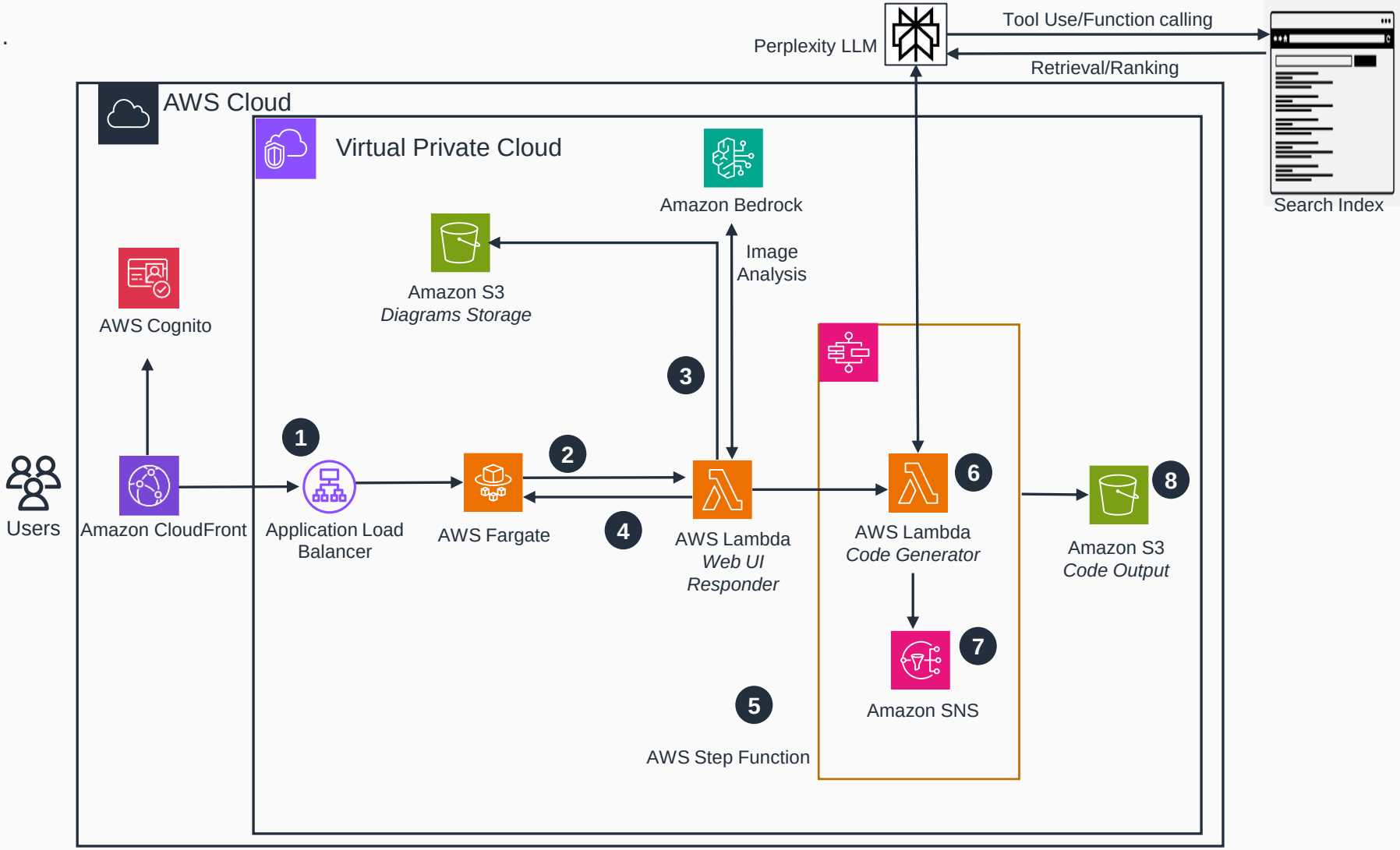
Guidance for Developing Infrastructure-as-Code Templates from Architecture Diagrams on AWS

This architecture diagram illustrates the Architec2Code AI application architecture on AWS. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality.

.



1. Users upload an AWS architecture diagram via a React Web UI hosted on **AWS Fargate**, fronted by **Amazon CloudFront** using **AWS Cognito** authentication, supported by an Application Load Balancer.
2. The uploaded diagram is sent to an **AWS Lambda** function within the Fargate Target group, that is invoked via the WebUI, using path based routing.
3. The Web UI responder Lambda invokes **Amazon Bedrock** API, leverages Claude 3.7 Sonnet, to perform Image analysis. The image is also stored in **Amazon Simple Storage Service (Amazon S3).**
4. A comprehensive summary of the architecture describing the use case, workflow and the different AWS services used is generated and returned to the Web UI.
5. The Web UI Responder invokes an **AWS Step Function.** The Step Function passes the architecture description to an agentic workflow within the Code Generator Lambda function for **AWS CDK** Stack generation..
6. The code generator lambda invokes the Perplexity LLM repeatedly in a sequential manner using pre-engineered ,elaborate prompts to understand individual modules from the image analysis, evolve the CDK Stack and associated IAM roles, policies for each module in parallel. The LLM leverages inbuilt tool use capabilities to access a search index and retrieve the most recent CDK constructs and IAM permissions to generate deployable code.
7. The generated code is stored in an **Amazon S3** bucket.
8. After a successful execution of the workflow, the user receives a **Amazon Simple Notification Service (Amazon SNS)** notification via E-mail.

Reviewed for technical accuracy August 21, 2025

AWS Reference Architecture