

Presentation Coaching Platform — Project Overview

What It Is

A cloud-native platform that accepts audio recordings of presentations, analyzes them through multiple AI evaluation agents, and produces a comprehensive PDF coaching report. The system runs entirely on AWS, using Amazon Bedrock for AI reasoning and a multi-agent architecture built with the Strands Agents SDK.

A user uploads an audio file (mp3, wav, m4a, aac) through a web frontend. The system processes the audio, chunks it, creates vector embeddings, and then passes it to seven specialized evaluation agents — each assessing a different dimension of presentation quality. The output is a structured coaching report with scores, findings, strengths, and improvement suggestions across all dimensions.

How It Works (End to End)

The platform has three processing stages:

Stage 1: Upload and Storage A single-page web app hosted on CloudFront handles authentication via Cognito and file uploads to S3 through presigned URLs. An API Gateway backed by Lambda functions manages submission records in DynamoDB. When a file lands in S3, an event triggers the next stage.

Stage 2: Preparation Workflow An AWS Step Functions state machine orchestrates the preparation pipeline. It validates the file format, transcribes the audio using Amazon Transcribe (with automatic language detection), and optionally chunks the audio into 30-second segments with 5-second overlap and calls Amazon Bedrock (Nova Multimodal Embeddings v2) to create vector embeddings for each chunk. Embeddings are controlled by a feature flag (`embeddings-enabled`) and stored as JSON files in S3 when enabled. The workflow publishes a handoff

message (including the transcript S3 key) to an SQS FIFO queue. This stage typically completes in 1-3 minutes.

Stage 3: Agentic Evaluation An ECS Fargate Spot task picks up the handoff message and orchestrates the evaluation. Seven independent evaluation agents (delivery, structure, executive presence, technical communication, audience engagement, pacing, persuasion) each analyze the presentation through their specific lens. Each agent calls Claude Sonnet via Bedrock to perform its assessment. The results are stored in S3 as JSON, then a Report Generator assembles them into a PDF coaching report using ReportLab.

Architecture at a Glance

Frontend: Static SPA on CloudFront + Cognito authentication Backend API: API Gateway + Lambda + DynamoDB Preparation: Step Functions + Lambda (10 functions) + SQS + Bedrock Embeddings + Amazon Transcribe Evaluation: ECS Fargate Spot + Strands Agents SDK + Bedrock Claude Sonnet Storage: S3 (uploads, chunks, embeddings, evaluation results, PDF reports) Messaging: SQS FIFO queues with dead-letter queues Notifications: SNS error topic + CloudWatch alarms Configuration: SSM Parameter Store CI/CD: AWS CodePipeline + CodeBuild (3 pipeline stacks, 9 total pipelines)

The Seven Evaluation Dimensions

Each dimension has its own agent with a specialized system prompt:

1. Delivery — vocal variety, pace, pauses, filler words, energy, projection
2. Structure — logical flow, transitions, organization, introduction, conclusion
3. Executive Presence — confidence, authority, gravitas, composure, leadership
4. Technical Communication — clarity, terminology, complexity management
5. Audience Engagement — interaction, storytelling, rhetorical questions, attention
6. Pacing — timing, rhythm, speed variation, pauses, segment balance
7. Persuasion — argument strength, evidence, call to action, emotional appeal

Agents are defined in a JSON manifest and can be enabled/disabled at runtime without code changes.

Cost Model

The evaluation compute runs on ECS Fargate Spot (70% discount). Tasks launch on demand when messages arrive and exit after 30 minutes of inactivity. At low volume (5 evaluations/day), the compute cost is approximately \$0.10/month. The dominant cost is Bedrock model invocations (7 agents × multiple calls per evaluation).

Technology Choices

Agent Framework: Strands Agents SDK — provides the multi-agent "Agents as Tools" orchestration pattern. Agents run in local mode (in-process inside the ECS container) making direct Bedrock API calls. The architecture supports future migration to Amazon Bedrock AgentCore for managed deployment, session memory, and per-agent scaling.

Foundation Model: Claude Sonnet 4 via Bedrock (us.anthropic.claude-sonnet-4-6) for evaluation reasoning. Amazon Nova Multimodal Embeddings v2 for audio vectorization.

Compute: ECS Fargate Spot with EventBridge + Lambda trigger chain. Zero idle cost — tasks only run when there's work. Supports concurrent processing of up to 5 submissions simultaneously.

Infrastructure as Code: AWS CDK (Python) for all stacks. Fully automated CI/CD via CodePipeline.

Current State

The infrastructure is complete and deployed. The end-to-end pipeline runs: upload → preparation → handoff → evaluation → report generation → completion. Remaining work is application-level fixes to the evaluation agents' content retrieval (currently using a wrong API for vector store access) and response parsing (the coaching supervisor's agent orchestration returns results in a format the parser doesn't handle). These are documented in remaining-issues.md.

Repository Structure

kiro-agentcore-app/ webapp/ — Frontend SPA (HTML/CSS/JS) upload-service/ — Backend API (Lambda + CDK) preparation-workflow/ — Step Functions pipeline (Lambda + CDK) agentic-

evaluation/ — Evaluation agents (ECS + CDK) cicd/ — CI/CD pipeline definitions (3 CDK stacks)
installations/ — Deployment guides and run instructions documentation/ — Architecture docs,
processing flows, config guides

AWS Services Used

Compute: AWS Lambda — Upload handlers, preparation workflow functions, ECS task launcher
Amazon ECS (Fargate) — Evaluation agent container runtime (Spot capacity) AWS Step
Functions — Preparation workflow orchestration (Standard Workflow)

AI/ML: Amazon Bedrock — Foundation model access (Claude Sonnet 4, Nova Embeddings v2)
Amazon Transcribe — Speech-to-text transcription with automatic language detection

Storage: Amazon S3 — File uploads, audio chunks, embeddings, evaluation results, PDF reports
Amazon DynamoDB — Submission records, processing status tracking

Messaging: Amazon SQS — FIFO queues (preparation input, handoff, dead-letter queues)
Amazon SNS — Error notifications, DLQ threshold alerts

Networking and Content Delivery: Amazon CloudFront — Frontend SPA hosting (CDN) Amazon
API Gateway — REST API for upload service (HTTP API)

Security and Identity: Amazon Cognito — User authentication (hosted UI, JWT tokens) AWS IAM
— Service roles, task roles, execution roles

Configuration and Secrets: AWS Systems Manager — Parameter Store (runtime configuration for
all modules) AWS Secrets Manager — GitHub personal access token for CI/CD

Container Registry: Amazon ECR — Docker image storage for evaluation container

Monitoring and Observability: Amazon CloudWatch — Logs (Lambda, Step Functions, ECS),
Metrics, Alarms Amazon EventBridge — Rules triggering ECS task launch on queue activity

CI/CD: AWS CodePipeline — Pipeline orchestration (9 pipelines across 3 stacks) AWS CodeBuild
— Build and deploy execution (test, Docker build, CDK deploy) AWS CloudFormation —
Infrastructure provisioning (via CDK)

Total: 20 AWS services