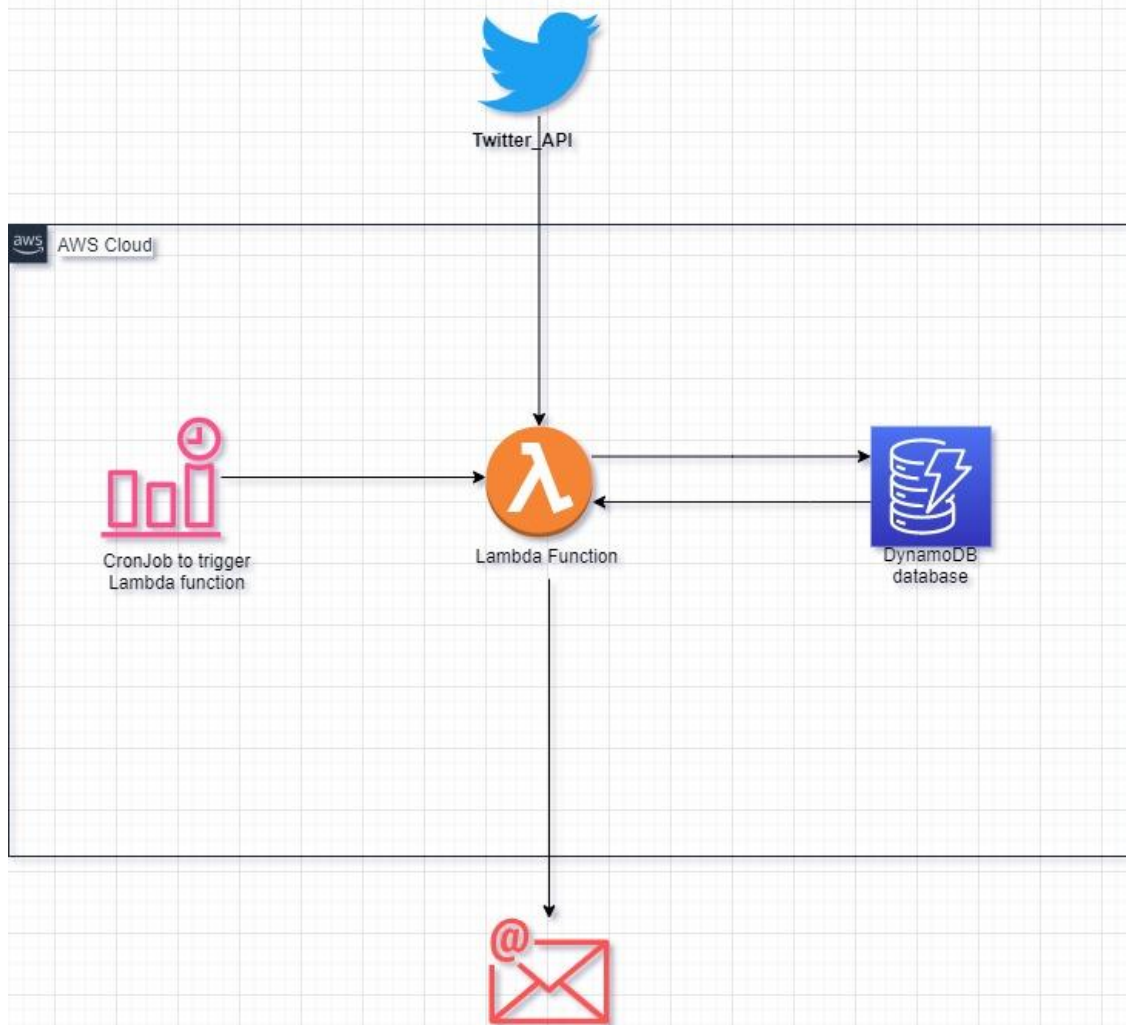


TWITTER UNFOLLOWER NOTIFICATION ARCHITECTURE



Steps:

1. Create required IAM role for Lambda function
2. Create DynamoDB database to store followers' data
3. Create a Lambda Function
4. Create Cron job to trigger the lambda function every one hour.

Repeat steps 2 to 4 and update the twitter account details in the Lambda function code to use this feature for other twitter accounts

1) Create required IAM role for Lambda function:

(Purpose: IAM role is an identity with permission policies that determine what all services can an AWS service can access)

- Go to AWS console, search for IAM and select 'Roles' from the dashboard.
- Click 'Create Role', select 'Select type of trusted entity' as 'AWS Service', Choose use case as 'Lambda', click 'Next Permissions'.
- Search and select '[AmazonDynamoDBFullAccess](#)', '[CloudWatchFullAccess](#)'. Click 'Next Tags'. Click 'Next: Review', enter 'Role name' as 'Lambda_to_dynamodb', Click on 'Create Role' button.

2) Create DynamoDB database to store followers' data

(Purpose: DynamoDB is NoSQL database, we are storing twitter followers data in DynamoDB)

- Go to AWS Console, search for DynamoDB, from the dashboard select 'Tables' and click on 'create table'.
- Enter 'table name' as 'Twitter_jrick'. Enter 'Partition Key' as 'sn', 'sort key' as 'user_id' as shown in below image.

Table name
This will be used to identify your table.

twitter_jrick

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from y hosts for scalability and availability.

sn String ▼

1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or sea same partition key.

user_id String ▼

1 to 255 characters and case sensitive.

- Click on 'Create Table' button on the bottom of the screen.

3) Create a Lambda Function

(Purpose: Lambda function is used to run piece of code which checks if someone unfollowed us on twitter)

- Go to AWS Console, search for 'Lambda', click on 'create function'.
- Enter function name as 'twitter_jrick', Runtime as 'Python 3.7', click on 'Change default execution role' and select 'use an existing role' and select 'Lambda_to_dynamodb' role you created in step 1. Click on 'Create function'
- Paste the entire code I've shared with you and change the credentials in it. Scroll to the bottom and click 'Add a layer', select custom layers and select 'deployment_packages_twitter' and version as '1'. Click on 'add' button.
- Save the code using 'CTRL+S' shortcut keys and Click on 'Test' button, enter the 'event name' as 'Test'. Click on 'Create'
- Click on 'deploy' button to deploy your code. Again, click on 'Test' button to test your code.
- Click on Configuration -> General configuration -> Edit -> enter Memory as 10240, Timeout as 15min, 0 sec.

4) Create Cron job to trigger the lambda function every one hour.

(Purpose: Event rules helps us to schedule the Lambda function at desired time)

- Go to AWS Console and search for Cloudwatch. In Dashboard select 'Rules' under 'Events'.
- Click on 'Create Rule', Select 'Schedule'. Enter Cron expression as `0 1 * * ? *`
- Click on Add Target on the right-side bar, select your lambda function from step 3 from the dropdown. Click on 'Configure Details', Enter the name as 'twitter_jrick_cron' and click on 'Create rule' button.