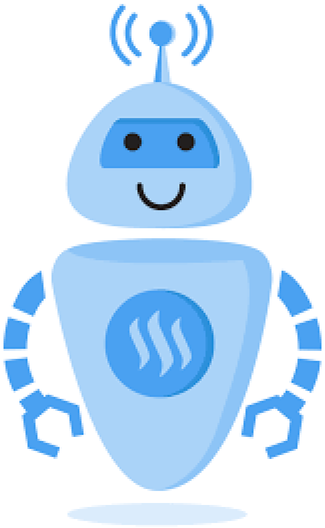
MarketLytics

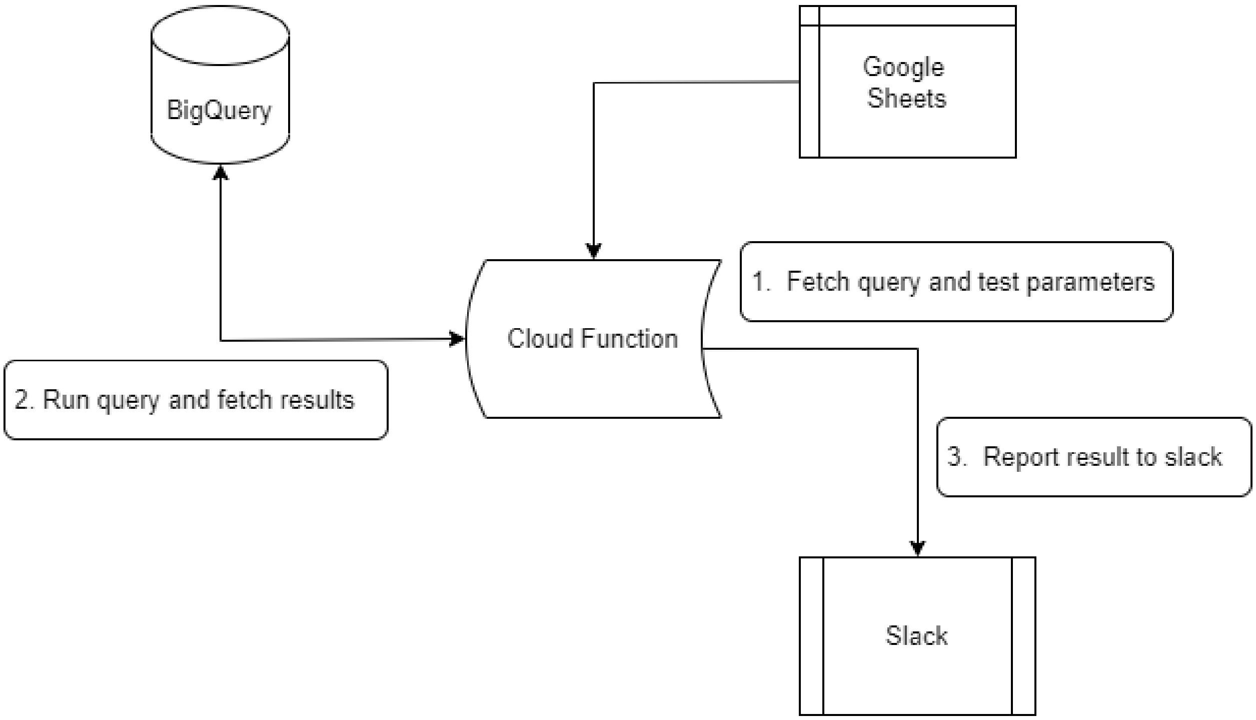


# Anoma Bot - User Manual

# Introduction

The aim of the Anoma bot is to notify data anomalies, the number of rows added in a day, and any data arrived in a day or not for a particular table of the database.
It is developed in such a way that it can be easily integrated with any project without changing much of the codebase.

# Architecture


BigQuery
Google Sheets
Cloud Function
1. Fetch query and test parameters
2. Run query and fetch results
3. Report result to slack
Slack

All the configuration, project ids, tests, and query parameters are configured in a google sheet. This sheet can hold parameters for multiple tests for multiple different projects.

The code is hosted in a cloud function that on getting triggered reads the sheet or config file, picks the specified test parameters, runs it, and reports the results to slack.

# Prerequisites

## Service Accounts

### Scheduler Service Account:

Should be set as an environment variable with name
ANOMALY_TESTS_SCHEDULER_SERVICE_ACC_PATH.
This is used for scheduling tests as cloud scheduler jobs.

### Tests Runner Service Account:

Should be set as an environment variable with name
ANOMALY_TESTS_RUNNER_SERVICE_ACC_PATH.
This is used as running those tests scheduled as jobs.

### Permissions above service accounts should have:

- Permissions to access Google Cloud Storage
- Permissions to schedule Google Cloud Scheduler Jobs
- Permissions to Access BigQuery Resources.
- Permissions for Google Cloud Functions

## Gitlab Repository

This is used to upload the pictures of results of tests, so that can be sent to Slack.
You need to set two environment variables for this.

- **GIT_PROJECT_ID:** The project ID of the Git Repository where results will be uploaded
- **GIT_TOKEN:** Token that has access to the repository with relevant permissions.

## Google Sheet

Configuration spreadsheet for the tests. You need to change the “self.sheet_id” variable in Utils class’ constructor in the utils.py file. Set this variable to your sheet id.

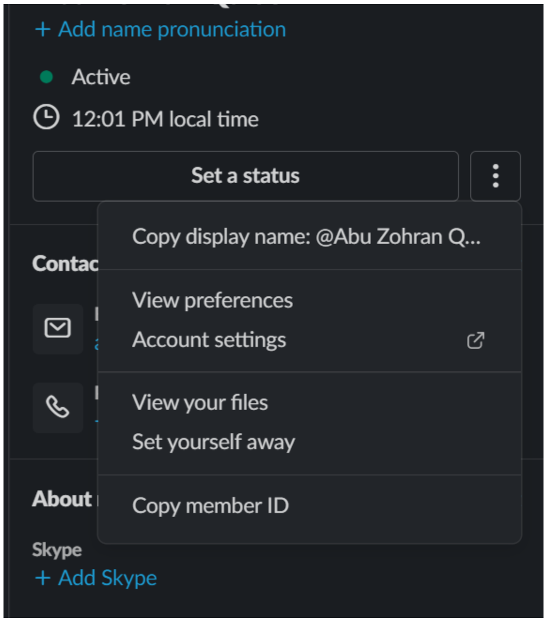
## Slack Channel Webhook URL

The Project uses a Slack channel webhook to send messages/alerts for tests’ results.
Set “self.webhook” variable in Slack class to your webhook url in alerts.py file.

# Steps to configure Anoma bot by using Google Sheet

## 1. Configure Google Sheet

**Slack_member_id:** It is the unique slack ID of an individual who configures the bot so that the individual gets tagged by the bot on the slack channel.
To find your member ID on slack: **Click on profile > Click on three dot dropdown menu > copy member ID**



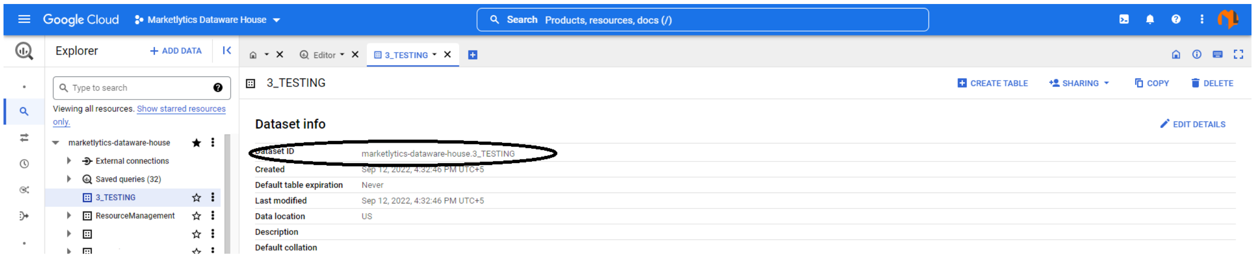
**Project_name:** It is the exact GCP project name that contains the required table on google cloud.
**test_id:** It is the unique integer ID that must be assigned by the one who is configuring the bot.
**Test_name:** It is the name of the test/task that an individual will assign.
**test_type:** There are three tests that the bot can perform;

1. Anomaly
2. Data_arrived_or_not
3. No_of_rows

**threshold:** This parameter will only be passed if the test type is set as “Anomaly” otherwise leave it blank. It is the integer number that identifies the threshold for marking anomalies in the dataset.
**timezone:** It is the timezone that will be provided by an individual. Since we are sitting in Karachi so it will be “Asia/Karachi”.

**cron_schedule:** It is the time of the scheduled test in cron format.
**main_table_name:** It is the name of the table for which the bot will run the test.

To find the main table name: **Go on Google Cloud > Click on the project > Click on the dataset > Click on the table for which bot will run > Click on the details tab > Copy Table ID**



3_TESTING

Dataset info

| | |
|---|---|
| Dataset ID | marketlytics-dataware-house.3_TESTING |
| Created | Sep 12, 2022, 4:32:46 PM UTC+5 |
| Default table expiration | Never |
| Last modified | Sep 12, 2022, 4:32:46 PM UTC+5 |
| Data location | US |
| Description | |
| Default collation | |

**date_column_name:** It is the name of the date column of the selected table.

## 2. Test the bot

- First, join the slack channel which webhook url you provided in prerequisites.
- Deploy the code as a cloud function with “get_anomalies” function as entry point.
- Send an HTTP request on cloud function URL with test_id query parameter set to the test you want to run. It will run and the result will be sent on the Slack channel.