AI RAG System

Getting Started Guide

Install and run the AI RAG System on any device
with a single command. No setup required.

Alexandru Stefanescu

14 February 2026

Nufringen, Germany

Table of Contents

1. What is the AI RAG System?

The AI RAG System is a local Retrieval-Augmented Generation application. It lets you ingest your own documents (PDF, TXT, Markdown) and then ask questions — answers are grounded in your data with source citations.

The system runs entirely on your machine using Ollama for the LLM and ChromaDB for vector storage. No data leaves your environment.

How it works

1. You upload documents (PDF, TXT, or Markdown files)
2. Documents are split into chunks and stored as vectors
3. When you ask a question, the system finds the most relevant chunks
4. The LLM generates an answer using only those relevant chunks
5. Sources are cited so you can verify the answer

Architecture

```
Documents -> Chunker -> ChromaDB (vectors)
|
User Query -> Retrieve top chunks -> Ollama LLM -> Answer + Sources
```

2. Install

Run this single command on any machine — Raspberry Pi, Ubuntu, Debian, Fedora, or macOS. Everything is installed automatically, including Docker if needed.

```
curl -fsSL https://alexandrustefanescu.github.io/ai-rag-system/install.sh | bash
```

That's it. The installer will:

- Install Docker and Docker Compose (if not already installed)
- Check available disk space and memory
- Create the project directory at `~/ai-rag-system`
- Pull and start the application containers
- Download the AI models (gemma3:1b and llama3.2:1b) in the background

> **Warning:** The first run takes a few minutes because the AI models (~1-2 GB) need to be downloaded. Subsequent starts are fast.

Once complete, open your browser:

https://localhost:8443

> **Tip:** The app uses a self-signed SSL certificate. Your browser will show a security warning — click 'Advanced' then 'Proceed' to continue. This is expected.

System requirements

	Minimum	Recommended
RAM	4 GB	8 GB

Disk space	4 GB free	8 GB free
OS	Linux (ARM64/x86) or macOS	Raspberry Pi OS, Ubuntu, Debian

3. Add Your Documents

Option A: Through the web interface

1. Open `https://localhost:8443`
2. Click the upload area or drag-and-drop your files
3. Documents are automatically chunked and indexed

Option B: Copy files to the documents folder

```
cp ~/my-notes.pdf ~/ai-rag-system/documents/
cp ~/report.txt ~/ai-rag-system/documents/
```

Then trigger ingestion:

```
curl -k -X POST https://localhost:8443/api/v1/ingest
```

Supported file formats

Format	Extension
Plain text	.txt
PDF	.pdf
Markdown	.md

4. Using the Web Interface

Chat

1. Type your question in the text box at the bottom
2. Press Enter or click Send
3. The AI answers based on your uploaded documents
4. Each answer includes expandable source citations with relevance scores

Model selection

Use the dropdown in the header to switch between available models (e.g., gemma3:1b, llama3.2:1b).

Manage documents

The side panel shows all indexed documents with file name, size, chunk count, and a delete button to remove individual documents.

5. Using the API

Interactive API docs are available at `https://localhost:8443/api/docs`

Method	Endpoint	Description
GET	/api/v1/health	Health check
GET	/api/v1/status	System status
POST	/api/v1/ask	Ask a question
POST	/api/v1/upload	Upload documents
POST	/api/v1/ingest	Re-ingest documents folder
GET	/api/v1/documents	List indexed documents
DELETE	/api/v1/documents/{name}	Delete a document

Ask a question

```
curl -k -X POST https://localhost:8443/api/v1/ask \
-H "Content-Type: application/json" \
-d '{"question": "What is machine learning?"}'
```

Upload a document

```
curl -k -X POST https://localhost:8443/api/v1/upload \
-F "files=@my_document.pdf"
```

Tip: The -k flag tells curl to accept the self-signed certificate.

6. Configuration

To customize settings, edit `~/ai-rag-system/docker-compose.yml` and add environment variables under the `rag` service, then restart:

```
cd ~/ai-rag-system && docker compose restart rag
```

Variable	Default	Description
LLM_MODEL	gemma3:1b	Default LLM model
LLM_AVAILABLE_MODELS	gemma3:1b, llama3.2:1b	Models in dropdown
LLM_TEMPERATURE	0.3	Generation temperature
LLM_MAX_TOKENS	512	Max response tokens
CHUNK_SIZE	500	Characters per chunk
CHUNK_OVERLAP	100	Overlap between chunks
VS_QUERY_RESULTS	5	Candidates per query

7. Troubleshooting

Browser shows security warning

This is expected (self-signed certificate). Click 'Advanced' then 'Proceed to localhost'.

Models are still downloading

Run `cd ~/ai-rag-system && docker compose logs ollama-pull` to check progress. First download takes several minutes.

App is not responding

Check if containers are running: `cd ~/ai-rag-system && docker compose ps`

No answer or empty response

Make sure you have documents uploaded. Check the document panel in the web UI.

Port already in use

Edit `~/ai-rag-system/docker-compose.yml`, change `8443:8443` to `9443:8443`, then restart.

Useful commands

```
cd ~/ai-rag-system

# View logs
docker compose logs -f

# Restart
docker compose restart

# Stop
docker compose down

# Uninstall (removes all data)
docker compose down -v
```

Warning: 'docker compose down -v' deletes all downloaded models and indexed documents. Only use for a clean reset.