AI Compliance Portal – Architecture & Implementation Guide

Overview

This document describes a production-ready, fast-to-deploy portal for ingesting compliance documents (e.g., CCPA/GDPR) into an Amazon Bedrock Knowledge Base backed by OpenSearch Serverless vectors, and exposing a grounded Q&A; chatbot with citations plus batch questionnaire answering.

Reference Architecture

Frontend (S3+CloudFront) → API Gateway (HTTP) → Lambda (FastAPI via Mangum). Data in S3 (raw/uploads, exports). Bedrock Knowledge Base (Titan embeddings) with OpenSearch Serverless vector store. Retrieve-and-Generate endpoint enforces grounding and citations.

MVP Scope

• Upload docs → start KB ingestion • Ask grounded questions with citations • Batch answer CSV questionnaires

Security & Compliance

Use KMS for encryption at rest. Configure IAM least privilege for Lambda S3 access and Bedrock APIs. Enable CORS only for your domain. Optionally add Bedrock Guardrails and Comprehend PII redaction for displayed snippets.

Deployment Steps (SAM)

• 1) Create two S3 buckets (RAW and EXPORTS).

• 2) Provision a Bedrock Knowledge Base using OpenSearch Serverless (Vector) and Titan embeddings; add a Data Source for the RAW bucket prefix `uploads/`. Record KB_ID and DATA_SOURCE_ID.

• 3) Deploy the API with SAM using `infrastructure/template-sam.yaml` and set parameters for buckets, KB, Data Source, and ModelArn.

• 4) Host `frontend/index.html` (S3 website or any static server). Paste your API endpoint into the page and test.

• 5) Upload a PDF/DOCX/XLSX and start an ingestion job. When complete, ask questions in the Chat tab.

• 6) For batch mode, upload a CSV to S3 with headers `question_id,question_text`, then run `/batch` and download results.

Key Backend Files

backend/app.py

FastAPI app exposing `/ingest/start`, `/ask`, `/batch` for document ingestion, grounded Q&A;, and questionnaire answering.

backend/bedrock_client.py

Bedrock control and runtime helpers: start ingestion job and Retrieve&Generate; with citations.

backend/s3_utils.py

S3 helpers for uploads, presigning, and reading CSV.

Environment Variables

KB_ID	**Knowledge Base ID (from Bedrock KB)**
DATA_SOURCE_ID	KB Data Source ID for RAW bucket
MODEL_ARN	Foundation model ARN (e.g., Claude 3 Haiku or Amazon Nova Lite/Micro)
RAW_BUCKET	S3 bucket for raw uploads (prefix `uploads/`)
EXPORT_BUCKET	S3 bucket for batch outputs

Prompting & Anti-Hallucination Measures

The Retrieve-and-Generate config enforces: low temperature, grounding-only answers, abstain when unsupported, and returns citations for verifiability. Retrieval uses hybrid search (vector + BM25) with top_k≈8.

Extensions

• Cognito auth • CloudFront+WAF • PII Redaction • Detailed audit logging • RAG evaluations (faithfulness, citation accuracy).