High Level Design (HLD) - Mini SOC Platform

Presented by: Azeddine Hamdaoui

Overview

This document describes the high-level design for the Mini SOC (Security Operations Center) platform using Wazuh SIEM (Security Information and Event Management), docker swarm (container orchestration), and DevOps practices.

Scope

- Deployment of Wazuh stack on Docker Swarm
- high availability (HA), disaster recovery (DR), fault tolerance, and scalability

Wazuh SIEM Overview

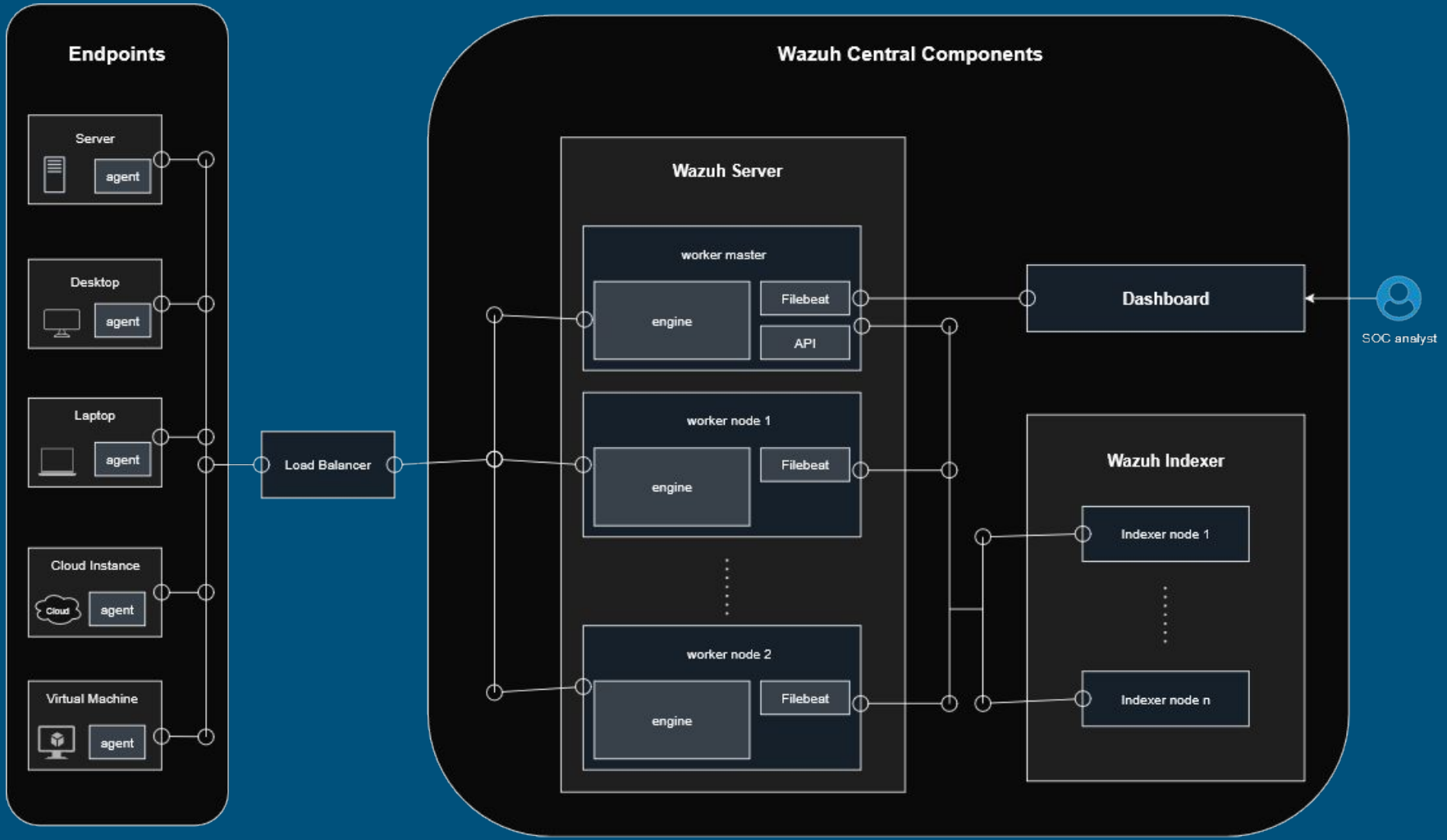
What is Wazuh?

- Wazuh is an **open-source security platform** that provides **log analysis**, **intrusion detection**, **vulnerability detection**, and **compliance monitoring**.
- It acts as a **Security Information** and **Event Management** (SIEM) system, collecting and analyzing data from endpoints, servers, and network devices

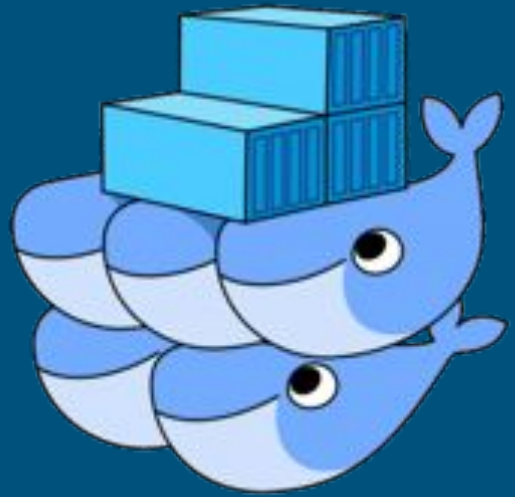
Key Components

- **Wazuh Manager**: Central component that analyzes **security events**, generates **alerts**, and manages **agents**.
- **Wazuh Agents**: Installed on endpoints or servers to collect logs, monitor files, detect anomalies, and send data to the manager.
- **Elasticsearch / Indexer**: Stores and indexes the collected data for fast search and analytics.
- **Kibana**: Provides dashboards and visualizations for monitoring and reporting.
- **Filebeat / Log Shippers**: Forward logs from agents to Elasticsearch in real time.

Wazuh stack Architecture


Endpoints
Server
agent
Desktop
agent
Laptop
agent
Cloud Instance
Cloud
agent
Virtual Machine
agent
Load Balancer
Wazuh Central Components
Wazuh Server
worker master
engine
Filebeat
API
worker node 1
engine
Filebeat
worker node 2
engine
Filebeat
Dashboard
SOC analyst
Wazuh Indexer
Indexer node 1
Indexer node n

Container Orchestration with Docker Swarm



Docker Swarm Overview

- **Container orchestration tool built into Docker**
  Enables you to manage multiple containers across different hosts as a single cluster.
  Unlike standalone Docker, Swarm automates deployment, scaling, and container lifecycle management.

- **Manages clusters of Docker nodes (machines) as a single virtual system**
  Nodes (physical or virtual machines) are grouped under one “swarm,” appearing as one logical Docker host.
  This allows distributed workloads to be managed from a single control point.

- **Provides high availability, scalability, and load balancing**
  Swarm automatically replicates services across nodes to prevent downtime.
  Traffic is distributed evenly between containers using built-in load balancing.
  You can easily scale services up or down with one command (docker service scale).

- **Secure and integrated by design**
  Uses mutual TLS for secure node communication.
  Works natively with existing Docker CLI and Compose files, requiring minimal new tooling.

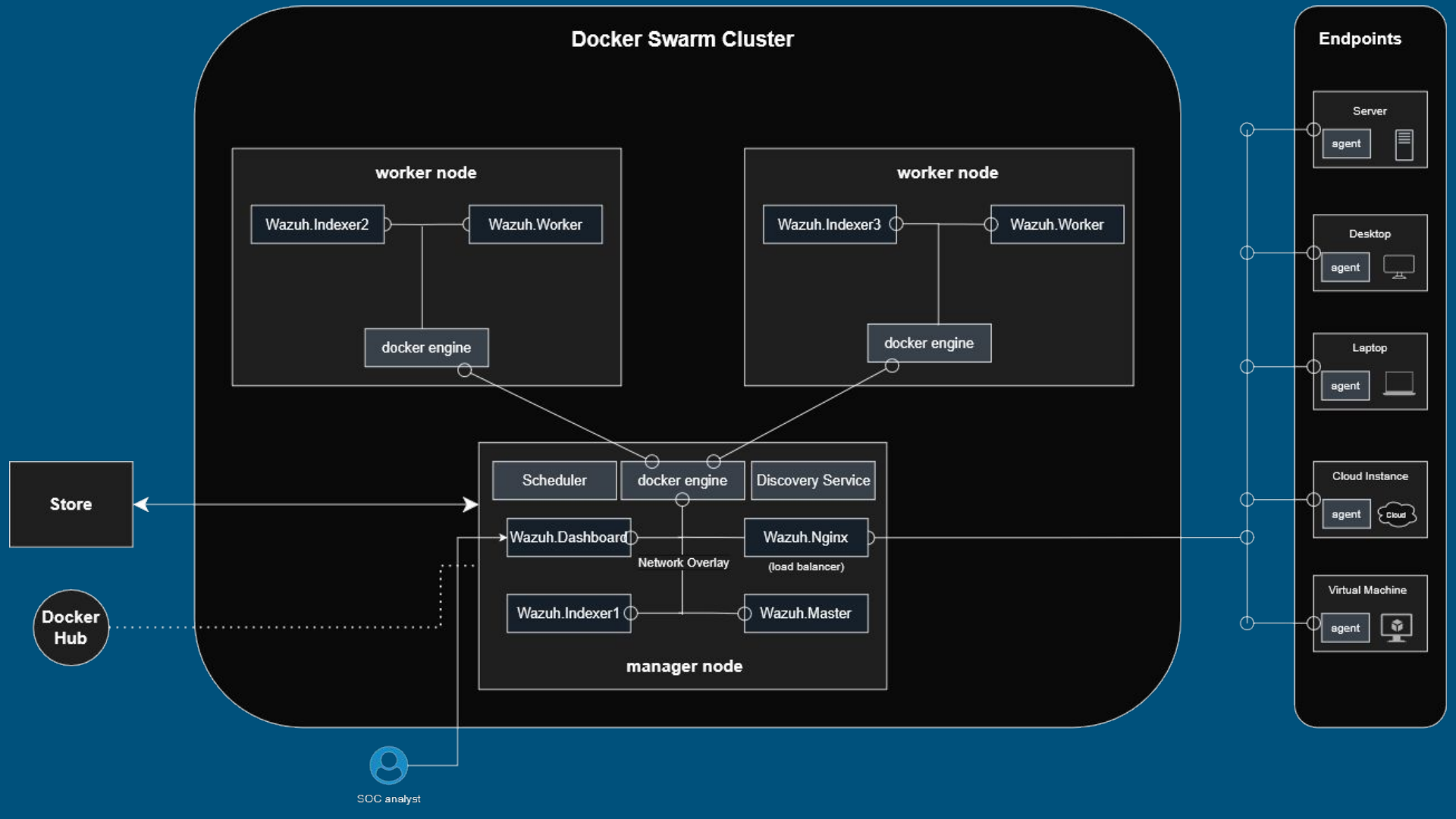
⚙️ How It Works

- **Manager Nodes**: Control the cluster, schedule services, handle orchestration.
- **Worker Nodes**: Run the containers (services) assigned by the managers.
- **Overlay Networks**: Enable container communication across multiple hosts.
- **Service Replication**: Automatically restarts failed containers.

Why Use Docker Swarm in This Project

- Simplifies multi-container deployment (e.g., Wazuh stack).
- Ensures high availability for critical services.
- Seamlessly scales horizontally when load increases.
- Built-in integration with Docker CLI (no external setup like Kubernetes).

Wazuh Stack Architecture on Docker Swarm (3-Node Cluster)


Docker Swarm Cluster
worker node
Wazuh.Indexer2
Wazuh.Worker
docker engine
worker node
Wazuh.Indexer3
Wazuh.Worker
docker engine
Store
Scheduler
docker engine
Discovery Service
Wazuh.Dashboard
Wazuh.Nginx
Network Overlay
(load balancer)
Wazuh.Indexer1
Wazuh.Master
manager node
Docker Hub
SOC analyst
Endpoints
Server
agent
Desktop
agent
Laptop
agent
Cloud Instance
agent
Cloud
Virtual Machine
agent

High Availability (HA), Disaster Recovery (DR), Fault Tolerance & Scalability

Ensuring continuous operation and resilience of the Wazuh Stack on Docker Swarm.

Introduction

Why Reliability Matters

- Security platforms must operate 24 / 7 / 365.
- Outages or data loss reduce visibility and response capability.
- This section covers the design strategies that make the environment resilient, recoverable, and scalable.

High Availability (HA)

Definition:

Ability of the system to remain operational despite node or service failures.

Implementation:

- Multiple Swarm nodes (1 manager + 2 workers) for cluster redundancy.
- Wazuh Indexer replicated across nodes for continuous log indexing.
- Wazuh Nginx acts as a load balancer for dashboard/API traffic.
- Overlay networks ensure service communication even if one node fails.

Disaster Recovery (DR)

Definition:

Process for restoring services and data after a catastrophic event.

Implementation:

- Automated backups of Indexer data and configurations.
- Off-site or remote storage of backups (e.g., S3 / NAS / remote volume).
- Recovery scripts or CI/CD jobs to redeploy Swarm stacks quickly.
- Option for secondary Swarm cluster or DR site replication.

Fault Tolerance

Definition:

Capability to continue operation when individual components fail.

Implementation:

- Swarm service replication keeps containers running if one crashes.
- Health-check & restart policies automatically recover failing services.
- Decoupled services (Manager, Indexer, Dashboard) prevent cascade failure.
- Persistent volumes protect stored data during container restarts.

Scalability

Definition:
Ability to increase capacity or performance as demand grows.

Implementation:

- **Horizontal scaling:** add worker nodes to the Swarm cluster.
- **Vertical scaling:** allocate more CPU/RAM to existing nodes.
- **Dynamic scaling via CI/CD:** pipelines deploy additional replicas automatically.
- **Load balancing** distributes traffic among Indexer and Worker nodes.