MSA 금융권의 Mission Critical Biz에서 Transaction은 ?

Axon Server

Axon Framework란 **EventSourcing, CQRS, DDD 세가지 Concept을 중심으로 어플리케이션을 개발**할 수 있게 도와주는 프레임워크이다.

Axon Framework가 Event Sourcing, CQRS, DDD를 중심으로 애플리케이션을 작성하게 도와주는 프레임워크라면, Axon Server는 그러한 프레임워크를 작동하게 하는 서버이다.

Axon Server는 메세지 라우팅, 이벤트 저장소 등의 역할을 수행한다.

Axon Framework + Axon Server 조합이 필수는 아니다. Axon Server대신 Kafka + NoSQL 등을 사용할 수도 있다. 하지만, Axon Server는 Event Store에 특화되어있으므로, 다른 것으로 대체하기엔 무리가 있을 수 있다.Event Store의 역할은 이벤트를 저장하는 것은 물론이고, 이벤트 전파, 스냅샷 저장등의 특성을 가져야 하기 때문이다.

Axon Server는 다음과 같은 특징이 있다.

- 높은 가용성
- 클러스터 모드를 지원함
- 핸들러가 먹통이 되었더라도 큐에 메세지를 저장
- 서비스간 통신은 gRPC 방식을 사용

Axon Server 띄우기

Axon 서버는 Axon 공식홈페이지(https://axoniq.io/)에서 다운받아 띄울 수도 있지만, docker hub에도 이미지가 등록되어 있기 때문에, docker만 설치되어있다면 아래와 같이 간단하게 실행이 가능하다.

```
$ docker run -d --name axonserver -p 8024:8024 -p 8124:8124 axoniq/axonserver
```

메세지 종류

Axon은 메세지에 종류가 있다. 또한 이러한 메세지는 명시적으로 Label을 가진다.
예를 들어, 애플리케이션의 자바클래스가 Label처럼 발행된다. 이러한 명시적 메세지는 약간의 오버헤드가 발생하지만, 장점이 더욱 많다. 또한, 메세지는 요구되는 의도에 따라 라우팅 패턴이 다르다.

Commands

- 액션을 취할 때(= Client가 명령을 내릴 때) 사용된다.
- 커맨드는 하나의 목적지로 라우팅되고, 응답을 제공할 수 있다.
- 커맨드는 동시성을 피하기 위해 항상 같은 애플리케이션 인스턴스로 전달한다.

Queries

- 액션이 된 정보가 필요할 때 사용된다.
- 전략에 따라 하나 이상의 목적지로 라우팅된다.
- Multiple 인스턴스로 구성했다면, 그 중 하나로만 쿼리가 전달된다.

Events

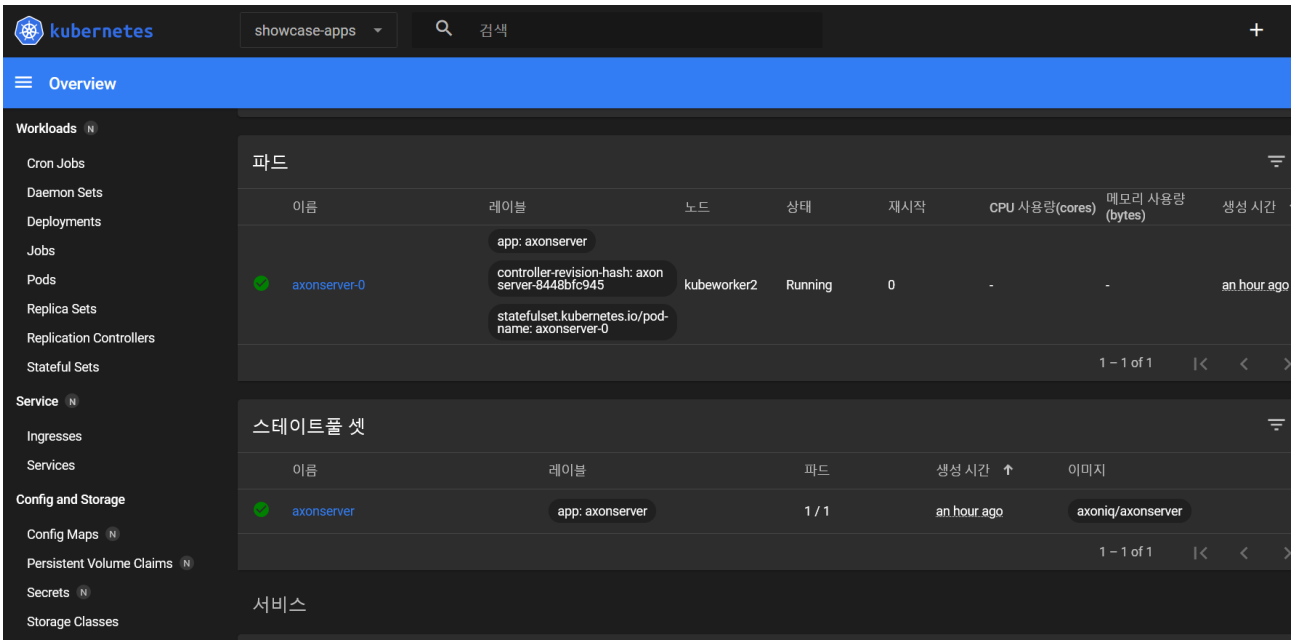
- 발생된 액션을 알릴 때 사용된다.
- 이벤트는 이벤트 저장소에 저장되고, 구독하고 있는 모든 목적지에 라우팅된다.
- 이벤트에 보통 비즈니스 결정사항을 담지 않는다.

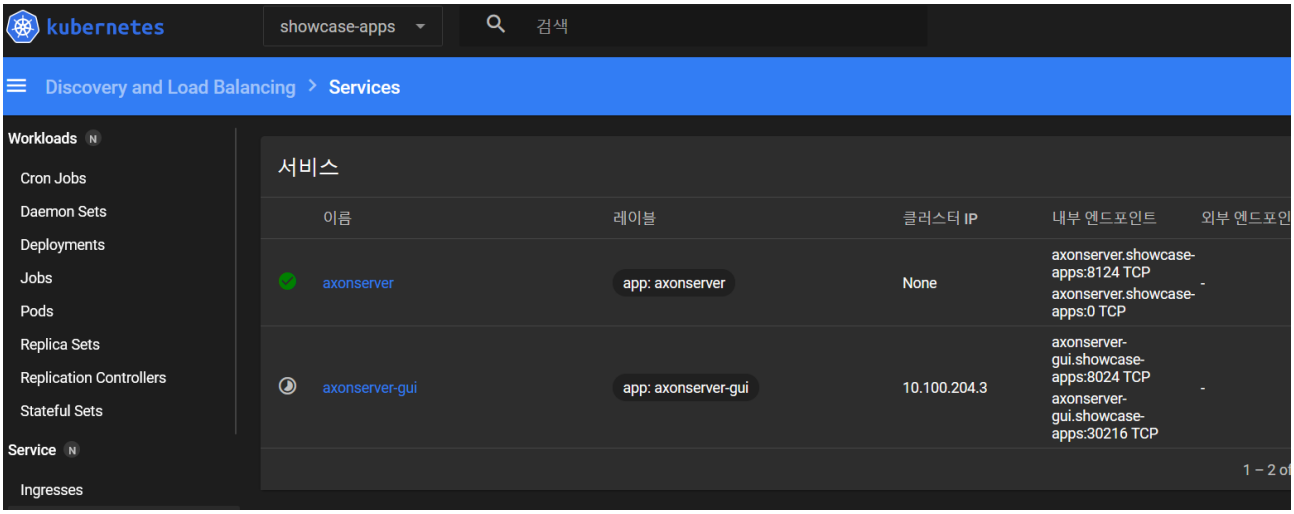
이렇게 명시적으로 분리된 메세지 컴포넌트는 각자 역할에 관심이 없다. 이러한 관심사의 분리는 MSA에도 잘맞는 개념이다.
(Axon 에서는 MSA환경에도 물론 잘맞는다고 하지만, 모놀리스 환경에서도 유용하게 사용될 수 있다고 말한다.)

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: axonserver
  namespace: showcase-apps
  labels:
    app: axonserver
spec:
  serviceName: axonserver
  replicas: 1
  selector:
    matchLabels:
      app: axonserver
  template:
    metadata:
      labels:
        app: axonserver
    spec:
      containers:
      - name: axonserver
        image: axoniq/axonserver
        imagePullPolicy: Always
        ports:
        - name: grpc
          containerPort: 8124
          protocol: TCP
```

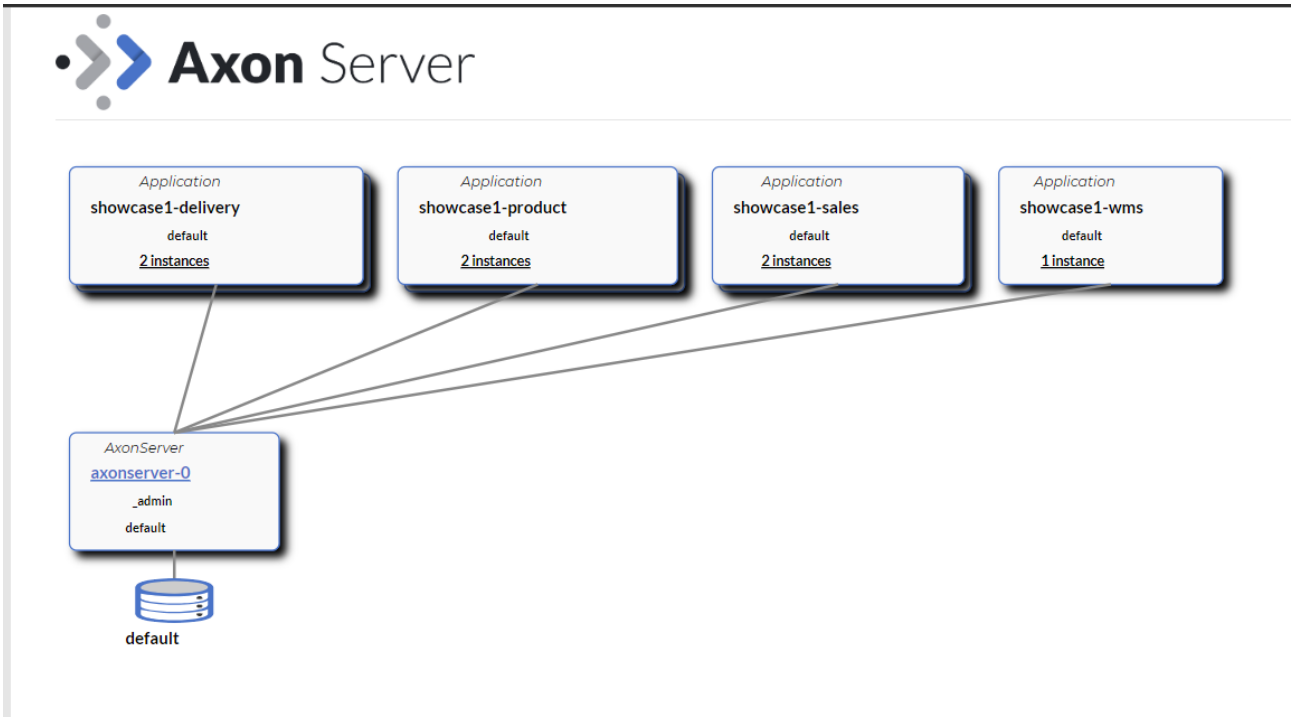
```
        - name: gui
          containerPort: 8024
          protocol: TCP
        readinessProbe:
          httpGet:
            port: 8024
            path: /actuator/health
          initialDelaySeconds: 5
          periodSeconds: 5
          timeoutSeconds: 1
---
apiVersion: v1
kind: Service
metadata:
  name: axonserver-gui
  namespace: showcase-apps
  labels:
    app: axonserver-gui
spec:
  ports:
  - name: gui
    port: 8024
    targetPort: 8024
  selector:
    app: axonserver
  type: LoadBalancer
---
apiVersion: v1
kind: Service
metadata:
  name: axonserver
  namespace: showcase-apps
  labels:
    app: axonserver
spec:
  ports:
  - name: grpc
    port: 8124
    targetPort: 8124
  clusterIP: None
  selector:
    app: axonserver
---
```

http://192.168.0.100:30216/


kubernetes
showcase-apps
검색
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
파드
이름
레이블
노드
상태
재시작
CPU 사용량(cores)
메모리 사용량 (bytes)
생성 시간
axonserver-0
app: axonserver
controller-revision-hash: axonserver-8448bfc945
statefulset.kubernetes.io/pod-name: axonserver-0
kubeworker2
Running
0
-
-
an hour ago
1 – 1 of 1
스테이트풀 셋
이름
레이블
파드
생성 시간
이미지
axonserver
app: axonserver
1 / 1
an hour ago
axoniq/axonserver
1 – 1 of 1
서비스


kubernetes
showcase-apps
검색
Discovery and Load Balancing
Services
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
서비스
이름
레이블
클러스터 IP
내부 엔드포인트
외부 엔드포인
axonserver
app: axonserver
None
axonserver.showcase-apps:8124 TCP
axonserver.showcase-apps:0 TCP
-
axonserver-gui
app: axonserver-gui
10.100.204.3
axonserver-gui.showcase-apps:8024 TCP
axonserver-gui.showcase-apps:30216 TCP
-
1 – 2 of

설치 후 화면



예제를 구성한 모습

각가의 모듈을 1 또는 2개의 Pod로 구성한 모습

Revision #6
★Created 2023-09-15 06:14:24 UTC by Admin
Updated 2023-12-14 08:45:03 UTC by Admin