**Project (LogDNA)**

Ample Technology is planning to use a log management tool which manages all the logs coming from different sources and displays the real time metrics and logs on a single dashboard. You are working as a solution Architect and your job is to select one log management tool such as New relic, LogDNA (mezmo) and create one use case to achieve the outcomes. You have to build a POC where you can use the listed services below.

1. Code Build
2. Lambda
3. CloudWatch
4. Github
5. LogDNA (mezmo)

**To-Dos**

* Suggest your approach
* Design your plan
* Implement your plan
* Verify if logs are flowing in the Dashboard real time

**Deliverables**

* Submit your design document and demonstrate you POC.

**Approach**

I have decided to use LogDNA due to being centralized log management solution designed to help DevOps, engineering, and SRE teams collect, aggregate, search, and analyze logs from across applications, infrastructure, containers, Kubernetes environments, and more—all within a single, intuitive platform

What is LogDNA?

LogDNA (now known as Mezmo)

* A Log management SaaS
* Index, Aggregate and Analyze logs
* Easy Integration with Log sources
* Live-tail, search and Parse Logs
* Create Dashboards, graphs & Alerts
* Effective Filters based on Apps & Level
* Centralized log data & Quick setup

A screenshot of a computer

AI-generated content may be incorrect.

**Step Followed**

Pushed Code to Github Source Control

Integrated Github with CodBuild

Added CloudWatch Trigger to AWS Lambda Function

Started CodeBuild with fetches code from Github and build it.

CodeBuild produces logs in CloudWatch and CloudWatch triggers Lambda

AWS Lamda starts log ingestion into LogDNA

**The Design**

A screenshot of a computer

AI-generated content may be incorrect.

**Implementation**

Go to mezmo.com and sign up for a free trial account. This is a log display tool website. It does not connect to AWS.

A screenshot of a computer

AI-generated content may be incorrect.

I copied the following repository where the code from the developer is located to my GitHub account.

<https://github.com/namanjainyr/codebuild_terraform>.

Then click on forks to copy the code to my repository

A screenshot of a computer

AI-generated content may be incorrect.

Go back to Codebuild service and integrate it with github in AWS console.

Once You click on start build, Codebuild will fetch the code from github and build it. This will appear in tail logs in CodeBuild.

A screenshot of a computer

AI-generated content may be incorrect.

CodeBuild Tail logs

A screenshot of a computer program

AI-generated content may be incorrect.

Add the CloudWatch Trigger to AWS Lambda Function

As CloudWatch stores the logs, it triggers the lambda function

AWS Lamda starts log ingestion into LogDNA (mezmo)

A screenshot of a computer

AI-generated content may be incorrect.

But in order for the CodeBuild, CloudWatch, Lambda Function to communicate, I AM Roles needed to be added these services.

1. AWSLambdaBasicExecutionRole
2. CloudWatchReadOnlyAccess

A screenshot of a computer

AI-generated content may be incorrect.

AWS Lamda starts log ingestion into LogDNA with the result below.

**Result**

A screen shot of a computer

AI-generated content may be incorrect.

**Challenges**

Manage account credentials to authorize the use of Github from the Codebuild. Basically, to Connect Codebuild to Github terraform project in the github account. But I have to click on the fork to copy the code to my repository.