

Abgabe verlängert auf Montagabend!:

Ihr Habt die Wahl

Aufgabenblock A:

Ergänze die vorherigen Aufgaben:

- 1) Die POST-Funktion soll die Nachricht in einer Datei speichern, welche einen zufälligen Namen hat.
- 2) Die POST-Funktion soll die genaue URL wiedergeben, die die GET-Funktion triggert.
- 3) Die GET-Funktion soll die gelesene Datei löschen.
- 4) Bitte erstelle eine kleine Zeichnung deiner Lösung und lade Sie mit einer Erläuterung und dem Code der Funktionen in dein Github-Repo.

Abgabe: Screenshot oder Link zum Repo

Optional: die Nachricht soll verschlüsselt werden, bevor Sie gespeichert wird

Optional: Baue die Lösung mit DynamoDB anstelle von S3

Optional: Baue ein kleines Frontend welches auf einem S3 gehostet wird, um die Nachricht einzugeben, sowie den Link auszugeben

Aufgabenblock B:

Wochenprojekt: Baue bis Ende der Woche eine kleine Architektur, welche die Services **API Gateway, Lambda, S3 und optional DynamoDB** kombiniert.

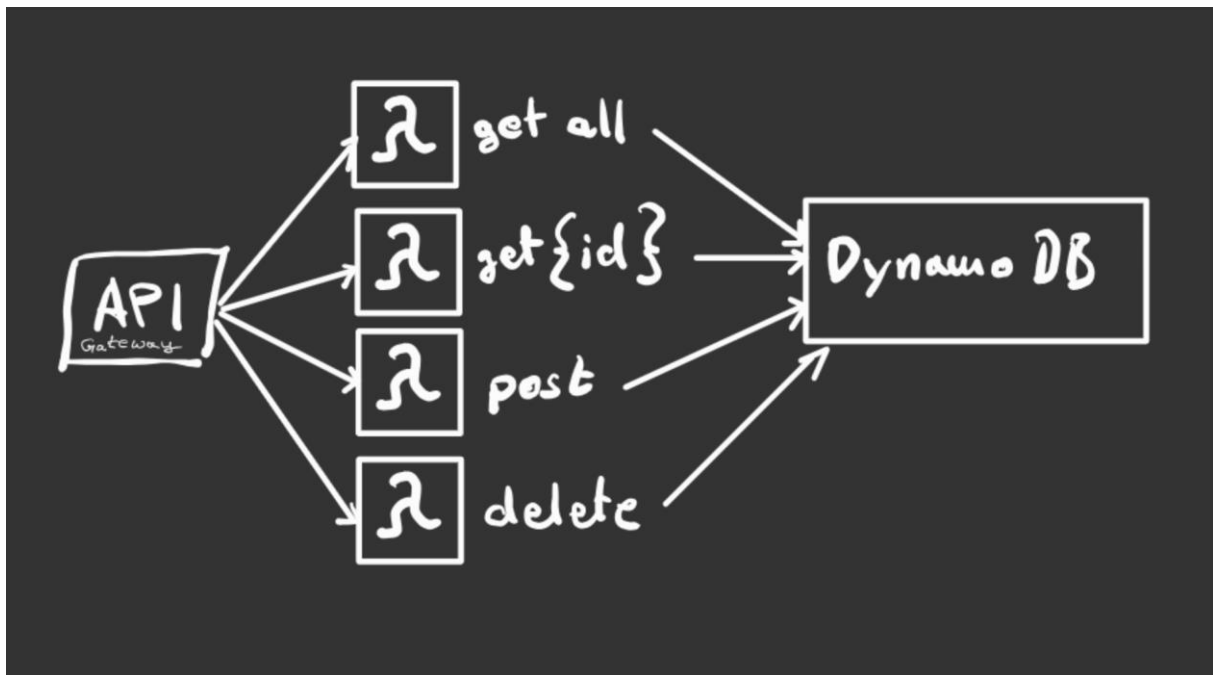
Erstelle hierzu einen Ordner in deinem Github Repo, wo du dein Ergebnis für einen potenziellen Arbeitgeber nachvollziehbar präsentierst (Zeichnung, Code aus Lambda, weitere Erläuterungen).

Zwei Ideen:

Siehe Seite zwei.

1)

Eine Datenbank die User speichert. Über das API Gateway können User hinzugefügt und gelöscht werden. Auch können alle User oder einzelne User mit einem GET ermittelt werden:

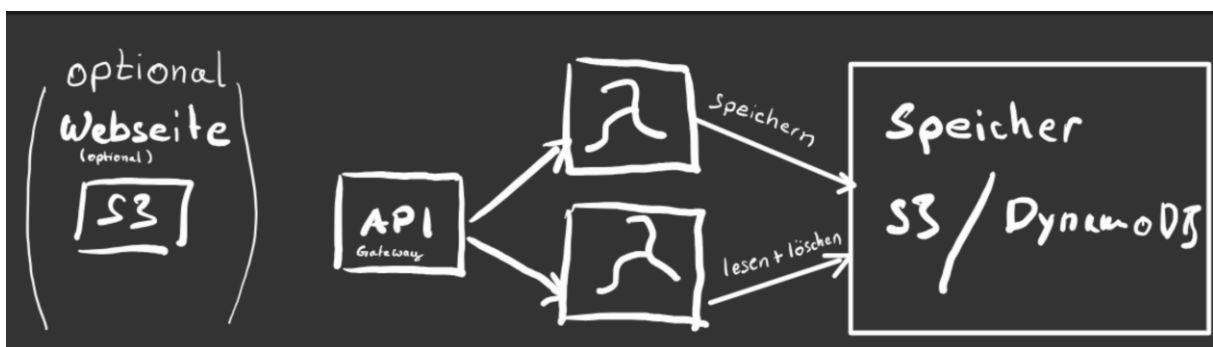


2)

Burn after Read:

Ein API Gateway und zwei Lambdafunktionen. Die erste Speichert eine Nachricht die mit einem POST kam in einen Speicherservice S3/DynamoDB und erzeugt einen Link mit dem sie über die zweite Funktion wieder Abgerufen werden kann. Beim Abrufen soll sie wieder gelöscht werden.

Optional: wer möchte kann gerne noch ein Minimalistisches Front-End Bauen und auf S3 hosten.



Das Projekt ist bis zum Ende der Woche gedacht, bitte gebt als Hausaufgabe ein Kurs Feedback an was Ihr heute Nachmittag gearbeitet habt mit einem Screenshot.