

ローカル環境で Dify を実行する（ただし 16GB～24GB の GPU が必要）
(2025/11)

0. 具体的な方法は書籍を購入して参考にするのがよい。ローカル環境での構築が記載されているものを下記に示す。

技術評論社：生成 AI アプリ開発大全－Dify の探求と実践活用（最も詳細でお勧め）

マイナビ：Dify ではじめる 業務効率化 AI アプリ開発 AI を会社に根付かせる実践ガイド

技術評論社：ゼロからわかる Dify の教科書 ~生成 AI×ノーコードでかんたん業務効率化

その他、ネットで検索したり生成 AI に聞いたりすると、手順を示してくれる

以下の記述は基本的な流れになる

モデルの登録がクラウド版ほど手軽ではなく 4.で解説するが、使い方はクラウド版と同じ

1. Docker Desktop をインストールする

<https://www.docker.com/get-started/>

Docker ディスクトップダウンロードから Windows 用ダウンロード ARM64

ダウンロードフォルダに「Docker Desktop Installer.exe」保存されるので、このファイルをクリックする

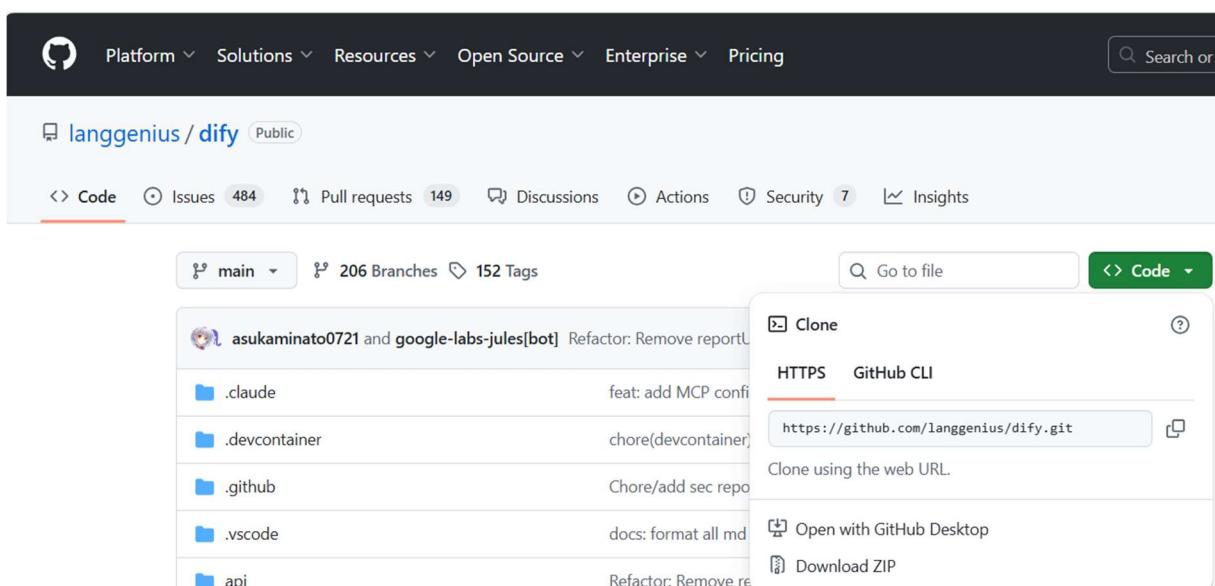
ディスクトップにアイコンができるので、これをクリックして実行する

2. Dify を Docker 上でインストールする

ソースコード (python) を取得する

<https://github.com/langgenius/dify.git>

①「Code」にクリックし、②「Download ZIP」をクリックする



ダウンロードフォルダに「dify-main.zip」が保存されるので、このファイルを展開して、適当なフォルダにおく

フォルダ名を dify-main から dify に変更する

3. ローカル型で動作する言語モデルを使う場合はこれをダウンロードし ollama で使えるように

し Dify と連携させる

- Ollama のインストール

<https://ollama.com/download>

Download for Windows をクリックして「OllamaSetup.exe」ファイルをダウンロードして実行する

ollama で使えるモデル一覧が下記の URL で表示される

<https://ollama.com/library>

ここに記載さいれている大規模言語モデル (LLM) は、pull でダウンロードできる

ollama pull schroneko/gemma-2-2b-jpn-it

記載されていない elyza8b モデルはネットで探してダウンロードする

自分でダウンロードしたモデルを使えるようにするには、Modelfile.txt を作って実行する

ollama create elyza:jp8b -f Modelfile.txt

Modelfile.txt の中身

```
FROM ./Llama-3-ELYZA-JP-8B-q4_k_m.gguf
TEMPLATE """{{ if .System }}<|start_header_id|>system<|end_header_id|>
{{ .System }}<|eot_id|>{{ end }}{{ if .Prompt }}<|start_header_id|>user<|end_header_id|>
{{ .Prompt }}<|eot_id|>{{ end }}<|start_header_id|>assistant<|end_header_id|>
{{ .Response }}<|eot_id|>"""
PARAMETER stop "<|start_header_id|>"
PARAMETER stop "<|end_header_id|>"
PARAMETER stop "<|eot_id|>"
PARAMETER stop "<|reserved_special_token|>"
```

エンベディングモデル（関数）のダウンロード 文章をベクトル化する関数

ollama pull mxbai-embed-large

同じように登録する

下記のコマンドで登録されたモデルを確認できる

ollama list

schroneko/gemma-2-2b-jpn-it:latest	fcfc848fe62a	2.8 GB	44 seconds ago
elyza:jp8b	5e3ed7a1e201	4.9 GB	7 weeks ago

4. Dify を起動し、チャットボットを作り、モデルを登録する

登録方法はローカル型特有の方法を使うが、チャットボットの作り方はクラウド型と同じ
コマンドプロンプトで

cd dify/docker

docker compose up -d

インストールが終了したら次のコマンドで動作を確認する

docker compose ps

<http://localhost/install> で実行

- ・モデルの登録

クラウド型の資料を参考に、先ずは ollama をインストールする

The screenshot shows the LangChain application interface. On the left, there's a sidebar with 'Setting' (設定) and several other options like 'Workspace' (ワークスペース), 'Model Provider' (モデルプロバイダー), 'Members' (メンバー), 'Request' (請求), 'DataSource' (データソース), 'API Extension' (API拡張), and 'Customize' (カスタマイズ). The main area is titled 'Model Provider' (モデルプロバイダー) and lists various AI models. The 'Ollama' model, which has 346,415 users, is highlighted with a red border. Other models listed include 'Azure OpenAI' (47,922 users), 'Azure AI Studio' (8,537 users), 'Cohere' (14,806 users), 'Gemini' (122,356 users), 'Hugging Face Hub' (25,555 users), 'Replicate' (2,100 users), and 'DeepSeek' (222,689 users). Each model entry includes its name, provider (langgenius), user count, and a brief description.

大規模言語モデル（LLM）の追加は下記のように設定する

文章をベクトル化するエンベディングモデルは、下記画面で Model Type : Text Embedding を選択

セットアップ Ollama

Model Type *

LLM Text Embedding

Model Name *

elyzajp8b

Base URL *

http://host.docker.internal:11434

Completion mode *

Chat

Model context size *

4096

Upper bound for max tokens *

4096

Vision support

Yes No

Function call support

Yes No

How to integrate with Ollama ↗

削除 キャンセル 保存

How to integrate with Ollama ↗

削除 キャンセル 保存

択し、Model Name : mxbai-embed-large とする