

Set AI with Gemini

"Inference Jobs" are Karakeep's AI features: after you save a bookmark, it sends the content to a language model (OpenAI or a local Ollama) to generate tags and a summary. **Failed: 36** means 36 of those requests errored out and ran out of retries. Your bookmarks are still saved, they just didn't get AI tags or summaries. Nothing else is broken.

Find the actual error

bash

```
docker logs --tail 200 karakeep 2>&1 | grep -iE "inference|error|fail|ollama|openai"
```

Use your real container name from `docker ps`. The error message usually tells you which of these it is:

Error	Cause	Fix
<code>401</code> / <code>Incorrect API key</code>	Wrong or missing <code>OPENAI_API_KEY</code>	Check the key in your <code>.env</code> or compose file, then recreate the container
<code>429</code> / <code>quota</code> / <code>rate limit</code>	No credit left or too many requests	Add billing credit, or throttle
<code>ECONREFUSED</code> / <code>fetch failed</code>	Ollama unreachable from inside Docker	See below
<code>model not found</code>	Model name doesn't match what's pulled	<code>ollama pull <model></code> and match <code>INFERENCE_TEXT_MODEL</code> exactly
<code>timeout</code> / <code>aborted</code>	Model too slow (very common on CPU-only machines)	Raise the timeout
JSON/parse or schema error	Small local models often fail to produce valid structured output	Set <code>INFERENCE_OUTPUT_SCHEMA=plain</code> or use a bigger model

If you use Ollama

Inside a container, `localhost` is the container itself, not your host. Use the host's address instead:

yaml

```
environment:
  OLLAMA_BASE_URL: http://host.docker.internal:11434
  INFERENCE_TEXT_MODEL: llama3.2:3b
  INFERENCE_IMAGE_MODEL: llava
  INFERENCE_JOB_TIMEOUT_SEC: 300
  INFERENCE_CONTEXT_LENGTH: 4096
extra_hosts:
  - "host.docker.internal:host-gateway"
```

Ollama also listens only on `127.0.0.1` by default. For Docker to reach it, set `OLLAMA_HOST=0.0.0.0` (e.g. `systemctl edit ollama` and add `Environment="OLLAMA_HOST=0.0.0.0"`). Firewall port 11434 so it's not exposed publicly. Test from inside the container:

bash

```
docker exec karakeep wget -q0- http://host.docker.internal:11434/api/tags
```

The default job timeout is short (about 30 seconds), and on laptop-class CPUs a local model easily takes longer, so raising `INFERENCE_JOB_TIMEOUT_SEC` is often the whole fix.

If you use OpenAI

Check `OPENAI_API_KEY`, that the account has credit, and that `INFERENCE_TEXT_MODEL` is a model your key can access. If you route through a proxy or compatible provider, `OPENAI_BASE_URL` has to be correct.

Clearing the 36 failures

1. Fix the cause and recreate the container: `docker compose up -d`.
2. On the admin page, use the admin actions to re-run the failed inference jobs (the wording varies by version). Otherwise, bookmarks only get re-tagged when you trigger it per bookmark.

If you don't want AI features

Remove the `OPENAI_API_KEY` / `OLLAMA_BASE_URL` settings, or set `INFERENCE_ENABLE_AUTO_TAGGING=false` (and the summarization equivalent if your version has it), and the jobs stop being queued.

Which provider are you using, OpenAI or Ollama? If you paste the log lines from the grep above, I can tell you exactly which row applies.

Investigating why embeddings run without an explicit setting.

That confirms it. Karakeep is asking Gemini for `text-embedding-3-small`, which Google doesn't have. You have no `EMBEDDING_*` variables set, so it's using the OpenAI default. The fix is to point it at a Gemini embedding model.

1. Find the model's dimension

bash

```
curl -s https://generativelanguage.googleapis.com/v1beta/openai/embeddings \
  -H "Authorization: Bearer $KEY" -H "Content-Type: application/json" \
  -d '{"model":"gemini-embedding-2","input":"hello"}' \
  | python3 -c "import sys,json; print(len(json.load(sys.stdin)['data'][0]['embedding']))"
```

If it prints a number, that's your dimension. If it prints an error, such as a 503 or 429, retry shortly. If the model isn't accepted, repeat with `gemini-embedding-001`.

2. Add the settings

In your compose file, using the model that worked and the number from step 1:

yaml

```
EMBEDDING_TEXT_MODEL: gemini-embedding-2
EMBEDDING_DIMENSIONS: <number from step 1>
```

Karakeep's default dimension is sized for the OpenAI model, so leaving `EMBEDDING_DIMENSIONS` unset would cause a mismatch. Karakeep's docs say a wrong value makes it fail to start.

3. Apply and regenerate

bash

```
docker compose up -d
docker logs --since 1m link.meson.kr 2>&1 | grep -iE "embedding|error" | head
```

If the container starts normally, go to the admin page and click **Regenerate Embeddings for All Bookmarks**. Use "All", not "Failed Only", since the old vector setup assumed a different model and size. Then watch for errors:

bash

```
docker logs -f --since 1m link.meson.kr 2>&1 | grep -iE "embeddings"
```

If you'd rather not use embeddings

I can't tell from your variables how auto-indexing got enabled, because `EMBEDDING_ENABLE_AUTO_INDEXING` isn't in your environment, yet jobs run. In some versions it appears to default on. To turn it off, add `EMBEDDING_ENABLE_AUTO_INDEXING: "false"`, recreate the container, and see whether new bookmarks still queue embedding jobs. I'm not certain that variable stops everything in your version, so verify in the logs.

Please tell me the dimension number and whether the container starts cleanly after the change.

```
meson@HERMiT: 15:59:05 /home/app-data/link.meson.kr$ docker logs --since 10m link.meson.kr
2>&1 | grep -i "embeddings" | head -5
2026-10-08T06:59:10.554Z info: Starting embeddings worker ...
2026-10-08T07:02:14.525Z info: <-- POST
/api/trpc/admin.regenerateAllBookmarkEmbeddings?batch=1
2026-10-08T07:02:16.499Z debug: [embeddings][579] Embedding text length: 1559 characters
2026-10-08T07:02:16.628Z error: [embeddings][579] embeddings job failed: Error: 404
models/text-embedding-3-small is not found for API version v1main, or is not supported for
embedContent. Call ModelService.ListModels to see the list of available models and their supported
methods.
at async runEmbeddings (file:///app/apps/workers/dist/index.js:87919:4)
meson@HERMiT: 16:04:38 /home/app-data/link.meson.kr$ docker exec link.meson.kr env | grep -iE
"EMBEDDING"
meson@HERMiT: 16:04:44 /home/app-data/link.meson.kr$ curl -s
https://generativelanguage.googleapis.com/v1beta/openai/embeddings \
-H "Authorization: Bearer $KEY" -H "Content-Type: application/json" \
-d '{"model":"gemini-embedding-2","input":"hello"}' \
| python3 -c "import sys,json; print(len(json.load(sys.stdin)['data'][0]['embedding']))"
3072
```

Working out the right embedding dimension setting.

The dimension is **3072**, so here is the change.

Update the compose file

yaml

```
EMBEDDING_TEXT_MODEL: gemini-embedding-2
EMBEDDING_DIMENSIONS: 3072
```

Then recreate the container and check that it starts cleanly:

bash

```
docker compose up -d
docker logs --since 1m link.meson.kr 2>&1 | grep -iE "embedding|dimension|error|fail" | head
```

Regenerate

On the admin page, click **Regenerate Embeddings for All Bookmarks**, then watch the log:

bash

```
docker logs -f --since 1m link.meson.kr 2>&1 | grep -iE "embeddings"
```

A successful job should show `Completed successfully` rather than the 404. Use "All" rather than "Failed Only", because the vector store was set up for the old dimension and Karakeep's release notes say a model or dimension change needs a full regeneration.

If something goes wrong

- **Container fails to start or restarts in a loop with a dimension error:** the vector store was created with the old size. Check `docker logs link.meson.kr` for the exact message and paste it here. I don't want to guess at a database fix without seeing it, and I'd back up the data directory first (`/home/app-data/link.meson.kr` and the Docker volumes).
- **429 or 503 on some embedding jobs:** that's the free-tier limit. Wait a few minutes, then use **Regenerate Embeddings for Failed Bookmarks Only**.
- **Semantic search:** it's labeled experimental in Karakeep, so judge it on your own bookmarks. Normal keyword search works the same as before.

Tell me if the container starts cleanly and whether the embedding jobs complete.

```
“ EMBEDDING_TEXT_MODEL: gemini-embedding-2 EMBEDDING_DIMENSIONS: 3072
```

in `.env` or `docker-compose.yml` ?

Clarifying the correct syntax for environment variable configuration.

Put them wherever your other `INFERENCE_*` settings already are. That's the safest choice, because Compose then applies them the same way. Which file that is depends on how your setup was built.

Check where your other settings live

bash

```
cd /home/app-data/link.meson.kr
grep -n "INFERENCE_TEXT_MODEL" docker-compose.yml .env 2>/dev/null
```

- **Found in `docker-compose.yml`**: add the lines there, in the same `environment:` block, using the YAML format:

yaml

```
environment:
  INFERENCE_TEXT_MODEL: gemini-2.5-flash-lite   # whatever you use
  EMBEDDING_TEXT_MODEL: gemini-embedding-2
  EMBEDDING_DIMENSIONS: 3072
```

- **Found in `.env`**: add them there, using `KEY=value` with no spaces and no colon:

```
EMBEDDING_TEXT_MODEL=gemini-embedding-2
EMBEDDING_DIMENSIONS=3072
```

This works only if the compose file has `env_file: .env` for the Karakeep service. The official Karakeep compose file does this, but if your service has only an `environment:` block, a plain `.env` entry won't reach the container. In that case reference it as `EMBEDDING_TEXT_MODEL: ${EMBEDDING_TEXT_MODEL}` or just put the value in the compose file.

The `KEY: value` format I showed you is YAML, so it only works in `docker-compose.yml`. If you paste it into `.env`, it won't be read correctly.

Which is better?

The model and dimension aren't secrets, so either file works. Keep your API key in `.env` with `chmod 600` and the plain settings in the compose file. That makes it easy to see your whole configuration in one place.

Verify it took effect

bash

```
docker compose up -d
docker exec link.meson.kr env | grep -iE "EMBEDDING"
```

You should see both lines. If the output is empty, the change didn't reach the container, so check which file you edited and the indentation (YAML needs spaces, not tabs).

Revision #1

Created 2026-10-08 07:13:22 UTC by meson

Updated 2026-10-08 07:13:49 UTC by meson