

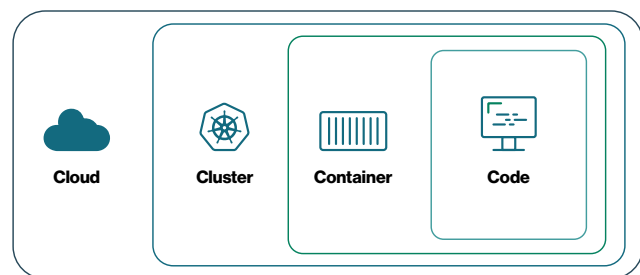
Compliance, Sovereignty, and Trust



Executive overview: security built into the platform

Security for databases running on Kubernetes is not achieved by adding controls after deployment. In this model, security begins with the platform itself. Red Hat® OpenShift® establishes a hardened baseline aligned with public-sector and government-grade expectations. EDB workloads run on top of that already secure, operationally mature foundation.

This combination is deliberate. Platform security and database security work together rather than operating independently. The result is a model designed from the outset for mission-critical and regulated workloads where control, auditability, and consistency matter as much as performance.



The Kubernetes 4C security model

Security teams often describe Kubernetes protection through the industry-standard 4C model:

- Cloud layer
- Cluster layer
- Container layer
- Code layer

This framework provides a clear way to understand how responsibilities are layered across Red Hat OpenShift and EDB.

Responsibilities by layer (EDB + Red Hat)



Cloud layer

This represents the underlying infrastructure where the platform runs. Security remains the responsibility of the cloud provider or the customer's for on-premises environments. Red Hat OpenShift assumes this secured foundation.



Container layer (shared responsibility)

In this layer, EDB follows supply chain and runtime security best practices. Containers use hardened images, are scanned and signed, and run as immutable units. Each container runs single process only (Postgres), limiting the attack surface and removing unnecessary services.



Cluster layer (OpenShift)

Red Hat OpenShift enforces Role-Based Access Control (RBAC), Security Context Constraints (SCCs), admission controls, and certified operator requirements. These policies apply consistently across hybrid environments. Databases deployed through the EDB Operator run within these guardrails without exception paths.



Code layer (EDB)

Secure development practices, shift-left testing, and least-privilege design govern the database software and the operator. Both are designed to run as non-root and avoid elevated privileges.

This foundation is extended through robust credential and key management. The EDB operator is engineered to integrate with external vaults or Red Hat OpenShift's native secret manager, ensuring database creds and encryption keys are handled securely outside the database instance itself. Furthermore, capabilities such as Transparent Data Encryption (TDE) ensure keys are never stored alongside the data they protect, which is essential for strong security posture.

These layers overlap intentionally, creating a defense in depth strategy, rather than relying on a single control point.



Operational security: how risk is reduced day to day

Secure defaults and least privilege

Deployments begin from secure-by-default configurations. Containers run with restricted privileges, don't include unnecessary services, and reduce runtime isolation on the available attack surface from the outset.

Operator and operand image security

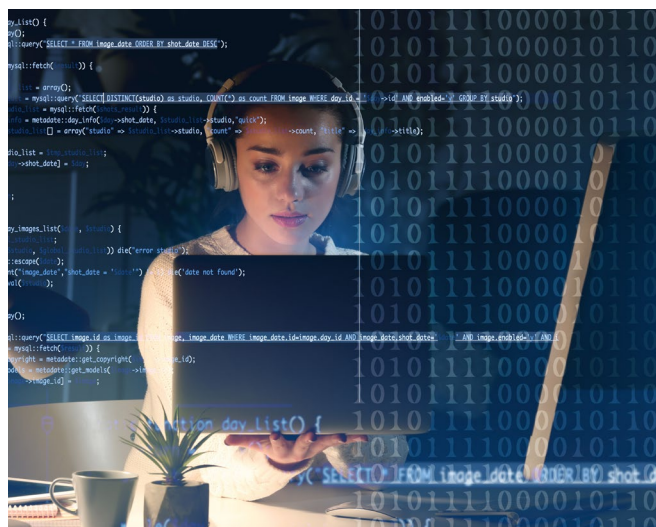
There is a clear separation between:

- Operator images (control plane responsibilities)
- Operand images (Postgres runtime)

Both image types are:

- Built on Red Hat Universal Base Images (UBI)
- Scanned, signed, and verified
- Delivered with SBOMs and provenance data

UBI provides a minimal, well-understood base with fewer packages, a smaller attack surface, and extended Red Hat supportability inside the running container.



Secure supply chain and patching

Images move through controlled build pipelines governed by policy as code requirements, incorporating enforced continuous security scanning and gating at various stages. This ensures that security checks are non-negotiable and prevent the deployment of non-compliant code. Signed images and digest-based references mean unverified components won't reach production. There is also simplified patching through CI/CD and GitOps workflows present in applications on Red Hat OpenShift. This reduces configuration drift and human error.

These practices are in line with emerging regulations like the EU Cyber Resilience Act and modern supply chain security best practices.

Data security & sovereignty

Encryption and confidentiality

EDB Postgres® supports encryption in transit by default with TLS – with mutual TLS optional – and encryption at rest through Transparent Data Encryption (TDE). Teams can also operate containers with no local log persistence. This reduces insider risk and accidental data exposure.

Data sovereignty and residency

Red Hat OpenShift clusters are region-bound by design, so databases deployed within those clusters have the exact same residency guarantees. Workloads can run on-premises, and in private or public cloud without forcing data into hyperscaler-managed database services.

Data sovereignty becomes a matter of architectural control and not a compliance checkbox, which serves to reinforce long-term trust in how information is handled.



Compliance and audit readiness

Red Hat OpenShift hardened baselines align with standards such as NIST 800-53. Databases running on Red Hat OpenShift inherit this posture without needing to adhere to parallel compliance frameworks.

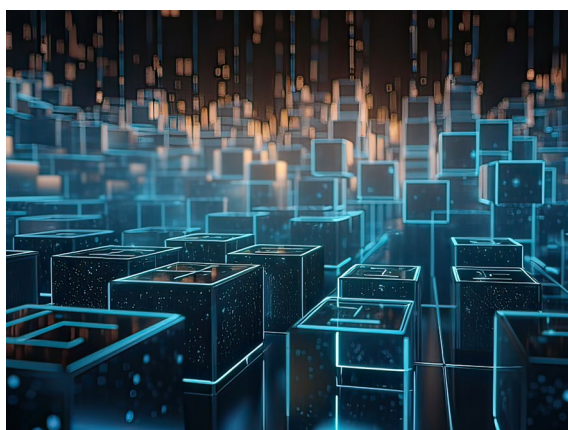
Infrastructure-as-code and declarative configuration provide repeatable controls, consistent audit trails, and simplified evidence collection. Audit teams can trace how environments are configured without relying on manual documentation. This reduces audit effort and associated cost.



Why security-conscious buyers trust EDB on Red Hat OpenShift

Security is part of how this platform works, not something added afterwards. Red Hat OpenShift enforces the controls at cluster level, and EDB ensures Postgres operates within those controls using hardened images, least-privilege execution, and verified build processes.

Responsibilities are clear. Red Hat OpenShift governs access, policies, and workload isolation. EDB focuses on how the database runs inside that framework. This alignment, combined with EDB's deep experience with Postgres, results in a security posture that fits naturally into regulated environments where consistency, accountability, and data control are essential.



Built for control, clarity, and compliance

EDB Postgres® on Red Hat OpenShift operates within a security model that is layered, deliberate, and consistent from infrastructure through to database runtime. OpenShift provides the cluster controls, policy enforcement, and workload isolation. EDB ensures Postgres runs inside that model using hardened images, least-privilege execution, and a verifiable software supply chain.

Data remains under clear organisational control. Encryption, residency, and portability are architectural characteristics, not afterthoughts. Administrative teams can manage systems without unnecessary data exposure, and audit teams can trace how environments are built and maintained through repeatable, versioned processes.

This combination of platform security and database security produces a posture that fits naturally into environments where compliance, sovereignty, and operational trust are non-negotiable.

Next steps

Explore how EDB and Red Hat work together to deliver an enterprise-grade Postgres foundation for OpenShift.

[Get in touch with EDB today](#) for more information.



www.enterprisedb.com

About EDB Postgres AI

EDB Postgres AI is the first open, enterprise-grade sovereign data and AI platform, with a secure, compliant, and fully scalable environment, on premises and across clouds. Supported by a global partner network, EDB Postgres AI unifies transactional, analytical, and AI workloads, enabling organizations to operationalize their data and LLMs where, when, and how they need them.

© EnterpriseDB Corporation 2026. All rights reserved

[List of marks] is/are [a] trademark[s] of Red Hat, LLC, registered in the United States and other countries.

