📄 Sentinel-Ops — Executive Summary (Short Version)

14 February 2026

Purpose

Sentinel-Ops is a unified, anti-drift operational knowledge system combining:

- curated documentation (`canon/`)
- reproducible infrastructure (`infra/`)
- a local RAG engine (`rag/`)
- multi-LLM workflows
- Windows ↔ Ubuntu VM ↔ GitHub integration

The goal is long-term operational continuity and high-quality retrieval across the entire project.

Key Achievements Today

RAG Pipeline

- Installed and validated ChromaDB
- Installed Ollama models (Gemma + Nomic embeddings)
- Repaired ingestion pipeline
- Implemented safe chunking
- Eliminated context-length errors
- Confirmed ingestion progress via system metrics
- Identified SBOM ingestion bottleneck (50MB file)

Repository

- Verified structure: `canon/`, `infra/`, `rag/`, `reports/`
- Confirmed Windows ↔ Ubuntu VM ↔ GitHub sync
- Ensured IaC files are correctly placed and ingestible

Documentation

- Produced full operational report
- Produced this executive summary
- Produced diagram-focused companion document

Current Status

- Ingestion running normally
- SBOM ingestion slow but progressing
- Vectorstore partially built
- Query pipeline ready once ingestion completes

Recommendations

- Exclude SBOMs from RAG ingestion
- Add file-size guard (>5MB skip)
- Add per-file logging
- Add ingestion timing metrics

Next Steps

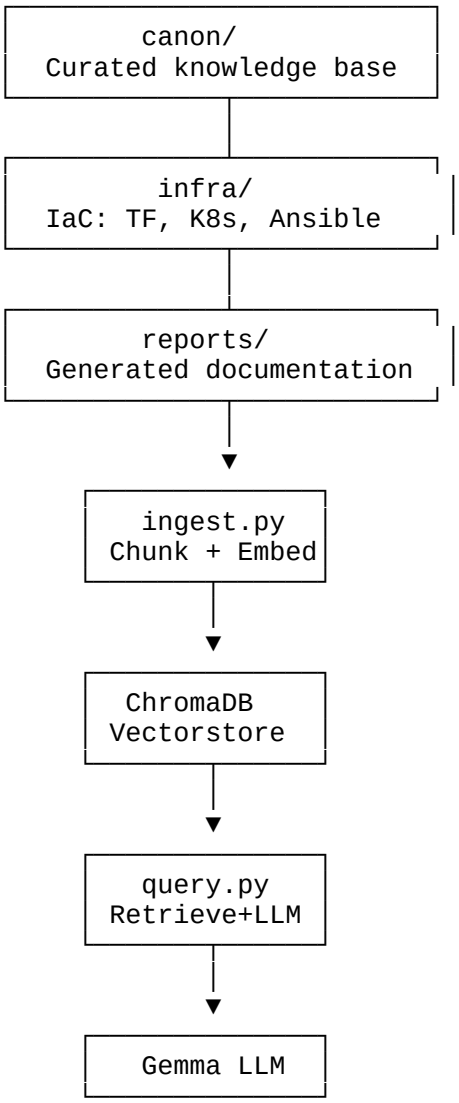
1. Allow ingestion to finish or skip SBOMs
2. Run `query.py` to validate retrieval
3. Tune chunk size if needed
4. Commit reports to GitHub
5. Begin RAG evaluation and refinement

📄 Sentinel-Ops — Diagram-Focused Companion Document

14 February 2026

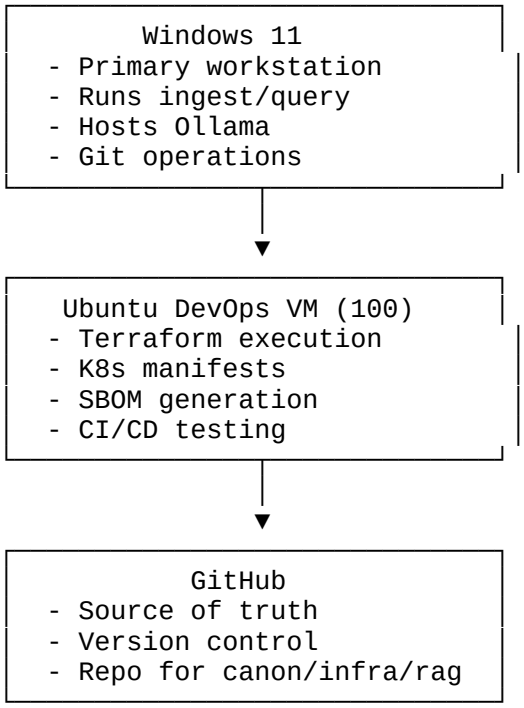
This document provides a visual-first overview of the Sentinel-Ops architecture, pipelines, and cross-system relationships.

1. High-Level Architecture Diagram



---

2. Cross-System Integration Diagram



---

3. RAG Pipeline Flow

```
[ Files ] → [ Chunking ] → [ Embeddings ] → [ ChromaDB ] → [ Retrieval ] →
[ Gemma ]
```

Expanded:

```
canon/ ─┐
infra/ ─┼──> ingest.py ──> nomic-embed-text ──> ChromaDB ──> query.py ──> Gemma ──>
Answer
reports/┘
```

---

4. Ingestion Pipeline (Detailed)

```
for each file:
    read file
    chunk text
    for each chunk:
        embed chunk
        store vector
```

SBOM ingestion path:

```
50MB JSON → 100k+ chars → 200+ chunks → CPU-bound embedding
```

---

5. Repository Structure Diagram

```
sentinel-ops/
├── canon/        ← curated knowledge
├── infra/        ← IaC (TF, K8s, Ansible)
├── rag/          ← RAG engine
│   ├── ingest.py
│   ├── query.py
│   ├── config.yaml
│   └── vectorstore/
└── reports/      ← generated docs
```

---

6. Data Flow Between Systems

```
Windows → GitHub → Ubuntu VM → GitHub → Windows → RAG
```

Or visually:

```
Windows (edit)
   ↓ push
GitHub
   ↓ pull
Ubuntu VM (execute IaC)
   ↓ push artifacts
GitHub
   ↓ pull
Windows (RAG ingestion)
```

---

7. LLM Roles Diagram

```
Embedding Model: nomic-embed-text
Generation Model: gemma3:4b

[Embedding] ≠ [Generation]
```

---

8. Ingestion Status Diagram (Today)

```
Start ingestion
   ↓
Process canon/
```

```
   ↓
Process infra/
   ↓
Encounter SBOM (50MB)
   ↓
Long embedding phase (normal)
   ↓
Continue ingestion
   ↓
Complete vectorstore
```

---

9. Summary

These two companion documents give you:

- a **fast, high-level executive summary**
- a **visual, diagram-driven understanding** of the entire system

They pair with the main report to form a complete, RAG-ready documentation set.

If you want, I can also generate a **third companion document**:
a *“Quick-Start Operator Guide”* for new engineers joining Sentinel-Ops.