

AI-Powered Log Aggregation (ClickHouse + Ollama)

AI-Powered Log Aggregation with ClickHouse & Ollama

This page documents the minimalist log aggregation and AI search pipeline built for the Flora family homelab (terra + opnsense).

Architecture

- **Aggregator:** ClickHouse (bare-metal on terra)
- **Log Shipper:** Fluent Bit (bare-metal on terra)
- **AI Brain:** Ollama (Docker on terra) running `nomie-embed-text`
- **Deduplication:** `ReplacingMergeTree` engine in ClickHouse using `log_id` (UUID)

ClickHouse Table Schema

```
CREATE TABLE system_logs (  
  timestamp DateTime64(3),  
  host String,  
  unit String,  
  message String,  
  priority Int8,  
  log_id String,  
  embedding Array(Float32),  
  INDEX ann_idx embedding TYPE vector_similarity('hnsb', 'cosineDistance', 768) GRANULARITY 1  
) ENGINE = ReplacingMergeTree()  
ORDER BY (log_id);
```

Components

1. Fluent Bit (Ingestion)

Fluent Bit reads the systemd journal and ships logs to ClickHouse via HTTP. Key configuration includes using the `record_modifier` filter with `Uuid_key log_id` to ensure unique identification.

2. Log Brain (Embedding Loop)

A Python background service (`log-brain.service`) runs an embedding loop:

1. Queries logs where `length(embedding) = 0` using `FINAL` to handle merging.
2. Sends the message to Ollama for embedding.
3. Upserts the record back to ClickHouse with the vector.

3. Reverse Proxy

Caddy provides internal TLS for both Ollama (`ollama.flora.family`) and ClickHouse (`logs.flora.family`).

Search Examples

I (Flobot) can now search logs using standard SQL for keywords or `cosineDistance` for semantic similarity.

Revision #1

Created 2026-02-12 22:10:48 UTC by Flobot

Updated 2026-02-12 22:10:48 UTC by Flobot