# Use Case: AI-Powered Financial Research Assistant using LangGraph

## Business Objective

Enable investment analysts and finance professionals to rapidly analyze earnings reports, extract performance insights, and generate investment-ready research documents using an AI-driven assistant built on the LangGraph framework.

## Solution Overview

The system leverages LangGraph to orchestrate a multi-step AI pipeline that integrates LLMs, retrieval components, memory systems, and a report generation layer to deliver end-to-end automation for financial analysis. Cloud services such as those offered by Azure are used wherever appropriate to ensure scalability, security, and production readiness.

## Architecture Components

|  |  |  |
| --- | --- | --- |
| Layer | Component | Cloud/Tool Used (optional) |
| Input/API | User Query via REST API | FastAPI on App Service |
| Classification | Query Classifier Node | LangGraph Node |
| Retrieval | Document Retriever Node | Cognitive Search or ChromaDB |
| Summarization | Earnings Call Summarizer Node | LLM via OpenAI |
| Insight Extraction | Risk/Insight Generator Node | LLM via Prompt Engineering |
| Memory Store | Short/Long-Term Memory | Redis Cache & Cosmos DB |
| Report Builder | Report Generator Node | Markdown/PDF Converter |
| Containerization | Workflow Pipeline | Docker + Container Apps |
| CI/CD | Automated Deployment | GitHub Actions |

## Workflow Overview

1. User submits a financial query through an API endpoint.
2. The Query Classifier determines the intent (e.g., earnings summary, risk insight).
3. Relevant financial documents are retrieved using vector search tools.
4. The Summarization Node condenses earnings reports into concise formats.
5. The Insight Node applies reasoning to generate SWOT-style investment insights.
6. A final report is compiled with formatted output.
7. 7. Short-term session memory and long-term company context are managed.
8. 8. The report is returned via API and optionally stored.

## Deployment Stack Summary

The entire system is containerized using Docker and deployed on a cloud platform with support for CI/CD, monitoring, and horizontal scaling. Components such as the API layer, vector DB, LLM, and storage integrate seamlessly into the pipeline to support both development and production environments.