This master report, configuration **SENTINEL-2026-0128-ADV-PLAYBOOK**, consolidates all previous architectural decisions from versions v3 and v4 while integrating advanced security, hardware isolation, and MLOps capabilities. It serves as a comprehensive technical playbook for the Sentinel cluster.

---

# 📑 Sentinel: Integrated Technical Playbook

**Configuration ID:** SENTINEL-2026-0128-ADV-PLAYBOOK

**Primary Anchor:** 192.168.1.50 (Static)

**Hardware:** Proxmox VM ID 100 | 24GB Dedicated RAM | 4 vCPUs

**Status:** Stabilized; Transitioning to Hardened AI Workloads.

---

## 🗂️ Master Index

1. **Phase 1: Git-Ops & Resource Baseline:** Initializing the "Source of Truth."
2. **Phase 2: Hardening & eBPF Security:** CIS compliance and Tetragon runtime enforcement.
3. **Phase 3: Out-of-Band Management (OOBM):** Raspberry Pi observer nodes.
4. **Phase 4: AI Control Plane (MCP):** LLM-to-cluster integration via Model Context Protocol.
5. **Phase 5: Zero Trust Architecture (NIST 800-207):** Secure identity and networking.
6. **Phase 6: Advanced Storage & Isolation:** Kata Containers and USB RAID.
7. **Phase 7: MLOps Pipeline:** Automated machine learning lifecycles.

---

## 🛠️ Detailed Implementation Playbook

### Phase 1: Git-Ops & Resource Baseline

**Goal:** Establish a version-controlled "Golden Image" of all configurations and verify the 24GB RAM upgrade.

**Step 1: Initialize the Infrastructure Repo:** Create a dedicated directory to track system state instead of the entire root partition.
Bash

```
mkdir -p /root/sentinel-infrastructure && cd /root/sentinel-infrastructure
git init
```

-

**Step 2: Snapshot Current Configuration:** Export the Proxmox VM configuration and cluster inventory.

Bash

```
qm config 100 > sentinel-v100.conf
pvesh get /cluster/resources --type vm --output-format yaml > cluster_inventory.yaml
```

-

**Step 3: Verify RAM Upgrade:** Ensure the 24GB allocation is active.

Bash

```
free -h  # Confirm 'Mem: 24Gi'
```

-

## Phase 2: Hardening & eBPF Security (Tetragon)

**Goal:** Implement real-time security observability and kernel-level enforcement using **Tetragon**.

**Implementation:** Install Tetragon via Helm to monitor process execution and file access.

Bash

```
helm repo add cilium https://helm.cilium.io
helm install tetragon cilium/tetragon -n kube-system
```

- 
- **Technical Detail:** Tetragon utilizes **eBPF** to block malicious events (e.g., unauthorized binary execution in /tmp) with $<1\%$ CPU overhead.
- **Reference:** Tetragon Official Documentation

## Phase 3: Out-of-Band Management (OOBM)

**Goal:** Use a Raspberry Pi as an external observer to maintain cluster visibility if the primary node fails.

- **Step 1: Portability:** Copy the kubeconfig to the Pi to allow remote kubectl access.
- **Step 2: Monitoring:** Deploy **Prometheus** and **Grafana** on the Pi to scrape metrics from the Sentinel node.
- **Benefit:** Provides a "Secondary Eye" that remains independent of the Proxmox host.

## Phase 4: AI Control Plane (MCP)

**Goal:** Enable local AI (Ollama) to safely inspect the cluster using the **Model Context Protocol (MCP)**.

**Implementation:** Deploy a read-only ServiceAccount for the MCP server.

Bash

```
kubectl create serviceaccount mcp-server -n default
kubectl create clusterrolebinding mcp-server-binding --clusterrole=view
--serviceaccount=default:mcp-server
```

- 
- **Technical Detail:** This allows an AI assistant to diagnose issues by reading logs and events without the risk of deleting production data.
- **Reference:** Model Context Protocol Introduction

## Phase 5: Zero Trust Architecture (ZTA)

**Goal:** Align the cluster with **NIST SP 800-207** standards, treating every connection as potentially compromised.

- **Step 1: Micro-segmentation:** Use **Netplan** to define isolated VLANs for Management, App traffic, and Storage.
- **Step 2: Identity-based Access:** Enforce **mTLS** for all pod-to-pod communication.
- **Reference:** NIST SP 800-207 Zero Trust Architecture

### Phase 6: Advanced Storage & Isolation

**Goal:** Implement hardware-level isolation for untrusted workloads and distributed storage.

**Kata Containers:** Configured in containerd to wrap pods in lightweight VMs for a second layer of defense.

YAML

```
# RuntimeClass for Kata
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: kata
handler: kata-qemu
```

-
- **Longhorn USB RAID:** Use **Longhorn** to manage distributed block storage across USB-mounted drives.
- **Reference:** Kata Containers Architecture

### Phase 7: MLOps Pipeline

**Goal:** Automate the AI lifecycle—from data engineering to deployment—on MicroK8s.

- **Key Components:** * **MLflow:** For tracking experiments and model versions.
    - **GPU Integration:** Setup nvidia-container-toolkit for hardware acceleration on Ubuntu 24.04.
- **Reference:** MLOps Principles and Guides

---

## 🗺️ Summary Roadmap Table

| Priority | Phase | Goal | Key Tools |
|---|---|---|---|
| **1** | **Git-Ops** | Version control for Netplan, RKE2, and OOBM scripts. | Git, Proxmox |

| | | | |
|---|---|---|---|
| **2** | **RAM Verify** | Finalize hardware allocation (24GB). | free -h |
| **3** | **SecOps** | eBPF-based runtime enforcement. | **Tetragon**, CIS Audit |
| **4** | **OOBM** | Independent Raspberry Pi monitoring. | Prometheus, Grafana |
| **5** | **Zero Trust** | Identity-based access and network isolation. | **NIST 800-207**, mTLS |
| **6** | **Isolation** | Hardware-backed pod sandboxing. | **Kata Containers** |
| **7** | **MLOps** | Automated AI model deployment pipeline. | **MLflow**, MicroK8s |

---

**Report Finalized.** Ready to proceed with Phase 2 (Tetragon) or Phase 5 (Zero Trust Design)?