**🔐 QuantumStream Data Encryption – Security Configuration Guide**

**📦 Feature:** Built-in AES-256 Encryption\ **🛠️ Applicable To:** All QuantumStream Projects (v2.3.0 and above)\ **📅 Last Updated:** July 2025\ **🎯 Objective:** *"Protect sensitive data in motion and at rest with enterprise-grade encryption."*

**🧠 Overview**

QuantumStream offers **native support for AES-256 encryption**, enabling secure handling of sensitive data across all stages of the streaming pipeline. This feature ensures compliance with industry standards and protects against unauthorized access, both in transit and at rest.

Encryption can be enabled at the project level through the configuration file, allowing teams to enforce security policies without modifying application logic.

**🔐 Encryption Standard**

* **Algorithm:** AES-256 (Advanced Encryption Standard, 256-bit key)
* **Mode:** GCM (Galois/Counter Mode) for authenticated encryption
* **Scope:** Applies to stream payloads, configuration secrets, and checkpoint data
* **Compliance:** Meets requirements for GDPR, HIPAA, and SOC 2

**⚙️ How to Enable Encryption**

To activate encryption in your QuantumStream project:

1. Open your project’s configuration file (qs.config.yaml).
2. Add or update the following section:

encryption:

enabled: true

key: "<your-256-bit-base64-key>"

1. Save the file and redeploy your stream using:

qs deploy

**🔑 Key Management Guidelines**

* The encryption key must be a **256-bit key**, encoded in **Base64**.
* Keys should be stored securely using a **Key Management System (KMS)** or environment variables.
* Never hard-code keys in version-controlled files.
* Rotate keys periodically and update the configuration accordingly.

Example of generating a secure key:

openssl rand -base64 32

**🧪 Validation & Diagnostics**

After enabling encryption:

* Run qs validate to ensure the configuration is correct.
* Use qs diag to confirm encryption is active and functioning.
* Monitor for any decryption errors or performance impacts via qs monitor.

**📊 Performance Considerations**

* Encryption introduces minimal latency (<5ms per record) due to hardware-accelerated AES support.
* For high-throughput environments, ensure CPU resources are sufficient to handle encryption overhead.
* Use the qs monitor command to track encryption-related metrics.

**🧭 FAQs**

**Q: What happens if the encryption key is invalid or missing?**\ A: The stream will fail to initialize, and an error will be logged during validation.

**Q: Can I disable encryption after enabling it?**\ A: Yes, but doing so will prevent decryption of previously encrypted data unless a fallback mechanism is implemented.

**Q: Is encryption applied to logs or metrics?**\ A: No. Logs and metrics are not encrypted by default. Use secure logging practices and encrypted transport layers (e.g., TLS).

**Q: Can I use a cloud KMS instead of a static key?**\ A: Yes. QuantumStream supports integration with AWS KMS, Azure Key Vault, and GCP Secret Manager via plugin modules.

QuantumStream’s encryption capabilities ensure your data remains secure, compliant, and resilient—no matter where it flows.