Project Progress Update

Date: February 13, 2026 Project: FoodProject SOC Platform (Wazuh + Shuffle + IRIS-web + SOC Integrator)

1) Executive Summary

The MVP platform is operational and running end-to-end in the lab environment. Core integrations are in place:

- Detection: Wazuh
- Automation: Shuffle
- Case management: IRIS-web (replacing DFIRTrack)
- Escalation (MVP): PagerDuty Stub
- Orchestration/API layer: soc-integrator

All major containers are currently up, and key health checks are passing.

2) Completed Work

Platform orchestration and operations

- Combined stack runner created and improved (`run-combined-stack.sh`)
- Added command support for:
  - `up` , `down` , `logs` , `status` , `help`
  - per-target control (`wazuh` , `iris` , `shuffle` , `pagerduty` , `integrator`)
- Added consolidated health/status script (`soc-status.sh`)

Integration architecture

- Connected Wazuh, Shuffle, IRIS-web, PagerDuty Stub, and soc-integrator on shared network
- Resolved startup conflicts and runtime issues (port, compose, routing compatibility)

SOC Integrator (MVP)

- Added/validated integration APIs for:
  - Wazuh

  - Shuffle
  - IRIS-web
  - PagerDuty Stub
- Implemented MVP orchestration endpoints:
  - `POST /mvp/incidents/ingest`
  - `POST /mvp/ioc/evaluate`
  - `POST /mvp/vpn/evaluate`
  - `GET /mvp/config/policies`
  - `PUT /mvp/config/policies`
  - `GET /mvp/health/dependencies`
- Added internal API-key protection for mutation endpoints

Persistence layer

- Added PostgreSQL service for soc-integrator (`soc-integrator-db`)
- Added incident/policy/audit schema and startup initialization
- Enabled deduplication and audit tracking for incident processing

Testing utilities and documentation

- Added Wazuh test-event injection script:
  - `scripts/send-wazuh-test-events.sh`
- Added root project docs:
  - `README.md`
- Added root ignore rules:
  - `.gitignore`

3) Current Live Status (Lab)

Current stack status: **UP**

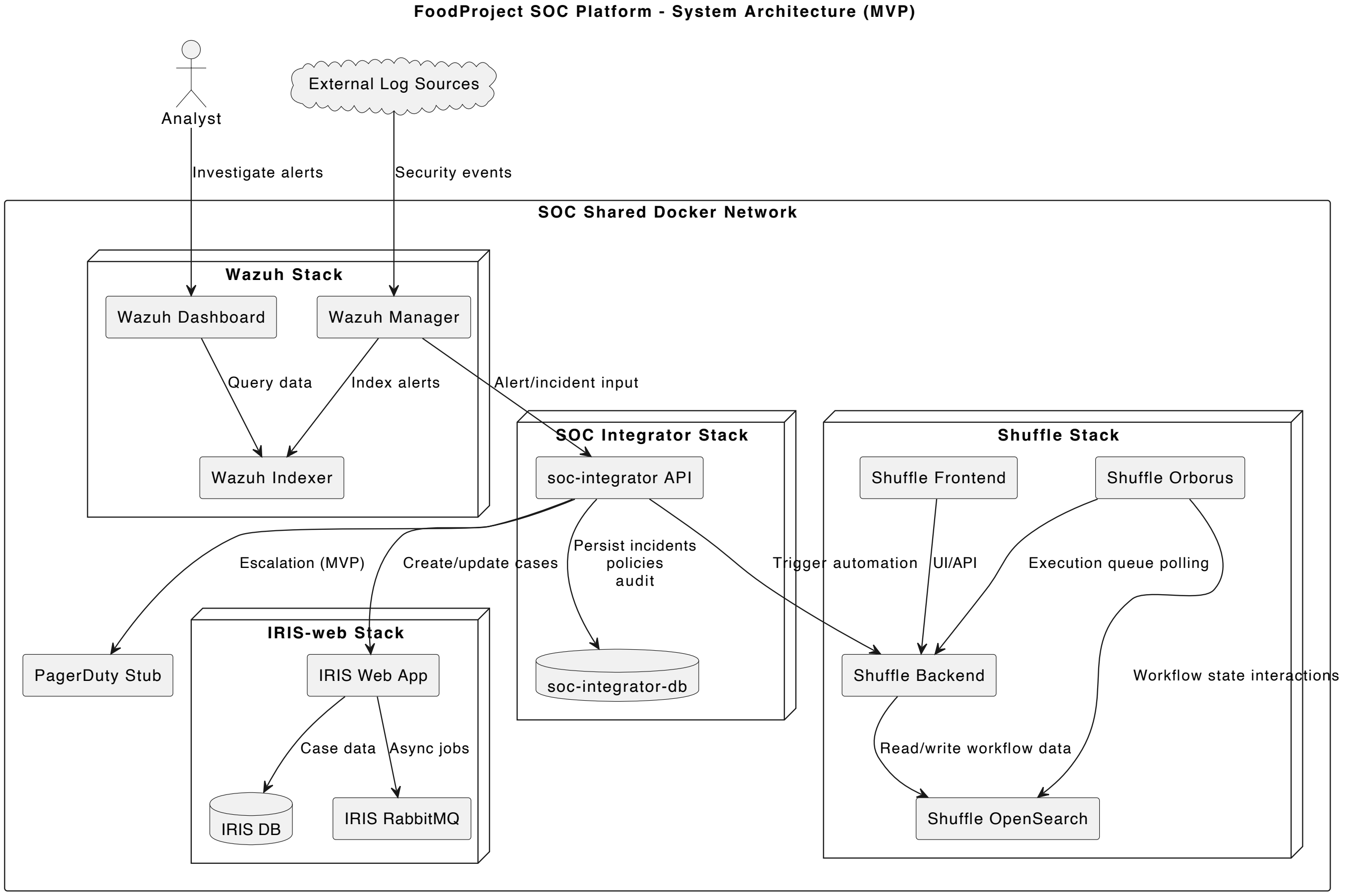
Healthy/available components:

- Wazuh manager, indexer, dashboard
- IRIS-web app/nginx/worker/db/rabbitmq
- Shuffle backend/frontend/opensearch/orborus
- PagerDuty Stub
- soc-integrator + soc-integrator-db

Endpoint checks:

- Wazuh Dashboard: OK
- Wazuh API: OK (auth-protected, expected 401 on unauthenticated root)
- IRIS Web: OK
- Shuffle Frontend: OK
- Shuffle Backend: reachable
- Shuffle OpenSearch: reachable (auth-protected)
- PagerDuty Stub: OK
- soc-integrator `/health`: OK

4) System Architecture Diagram (PlantUML)



5) In Progress / Remaining for Customer UAT

1. Detection content tuning

- Fine-tune Wazuh rules/decoders for customer log patterns and false-positive reduction

2. Use-case calibration

- Validate risk/severity mapping per approved use cases
- Tune exception list and threshold logic (especially VPN geo anomaly)

3. UAT evidence package

- Capture deterministic UAT scenarios and outputs for:
    - IOC flow
    - VPN outside-TH flow
    - IRIS case creation/update
    - PagerDuty Stub escalation path

4. Production hardening items

- Rotate default/local secrets used in lab config
- Lock down internal API keys and access boundaries

6) Risks / Notes

- Current escalation target is **PagerDuty Stub** by design for MVP. Real PagerDuty production integration is the next stage.
- Some Wazuh config certificate directories are root-owned in the local lab clone, which may affect local git add operations if not excluded/fixed.

7) Next Milestone (Proposed)

Next milestone: **MVP UAT Completion**

Target outputs:

- Approved UAT checklist execution
- Tuned policy thresholds for customer environment
- Signed-off incident lifecycle flow: Wazuh event -> soc-integrator decision -> IRIS case -> PagerDuty Stub escalation