# Use Case: Regulatory Compliance Assistant

## Business Objective

Financial institutions face increasing regulatory oversight from bodies like SEBI, RBI, FATCA, and GDPR. Compliance analysts must constantly parse complex circulars, regulatory updates, and legal notices to ensure policy adherence. This process is manual, time-consuming, and error-prone.

## Solution Overview

The Regulatory Compliance Assistant (RCA) is a LangChain-based application utilizing Retrieval-Augmented Generation (RAG) to automate policy retrieval, clause summarization, and compliance risk identification. It integrates cloud-native services to ensure scalability, security, and accessibility. Analysts can query updates like 'Summarize SEBI circulars from March 2024' or 'List new RBI reporting rules for NBFCs.'

## Architecture Components

|  |  |  |
| --- | --- | --- |
| Layer | Component | Tool |
| Input/API | User Query Endpoint | FastAPI on Azure App Service |
| Data Source | Regulatory Documents (PDF, Web) | Azure Blob Storage |
| Ingestion | Blob Event Trigger | Azure Event Grid + Azure Function |
| Indexing | Document Parsing & Embedding | LangChain + Azure OpenAI + PgVector |
| Retrieval | RAG Chain: Similarity Search | LangChain + Cognitive Search or PgVector |
| LLM Processing | Prompt Template for Compliance Q&A | Azure OpenAI GPT-4 |
| Memory | Query History / Past Citations | Azure Redis Cache / Cosmos DB |
| Output | Summary & Risk Report | Markdown/PDF via API |
| Containerization | Application Packaging | Docker + Azure Container Apps |
| CI/CD | Pipeline Automation | GitHub Actions |

## Workflow Overview

1. Analyst enters a regulatory query via a FastAPI endpoint.
2. Event Grid triggers ingestion upon new document upload to Azure Blob.
3. Azure Function invokes LangChain parser to chunk, embed, and index documents into PgVector or Cognitive Search.
4. RAG chain retrieves relevant passages using semantic similarity.
5. Azure OpenAI processes the result with a predefined Q&A prompt template.
6. Risk/Compliance highlights are extracted.
7. The final output is served via API in structured Markdown or PDF format.
8. Session memory is cached in Redis; long-term knowledge stored in Cosmos DB.

## Deployment Summary

The system is packaged with Docker, deployed on Azure Container Apps, and monitored using Azure Monitor and Application Insights. Azure Event Grid enables real-time ingestion automation, while GitHub Actions manages CI/CD. This architecture ensures security, scalability, and compliance with internal IT policies.