GitInfo-API

1. Voraussetzung

- AWS - Lambda
- AWS – API Gateway
- Python 3

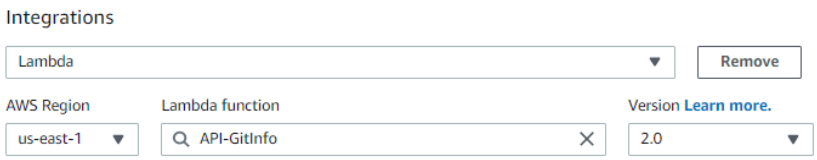
2. Aufgabe

Heutzutage ist es üblich das viele Softwareprojekte auf unterschiedlichen Anbieterplattformen liegen. Die bekanntesten dabei sind GitHub und GitLab.

Die API soll es ermöglichen Informationen über alle Projekte eines Nutzers von beiden Plattformen zu geben.

3. Ablauf Einrichtung / Konfiguration

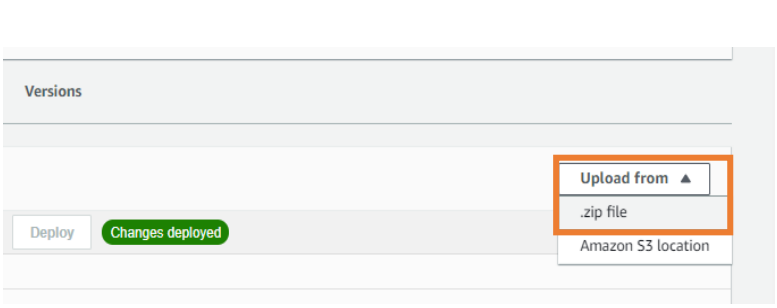
1. Anlegen einer Lambda Funktion
   - Sprache: Python 3.8
   - Rolle: Neue anlegen mit folgenden Berechtigungen
     - S3 Full-Access
     - Lambda Full-Access
     - API-Gateway Full-Access
2. API-Gateway hinzufügen (Create API)
   - HTTP-API > Build
   - Lambda verknüpfen

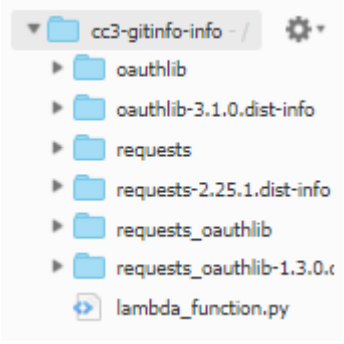


   - Route anpassen
     - Method: GET
     - (optional) Pfad ändern

>>> In der Lambda Funktion sollte nun das API Gateway verlinkt sein

3. libs.zip entpacken und wieder zusammen mit dem Python Skript als ZIP verpacken
4. ZIP Datei nach Lambda hochladen (wenn alles funktioniert hat, sollten das Python Skript und die Libs in Lambda zu sehen sein)





4. REST-Schnittstellen

4.1 Externe (GitHub / GitLab)

GitHub	https://api.github.com/users/
GitLab	https://gitlab.com/api/v4/users/

4.1.1 Nutzerinfos

GitHub - (GET) /users/{username}

Parameter	• username (benötigt) – Nutzername
Dokumentation	https://docs.github.com/en/rest/reference/users#get-a-user

GitLab - (GET) /users/{userid}

Parameter	• userid (benötigt) – ID des Nutzers
Dokumentation	https://docs.gitlab.com/ee/api/users.html#for-user

4.1.2 Repository Infos

GitHub - (GET) /users/{username}/repos

Parameter	• username (benötigt) – Nutzername
Dokumentation	https://docs.github.com/en/rest/reference/repos#list-repositories-for-a-user

GitLab - (GET) /users/{userid}/projects?private=true

Parameter	• userid (benötigt) – ID des Nutzers • private (optional) – Anzeige privater Repos
Dokumentation	https://docs.gitlab.com/ee/api/projects.html#list-user-projects

4.2 Interne (API-Gateway)

(GET) /info

Parameter	• gh_id (benötigt) – GitHub Name • gl_id (benötigt) – GitLab ID • gl_token (benötigt) – GitLab Token

Der GitLab Token kann in GitLab unter Preferences > Access Tokens erstellt werden

5. Ergebnis

Sobald wir ein API-Aufruf senden, bekommen wir folgendes JSON Zusammenfassung der beiden Plattformen zurück:


```
{
   "github":{
      "id":19265916,
      "name":"Kev",
      "alias":"Kr3b5",
      "url":"https://github.com/Kr3b5",
      "repos":[
         {
            "id":142672479,
            "name":"SpringReactStarter",
            "private":false,
            "url":"https://github.com/Kr3b5/SpringReactStarter",
            "forks":0,
            "language":"JavaScript"
         }
      ]
   },
   "gitlab":{
      "id":3180524,
      "name":"Kr3b5",
      "alias":"Kr3b5",
      "url":"https://gitlab.com/Kr3b5",
      "repos":[
         {
            "id":24957853,
            "name":"CC3",
            "private":"true",
            "url":"https://gitlab.com/Kr3b5/cc3",
            "forks":0
         }
      ]
   }
}
```


Falls weitere Infos hinzugefügt werden sollen, kann man die Listen (user-Aufruf / repo-Aufruf) im Skriptkopf nutzen

```
user_keys = ["id", "name", "alias", "url"]
user_github_keys = ["id", "name", "login", "html_url"]
user_gitlab_keys = ["id", "name", "username", "web_url"]

repo_keys = ["id", "name", "private", "url", "forks", "language"]
repo_github_keys = ["id", "name", "private", "html_url", "forks", "language"]
repo_gitlab_keys = ["id", "name", "visibility", "web_url", "forks_count", ""]
```

Diese sind für beide Aufrufe die JSON-Keys, welche man zurück haben will, die welche GitHub verwendet und als drittes welche GitLab nutzt. Falls ein Key nur bei einer Plattform existiert, einfach “” in die Liste eingeben, damit wird es bei der zutreffenden Plattform übersprungen.

6. Stolperfallen

- REST API vs HTTP API

Problem	REST API funktioniert nicht mit Lambda Funktion, wenn man sie in API Gateway verlinkt (Rechtefehler)
Lösung	1. REST API in Lambda Funktion erstellen *oder* 2. HTTP API nutzen

- Libs nicht vorhanden

Problem	Es fehlt die Request-Lib für Python
Lösung	1. Lokal herunterladen mit pip als Archiv 2. Zusammen mit dem Skript als Zip hochladen

- GitHub API-Error

Problem	GitHub gibt keine Daten mehr zurück
Lösung	Es sind nur 60 Request/h erlaubt ➔ Abwarten *oder* ➔ GitHub Key nutzen (5000 R/h)

- Timeout

Problem	Falls Lambda mehr als 3s für den Aufruf braucht gibt es einen Fehler zurück
Lösung	Lambda Timeout in Configuration > General Configuration > Timeout hochsetzen

- Rechte stimmen nicht

Problem	Fehlermeldung wegen fehlender Rechte
Lösung	Der Rolle die in Einrichtung aufgezählten Rechte eintragen