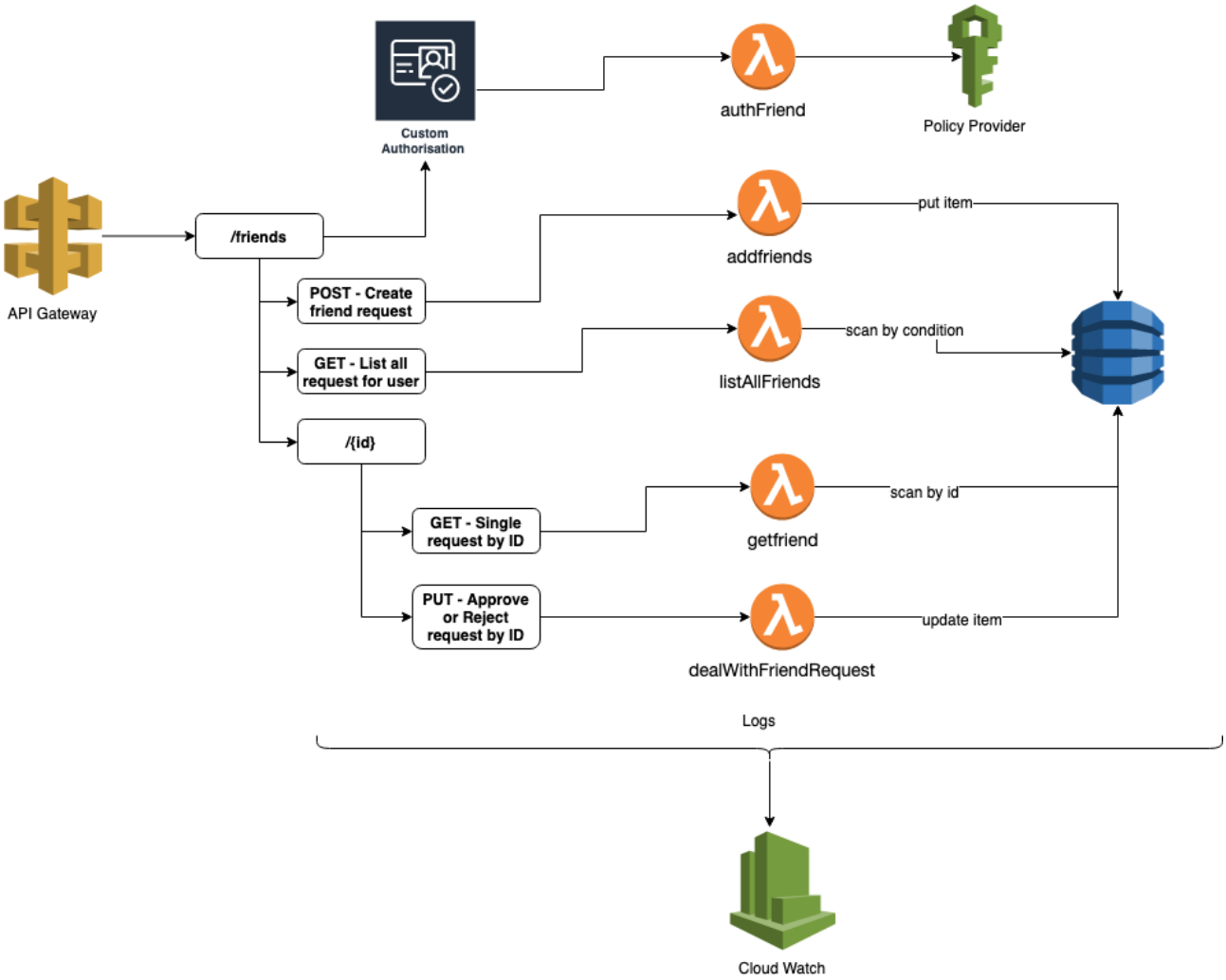
Project Name: DNA Friends App

This application expose rest endpoints for online friends app to create new friend request, allows user to list all friend request and either approve or reject request.

Application build around **serverless** architecture by using **AWS** service such as **API Gateway** (Rest interface), **Lambda** (NodeJS) and **DynamoDB** (storage).

All rest endpoints are secured using **custom authorisation** module which generate appropriate policies for valid users.



Service Description

Create Friend Request	POST https://hoeezamirc.execute-api.eu-west-2.amazonaws.com/development/friends	Header:
		
		Content-Type: application/json
		
		Authorization: allow
		
		Notes: Authorization is a simulation, you can change allow to deny to simulate unauthorization scenario.
			Request Body (Sample):
			
			{
			"sender" : {
			"id" : "ABC999",
			"name" : "Ashely"
			},
			"receiver" : {
			"id" : "ABC34343",
			"name" : "Mick"
			}
			}
			
			Response: 200 OK with
			
			{
			"statusCode": 200,
			"body": "{\"location\": \"friendsdna_ofvftp95gmo0\",\"sender\": \"Ashely\",\"status\": \"Pending\"}"
			}

List all friends by user	GET https://hoeezamirc.execute-api.eu-west-2.amazonaws.com/development/friends/	Content-Type: application/json
		
		Authorization: allow
		
		auth-user: <receiver user name used in previous request>, this is to show that the user is extracted from header. in real scenario user will be coming from authorisation token.
			Response:
			
			[
			{
			"requestId": "dna_ofvftp95gmo0",
			"sender": {
			"id": "ABC999",
			"name": "Ashely"
			},
			"receiver": {
			"id": "ABC34343",
			"name": "Mick"
			},
			"status": "Pending"
			}
			]
			
			Note: more down request id, this will be used in future request when we approved or reject friend requests.

Get single friend request by id	https://hoeezamirc.execute-api.eu-west-2.amazonaws.com/development/friends/dna_ofvftp95gmo0
	
	Note: replace dna_id with the id from previous response.
		Content-Type: application/json
		
		Authorization: allow
			Response:
			
			{
			"requestId": "dna_ofvftp95gmo0",
			"sender": {
			"id": "ABC999",
			"name": "Ashely"
			},
			"receiver": {
			"id": "ABC34343",
			"name": "Mick"
			},
			"status": "Pending"
			}

Approved or Reject friend request	https://hoeezamirc.execute-api.eu-west-2.amazonaws.com/development/friends/dna_ofvftp95gmo0
	
	Note: replace dna_id with the id from previous response.
		Content-Type: application/json
		
		Authorization: allow
			Request Body:
			
			{
			"status" : "Approved"
			}
			
			or
			
			{
			"status" : "Rejected"
			}
			
			This is also validated against model which is configured in API Gateway and has only 2 value “Approved/ Rejected”, you can test by giving any other value and you should see 400 bad request error.
			
			Response for approval:
			
			"{\"message\":\"Request has been Approved\”}"
			
			Response for rejection:
			
			"{\"message\":\"Request has been Rejected\”}"

Postman configuration:

Create friend request

POST **/development/friends** HTTP/1.1
Host: hoeezamirc.execute-api.eu-west-2.amazonaws.com
Content-Type: application/json
Authorization: allow
Cache-Control: no-cache
Postman-Token: aa5ce4ad-04e1-4a7a-8dd8-d996448110af

```
{
  "sender" : {
     "id" : "ABC9991",
     "name" : "Ashely"
  },
  "receiver" : {
     "id" : "ABC343431",
     "name" : “Adrian"
  }
}
```

List all request for the user

GET **/development/friends/** HTTP/1.1
Host: hoeezamirc.execute-api.eu-west-2.amazonaws.com
Content-Type: application/json
Authorization: allow
auth-user: Adrian
Cache-Control: no-cache
Postman-Token: c9a5686b-ef67-45e6-b7e4-19f7d95b1a96

Get single request by request id

GET **/development/friends/dna_ofvftp95gmo0** HTTP/1.1
Host: hoeezamirc.execute-api.eu-west-2.amazonaws.com
Content-Type: application/json
Authorization: allow
Cache-Control: no-cache
Postman-Token: 52af52fd-4b39-4a9f-b275-6887849560f5

Approve or reject friend request

PUT **/development/friends/dna_ofvftp95gmo0** HTTP/1.1
Host: hoeezamirc.execute-api.eu-west-2.amazonaws.com
Content-Type: application/json
Authorization: allow
Cache-Control: no-cache
Postman-Token: b6d77f34-51b9-49a0-b249-b972c80d0384

```
{
        "status" : "Approved"
}
```

Artefact In github: https://github.com/amitmahajan78/friend-request-service

API Documentation:

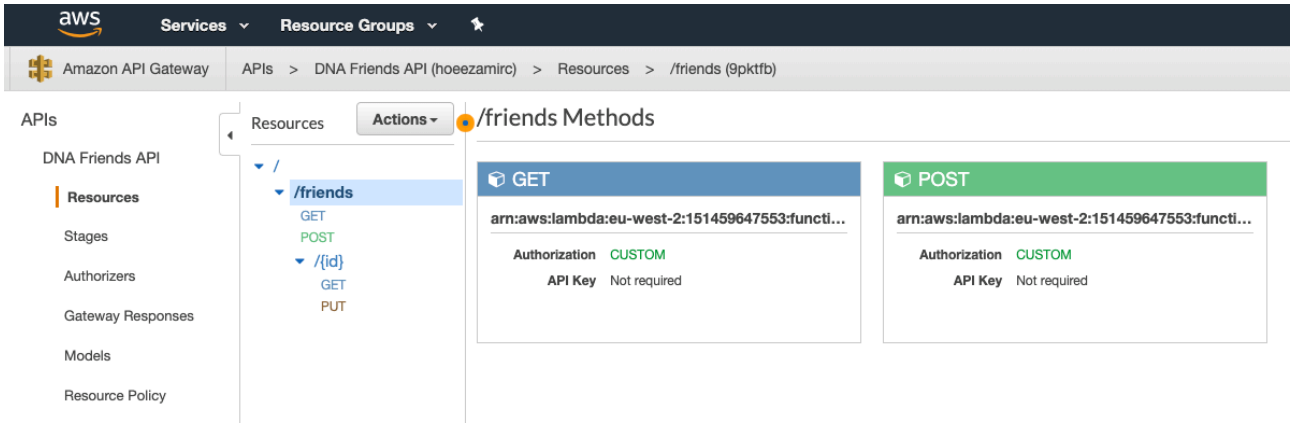
1. **DNA Friends API-development-oas30-postman.json**
2. **DNA Friends API-development-oas30.json**

Lambda Function export

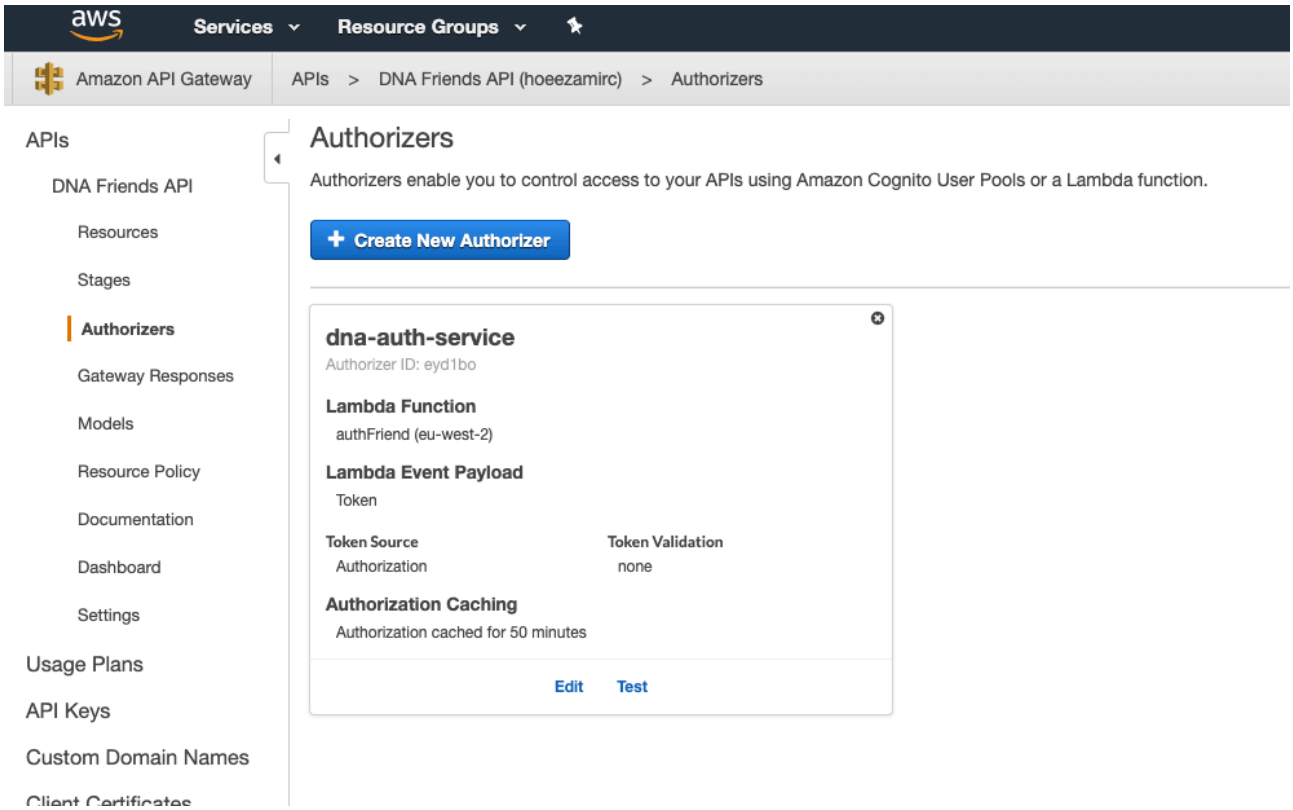
1. **addfriends.yaml**
2. **authFriend.yaml**
3. **dealWithFriendRequest.yaml**
4. **getfriend.yaml**
5. **listAllFriends.yaml**

Screenshots

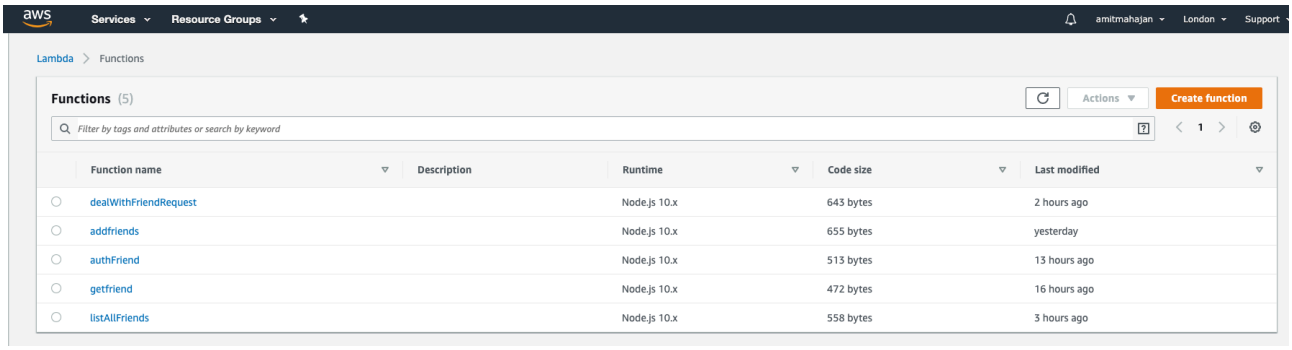
API Gateway



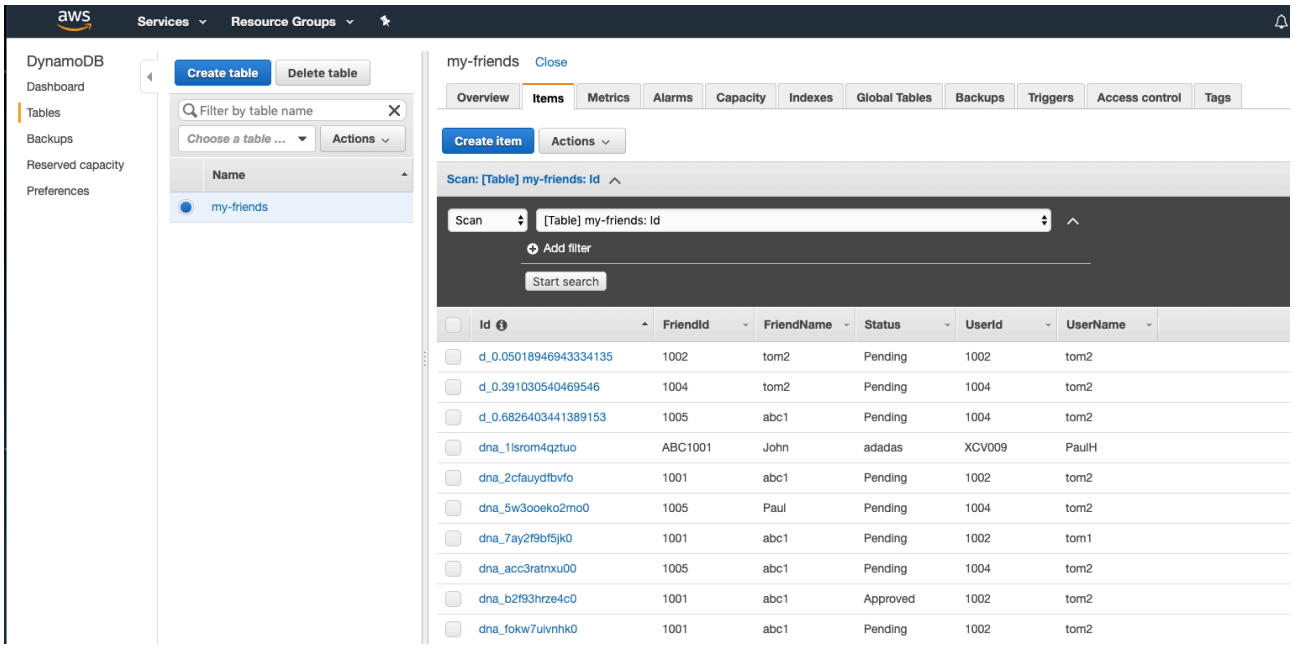
API Gateway Authorization module



Lambda



DynamoDB



CloudWatch