Axon Integration for Watching Kubernetes Events

Overview

This document outlines a comprehensive approach to integrating Axon Framework with Kubernetes to capture and process Kubernetes events in real time. Axon provides Event Sourcing and CQRS capabilities, making it an excellent choice for event-driven architectures. By leveraging Axon Server (or a custom event store), Kubernetes events (such as pod creation, scaling, failures, etc.) can be captured, stored, and processed for monitoring, auditing, or reactive automation.

Kubernetes Events

Real-time cluster activities including pod creation, scaling, and failures

Event Sourcing

Capturing and storing all events as the source of truth

CQRS

Command Query Responsibility Segregation for efficient processing

Architecture

1

Components

- *Kubernetes Cluster: Source of events (Pod, Deployment, Node, etc.)*
- *Axon Server (or custom Event Store): Stores and distributes events*
- *Kubernetes Event Watcher: A service that captures Kubernetes events and forwards them to Axon*
- *Event Processors: Axon-based services that handle events (e.g., projections, sagas, notifications)*

2

Data Flow

1. *Kubernetes API emits events (e.g., PodScheduled, DeploymentScaled)*
2. *Event Watcher (custom service) captures these events*
3. *Events are published as Axon Events (stored in Axon Server)*
4. *Event Processors subscribe and react to events (e.g., triggering alerts, updating read models)*

Implementation Steps

Setting Up Axon

Option 1: *Use Axon Server (standalone or Kubernetes deployment):*

```
helm install axon-server axonframework/axon-server -n axon
```

Option 2: *Use a custom event store (e.g., PostgreSQL with Axon Framework).*

Capturing Kubernetes Events

Option 1: *Custom Kubernetes Event Watcher (Java/Spring + Axon)*

Event Watcher Implementation

Create a custom service that uses the Kubernetes Java client to watch for events and publishes them to Axon Server using EventGateway.

This service acts as the bridge between your Kubernetes cluster and the Axon event-processing pipeline.

Monitoring & Scaling

Axon Dashboard

Monitor event throughput and lag

Kubernetes Health

Ensure the event watcher pod is running

Auto-Scaling

Scale event processors based on load

Proper monitoring ensures your event processing pipeline remains healthy and responsive. Set up alerts for event processing lag and configure horizontal pod autoscaling for the event processors to handle varying loads.

Conclusion

Integrating Axon with Kubernetes events enables a scalable, event-driven architecture with full auditability and real-time reactivity. This setup is ideal for:

Observability

Tracking cluster changes

Automation

Self-healing systems

Audit Logging

Compliance-friendly event storage