

Ritual Roast Coffee Game

Welcome to the **Ritual Roast Coffee Game** – a fun, interactive project built to showcase AWS serverless services while keeping the coffee theme alive.

This project uses **AWS Step Functions**, **Lambda**, and supporting AWS services to simulate a mini coffee shop ordering experience where outcomes are determined by state transitions.

Game Concept

You step into Ritual Roast Coffee, choose a coffee bean and then suggest the best roast method for that bean - see if your ritual delivers the perfect cup... or a spilled disaster.

- **WIN** → Acknowledgement of your barista skills
- **LOSE** → A disappointing revelation of your lack of skills

The randomness and flow are handled by AWS Step Functions and Lambda.

Tech Stack

- **AWS Step Functions** – orchestrates the game logic
 - **AWS Lambda** – backend logic for win/lose state
 - **Amazon S3** – static hosting and distribution of the game web app
 - **JavaScript/HTML/CSS** – lightweight frontend animations
-

Repo Contents

- `/frontend` → HTML, CSS, JavaScript for the Ritual Roast Coffee game UI
- `/state-machine` → AWS Step Functions JSON definition
- `/lambdas` → Sample AWS Lambda functions (Node.js) for handling game states
- `/policies` → sample policies for IAM roles

Repo is delivered a compressed zip file – you must unzip it to access its contents



Getting Started

1. Clone the repo:
 2. `git clone fronten`
 3. `cd ritual-roast-game`
 4. Deploy backend resources:
 - Create the Step Functions state machine using the JSON in `/state-machine`.
 - Deploy Lambda functions from `/lambdas`.
 5. Host the frontend:
 - Upload `/frontend` to an S3 bucket configured for static hosting (or behind CloudFront).
 6. Play the game:
 - Open the game in your browser and click *Order Coffee*.
-



Demo & Tutorial

Watch the full **walkthrough and demo on YouTube:**

[AWS Coffee Game – Ritual Roast Edition](#)



License

This project is for **educational and fun purposes only**.