load-balancers directing traffic to your service.

Failed to commit changes: Invalid 'prisma.user.create()'

Certain versions may have trouble creating a user if the field `metadata` is empty. Using an empty json object `{}` as the field value should resolve this issue. Also, the `id` field will autoincrement, so you may also try leaving the value of `id` as empty.

CLIENT_FETCH_ERROR

If you experience this error, it may be the way the default Auth callback in the server is using the WEBAPP_URL as a base url. The container does not necessarily have access to the same DNS as your local machine, and therefor needs to be configured to resolve to itself. You may be able to correct this by configuring `NEXTAUTH_URL=http://localhost:3000/api/auth`, to help the backend loop back to itself.

```
docker-calcom-1  | @calcom/web:start: [next-auth][error][CLIENT_FETCH_ERROR]
docker-calcom-1  | @calcom/web:start: https://next-auth.js.org/errors#client_fetch_error request to http://testing.localhost:3000/api/auth/session failed, reason: getaddrinfo ENOTFOUND testing.localhost {
docker-calcom-1  | @calcom/web:start:   error: {
docker-calcom-1  | @calcom/web:start:     message: 'request to http://testing.localhost:3000/api/auth/session failed, reason: getaddrinfo ENOTFOUND testing.localhost',
docker-calcom-1  | @calcom/web:start:     stack: 'FetchError: request to http://testing.localhost:3000/api/auth/session failed, reason: getaddrinfo ENOTFOUND testing.localhost\n' +
docker-calcom-1  | @calcom/web:start:       '    at ClientRequest.<anonymous> (/calcom/node_modules/next/dist/compiled/node-fetch/index.js:1:65756)\n' +
docker-calcom-1  | @calcom/web:start:       '    at ClientRequest.emit (node:events:513:28)\n' +
docker-calcom-1  | @calcom/web:start:       '    at ClientRequest.emit (node:domain:489:12)\n' +
docker-calcom-1  | @calcom/web:start:       '    at Socket.socketErrorListener (node:_http_client:494:9)\n' +
docker-calcom-1  | @calcom/web:start:       '    at Socket.emit (node:events:513:28)\n' +
docker-calcom-1  | @calcom/web:start:       '    at Socket.emit (node:domain:489:12)\n' +
docker-calcom-1  | @calcom/web:start:       '    at emitErrorNT (node:internal/streams/destroy:157:8)\n' +
```

```
docker-calcom-1  | @calcom/web:start:       '    at emitErrorCloseNT (node:internal/streams/destroy:122:3)\n' +
docker-calcom-1  | @calcom/web:start:       '    at processTicksAndRejections
(node:internal/process/task_queues:83:21)',
docker-calcom-1  | @calcom/web:start:     name: 'FetchError'
docker-calcom-1  | @calcom/web:start:   },
docker-calcom-1  | @calcom/web:start:   url: 'http://testing.localhost:3000/api/auth/session',
docker-calcom-1  | @calcom/web:start:   message: 'request to http://testing.localhost:3000/api/auth/session
failed, reason: getaddrinfo ENOTFOUND testing.localhost'
docker-calcom-1  | @calcom/web:start: }
```

Instalação Calcom docs

Link: https://cal.com/docs/introduction/quick-start/self-hosting/docker#requirements

Docker

The Docker configuration for Cal is an effort powered by people within the community. Cal.com, Inc. does not provide official support for Docker, but we will accept fixes and documentation. Use at your own risk.

If you want to contribute to the Docker repository, reply here.

The Docker configuration can be found in our docker repository.

Requirements

Make sure you have `docker` & `docker compose` installed on the server / system.

Note: `docker compose` without the hyphen is now the primary method of using docker-compose, per the Docker documentation.

Getting Started

1. Clone calcom-docker

```
git clone --recursive https://github.com/calcom/docker.git calcom-docker
```

2. Change into the directory

```
cd calcom-docker
```

3. Update the calcom submodule

```
git submodule update --remote --init
```

4. Rename `.env.example` to `.env` and then update `.env`

5. Build and start Cal.com via docker compose

```
docker compose up --build
```

6. (First Run) Open a browser to http://localhost:5555 to look at or modify the database content.
   a. Click on the `User` model to add a new user record.
   b. Fill out the fields (remembering to encrypt your password with BCrypt) and click `Save 1 Record` to create your first user.
7. Open a browser to http://localhost:3000 and login with your just created, first user.

Configuration

Build-time variables

These variables must be provided at the time of the docker build, and can be provided by updating the .env file. Changing these is not required for evaluation, but may be required for running in production. Currently, if you require changes to these variables, you must follow the instructions to build and publish your own image.

- NEXT_PUBLIC_WEBAPP_URL
- NEXT_PUBLIC_LICENSE_CONSENT
- NEXT_PUBLIC_TELEMETRY_KEY

Important Run-time variables

- NEXTAUTH_SECRET

Git Submodules

This repository uses a git submodule.

To update the calcom submodule, use the following command:

```
git submodule update --remote --init
```

For more advanced usage, please refer to the git documentation: https://git-scm.com/book/en/v2/Git-Tools-Submodules

Troubleshooting

- SSL edge termination: If running behind a load balancer which handles SSL certificates, you will need to add the environmental variable `NODE_TLS_REJECT_UNAUTHORIZED=0` to prevent requests from being rejected. Only do this if you know what you are doing and trust the services/load-balancers directing traffic to your service.
- Failed to commit changes: Invalid 'prisma.user.create()': Certain versions may have trouble creating a user if the field `metadata` is empty. Using an empty json object `{}` as the field value should resolve this issue. Also, the `id` field will autoincrement, so you may also try leaving the value of `id` as empty.