

Self-hosting Dify: the Community Edition on your own computer or server



Appendix to the workshop AI Agents

LeX Consultancy B.V. · 20 September 2026

In the workshop and in every guide we use the online version at cloud.dify.ai: sign up, build, done. Dify is also open source. You can run the same software yourself: on your own laptop to practise, or on a school server so that data never leaves the organisation. This appendix describes how to do that on Windows, Mac and Linux.

The steps are written for someone who has opened a terminal before but is not a system administrator. Allow half an hour the first time.

When to, when not to. For the workshop and for learning to build, the online version is the better choice: nothing to install, everything works straight away. Self-hosting becomes interesting when you want to work with real student data, when the school does not want data outside its own region, or when more than a handful of people will use Dify as a matter of routine. Note that self-hosting does not remove your obligations. The language model (OpenAI, Anthropic, Mistral or a local model) is still where your text is sent; that is where a data processing agreement has to be in place, not only for Dify.

What you get

The Community Edition is the same Dify as online, with the same Studio, workflows, chatflows, knowledge bases and tools. A few differences:

	Online (cloud.dify.ai)	Self-hosted (Community Edition)
Cost	Sandbox free with credits; subscription after that	Free software; you pay for your own model usage and your own computer or server
Language model	OpenAI by default, paid with Dify credits	You connect a provider yourself: OpenAI, Anthropic, Mistral, Azure, or a local model via Ollama
Data	With Dify (cloud) and with the model provider	On your computer or server, and with the model provider
Sharing	The web app link works everywhere	The web app link only works where your computer or server can be reached
Updates	Automatic	Your job (see “Updating”)
Agent Console	Available (beta)	Follows the releases; may lag behind the cloud

Every guide in this series also works on your own installation. The only difference is the address in the browser: `http://localhost` instead of `cloud.dify.ai`.

What you need

- A computer with at least **2 CPU cores and 4 GB of memory** free for Dify. In practice 8 GB in total is comfortable, 16 GB is relaxed.
- About **10 GB of free disk space**.
- **Docker**: the program that runs Dify and its parts (database, search index, web server) in separate “containers”. Dify consists of about fifteen of them; Docker starts them all in one go.
- **Git** to fetch the Dify files. Optional: you can download a zip instead.
- An **API key from a model provider** (OpenAI, for example), or Ollama for a local model.

The installation runs on Windows 10/11 (through WSL 2), macOS 10.14 or later (Intel and Apple Silicon) and any common Linux.

Step 1. Install Docker

Windows

1. Go to docker.com and download **Docker Desktop for Windows**.
2. Run the installer. Leave **Use WSL 2** ticked; that is the Linux layer the containers run in. Windows installs WSL 2 itself if it is missing; a restart is sometimes needed.
3. Start Docker Desktop and wait until the bottom left says “Engine running”.

4. Open **Settings** → **Resources** and give Docker at least 8 GB of memory if your computer has it.

Keep the Dify files in the Linux environment later on (for example in `\\wsl$\\Ubuntu\\home\\yourname`), not on `C:\\Users\\...`. Dify runs noticeably slower from a Windows folder and sometimes gives permission errors.

Mac

1. Go to docker.com and download **Docker Desktop for Mac**. Pick the version for your processor (Apple Silicon or Intel; see → About This Mac).
2. Drag Docker to Applications and start it. Grant the permissions it asks for.
3. Open **Settings** → **Resources** and set memory to at least 8 GB and CPUs to at least 2.

Linux

On Ubuntu or Debian, in a terminal:

TYPE IN

```
sudo apt update
sudo apt install docker.io docker-compose-v2 git curl
sudo usermod -aG docker $USER
```

Log out and back in so the last line takes effect (you can then use Docker without sudo). For Fedora, Arch and other distributions see docs.docker.com/engine/install. Dify needs Docker 19.03 or later and Docker Compose 2.24 or later. Check:

TYPE IN

```
docker --version
docker compose version
```

Step 2. Get Dify

Open a terminal. On Windows that is the **Ubuntu** app (WSL) or PowerShell; on Mac **Terminal**; on Linux your usual terminal. Pick a folder where Dify may live and go there.

The developers recommend fetching the latest **released version**, not the development branch:

TYPE IN

```
git clone --branch "$(curl -s
https://api.github.com/repos/langgenius/dify/releases/latest | jq -r
.tag_name)" https://github.com/langgenius/dify.git
```

That line first asks GitHub for the number of the latest release and fetches exactly that one. It needs `git`, `curl` and `jq`. If it fails (because `jq` is missing, for instance), do it in two steps:

look at github.com/langgenius/dify/releases for the version at the top and use that (on 20 September 2026 it was 1.17.1):

TYPE IN

```
git clone --branch 1.17.1 https://github.com/langgenius/dify.git
```

Without git: on the same page, under **Assets**, download the zip of the latest release and unpack it.

Then go to the folder with the Docker files:

TYPE IN

```
cd dify/docker
```

Step 3. Prepare the settings

Dify reads its settings from a file called `.env`. An example is provided; copy it:

TYPE IN

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

On Windows in PowerShell: `copy .env.example .env`.

For a first installation on your own computer you do not need to change anything. A few lines you will want to know about later:

Line in <code>.env</code>	What it does
<code>EXPOSE_NGINX_PORT=80</code>	The port Dify is reachable on. If 80 is already in use (by another web server, say), set <code>8080</code> ; the address then becomes <code>http://localhost:8080</code> .
<code>SECRET_KEY=</code>	The key Dify uses to protect passwords and sessions. On a server: put a long random string here before the first start.
<code>CONSOLE_WEB_URL=</code> and <code>APP_WEB_URL=</code>	The public address of your installation. Empty is fine on your own computer; on a server put the address your colleagues will use.
<code>UPLOAD_FILE_SIZE_LIMIT=15</code>	Maximum file size in MB for uploads to a knowledge base. For the transcription tool (guide 05) with recordings of tens of MB, raise this.

The `.env` file will contain sensitive data. Never put it in git or on a shared drive.

Step 4. Start

TYPE IN

```
docker compose up -d
```

The first time, Docker downloads all the parts; that takes a few minutes depending on your connection. `-d` means “in the background”: your terminal comes back and Dify keeps running.

Check that everything is up:

TYPE IN

```
docker compose ps
```

Every line should say **Up** or **healthy**. A container that keeps saying **Restarting** is usually short of memory: give Docker more memory (step 1) and start again.

Step 5. Create the admin account

Open in your browser:

TYPE IN

```
http://localhost/install
```

(With another port: `http://localhost:8080/install`. On a server: `http://<server-address>/install`.)

Here you create the first account: that is the administrator of the workspace. After that you land on `http://localhost`, in the same Studio as online.

This install page works once. After that you simply log in at `http://localhost`.

Step 6. Connect a language model

Unlike the online version, your own Dify has no credits and no default model. You connect a provider yourself.

1. Click your account name at the top right and choose **Settings → Model Provider**.
2. Pick a provider (OpenAI, Anthropic, Mistral, Azure OpenAI, Google) and click **Install** if the plugin is not there yet.
3. Click **Setup** and paste your API key. Dify checks the key immediately.
4. Under **System Model Settings**, choose which model new apps use by default.

From here on, every guide in this series can be followed literally. Where a guide picks **gpt-5-mini**, pick the same model with your provider.

Your choice: a local model. If you want nothing to leave the building at all, you can run a model on your own computer with **Ollama** (ollama.com). Install Ollama, pull a model (`ollama pull gemma3:12b` or another one that fits your machine) and add the provider **Ollama** in Dify with the address `http://host.docker.internal:11434`. Expect to need at least 16 GB of memory and some patience: quality and speed are below those of the big providers, but for practising and for sensitive texts it can be enough.

Stopping, starting, updating

Stop (data is kept):

TYPE IN

```
docker compose down
```

Start again:

TYPE IN

```
docker compose up -d
```

Updating to a new version. First read the release notes at github.com/langgenius/dify/releases: sometimes there is an extra step. Then, in the `dify/docker` folder:

TYPE IN

```
docker compose down
git fetch --tags
git checkout <new version number>
docker compose pull
docker compose up -d
```

After an update, compare `.env.example` with your own `.env`: new settings only appear in the example file. If you changed `docker-compose.yml` yourself, reapply those changes.

Remove everything, including all apps, knowledge bases and users:

TYPE IN

```
docker compose down -v
```

The `-v` also deletes the storage. Without `-v`, everything stays.

Where your data lives

Everything you make in Dify (apps, knowledge bases, conversations, uploads) is in the folder `dify/docker/volumes`. A backup is a copy of that folder while Dify is stopped. Apps can also be exported one by one as a DSL file (in Studio: the three dots on an app → **Export DSL**). Such a file can be imported into any other Dify, the online one included. That is also how you move an app you built at home to the school installation.

On a server for the school

On your own laptop the above just works. For use by colleagues there is an extra layer, and that is work for IT:

- A server (own hardware or a virtual machine at a provider in your region) with 4 cores, 8 to 16 GB of memory and a fixed name such as `dify.school.org`.
- **HTTPS**: a certificate and a reverse proxy (Nginx, Caddy, Traefik) in front. In `.env`, set `CONSOLE_WEB_URL` and `APP_WEB_URL` to `https://dify.school.org`.
- **Signing in with school accounts** (Microsoft Entra or Google Workspace) via the SSO options in `.env`; without SSO the administrator creates the accounts.
- **Backups** of `volumes`, daily, and one test restore.
- **A data processing agreement** with the model provider, and a decision about which data may and may not go in. Self-hosting does not make that conversation unnecessary; it does make it simpler, because there is one party fewer in the chain.

The full documentation for production use is at docs.dify.ai under **Self-Host**.

If it does not work

What you see	What is going on
<code>docker: command not found</code>	Docker is not installed, or (Linux) you have not logged in again after <code>usermod</code>
<code>port is already allocated</code>	Port 80 is taken; set <code>EXPOSE_NGINX_PORT=8080</code> in <code>.env</code> and start again
Page keeps loading, <code>docker compose ps</code> says Restarting	Not enough memory for Docker; raise it in Docker Desktop or close other programs
<code>http://localhost/install</code> says "already installed"	There is an account already; go to <code>http://localhost</code> and log in
Model key is rejected	Key pasted wrongly, or the provider has no credit on that key
Knowledge base does not index	The <code>weaviate</code> container (or the vector database you chose) is not running; look with <code>docker compose logs weaviate</code>
Very slow on Windows	The files are on <code>C:\</code> ; move the <code>dify</code> folder to the Linux side of WSL

Viewing the log of one part, for example the API:

TYPE IN

```
docker compose logs -f api
```

Licence: CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>). You may copy, adapt and reuse this document, including at your own school, provided you credit the source (Guido van Dijk, LeX Consultancy B.V., <https://git.lexconsultancy.nl/guido/dify-guides>) and share your adaptation under the same licence. The LeX Consultancy logo is excluded; product names are trademarks of their owners. Schools, people and documents in the examples are fictional.

Owner and contact: Guido van Dijk, LeX Consultancy B.V. · guido@l-e-x.nl



LeX Consultancy B.V. · l-e-x.nl · Self-hosting Dify: the Community Edition on your own computer or server