Solana Watchtower: Project Overview

Purpose

Solana Watchtower is an end-to-end monitoring and alerting toolkit designed for teams running programs on the Solana blockchain. The project aims to make it easy to watch on-chain activity, detect security issues, and notify operators before problems escalate.

Goals

Provide real-time visibility into Solana program activity

Enable customizable security and business rules

Offer multi-channel notifications for rapid response

Supply metrics for performance analysis and capacity planning

Deliver a dashboard for operators and security teams

Core Components

The repository is organized into several Rust crates that work together:

| Crate | Description |

|-------|-------------|

| **subscriber** | Connects to Solana RPC WebSockets and Geyser plugins to stream events. Handles subscription management and filtering. |

| **engine** | Implements the rule engine, evaluating incoming events against built-in or custom rules. Also aggregates metrics and manages alerts. |

| **notifier** | Provides email, Telegram, Slack, and Discord channels with rate limiting. Alerts generated by the engine are dispatched here. |

| **dashboard** | Exposes a web UI for real-time monitoring and alert management. It also serves Prometheus metrics for external scraping. |

| **cli** | Command-line interface used to start Watchtower, validate configuration, run tests, and manage rules. |

These crates are combined in the watchtower binary, allowing the system to be launched with a single command.

Monitoring Flow

Subscriber receives logs and account updates from Solana nodes.

Events are fed into the **Engine** where rule checks and metrics calculations happen.

Detected issues are passed to the **Notifier** which batches and sends alerts.

The **Dashboard** and Prometheus endpoint expose system state for operators.

This design lets you plug in new rules or notification channels without modifying the core logic.

Configuration

Watchtower uses a single TOML configuration file. You can enable or disable built-in rules, adjust thresholds, and specify credentials for notification channels. The example config in configs/watchtower.toml covers all available settings.

Command Highlights

watchtower start – launch monitoring with optional dashboard and metrics ports

watchtower test-notifications – send test messages to all configured channels

watchtower validate-config – verify the syntax and values of a config file

watchtower rules – list built-in rules or test them against sample data

Deployment Options

Local binary – Build with Cargo and run watchtower start

Docker Compose – Use the files in docker/ for containerized deployment. Copy docker/env.example to .env and adjust RPC endpoints and secrets before running docker-compose up -d.

Business Value

Continuous monitoring helps projects detect suspicious transactions, liquidity risks, or performance issues. Integrating with common chat tools shortens the response time during incidents. Because Watchtower is open source under the MIT license, teams can extend it with custom rules or integrate it with existing security pipelines without vendor lock■in.

License