Sentinel-Ops — Quick-Start Operator Guide

For New Engineers Joining the Project

14 February 2026

---

1. Purpose of This Guide

This guide provides a fast, practical onboarding path for new engineers joining the Sentinel-Ops project. It explains:

- What Sentinel-Ops is
- How the repository is structured
- How the RAG engine works
- How to run ingestion and queries
- How Windows, Ubuntu VM 100, and GitHub fit together
- What tools you need
- What to do on Day 1

It is intentionally concise, operational, and action-oriented.

---

2. What Sentinel-Ops Is

Sentinel-Ops is a unified operational knowledge system combining:

- **Curated documentation** (`canon/`)
- **Infrastructure-as-Code** (`infra/`)
- **A local RAG engine** (`rag/`)
- **Multi-LLM workflows**
- **Cross-platform execution** (Windows + Ubuntu VM 100)
- **GitHub as the source of truth**

Its purpose is to provide:

- long-term operational continuity
- anti-drift documentation
- reproducible infrastructure
- high-quality retrieval using local LLMs

---

3. Repository Structure (Mental Model)

```
sentinel-ops/
├── canon/        ← curated, authoritative documents
├── infra/        ← Terraform, Kubernetes, Ansible, SBOMs
├── rag/          ← ingestion + retrieval engine
└── reports/      ← generated documentation
```

If you understand this structure, you understand the project.

---

4. Your Working Environments

4.1 Windows 11 (Primary Workstation)

Used for:

- Editing files
- Running the RAG engine
- Running Ollama
- Running Python ingestion
- Git operations

Key paths:

```
C:\Users\Admin\Documents\sentinel-ops\
C:\Users\Admin\.ollama\models\
```

4.2 Ubuntu DevOps VM (VM 100)

Used for:

- Terraform execution
- Kubernetes manifests
- SBOM generation
- CI/CD testing

Key paths:

```
/home/andy/sentinel-ops/
/home/andy/.ssh/
```

4.3 GitHub (Source of Truth)

Everything flows through GitHub:

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

---

5. Tools You Need Installed

On Windows

- Python 3.12
- Ollama
- ChromaDB (Python package)
- Git
- VS Code (recommended)

On Ubuntu VM

- Terraform
- kubectl
- Ansible
- Git
- SBOM tooling (Syft/Grype)

---

6. RAG Engine Overview

The RAG engine lives in:

```
sentinel-ops/rag/
```

It has two main scripts:

6.1 ingest.py

- Walks the repo
- Reads files
- Chunks text
- Embeds chunks using `nomic-embed-text`
- Stores vectors in ChromaDB

6.2 query.py

- Accepts a question

- Retrieves relevant chunks
- Sends them to Gemma
- Returns an answer

6.3 config.yaml

Controls:

- which folders to ingest
- chunk size
- overlap
- model selection
- vectorstore path

---

7. How to Run the RAG Engine

7.1 Step 1 — Open a terminal in the rag/ folder

```
cd C:\Users\Admin\Documents\sentinel-ops\rag
```

7.2 Step 2 — Run ingestion

```
python ingest.py
```

This will:

- walk canon/, infra/, reports/
- chunk files
- embed them
- write vectors to `vectorstore/`

Important:
Large files (e.g., 50MB SBOMs) take time.
This is normal.

7.3 Step 3 — Run a query

After ingestion completes:

```
python query.py
```

You’ll be prompted for a question.

Example:

```
What is the Sentinel-Ops IaC baseline?
```

Gemma will answer using your repo content.

---

8. Best Practices for Operators

8.1 Keep canon/ clean

Only authoritative documents belong here.

8.2 Keep infra/ reproducible

Terraform, Kubernetes, and Ansible must remain deterministic.

8.3 Avoid polluting the vectorstore

Skip:

- SBOMs
- binaries
- large JSON files
- logs

8.4 Commit early and often

GitHub is the source of truth.

8.5 Regenerate ingestion after major changes

Any update to canon/ or infra/ requires a fresh ingestion.

---

9. Troubleshooting

Problem: ingestion is slow

Cause: large files (e.g., SBOMs).
Fix: skip them.

Problem: context length exceeded

Cause: chunk too large.
Fix: reduce `chunk_size` in config.yaml.

Problem: vectorstore corrupted

Fix:

```
rm -r vectorstore
python ingest.py
```

Problem: Gemma not responding

Fix: restart Ollama.

---

10. Quick Reference Commands

Windows

```
python ingest.py
python query.py
ollama list
git status
git pull
git push
```

Ubuntu VM

```
terraform apply
kubectl get pods
ansible-playbook site.yml
syft /host > sbom.json
```

---

11. Onboarding Checklist (Day 1)

Task	Status
Clone repo from GitHub	☐
Install Python + Ollama	☐
Install ChromaDB	☐
Run ingest.py	☐
Run query.py	☐
Explore canon/	☐
Explore infra/	☐
Explore rag/	☐
Connect to Ubuntu VM	☐
Run Terraform plan	☐

---

12. Final Notes for New Engineers

Sentinel-Ops is designed to be:

- predictable
- reproducible
- transparent
- version-controlled
- LLM-augmented

If you understand:

- the repo structure
- the ingestion pipeline
- the Windows ↔ Ubuntu ↔ GitHub flow

you can operate the entire system confidently.

If you'd like, I can also prepare a **fourth companion document**:
a *"Sentinel-Ops Architecture Deep Dive"* with diagrams, flows, and design rationale.