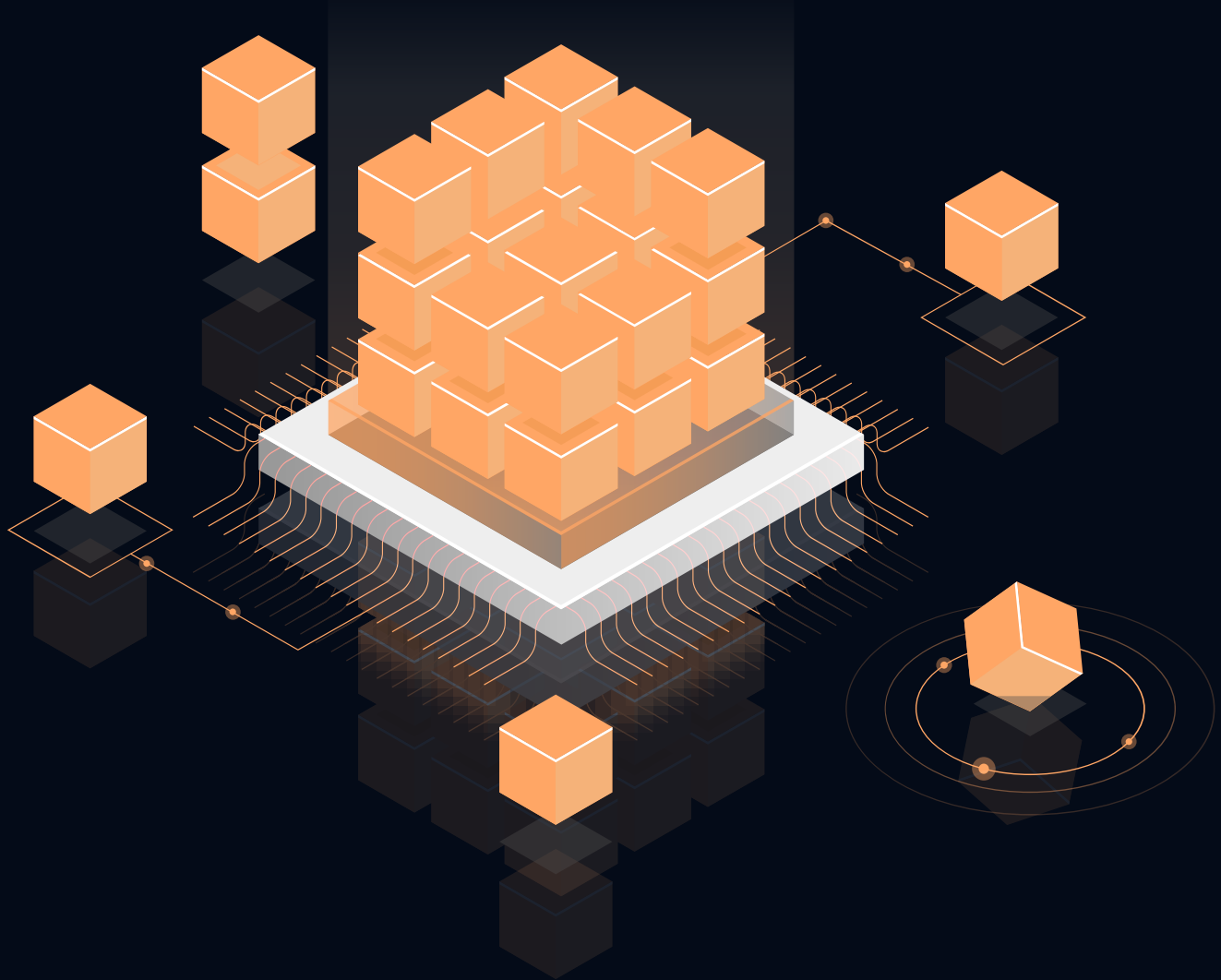
enclaive

Red Hat

Solution Brief

Multi-Tenancy, Secure Cloud Transitioning and Sovereignty with OpenShift and vHSM Key Management



Executive Summary

As organizations scale on shared Kubernetes to cut costs and move faster, safeguarding sensitive data gets harder—especially in hybrid footprints or on external infrastructure where **digital sovereignty** and **zero-trust** matter. Traditional encryption at rest and in transit is no longer enough to counter insider risk, cross-tenant exposure, and container-level breaches.

Red Hat and enclaive deliver a joint solution that pairs **Red Hat OpenShift** with **confidential computing** and **externalized key & workload identity management**. This extends today's encryption baseline with **hardware-backed confidentiality and integrity for Kubernetes applications**, enabling a sovereign, zero-trust, multi-tenant Kubernetes platform for workload-sensitive environments and regulatory demands.

Outcomes: a cloud-native foundation that drives

- **Stronger multi-tenant isolation** at node or pod granularity
- **Risk reduction** through enclave-protected workloads
- **Lower integration effort** using OpenShift-native tooling and policies, and
- **Faster time-to-value** via a validated, co-engineered stack
- **Audit readiness** with verifiable attestation signals ("compliance-as-code")

Acknowledgement

This solution was supported by numerous Red Hat team members.

Table of Contents

1 Introduction

Across the public and private sectors, organizations are moving fast to cloud-native platforms and Kubernetes to modernize, scale, and cut costs. Multi-tenancy is central to that strategy—but running many teams and workloads on shared infrastructure raises the stakes for security and sovereignty, especially when leveraging external service providers or public clouds.

This solution brief shows how **Red Hat OpenShift** combined with e**nclaive's virtual Hardware Security Module (vHSM)** and Confidential Computing delivers a zero-trust, sovereign Kubernetes foundation for sensitive, multi-tenant workloads—protecting data, code and AI models **in use**, not just at rest or in transit.

1.1 The Challenge: Securing Multi-Tenant Kubernetes

Organizations face tough regulatory and compliance demands for data protection. In shared Kubernetes environments, the security bar is higher:

- **Adopt Zero Trust—everywhere**
  Perimeters aren't enough. Clusters must default to "trust no one," with:
  - Continuous verification of workload identity and behavior
  - Strict, policy-driven access controls between services
  - Cryptographic attestation of platforms and components
  - Least-privilege access from user to process
- **Protect the host and control plane**
  Prevent container escape and privilege escalation from compromising other tenants or the underlying infrastructure.
- **Keep data, code and AI models confidential—even from admins**
  Ensure sensitive workload processed inside containers remains protected from cloud operators, privileged insiders, and bad actors.

Traditional controls help, but they **don't fully address data-in-use risk**, insider threats, or advanced attacks at the container and infrastructure layers. A new approach—rooted in Confidential Computing, attestation, and externalized key management—is required to make multi-tenant Kubernetes truly zero-trust and sovereignty-ready.

1.2 Solution: OpenShift and vHSM with Confidential Computing

Unite the power of **Red Hat OpenShift**—the enterprise Kubernetes standard—with **Red Hat Enterprise Linux** and **enclaive vHSM**, a virtual HSM that delivers externalized keys and workload identity with hardware-grade protection.

Confidential Computing, powered by hardware Trusted Execution Environments (TEEs), isolates workloads so that **data and code stay encrypted even while in use**—shielded from the host OS, hypervisor, and other privileged software. In parallel, virtualization helps protect the host from container escalations in shared, multi-tenant clusters. Combined with encryption **at rest** and **in transit**, plus **externalized keys and attestation**, the platform creates a **three-dimensional encryption posture**—covering memory, network, and storage by design.

Red Hat Enterprise Linux—widely certified across clouds, hardware, and software ecosystems—provides a stable, secure substrate for running applications inside protected enclaves, **often without code or configuration changes**.

enclaive vHSM extends those enclaves with integrated **key lifecycle management**, **workload identity**, cryptographic operations, and protected secrets storage **inside** the TEE. Access is gated by **attestation** and policy, ensuring sensitive material never leaves the enclave boundary.

Red Hat OpenShift delivers the robust, scalable Kubernetes platform to deploy and operate it all—complete with **enterprise security**, **developer tooling**, and **day-2 operations**—making it ideal for workload-sensitive environments.

What this means for you

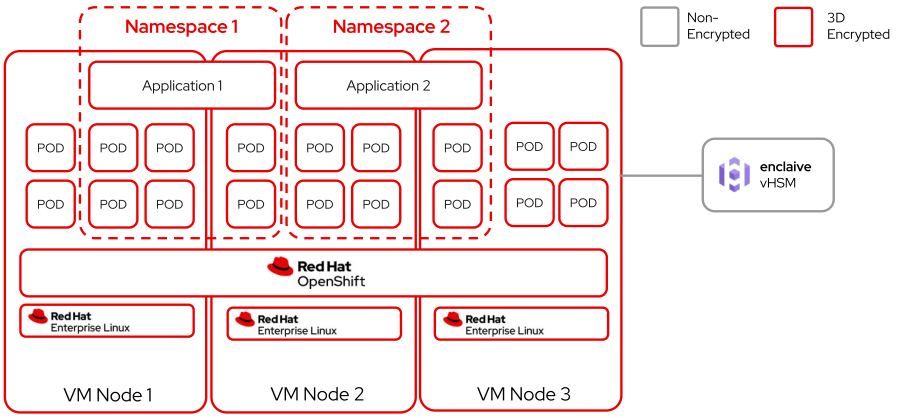
- **Data-in-use, at rest and in-transit protection ("3D encryption")** with hardware-graded confidentiality and integrity
- **Sovereign control** via externalized keys and attested identities
- **Zero-trust multi-tenancy** with strong isolation, end-to-end
- **Faster adoption** using proven OpenShift tooling and workflows

2 Reference Architecture

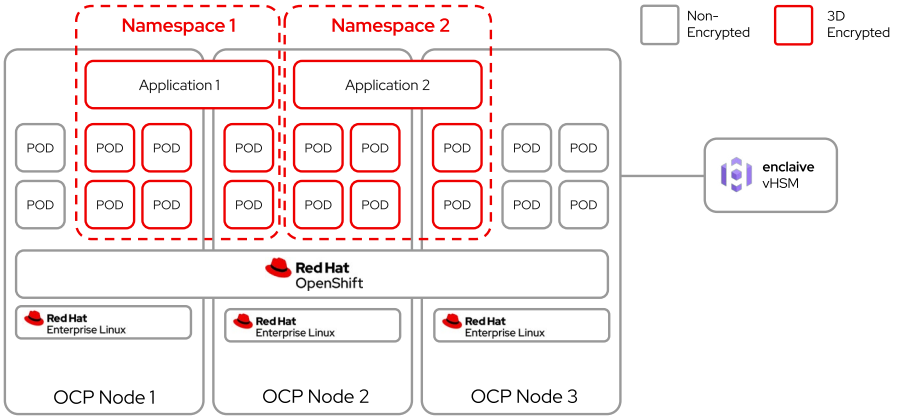
In a nutshell, the solution works as follows:

1. **Confidential Computing Infrastructure:** OpenShift clusters are deployed on infrastructure that supports Confidential Computing (e.g., servers with Intel TDX, AMD SEV-SNP, NVIDIA CC technology).
2. **Enclave Protection:** Critical OpenShift components of the control and data plane as well as sensitive application pods are configured to run within 3D encrypted secure enclaves.
3. **enclaive vHSM Integration:** The enclaive vHSM provides secure key and workload identity management services to Kubernetes and container components that require them (e.g, via Kubernetes secrets or sidecar injection).
4. **Access Policy Enforcement:** Policies are established to dictate which workloads run in enclaves and how they interact with the vHSM to access secrets. Optionally, policies at the pod level can be established to prevent a kubernetes admin from executing into pods.

2.1 Two OpenShift Usage Patterns



- **Pattern A — "Confidential Node"**: selected **Kubernetes nodes** run inside a **confidential VM**, while the control plane may or may not be confidential; pods are regular Linux containers on that VM.



- **Pattern B — "Confidential Container"**: selected pods run inside their **own confidential VM** (via Kata/CoCo), while the control plane may or may not be confidential; pods are regular Linux containers on that VM.

2.2 Use Case (Confidential Node): Public Sector Multi-Tenant Kubernetes

Scenario: A national government runs a centralized **Red Hat OpenShift** platform as a modern shared service to host applications for ministries, agencies, and municipalities under common regulations. Each tenant subsystem—typically a single public entity or "**Fachverfahren**"—is self-administered but confined to its own assigned resources, preserving autonomy without sacrificing platform consistency.

Problem: Agencies must isolate and protect data processing to meet legal obligations and handle classified information, yet they don't want the overhead of dedicated clusters per tenant. While OpenShift already enforces separation through MAC, RBAC, namespaces, node pinning, and encryption at rest and in transit, regulators and sovereignty strategies increasingly require **hardware-based confidentiality**, **external key custody**, and protection of **data in use**—ensuring even privileged administrators cannot access sensitive workloads.

Solution: Deploy **OpenShift on Confidential Nodes** combined with **enclaive vHSM** for external key management and attestation. Workloads run inside hardware-backed trusted execution environments, so code and data remain encrypted in memory and opaque to hosts, hypervisors, and platform admins. Keys are issued only to attested workloads that match policy, and key management can operate outside the OpenShift domain to physically and logically separate platform operations from key custody. This augments OpenShift's native hardening with sovereign control over identity, keys, and workload trust—without forcing application rewrites.

Result: The state achieves the **economics of a shared cloud** with the **assurance of physical separation**: strong tenant isolation, verifiable sovereignty, and audit-ready operations—delivered on a single, efficient OpenShift platform.

2.3 Use Case (Confidential Container): Health Sector Multi-Tenant Kubernetes

Scenario: National eHealth services ("**Fachdienste**") run on a **multi-tenant OpenShift**. These services process **electronic patient data** and must uphold **digital sovereignty**: providers want the agility of shared Kubernetes without exposing sensitive data to platform operators or external infrastructure admins.

Problem: Traditional controls—encryption **at rest** and **in transit**—don't stop powerful insiders. In shared clusters, a **kubeadmin** (or host admin) can still gain visibility via node access, debug shells, or side-channel tooling. Regulators require that **administrators must not be able to read or extract patient data** from pods that store or process it. Standard containers and in-cluster secrets don't meet this bar.

Solution: The solution combines **OpenShift with Confidential Containers and external key management with attestation** to protect sensitive Fachdienste end-to-end. Each designated service runs inside a per-pod TEE (micro-VM), so data and code remain encrypted while in use and stay opaque to hosts, hypervisors, and even cluster administrators. An external KMS/vHSM enforces attestation-gated access, releasing keys only to workloads whose measured identity matches policy; secrets never leave the enclave boundary. OpenShift RuntimeClass and admission controls ensure that specified Fachdienste must run confidentially, while non-sensitive services can remain standard to preserve efficiency. Because the model uses confidential runtime classes and familiar CSI/secret interfaces, applications typically require no—or only minimal—changes, accelerating adoption without disrupting existing workflows.

Result: The outcome is a zero-trust, multi-tenant OpenShift platform where designated Fachdienste keep data encrypted in use and demonstrably inaccessible—even to cluster or host administrators. Verifiable attestation artifacts and enforced separation of duties streamline audits and help meet stringent regulatory expectations, including GDPR and BSI guidance. Externalized key custody and enclave boundaries deliver digital sovereignty while preserving developer velocity: teams keep familiar OpenShift workflows and apply confidentiality only where needed, accelerating time-to-value and controlling cost. Per-pod isolation strengthens resilience by shrinking the blast radius and curbing lateral movement, so sensitive eHealth services operate with higher confidence and predictability.

2.4 Side-by-side Comparison

Dimension	OpenShift on Confidential Nodes (node in cVM)	OpenShift with Confidential Containers (pod in cVM)
Security boundary	TEE = **entire worker node**. All pods on that node share the node's TCB.	TEE = **per-pod micro-VM**. Each pod has its own kernel/ guest and isolated TCB, independent from the host/node.
Threat model coverage	Protects workloads from **cloud/ host admin** and **co-tenants on other nodes**. Less isolation between pods on the same node (still standard Linux/container and Mandatory Access Control/ SELinux isolation).	Adds **strong isolation between pods** on the same node (separate kernels). Limits blast radius to a single pod/VM.
Attestation unit	**Per-Node attestation** possible. Pod inherits node's attested state.	**Per-Pod attestation** possible. Pod can gate secrets on its own attestation report.
Key/secrets flow	**Node-level attestation** → node retrieves keys from vHSM; pods consume via standard secrets or CSI.	**Pod-level attestation** → pod retrieves its own keys directly from vHSM; pods consume via sidecar injector.
Workload compatibility	Highest: Works with most Linux containers, sidecars, service meshes, eBPF, device plugins—subject to TEE platform support.	Good but **stricter**: no host-PID/IPC, limited privileged modes, some eBPF/ptrace/debug patterns won't work. Sidecars must also be confidential or co-located in the same micro-VM.
Performance profile	Near node-native. **Cold start** like normal containers. Best **density** and **throughput**.	Slight **startup overhead** (micro-VM boot) and a smaller steady-state tax. Lower **density** per node vs standard containers.

Dimension	OpenShift on Confidential Nodes (node in cVM)	OpenShift with Confidential Containers (pod in cVM)
Scheduling/operations	Treat cVM nodes as a **node pool** (labels/taints). Standard OpenShift day-2 ops: upgrades, MCC, MCO, etc., with cVM-capable images.	**Per-pod** selection (annotations/RuntimeClass). Mixed clusters: some pods confidential, others standard. More knobs per workload.
Failure domains	Node is the isolation/failure unit. Node compromise affects all pods on it.	Pod is the isolation/failure unit. Stronger multi-tenant separation within the same node.
Networking	CNI operates normally. Encapsulation/encryption may rely on node kernel features inside the enclave.	CNIs work, but datapath runs inside the pod's guest kernel; check compatibility for service meshes, eBPF, and SR-IOV.
Storage	Node can enforce **always-encrypted** ephemeral and persistent volumes; CSI drivers run on nodes.	Pod guest handles encryption; CSI/csi-secrets-store must be supported in the micro-VM runtime.
GPU / devices	Pass-through/mediated devices supported if the **TEE+hypervisor** and OpenShift device operators support them.	Device pass-through must be supported **into the micro-VM**.
Observability & debug	Similar to standard: node logs/metrics inside TEE. Host-level introspection restricted (by design).	Pod kernel is separate; **fewer host-side hooks**. Use in-guest agents.
Image supply chain	Node OS and container images measured at node boot; standard image policies for pods.	Pod image + guest measured per run; can bind secrets/ artifacts to **that per-pod measurement**.
Cost & density	Better **cost/Pod** for homogeneous trusted tenants; fewer VMs, higher packing.	Higher cost/Pod (micro-VM per pod), but cost justified for **hard multi-tenancy** or high-sensitivity apps.
Use-case fit	"Lift-and-secure" most workloads; cluster/namespace multi-tenancy where tenants are mutually trusted or contractually bound.	Strong isolation for **zero-trust multi-tenancy**, regulated data, or untrusted third-party code/ plugins per pod.

2.5 When to choose Confidential Nodes (Cluster in cVMs)

Pick this when you want t**he simplest path to "confidential-by-default"** for a whole cluster.

- **Threat model**: You trust teams inside the cluster but **don't trust the cloud provider or its host OS/hypervisor**. You want to reduce your TCB to the node VM boundary.
- **Tenancy**: **Soft multi-tenancy** (one org, multiple app teams). Cross-pod isolation is "good enough."
- **Performance & density**: You need highest density, lowest overhead and fastest pod start times; you have latency-sensitive services, chatty microservices, or high-throughput networking.
- **Hardware/accelerators**: You need easier access to GPUs/DPUs/NIC offloads; passthrough and drivers are typically less fiddly at node scope.
- **Compatibility**: Minimal changes. Works with most CNIs, CSIs, service meshes, eBPF/DPDK (subject to the cVM's kernel policy).
- **Ops simplicity & cost**: Simpler to operate and cheaper (one TEE per node, not per pod). Easier logging/metrics/daemonsets.

2.6 When to choose Confidential Containers (Pods in cVMs)

Pick this for **hard multi-tenancy and strongest per-workload isolation**.

- **Threat model**: You must **isolate tenants from each other and from the node kernel**. Even if a node is compromised, a pod's TEE still protects its memory/state.
- **Tenancy**: **Hard multi-tenancy** (different customers/tenants sharing nodes), marketplace workloads, or regulated data from distinct controllers.
- **Attestation granularity**: You need per-workload attestation and evidence tied to a specific image/SBOM for audits or zero-trust admission.
- **Supply chain/IP protection**: Each pod measures its userspace, great for model/IP protection, cryptographic key isolation, or bring-your-own-runtime scenarios.
- **Blast radius**: A vulnerability in one pod doesn't expose others; great for running untrusted or third-party code.
- **Trade-offs**: Expect lower density, higher memory/CPU overhead, slower cold-starts (mitigations exist: snapshots, pooling), trickier device passthrough, and extra ops work.

2.7 Quick Decision Matrix

Use the following matrix to identify which solution is right for you.

Factor	Confidential Nodes	Confidential Containers
Isolation strength	Node-level (good)	Pod-level (strongest)
Multi-tenancy	Soft	Hard / Zero-trust

Factor	Confidential Nodes	Confidential Containers
Attestation	Per node	**Per workload (fine-grained)**
Startup time & density	Best	Reasonable
Performance (net/IO/latency)	**Better**	Overhead from VM boundary
GPU / special devices	**Easier**	Slightly more complex
Ops complexity	**Lower**	Higher (tooling/observability)
Cost	**Lower** (linear in number of nodes)	Higher (linear in number of pods)
Compliance/audit needs	Good	**Best** (granular evidence)

3 Deployment Options: From On-Premise to Hybrid and Public Cloud

The flexibility of Red Hat OpenShift, combined with the hardware-rooted security of Confidential Computing and enclaive vHSM, allows for versatile deployment models:

3.1 On-Premise Deployment

For organizations with strict data residency requirements or existing on-premise infrastructure, the solution can be deployed entirely within their own data centers. This involves:

- **Dedicated Hardware**: Utilizing servers equipped with Intel TDX or AMD SEV-SNP capabilities within the organization's private data center.
- **Full Control**: Maintaining complete control over the physical infrastructure, network, and data, while still benefiting from the enhanced security of enclaves and vHSM for workloads.
- **Compliance**: Meeting specific regulatory mandates that necessitate on-premise data processing and storage.

3.2 Hybrid Cloud Deployment

A hybrid cloud approach offers a balance of control and scalability, allowing organizations to leverage both on-premise resources and public cloud environments securely. This model enables:

- **Workload Portability**: Deploying sensitive workloads requiring the highest level of confidentiality within on-premise confidential computing environments, while less sensitive or bursting workloads can utilize public cloud resources.
- **Consistent Security Posture**: Extending the same security principles and controls/policies (Confidential Computing, enclaive vHSM, OpenShift) across both private and public cloud environments, ensuring a unified security framework.
- **Disaster Recovery and Scalability**: Utilizing public cloud for disaster recovery sites or to handle peak loads, without compromising the confidentiality of data in use. enclaive's multi-cloud capabilities further enhance this by providing consistent key management and attestation across diverse cloud providers.

3.3 Public Cloud Deployment (AWS, Azure, GCP, OVH)

For organizations looking to fully leverage the scalability and global reach of public cloud providers while maintaining strong security, the solution can be deployed directly on leading cloud platforms. This involves:

- **Cloud-Native Confidential Computing**: Utilizing the Confidential Computing offerings available from major cloud providers, such as AWS, Azure, and Google Cloud Confidential VMs.
- **Integrated Security**: Seamlessly integrating enclaive vHSM within the chosen public cloud environment to provide secure key management and attestation for workloads running in confidential VMs or containers.
- **Global Reach and Scalability**: Benefiting from the inherent scalability, elasticity, and global infrastructure of public clouds, while ensuring sensitive data remains protected in enclaves, meeting compliance requirements for cloud adoption.

Both deployment options ensure that the core benefits of Confidential Computing and enclaive vHSM – namely, the protection of data in use and secure key management – are consistently applied, regardless of where the workloads reside. This adaptability is crucial for public sector organizations navigating complex IT landscapes and evolving security requirements.

4 Comparison: Standard OpenShift vs. Confidential OpenShift with External vHSM

Category	Standard OpenShift + vHSM	Confidential OpenShift + vHSM
Execution Environment	Runs on traditional VMs, cloud VMs or bare-metal nodes without memory encryption.	Runs inside Confidential VMs (e.g., AMD SEV, Intel TDX), which provide hardware-encrypted memory and isolated execution.
Memory Protection	No protection of memory in use against hypervisor- or infrastructure-level access. Memory can be introspected by privileged users.	Memory is encrypted and inaccessible even to the host OS or hypervisor. Protects against some side-channel attacks and memory scraping.
Workload Isolation	Software implemented isolation only on Kubernetes namespace or cluster level. Container escapes and rogue containers are effectively blocked by Mandatory Access Control/SELinux, but no runtime isolation on hardware level.	Each workload runs inside hardware-isolated VMs.
Secret Management	Kubernetes Secrets stored in etcd (encrypted at rest with master key), potentially exposed if the key management system is compromised or the key is held by an untrusted organisation.	Secrets and cryptographic material managed through enclaive's vHSM, which isolates key material from the host and Kubernetes control plane.
Key Lifecycle Control	Master key is often managed by cloud KMS or by cluster admin.	vHSM performs key generation, signing, encryption - entirely within enclaive. Admins and infrastructure providers cannot extract keys.
Insider Threat Mitigation	Whereas tenant administrators are completely restricted to their own tenant namespaces, Kubernetes and infrastructure admins can access host OS, volumes, logs, and memory.	Confidential computing and vHSM remove trust from infrastructure admins. Only enclave-authorized code can access key material.

Category	Standard OpenShift + vHSM	Confidential OpenShift + vHSM
Compliance Readiness	Well suited for general or critical workloads within a BSI IT-Grundschutz-environment, but may require compensating controls in sensitive sectors with very high security or Digital Sovereignty requirements, where hardware based isolation mechanisms are useful or even mandatory.	Aligns with zero-trust models, "data in use" encryption and “operator exclusion” requirements. Enables even stronger compliance for classified, personal data and patient record handling.
Deployment Complexity	Mature and supported widely. Lower operational complexity. Already covers baseline security requirements.	Requires supported cloud providers and configuration for cVMs. vHSM integration introduces additional operational components.
Performance Impact	Native performance without hardware-level encryption overhead.	Slight 3-8% more CPU cycles overhead due to memory encryption and enclave transitions, typically acceptable for most workloads.
Security Boundaries	Software-enforced boundaries; vulnerable to privilege escalation or kernel exploits.	Hardware-enforced boundaries; even compromised kernel/ hypervisor cannot access enclave-protected workloads.

5 Summary

This integrated solution provides distinct and significant benefits for both infrastructure managers and Kubernetes managers:

5.1 Benefit for Kubernetes Management

- **Enhanced Host OS Security**: In a multi-tenant environment, the risk of a container escaping its boundaries or escalating privileges is a significant concern. With Confidential Computing, even if a container manages to compromise the guest OS or the Kubernetes control plane within its enclave, the underlying host Operating System remains protected and isolated. The enclave's memory and CPU state are encrypted and inaccessible from the host, preventing the compromised container from directly affecting the host OS.
- **Mitigation of Container Escalation**: This architecture provides a strong defense against potential container escalation attacks. The hardware-enforced isolation ensures that malicious code or com promised containers cannot „peep" into or manipulate the host OS, significantly reducing the attack surface and potential for lateral movement within the infrastructure.
- **Reduced Blast Radius**: By containing potential breaches within the confidential enclave, the blast radius of any security incident is severely limited, protecting the broader infrastructure from compromise.

5.2 Benefit for Kubernetes Application Deployment

- **Confidentiality of Workloads and Data**: For the Kubernetes application deployment, a primary concern in multi-tenant environments is ensuring that the data and applications running within their Kubernetes clusters are truly confidential, even from privileged infrastructure administrators. Confidential Computing, coupled with enclaive vHSM, provides **hardware-enforced assurance that infrastructure managers, even with root access to the host, cannot inspect, modify, or exfiltrate sensitive data or secrets** residing within the enclaves.
- **Secure Secrets Management**: The enclaive vHSM ensures that cryptographic keys and application secrets are generated, stored, and used exclusively within the secure enclave. This prevents the infrastructure manager from „peeping" at or accessing these critical secrets, thereby maintaining the integrity and confidentiality of the Kubernetes workloads.
- **Compliance and Trust**: This level of isolation and protection builds a higher degree of trust and helps meet stringent compliance requirements often mandated in the public sector, where the separation of duties and protection against insider threats are paramount. The Kubernetes manager can confidently assure tenants that their data is secure from unauthorized access at all layers.

6 Next Steps

Interested in securing your Kubernetes infrastructure?

- **Request a confidential Kubernetes PoC**
- **Join the Red Hat & enclaive webinar on confidential kubernetes**
- **Visit enclaive.io or redhat.com for more**

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) may represent or impact current enclaive product offerings and practices, which are subject to change without notice, and © does not create any commitments or assurances from enclaive and its affiliates, suppliers, or licensors. enclaive products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of enclaive to its customers are governed by enclaive agreements, and this document is not part of, nor does it modify, any agreement between enclaive and its customers.