ClarifyAI

Company Document
Insight Assistant

11726 GROUP 2

700771333 - GHATTAMANENI LIKHITHA

700773763
KOMATLAPALLI VENKATA NAGA SRI

700772413 - NIDHIN NINAN

700763677- ROHINI PATTURAJA

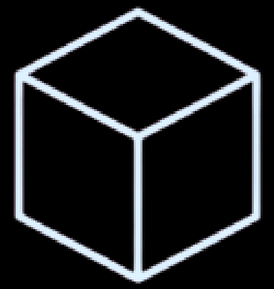
Introduction

- In many companies, employees waste time trying to hunt for the right document or the correct answer. Search tools often give confusing or wrong results.
- ClarifyAI was created to fix this problem by giving clean answers directly from the company's documents.
- It uses a modern RAG system on AWS to make the whole process reliable, secure, and easy to maintain.
- This project focuses on how ClarifyAI works, its architecture, and why it is a better solution than traditional search.

WHY LEGACY ENTERPRISE SYSTEM FAILS ?

Prompt Fragility

Users must guess the “right” keywords. Minor Phrasing changes yield inconsistent or failed results

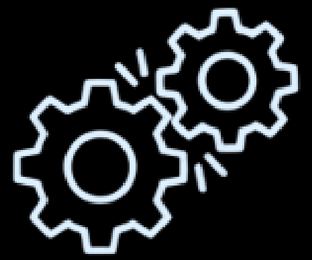


No provenance

Answers are provided without sources, making them impossible to verify and reducing auditability.

Domain Drifts & Hallucinations

General LLMs can invent facts or pull information from non-approve, external sources.



Operational friction

Updating the knowledge vase is often complex and slow, preventing frequent document updates

PROPOSED SYSTEM

ClarifyAI is an internal Question-Answering (Q&A) assistant for organizations that need trustworthy, answers sourced only from their approved documents. It pairs a simple React UI with a retrieval-augmented backend so employees can ask natural questions and immediately see the responses that support each query.

Grounded Responses

Retrieves relevant passages before generation and displays the source for every answer, ensuring transparency.

Secure, Role-Based Access

Differentiates between Admins (upload/sync) and Employees (query only) using Amazon Cognito, with the UI adapting to the user's role.

Low-Ops Serverless Stack

Built on a fully managed AWS stack for rapid deployment, scalability, and minimal operational overhead.

ClarifyAI - Company Document Insight Assistant

- Internal, role-aware question-answering system for organizations with controlled knowledge bases.
- Employees ask natural-language questions and receive precise, citation-backed responses derived only from sanctioned documents.
- Uses a Retrieval-Augmented Generation (RAG) pattern on AWS:
  - Bedrock Knowledge Bases over S3 Vector Store Serverless for managed semantic retrieval.
  - Bedrock foundation models for lightweight answer generation.
- React interface on Amplify, secured with Amazon Cognito, provides:
  - Simple Admin vs Employee access control.
  - Fast deployments and low operational overhead.

Technical Stack

Cloud Services

- Amazon S3
- S3 Vector Store Serverless
- Amazon Bedrock
- AWS Lambda (Python 3.11)
- Amazon API Gateway
- Amazon Cognito
- AWS Amplify Hosting

Frontend Stack

- React (SPA)
- aws-amplify (API + Auth integration)
- @aws-amplify/ui-react (prebuilt auth/UI components)

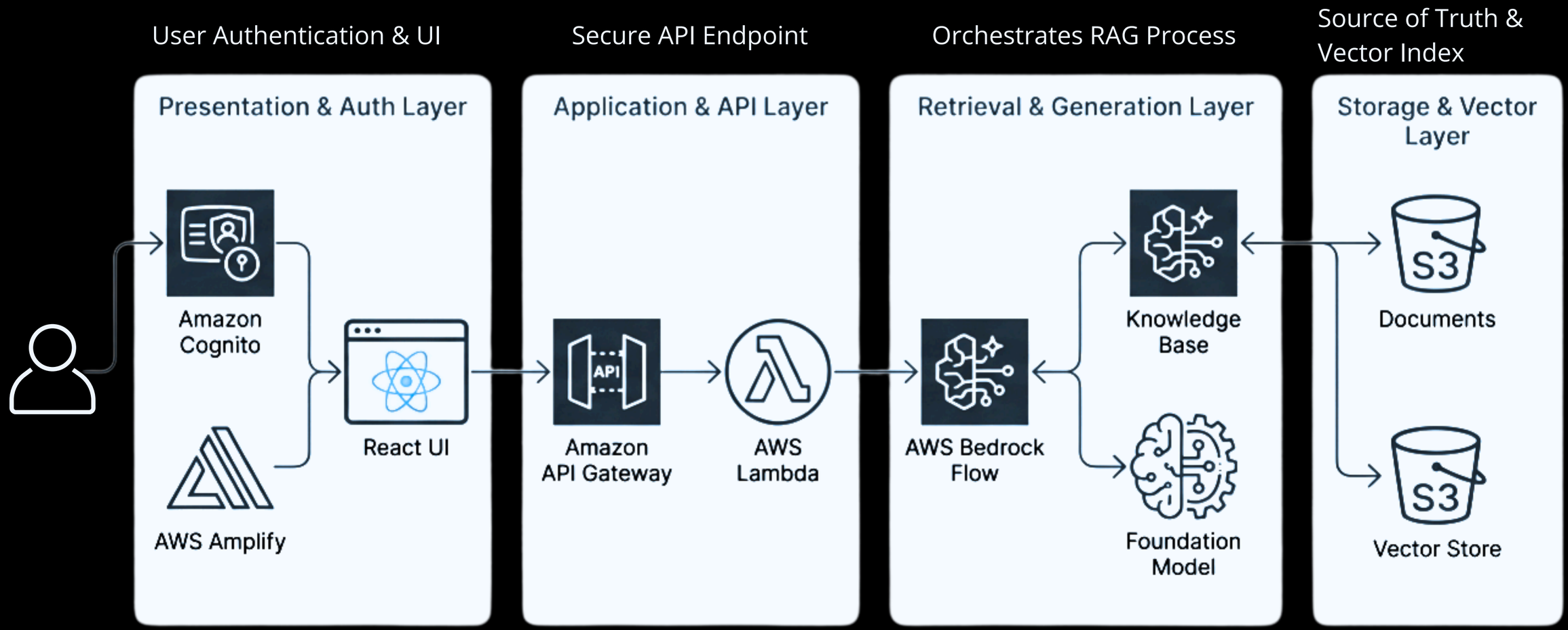
DevOps

- Git repository integrated with Amplify CI/CD
- Environment variables for endpoints, IDs, and secrets configuration

ClarifyAI
Company Document
Insight Assistant

SYSTEM ARCHITECTURE

ClarifyAI follows a layered RAG architecture that separates concerns for robustness and security



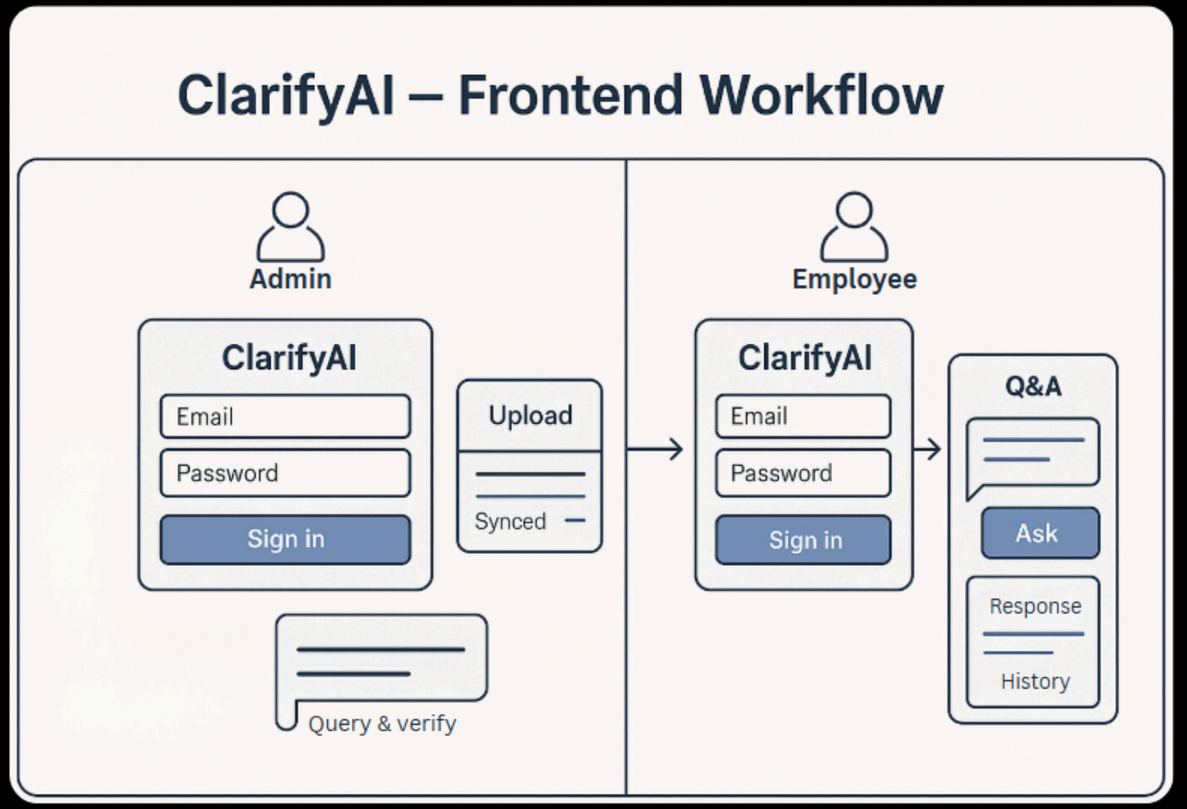
EXECUTION FLOW

Serverless, role-aware RAG assistant that answers only from an approved document corpus in Amazon S3.

Documents are indexed by Bedrock Knowledge Bases into S3 Vector Store Serverless for semantic retrieval.

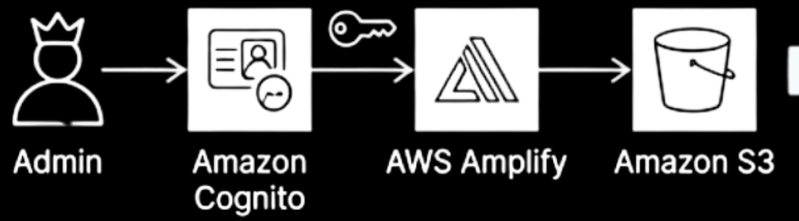
React UI on Amplify → Cognito-authenticated users → secured API (API Gateway → Lambda → Bedrock retrieve-and-generate).

Returns citation-backed, policy-aligned answers with clear admin vs employee access controls and low operational overhead.



THE ADMIN WORKFLOW: SECURELY MANAGING THE KNOWLEDGE CORPUS

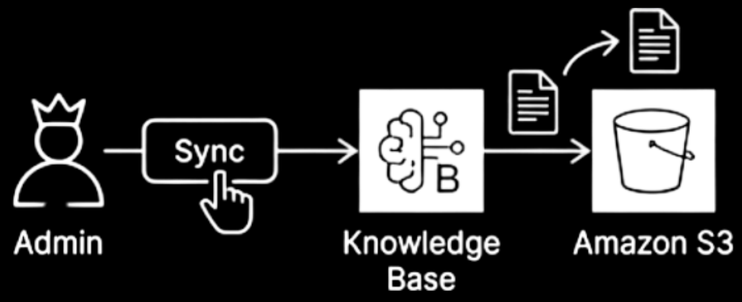
1. AUTHENTICATE & UPLOAD



Admin signs in via the Amplify UI, authenticated by Amazon Cognito. Cognito Identity Pool provides temporary IAM credentials granting S3 write access.

Admin uploads new documents (PDF, DOCX, etc.) through the UI to a designated S3 bucket.

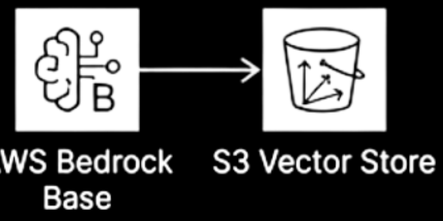
2. SYNCHRONIZE KNOWLEDGE BASE



Admin triggers a 'Sync' operation from the dashboard.

Bedrock Knowledge Base crawls the S3 bucket, chunks tne new documents, and generates embeddings using Amazon Titan.

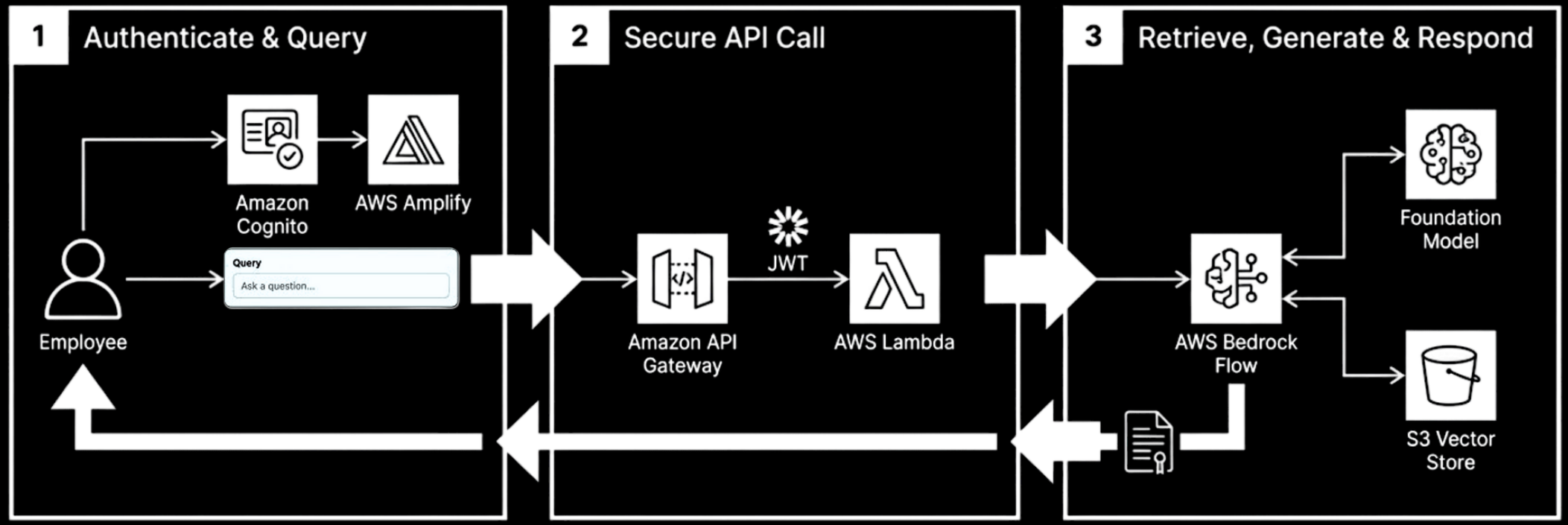
3. INDEX VECTORS



The new vector embeddings are written to the AWS S3 Vector Store, making the new content available for queries.

The admin controls exactly when new content becomes queryable.

THE EMPLOYEE WORKFLOW: QUERYING FOR TRUSTED, VERIFIABLE INSIGHTS



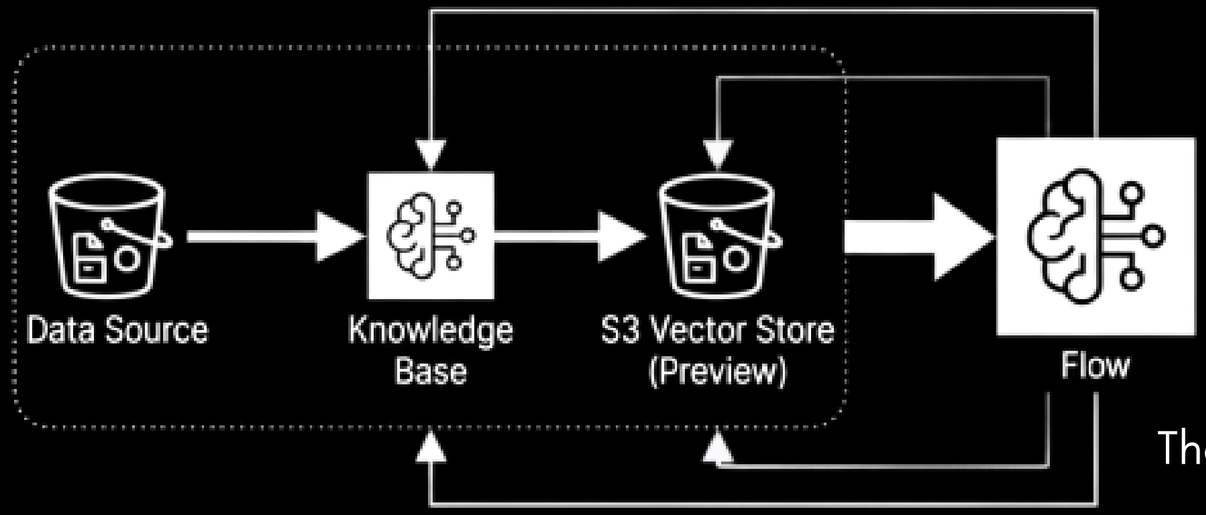
Employee signs in via the Amplify UI, authenticated by Amazon Cognito. The UI presents a simple query interface (no upload functionality). Employee submits a natural-language question.

The React app makes a POST request to API Gateway, including the user's Cognito ID token in the Authorization header. API Gateway validates the token with its Cognito Authorizer before invoking the Lambda function.

Lambda invokes an AWS Bedrock Flow, which orchestrates the RAG process. The flow retrieves the most relevant text chunks from the S3 Vector Store (via the Bedrock Knowledge Base) and provides them as context to a Bedrock Foundation Model to generate an answer. The final response, including the generated text and citations, is returned to the user.

THE TECHNOLOGY CORE: THE RAG ENGINE

Amazon S3 (Data Source) :
The authoritative, encrypted repository for all curated documents. This is the source of truth.

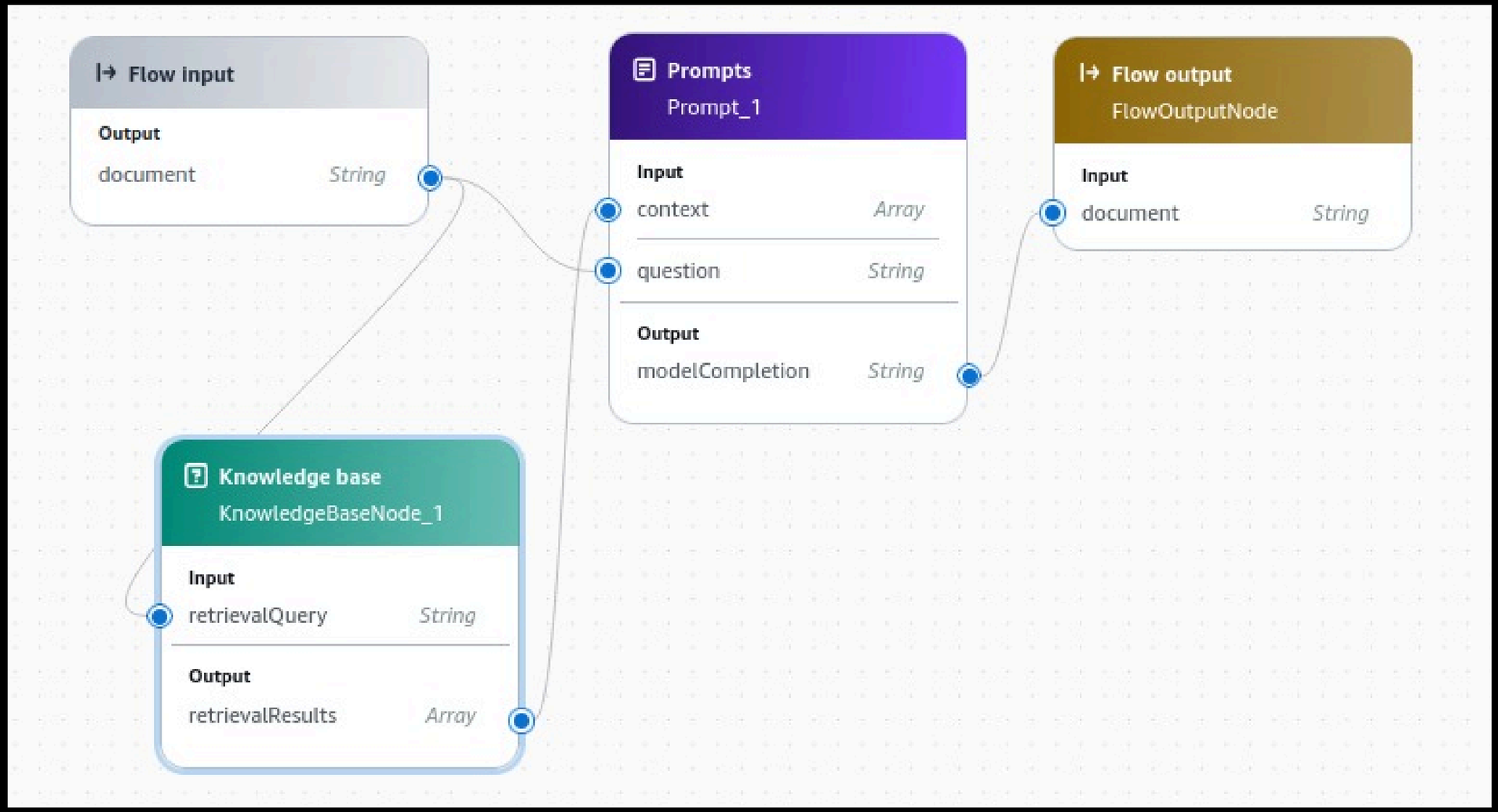


AWS S3 Vector Store :
A managed vector store that indexes document embeddings for efficient semantic search. Chosen for its lower cost and simple integration within the Bedrock ecosystem.

Bedrock Knowledge Base :
The managed ingestion pipeline. It automates document chunking, embedding generation, and connection to the vector store.

AWS Bedrock Flow :
The central orchestrator. It executes the retrieve_and_generate logic, seamlessly combining the retrieval step from the Knowledge Base with the generation step from a Foundation Model to produce grounded answers.

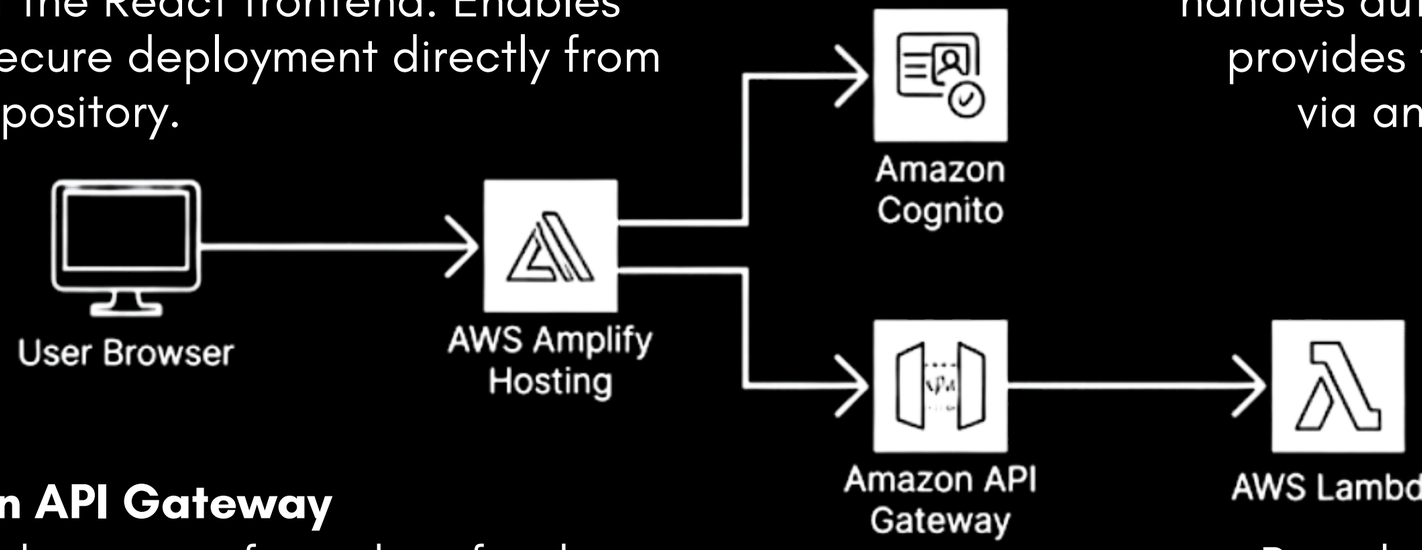
BEDROCK FLOW



THE TECHNOLOGY CORE: THE APPLICATION BACKBONE

AWS Amplify Hosting
Provides a CI/CD pipeline and global CDN for the React frontend. Enables rapid, secure deployment directly from a Git repository.

Amazon Cognito (User & Identity Pools)
A fully managed user directory that handles authentication (who you are) and provides temporary, scoped credentials via an Identity Pool for authorization (what you can do).



Amazon API Gateway
Acts as the secure front door for the application logic. Enforces authentication via a Cognito authorizer, manages CORS, and proxies requests to Lambda.

AWS Lambda (Python 3.11)
Provides the serverless compute layer. Its sole purpose is to receive the validated request, invoke the Bedrock Flow, and format the response, ensuring a minimal and secure attack surface.

A FOUNDATION OF TRUST: SECURITY BY DESIGN

ClarifyAI is built on a principle of least privilege, ensuring every component and user has only the permissions necessary for their role. Access is strictly controlled from the user's login to the data's storage.

Group-Based Access Control

Two Cognito Groups (`Admin`, `User`) clearly define roles. The UI and backend permissions are driven by group membership.

Credential Isolation

Cognito Identity Pools map authenticated users to IAM roles. Only users in the `Admins` group receive temporary credentials allowing `s3:PutObject actions.

API Safeguards

Every call to the`/query` endpoint is protected by an API Gateway Cognito authorizer, which validates the JWT token on every request. Unauthorized calls are rejected with a 401/403 error.

Data Protection

All documents in S3 are encrypted at rest by default. All data is encrypted in transit with TLS. Full audit trails are available via AWS CloudTrail.

IAM ROLES AND POLICIES ENFORCE LEAST PRIVILEGED ACROSS THE STACK

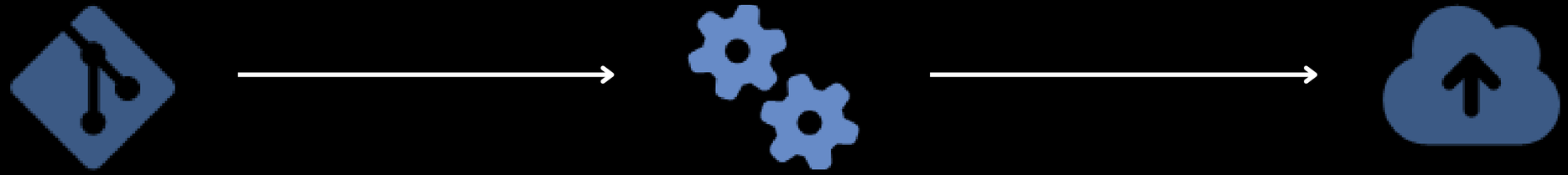
Name	Type	Assumed By	Purpose
Admin	Role	amplify.amazonaws.com	Amplify admin ops
AmazonBedrockExecutionRoleForFlows_WJPB2S22FY	Role	bedrock.amazonaws.com	Runs Bedrock Flows
AmazonBedrockExecutionRoleForKnowledgeBase_6fino	Role	bedrock.amazonaws.com	KB access
amplify-clarifaireactapp-main-a4ac8-authRole-idp	Role	lambda.amazonaws.com	Auth → Lambda
amplify-login-lambda-f6e9f59f	Role	lambda.amazonaws.com	Login Lambda

IAM ROLES AND POLICIES ENFORCE LEAST PRIVILEGED ACROSS THE STACK

Name	Type	Assumed By	Purpose
APIGatewayLog_CloudWatch	Role	apigateway.amazonaws.com	API logs
AWSServiceRoleForAPIGateway	Role	ops.apigateway.amazonaws.com	API Gateway ops
AWSServiceRoleForSSO	Role	sso.amazonaws.com	SSO ops
RAGQueryLambdaExecutionRole	Role	lambda.amazonaws.com	RAG Lambda
us-east-1_PQbe7dk47_Full-access	Role	cognito-identity.amazonaws.com	Full-access users

CI/CD Via Amplify Hosting

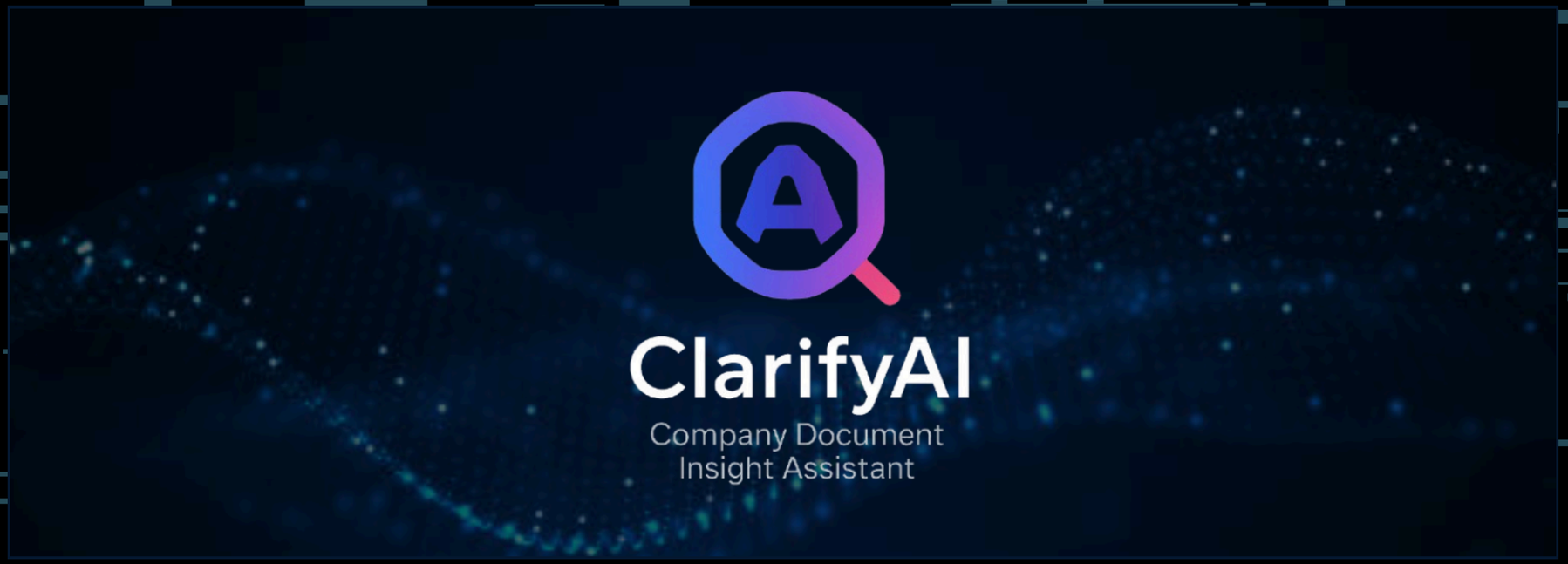
A streamed pipeline from code commit to global deployment, enabling rapid and reliable feature delivery.Zero downtime rollbacks are acheived by simply selecting a previous build in the Amplify console.



Git Commit
Push code changes

Amplify Build
Automated testing and Building

Global Deploy
Deploy to CloudFront



FRONTEND & USER EXPERIENCE

- AUTHENTICATION
- ADMIN
  - UPLOAD, SYNC & VERIFY
- EMPLOYEE
  - QUERY & RESPONSE

OutComes

Project Deployed Domain:

“https://nidhin-dev.d1ktenahlbip4m.amplifyapp.com/”

Github Repository Link :

“https://github.com/nidhinninan/clarifai-react-app/tree/nidhin-dev”

What We Learned

- Multiple AWS services—Bedrock KBs, S3, Vector Store, Lambda, and Amplify—must work together and cannot operate as standalone components.
- Bedrock Flow greatly simplified chaining prompts, context, and retrieval using a clear visual workflow.
- IAM permissions were more complex than expected, requiring detailed roles for each microservice.
- OpenSearch became too costly for long-term vector storage, so switching to S3 Vector Store saved significant cost.
- Amplify made authentication easier by integrating smoothly with Cognito and automatically handling user tokens.

CHALLENGES AND SOLUTIONS:

Challenges	Solution
Getting accurate document retrieval from long or messy files	Used proper chunk sizes and embeddings in S3 Vector Store
Ensuring answers only come from company documents	Enabled Bedrock Knowledge Base grounding with citations
Managing Admin vs Employee access in the RAG pipeline	Used Cognito user groups + IAM least-privilege roles
Connecting frontend queries securely to backend services	Used API Gateway + Lambda with CORS + token validation

Future Roadmap

- AUTOMATED SYNC TRIGGERS ON S3 UPLOAD EVENTS.
- MULTI-LANGUAGE SUPPORT FOR GLOBAL TEAMS.
- FINE-GRAINED DOCUMENT-LEVEL PERMISSIONS.
- INTEGRATED COST-ANOMALY DETECTION AND ALERTS.

Conclusion

- ClarifyAI showcases a secure, serverless RAG architecture that delivers accurate, citation-backed answers from trusted enterprise documents.
- Using AWS Bedrock, S3 Vector Store, Lambda, API Gateway, Cognito, and Amplify, it overcomes issues like hallucinations, prompt fragility, and missing provenance while keeping operations low-cost.
- The project highlighted key cloud insights, including the cost advantage of S3-based vector indexing and the need for precise IAM role design.
- With future enhancements such as multilingual features, expanded file support, analytics, and integrations with Slack or Teams, ClarifyAI can scale into a reliable solution for compliance, onboarding, customer support, and enterprise knowledge management.

ANY QUERIES?

THANK YOU

FOR TAKING THE TIME TO LEARN ABOUT OUR

CLARIFYAI