Assignment 6. Conversational Q&A using AzureOpenAI

Title: - Conversational Q&A System using Azure OpenAI and DockerHub

Objective

The objective of this assignment is to design and implement a Conversational Q&A system. Participants will gain hands-on experience with Azure OpenAI for embeddings, LangChain for chunking and orchestration, and FAISS for vector-based retrieval. They will also learn to build an interactive Streamlit UI, deploy it on Azure App Service, and finally dockerize the application for publishing on DockerHub.

This exercise simulates a real-world scenario where organizations use domain-specific datasets and large language models to build intelligent, searchable, and conversational applications.

Conversational Q & A Fitness Document

The chosen dataset is a comprehensive collection of specifications of Indian fitness plans and exercises across various brands. It includes attributes such as:

- Current Weight (kg)
- BMI
- Health goal (e.g., weight loss, muscle gain, maintenance)
- Health condition (e.g., diabetes, heart issues)
- Activity preference (e.g., yoga, cardio, strength training)
- Other relevant health details

This dataset is ideal for a conversational Q&A system because it contains structured attributes that naturally align with user queries.
Examples of queries:

- "Show me fitness plans and exercises diet plan for weight loss with diabetes-friendly meals."

  This dataset represents a structured fitness and wellness knowledge base, which makes it useful for building a virtual fitness plan or exercise consultant.

Real-world applications include:

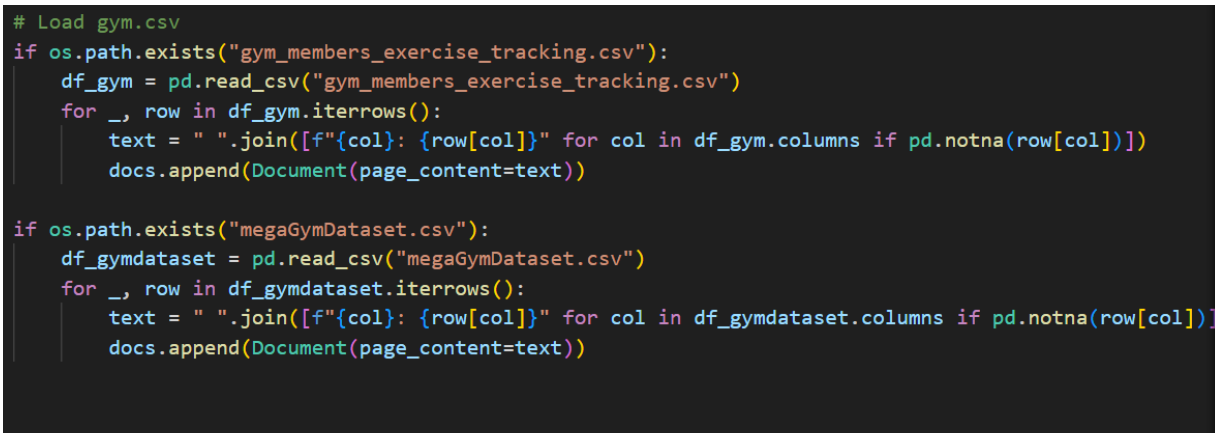
- Online fitness plan or exercise recommendation chatbots for gyms or fitness centers
- AI-powered comparison tools for fitness and wellness websites
- Customer service assistants for nutritionists or fitness trainers
- Decision-support tools for prospective users

Assignment Tasks

Task 1: Dataset Selection & Description

For this project, I selected an Indian fitness plans and exercises dataset containing attributes like Age, Gender,Weight (kg),Height (m),Max_BPM,Avg_BPM,Resting_BPM,Session_Duration (hours).

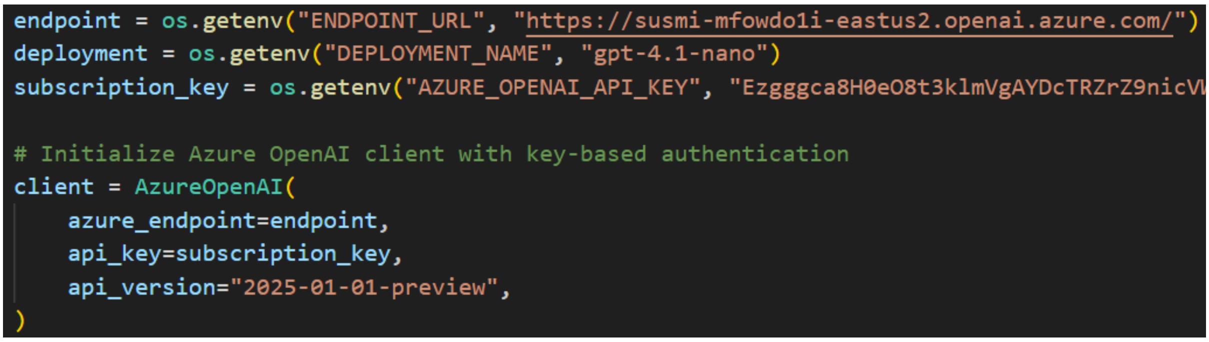
This dataset is well-suited for a Q&A system because it allows users to search for fitness plans and exercises based on multiple attributes simultaneously and supports intelligent recommendations. Such a system can help prospective users quickly find the best fitness plan or exercise that matches their preferences, reducing the need for manual filtering on fitness plan or exercise websites.



```
# Load gym.csv
if os.path.exists("gym_members_exercise_tracking.csv"):
    df_gym = pd.read_csv("gym_members_exercise_tracking.csv")
    for _, row in df_gym.iterrows():
        text = " ".join([f"{col}: {row[col]}" for col in df_gym.columns if pd.notna(row[col])])
        docs.append(Document(page_content=text))

if os.path.exists("megaGymDataset.csv"):
    df_gymdataset = pd.read_csv("megaGymDataset.csv")
    for _, row in df_gymdataset.iterrows():
        text = " ".join([f"{col}: {row[col]}" for col in df_gymdataset.columns if pd.notna(row[col])]
        docs.append(Document(page_content=text))
```

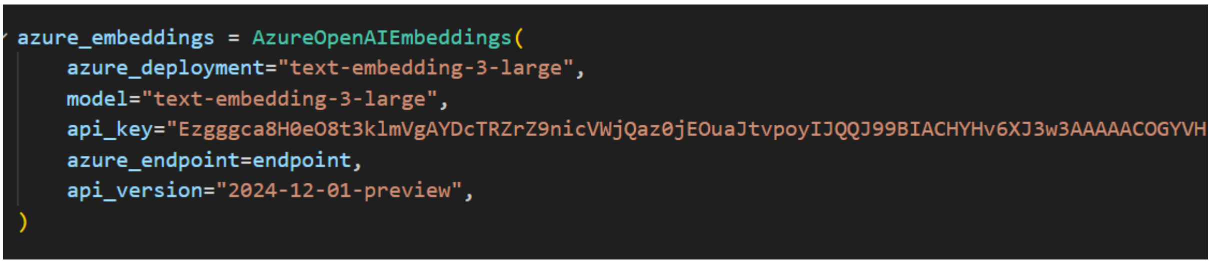
Task 2: Data Chunking & Embedding Generation

- Used **LangChain** to convert each row of the dataset into a text-based document.
- Generated embeddings using **Azure OpenAI's `text-embedding-3-small`** model.
- Stored embeddings locally to enable fast semantic search during queries.



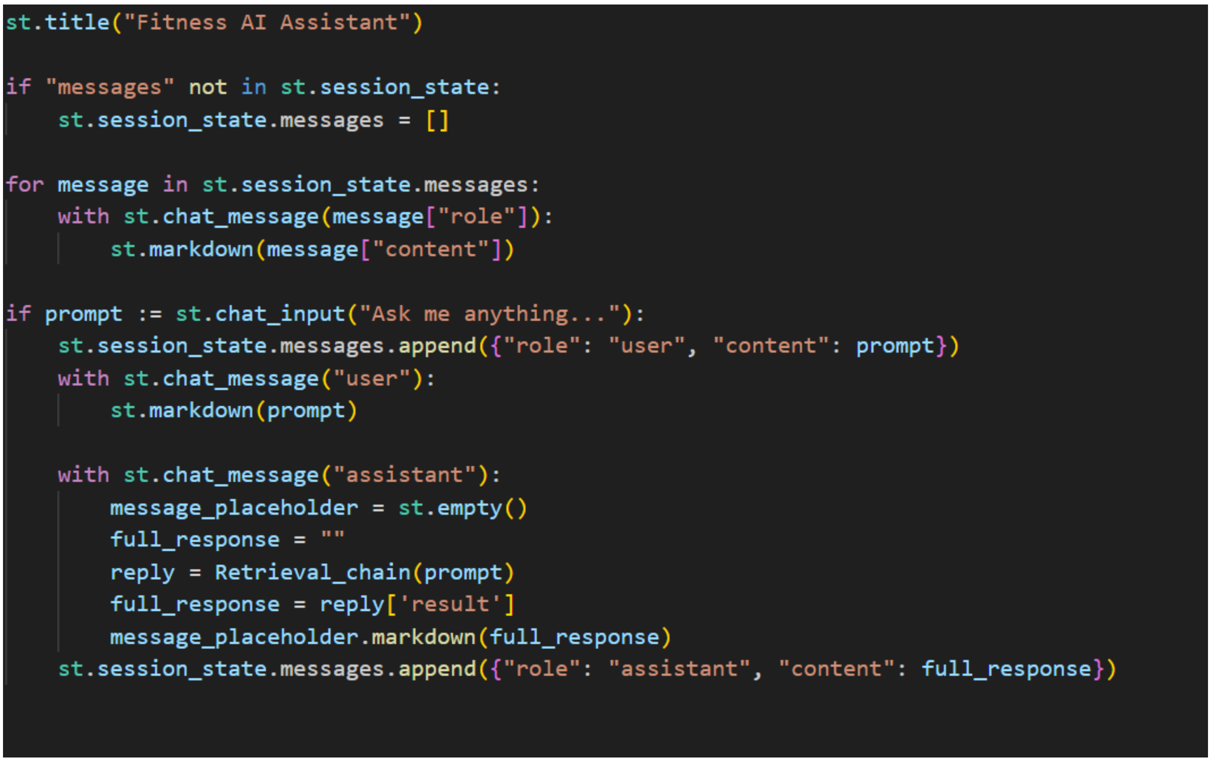
```
endpoint = os.getenv("ENDPOINT_URL", "https://susmi-mfowdo1i-eastus2.openai.azure.com/")
deployment = os.getenv("DEPLOYMENT_NAME", "gpt-4.1-nano")
subscription_key = os.getenv("AZURE_OPENAI_API_KEY", "Ezgggca8H0eO8t3klmVgAYDcTRZrZ9nicV

# Initialize Azure OpenAI client with key-based authentication
client = AzureOpenAI(
    azure_endpoint=endpoint,
    api_key=subscription_key,
    api_version="2025-01-01-preview",
)
```

Task 3: Build FAISS Index

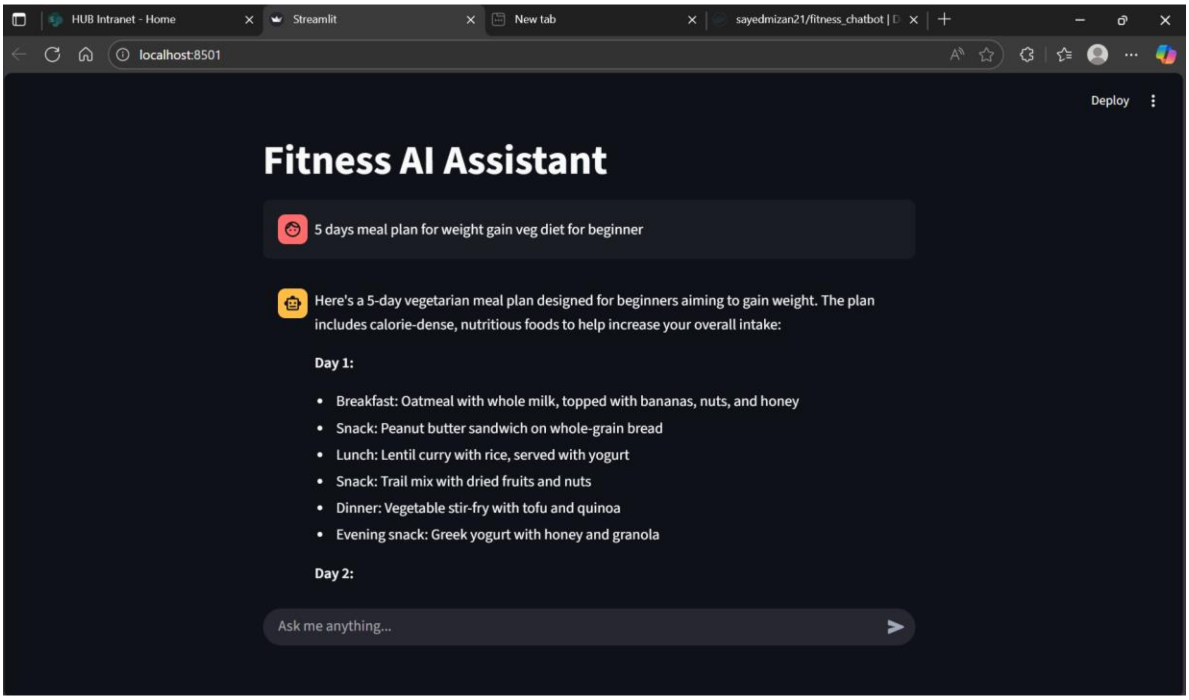


```
azure_embeddings = AzureOpenAIEmbeddings(
    azure_deployment="text-embedding-3-large",
    model="text-embedding-3-large",
    api_key="Ezggca8H0eO8t3klmVgAYDcTRZrZ9nicVWjQaz0jEOuaJtvpoyIJQQJ99BIACHYHv6XJ3w3AAAAACOGYVH
    azure_endpoint=endpoint,
    api_version="2024-12-01-preview",
)
```

Task 4: Create Streamlit UI



```
st.title("Fitness AI Assistant")

if "messages" not in st.session_state:
    st.session_state.messages = []

for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

if prompt := st.chat_input("Ask me anything..."):
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):
        st.markdown(prompt)

    with st.chat_message("assistant"):
        message_placeholder = st.empty()
        full_response = ""
        reply = Retrieval_chain(prompt)
        full_response = reply['result']
        message_placeholder.markdown(full_response)
    st.session_state.messages.append({"role": "assistant", "content": full_response})
```



Fitness AI Assistant

5 days meal plan for weight gain veg diet for beginner

Here's a 5-day vegetarian meal plan designed for beginners aiming to gain weight. The plan includes calorie-dense, nutritious foods to help increase your overall intake:

Day 1:

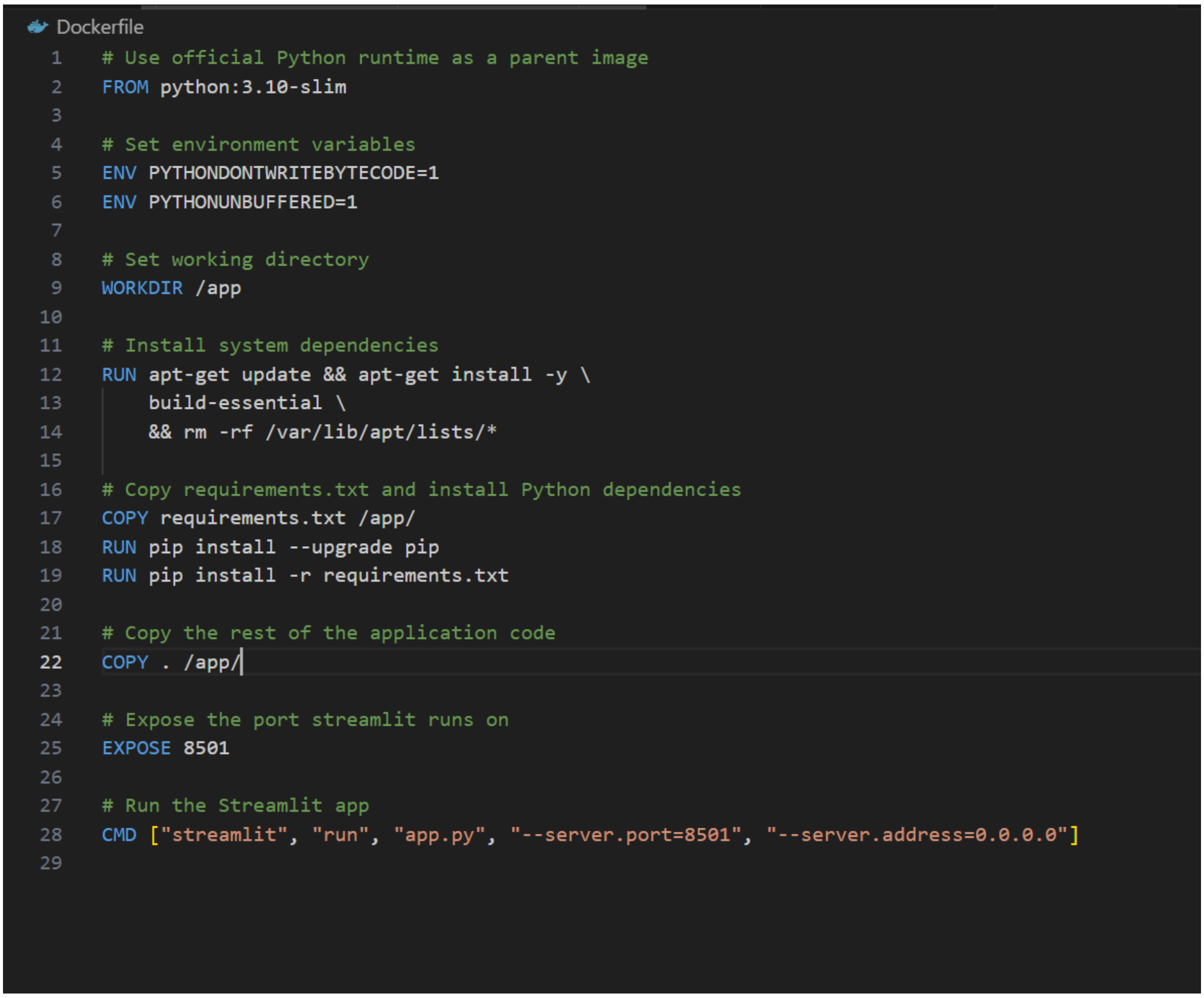
- Breakfast: Oatmeal with whole milk, topped with bananas, nuts, and honey
- Snack: Peanut butter sandwich on whole-grain bread
- Lunch: Lentil curry with rice, served with yogurt
- Snack: Trail mix with dried fruits and nuts
- Dinner: Vegetable stir-fry with tofu and quinoa
- Evening snack: Greek yogurt with honey and granola

Day 2:

Ask me anything...

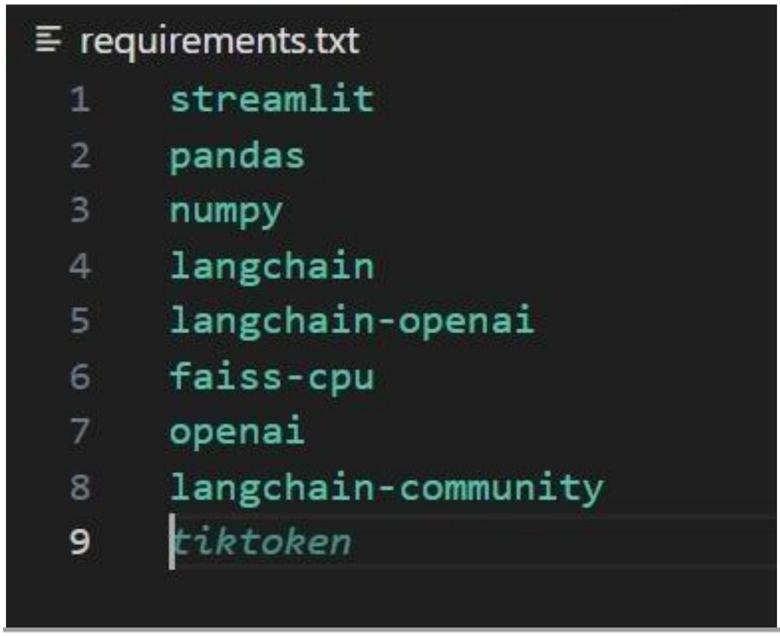
Task 5: Dockerization & Publishing on DockerHub

DockerFile



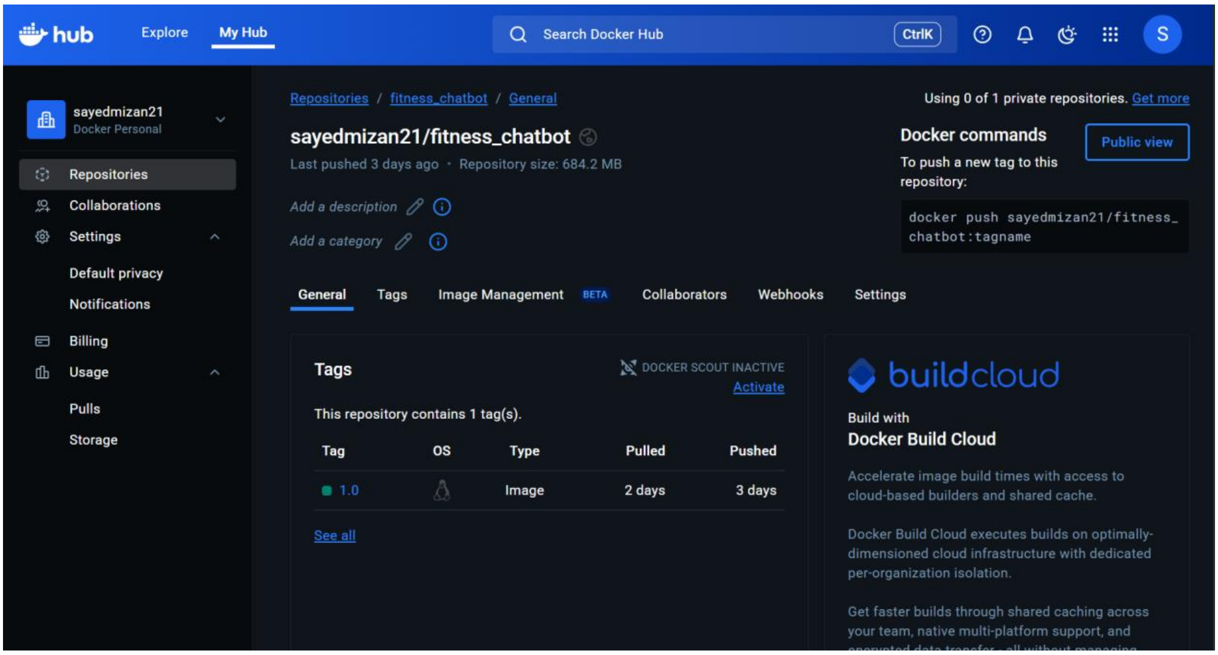
```dockerfile
# Use official Python runtime as a parent image
FROM python:3.10-slim

# Set environment variables
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

# Set working directory
WORKDIR /app

# Install system dependencies
RUN apt-get update && apt-get install -y \
    build-essential \
    && rm -rf /var/lib/apt/lists/*

# Copy requirements.txt and install Python dependencies
COPY requirements.txt /app/
RUN pip install --upgrade pip
RUN pip install -r requirements.txt

# Copy the rest of the application code
COPY . /app/

# Expose the port streamlit runs on
EXPOSE 8501

# Run the Streamlit app
CMD ["streamlit", "run", "app.py", "--server.port=8501", "--server.address=0.0.0.0"]

```

Requirements file



```
streamlit
pandas
numpy
langchain
langchain-openai
faiss-cpu
openai
langchain-community
tiktoken
```

Dcoker Hub



Repositories / fitness_chatbot / General

Using 0 of 1 private repositories. Get more

sayedmizan21/fitness_chatbot

Last pushed 3 days ago · Repository size: 684.2 MB

Add a description

Add a category

Docker commands

Public view

To push a new tag to this repository:

```
docker push sayedmizan21/fitness_
chatbot:tagname
```

Tags

DOCKER SCOUT INACTIVE
Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
● 1.0		Image	2 days	3 days

See all

buildcloud

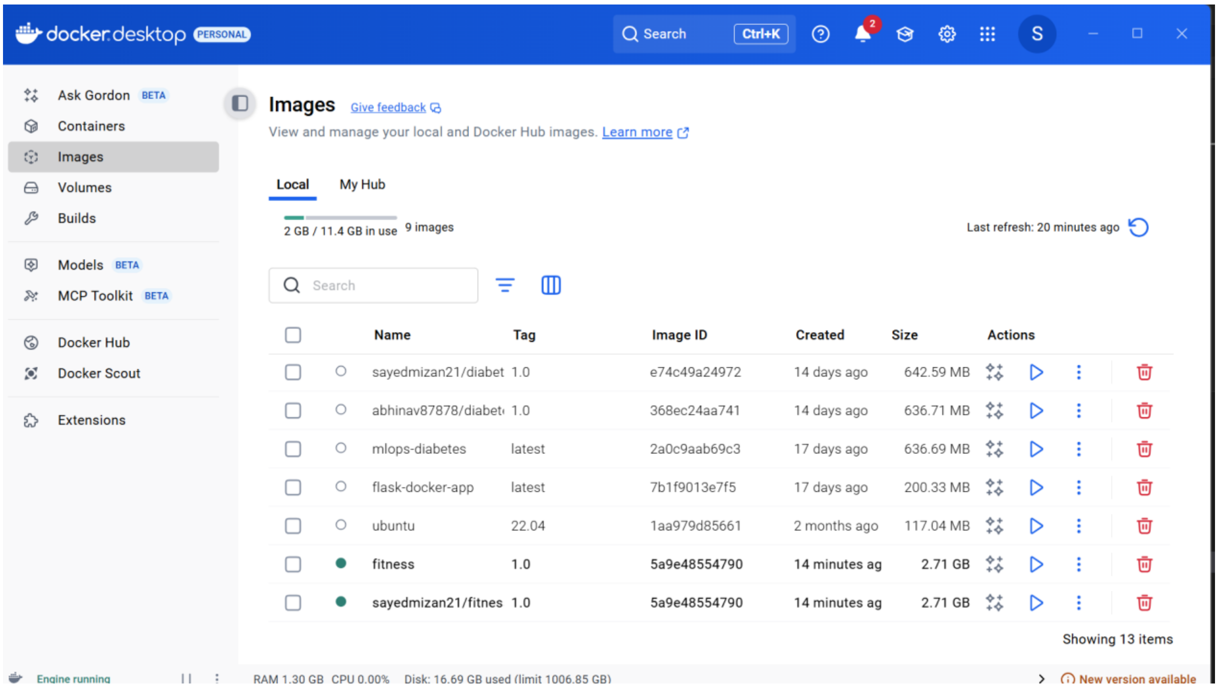
Build with
Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer – all without managing

Dcoker Desktop



Images Give feedback

View and manage your local and Docker Hub images. Learn more

2 GB / 11.4 GB in use 9 images

Last refresh: 20 minutes ago

□		Name	Tag	Image ID	Created	Size	Actions
□	○	sayedmizan21/diabet	1.0	e74c49a24972	14 days ago	642.59 MB	
□	○	abhinav87878/diabet	1.0	368ec24aa741	14 days ago	636.71 MB	
□	○	mlops-diabetes	latest	2a0c9aab69c3	17 days ago	636.69 MB	
□	○	flask-docker-app	latest	7b1f9013e7f5	17 days ago	200.33 MB	
□	○	ubuntu	22.04	1aa979d85661	2 months ago	117.04 MB	
□	●	fitness	1.0	5a9e48554790	14 minutes ag	2.71 GB	
□	●	sayedmizan21/fitnes	1.0	5a9e48554790	14 minutes ag	2.71 GB	

Showing 13 items

Engine running RAM 1.30 GB CPU 0.00% Disk: 16.69 GB used (limit 1006.85 GB) New version available