

AWS Deployment Guide for React + Cognito + Lambda + API Gateway + DynamoDB App

1. Prerequisites

- AWS Account
- Node.js & npm installed
- Vite-based React app ready
- IAM permissions to access Cognito, Lambda, API Gateway, DynamoDB, S3

2. Frontend Hosting on S3 + CloudFront

- Build your React app using: `npm run build`
- Go to S3 > Create Bucket (uncheck Block Public Access)
- Upload contents of `/dist` (from build)
- Enable Static Website Hosting under Properties
- Note endpoint URL

(Optional):

- Create CloudFront distribution for CDN performance

3. Cognito Setup

- Create a User Pool with email-based login
- Create App Client (no client secret)
- Enable Authorization Code Grant with callback to your app
- Set correct callback & sign-out URLs (e.g., `http://localhost:5173` or CloudFront domain)

4. API Gateway + Lambda

- Create a REST API in API Gateway

AWS Deployment Guide for React + Cognito + Lambda + API Gateway + DynamoDB App

- Add /Tasks and /Tasks/{id} resources
- Create methods: GET, POST, PUT, DELETE for both
- Enable CORS on all methods (OPTIONS + headers)
- Integrate each with its Lambda function
- Deploy API to 'prod' stage and note the Invoke URL

5. DynamoDB Setup

- Create a table 'TasksTable' with Partition Key: taskId (String)
- Lambda functions will use boto3 to interact with this table

6. Lambda Configuration

- Write 4 functions: CreateTask, GetTasks, UpdateTask, DeleteTask
- Each should:
 - Parse input from `event`
 - Read/write to DynamoDB
 - Return proper CORS headers and status codes
- Attach proper IAM roles to allow access to DynamoDB

7. Final App Configuration

- React app fetches data from API Gateway
- Each fetch call includes Authorization: Bearer {id_token}
- React uses react-oidc-context for authentication

Make sure:

- All endpoint paths in frontend match API Gateway (e.g., /Tasks/{id})
- CORS headers are returned from every Lambda

AWS Deployment Guide for React + Cognito + Lambda + API Gateway + DynamoDB App

8. Testing and Validation

- Test all CRUD operations in UI
- Check browser console for CORS or auth errors
- Use DynamoDB Console to confirm item creation/deletion