

Project Tech Stack & Architecture Overview

TPA-Facing Prototype/Wireframe

Please feel free to click around to get a feel. We would need to optimize form intake and set up actual actions but this is our vision for what the front-end looks and feels like from a UX/UI perspective.

<https://kyndly-portal.vercel.app/dashboard>

Tech Stack

Frontend

- **Framework:** Next.js
- **Language:** TypeScript
- **Styling/UI:** Tailwind CSS
- **Deployment:** AWS Amplify

Backend

- **Compute:** AWS Lambda Functions
- **Language:** Python
- **Authentication:** Auth0
- **Notification Service:** AWS Simple Email Service (SES)

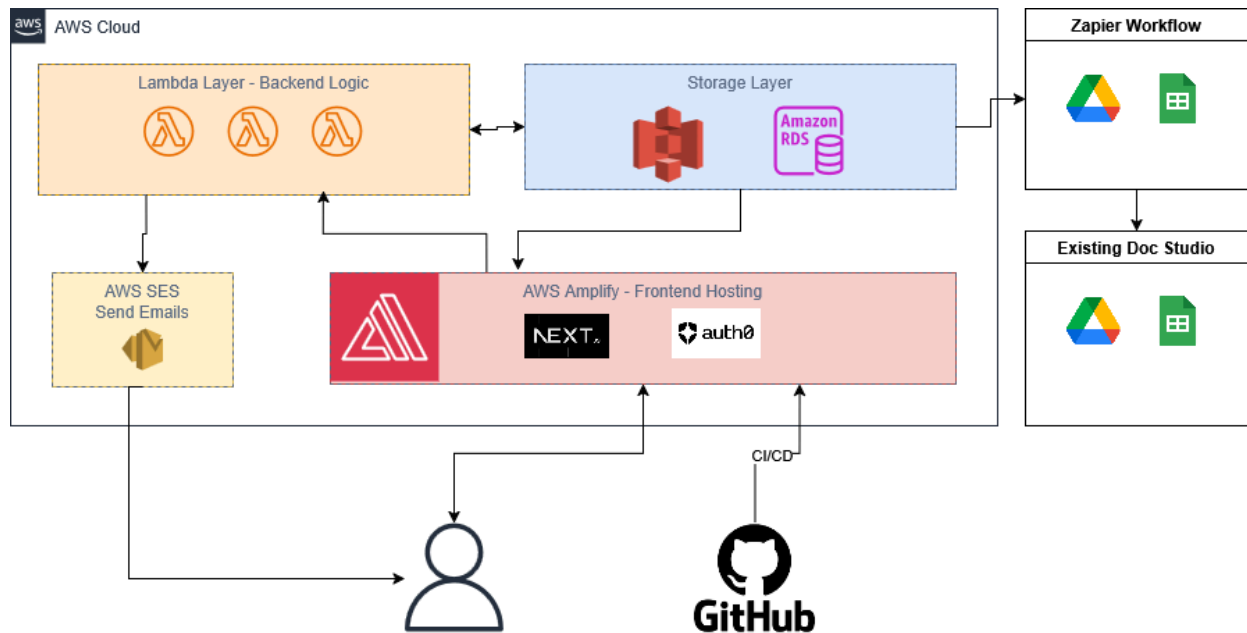
Storage

- **Static Files & Assets:** AWS S3
- **Relational Database:** AWS RDS (PostgreSQL/MySQL based on project requirements)

DevOps & Security

- **Version Control:** GitHub
- **Code Security:** GitHub Security Scans
- **CI/CD:** GitHub Integration with AWS Amplify

Detailed Architecture



Frontend

- Built with Next.js for optimized performance.
- Hosted and deployed via AWS Amplify, utilizing continuous deployment features integrated with GitHub.
- Tailwind CSS provides responsive, consistent UI/UX design.

Backend

- AWS Lambda functions handle business logic, data processing, and integrations.
- Functions written in Python for efficiency and maintainability.
- Auth0 manages secure user authentication and authorization.

- AWS SES sends transactional emails for notifications and user communication.

Data Storage

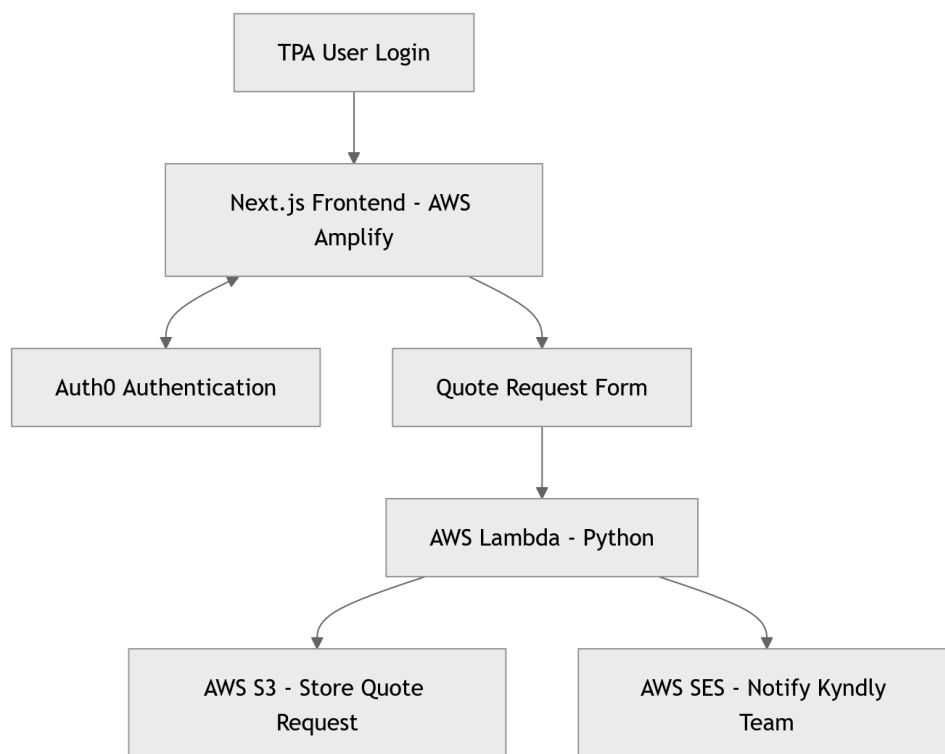
- AWS S3 handles file storage and retrieval, optimized for reliability and scalability.
- AWS RDS manages relational data, supporting complex queries and transactional integrity.

DevOps

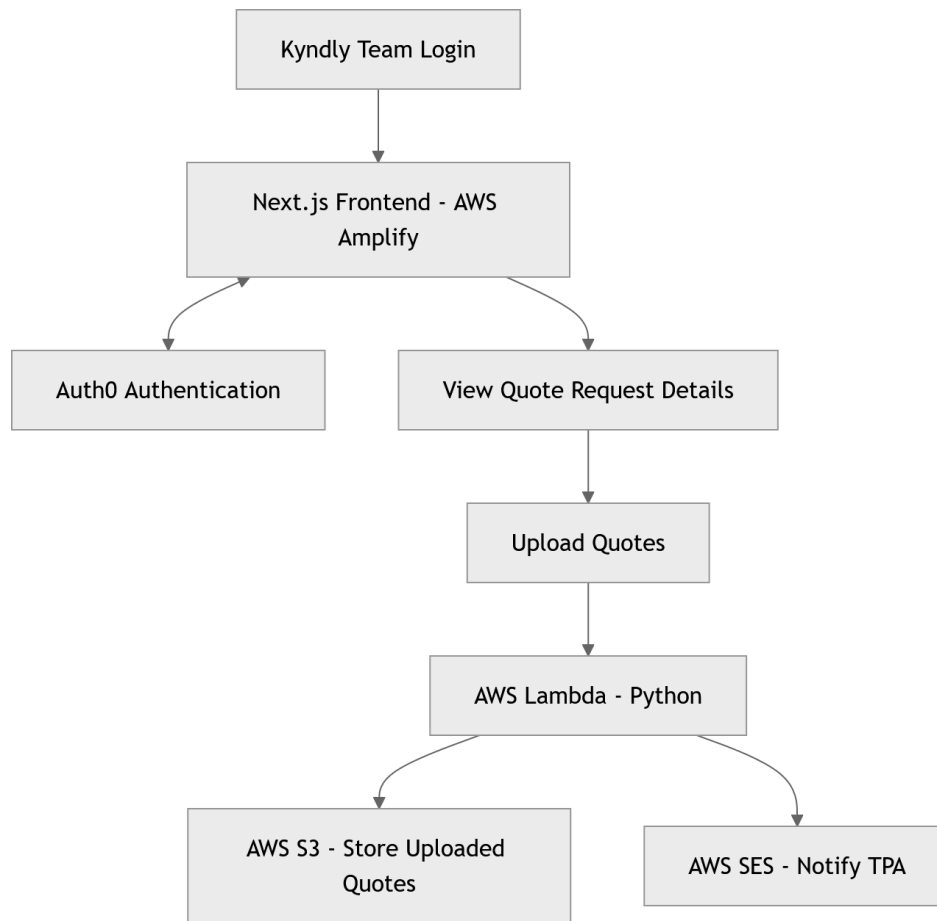
- GitHub is used for version control, code reviews, and collaboration.
- Security scans are integrated into GitHub workflows for code vulnerability assessments.
- Continuous Integration and Deployment (CI/CD) pipeline between GitHub and AWS Amplify ensures smooth deployments.

Data Flow Diagrams

TPA User Data Flow



Kyndly Team Data Flow



Explanation of Data Flows

- **TPA User Flow:**

- TPA logs into frontend.
- Authenticated via Auth0.
- Submits quote request form processed by AWS Lambda.
- Quote request stored in AWS S3.
- Notification email sent to Kyndly team via AWS SES.

- **Kyndly Team Flow:**

- Kyndly team logs into frontend.
- Authenticated via Auth0.

- Views details of quote requests retrieved from AWS S3.
- Uploads quotes through frontend, processed by AWS Lambda.
- Uploaded quotes stored back in AWS S3.
- Notification email sent to TPA via AWS SES.

Integration with Back Office Automation

This system will integrate with existing back office workflows already running in Google Workspace:

- **Existing Workflow Overview:**
 - A Google Form is currently used to trigger DocumentStudio workflows that automate back office processes using submitted responses.
- **Proposed Integration via Zapier:**
 - A **Zapier workflow** will monitor AWS S3 for new file uploads (from TPA form submissions).
 - When a file is detected, it will be **transferred to the designated Google Drive folder** used by the existing Google Form process.
 - A new **row will be written to the associated Google Sheet**, mimicking a form submission.
 - This will **trigger the existing DocumentStudio workflow**, allowing back office automations to continue seamlessly without changes.

This ensures the new AWS-hosted submission process integrates smoothly with the current automation environment, maintaining operational continuity while leveraging scalable cloud infrastructure.

S3 Data Partitioning Strategy

To support secure, scalable, and organized storage of quote submissions from TPAs on behalf of various employers, we will implement a structured S3 key naming and partitioning strategy.

Recommended Key Structure:

```
s3://your-bucket-name/submissions/{tpa_id}/{employer_id}/{submission_id}/file.pdf
```

Partitioning Principles:

- **TPA-Level Partitioning:** Each TPA gets a top-level folder (`tpa_id`), allowing for logical isolation of data by submitting organization.
- **Employer Subfolders:** Within each TPA folder, individual employers have their own subfolders (`employer_id`) to organize submissions.
- **Submission-Level Isolation:** Each submission is stored in its own folder identified by a unique `submission_id`, enabling fine-grained access control, auditing, and lifecycle management.

Benefits:

- **Security:** Enables IAM policies or S3 bucket policies that restrict access by `tpa_id` or `employer_id`.
- **Maintainability:** Keeps files logically grouped and easily traceable by source.
- **Scalability:** Supports millions of files across TPAs and employers without key collisions or performance degradation.
- **Automation-Friendly:** Zapier or Lambda triggers can easily detect and process files based on the key prefix.

AI Chatbot Integration

To enhance user support and engagement, we will implement an AI-powered chatbot within the web application. The solution will be designed for scalability, security, and natural user interaction.

Overview of Implementation

- Chat Interface: Embedded chat UI within the Next.js frontend application.
- Conversational Intelligence: Integration with a conversational AI model (e.g., OpenAI GPT-4) to provide natural language responses.
- Knowledge Base: Bot responses will be grounded in a structured knowledge base covering implementation, onboarding, compliance, and other relevant topics.
- Authentication: All access to the chatbot will be protected via Auth0 to ensure secure interactions based on user identity.
- Backend Handling:
 - Hosted via Next.js API routes or AWS Lambda functions.

- Handles POST requests from the chat UI component.
- Connects to the OpenAI API (or equivalent) to generate responses dynamically.

This chatbot will provide context-aware answers, reduce manual support effort, and serve as a valuable tool for onboarding and ongoing customer education.

Process overview - User Stories:

A TPA user logs into the portal, completes a Quote Request Form (uploading census files and supporting documents). Upon submission, a new row is created in the ICHRA Tracker with fields including Company Name, Date Entered, and Effective Date. Documents are organized in a Google Drive folder hierarchy. Kyndly team receives an email notification with quote metadata. The existing Kyndly system will perform the quote building process and then the quote will be uploaded manually through the admin portal and displayed in the portal for the TPA to share with the broker/company.

Requirement Type	User Story (behavior-driven)	Notes/Rules
UI/UX	As a TPA, I want to submit a quote request with supporting docs so that I can get a proposal generated.	Link to the Transperra Choice Quoting Tool:
Data Capture	Upon submission, the portal should create a new ICHRA Tracker row with captured form fields.	Should capture Company Name, Date entered (dd-Month-yyyy), ICHRA Effective date from proposal submission They are currently manually uploading the quote
Data Capture	Upon submission the portal should create a google drive folder for the client	Will use the existing workflow and trigger it from s3 > Google Drive via Zapier

Notification	An email notification must be triggered to the Kyndly team with quote metadata.	Metadata: <ul style="list-style-type: none"> - Email subject: {Priority}, {submitter name}, "has submitted a company to quote" - First line: {Submitter name} has just submitted a {GLI vs Non-GLI} for {Company Name} click {google drive link} to access the google drive - {Plan Effective date} - {PEPM} - {Target Deductible} - {Current Funding Strategy} - {Broker Name & Email} - {Additional Notes}
Proposal Generation	Handled by Kyndly team using existing processes	Proposal ID is primary key
File Management	As a Kyndly Admin, I want to select a completed proposal from a Google Drive folder and upload it into the portal, attaching it to the correct TPA quote record.	
File Management	As a Kyndly admin i want to be able to upload plan documents for a sold case	
File Management	As Kyndly admin I want to be able to upload knowledge base articles that our TPA's can download and share with their brokers/companies	