

JEDx Collector SAM Template – Plain-English Walkthrough

This document explains how each resource in the collector service works and how data flows through the system.

Top-level & Globals

- Uses AWS SAM transform with a default 3s Lambda timeout and JSON logging for consistent observability.

Parameters

- Prefix: prefixes all named resources so you can deploy multiple environments without name conflicts.

Monitoring & Grouping

- ApplicationResourceGroup groups the stack resources; ApplicationInsightsMonitoring enables CloudWatch Application Insights for this group.

Public API (Cognito-protected)

- JedxCollectorApi: API Gateway with CORS and a Cognito user pool authorizer. Exposes endpoints under /collector (e.g., /collector, /collector/record/{senderId}/{object_type}/{RefId}, /collector/records/{senderId}, /collector/login).

API Lambdas

- JedxCollectorApiFunction: Handles ANY /collector. Publishes events to Kinesis (KINESIS_STREAM_NAME) for async processing, and can also perform DynamoDB operations.
- JedxCollectorRecordApiFunction: CRUD/read paths for a specific record, using S3 (artifact storage) and DynamoDB (metadata).
- JedxCarRecordsApiFunction: Lists multiple records for a sender, using S3 and DynamoDB.
- JedxCarUserApiFunction: Handles POST /collector/login using DynamoDB only.

Data Pipeline

- JedxCollectorInputStream: Kinesis stream that receives events from the API handler for downstream validation.

- CollectorValidationFunction: Kinesis consumer. Validates records, writes success artifacts to JedxCollectorBucket, writes failures to JedxCollectorErrorBucket, and records outcomes to JedxCollectorTable (DynamoDB).

Storage

- JedxCollectorBucket: S3 bucket for successful/validated artifacts.
- JedxCollectorErrorBucket: S3 bucket for failed validations or errors.
- JedxCollectorTable: DynamoDB table (pk + sk) used for metadata/state.

Unique flow (Collector vs. CAR)

- Collector API → Kinesis: The collector service ingests via API and pushes directly to the Kinesis stream. The CAR service ingests via S3 object create → Kinesis.

Notes & Suggestions

- Consider DLQ or retries for CollectorValidationFunction if you want to avoid losing transient failures.
- Verify least-privilege on S3/DynamoDB actions, and consider scoping to prefixes where possible.