**🧪 QuantumStream CLI – Full Usage Guide**

**📦 Tool Name:** QuantumStream CLI\ **🛠️ Version:** v2.3.1\ **📅 Last Updated:** July 2025\ **🎯 Purpose:** *"Command-line control for stream management, configuration, and diagnostics in QuantumStream environments."*

**🧠 Overview**

The **QuantumStream CLI** is a powerful command-line interface designed to help developers, system administrators, and data engineers manage streaming data pipelines with precision and efficiency. It provides a suite of commands for initializing projects, configuring stream parameters, monitoring real-time metrics, and diagnosing system health.

Whether you're deploying a new stream, tuning performance, or troubleshooting issues, the CLI offers a streamlined, scriptable interface for full lifecycle management.

**⚙️ Installation**

To install the CLI:

pip install quantumstream-cli

Or download the binary from the QuantumStream Developer Portal.

**🚀 Quick Start**

Initialize a new QuantumStream project:

qs init

This command sets up the project directory, generates a default configuration file (qs.config.yaml), and prompts for stream parameters such as source, sink, and transformation logic.

**📋 Core Commands**

| **Command** | **Description** |
| --- | --- |
| qs init | Initializes a new QuantumStream project |
| qs config set <key> <value> | Updates a configuration parameter |
| qs config view | Displays the current configuration |
| qs deploy | Deploys the stream to the configured environment |
| qs monitor | Launches a real-time metrics dashboard |
| qs logs | Streams live logs from the active pipeline |
| qs validate | Checks configuration and stream logic for errors |
| qs stop | Gracefully stops a running stream |
| qs status | Displays the current status of the stream |
| qs diag | Runs a diagnostics scan and outputs a health report |

**📊 Monitoring & Diagnostics**

Use the qs monitor command to view real-time metrics such as:

* Throughput (records/sec)
* Latency (ms)
* Error rate (%)
* Memory and CPU usage
* Backpressure indicators

For troubleshooting, qs diag provides a detailed report including:

* Stream health status
* Configuration mismatches
* Node connectivity
* Resource bottlenecks

**🧩 Configuration Management**

All stream settings are stored in qs.config.yaml. You can edit this file manually or use CLI commands:

qs config set stream.name user\_activity\_stream

qs config set source.kafka.topic user-events

qs config view

**🧪 Validation & Testing**

Before deployment, validate your setup:

qs validate

This checks for missing fields, schema mismatches, and unsupported transformations.

**📦 Deployment**

Deploy your stream to the cloud or on-prem environment:

qs deploy --env production

Use --dry-run to simulate the deployment without making changes.

**🧭 Help & Documentation**

For help with any command:

qs help

qs help <command>

Or visit the QuantumStream CLI Docs for full documentation, examples, and troubleshooting guides.

**🧭 FAQs**

**Q: Can I use the CLI with multiple environments?**\ A: Yes, you can define multiple environments in your config file and switch using --env.

**Q: Does qs monitor require a GUI?**\ A: No, it runs in the terminal using a TUI (text-based user interface).

**Q: Is the CLI compatible with CI/CD pipelines?**\ A: Absolutely. All commands are scriptable and support non-interactive flags.

The QuantumStream CLI is your command-line gateway to building, managing, and scaling real-time data pipelines with confidence and control.