Guidebot RedShift

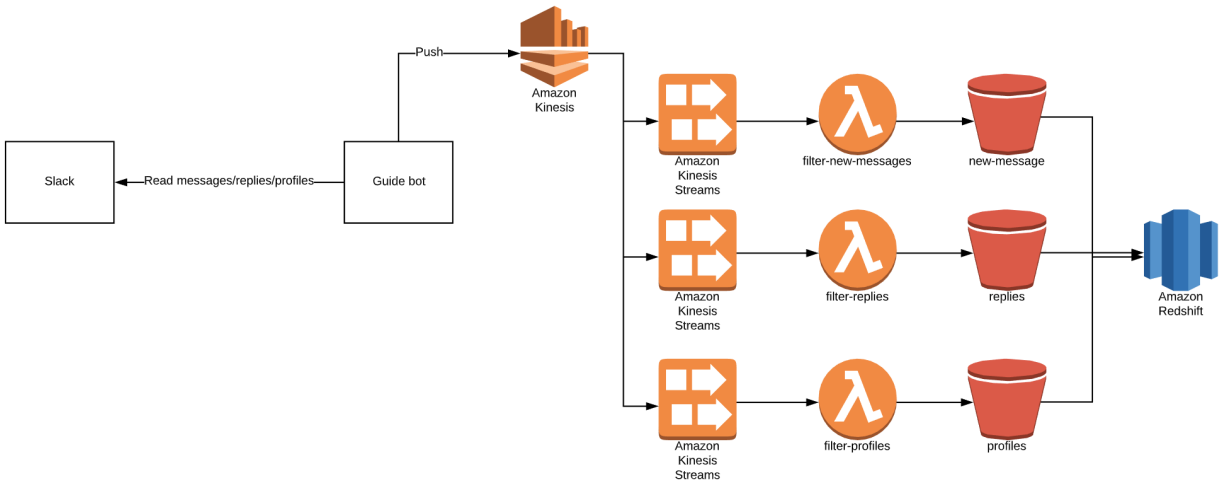
Repo:
https://github.com/auth0/tfl-guide-bot

Heroku:
https://dashboard.heroku.com/apps/tfl-bot

FTQ Export Design

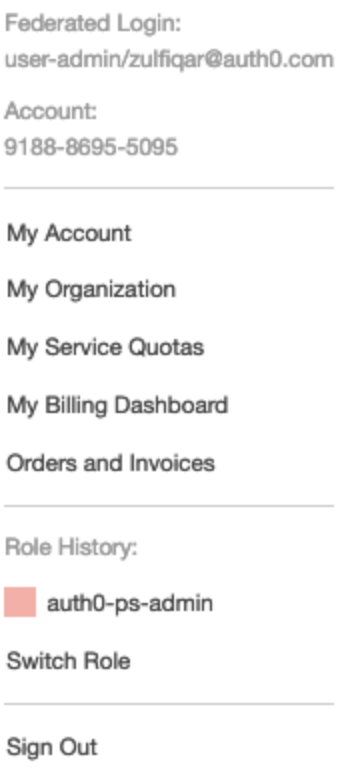
- The export process is extended to retrieve and isolate following 3 event types
  - messages (initial message)
  - User profiles
  - Message replies

These 3 event types are then pushed to the kinesis stream with 3 firehose delivery streams each filtering it's own event (using a lambda) and forwards it to S3, from where the messages will be imported in 3 redshift tables using Copy command. Copy to RedShift is done using the OOB Kinesis integration.



RedShift Schema

The artefacts are already deployed in AWS PS accounts in us-east-1 region.



Credentials:

```
--drop table public.ftq_analytics_initial_questions;
CREATE TABLE public.ftq_analytics_initial_questions (
        thread_ts VARCHAR(32) PRIMARY KEY UNIQUE NOT NULL,
   user_handle VARCHAR(16),
        reply_count INTEGER,
        reply_users_count INTEGER,
        latest_reply VARCHAR(32),
        text VARCHAR(32768)
)
```

```
COPY public.ftq_analytics_initial_questions (
  thread_ts,
  user_handle,
  reply_count,
  reply_users_count,
  latest_reply,
  text)
from 's3://guide-bot-export/test-run/ftq-initial-questions'
iam_role 'arn:aws:iam::625139634126:role/redshifts3read'
CSV
DELIMITER AS '~'
REGION as 'eu-central-1'
TIMEFORMAT as 'epochmillisecs';
```

```
--drop table public.ftq_analytics_user_profiles;
CREATE TABLE public.ftq_analytics_user_profiles (
    user_handle VARCHAR(16) PRIMARY KEY UNIQUE NOT NULL,
          display_name VARCHAR(255),
          department VARCHAR(255),
          division VARCHAR(255),
          city VARCHAR(255),
          first_name VARCHAR(255),
          last_name VARCHAR(255),
          real_name VARCHAR(255),
          title VARCHAR(255),
          phone VARCHAR(255)
)
```

```
COPY public.ftq_analytics_user_profiles (
    user_handle,
    display_name,
    first_name,
    last_name,
    real_name,
    phone,
    title,
    department,
    division)
from 's3://guide-bot-export/test-run/ftq-user-profiles'
```

```
iam_role 'arn:aws:iam::625139634126:role/redshifts3read'
CSV
DELIMITER AS '~'
REGION as 'eu-central-1'
TIMEFORMAT as 'epochmillisecs';
```

```
--drop table public.ftq_analytics_question_replies;
CREATE TABLE public.ftq_analytics_question_replies (
	user_handle VARCHAR(16),
	thread_ts TIMESTAMP,
	text VARCHAR(32768)
)
```

```
COPY public.ftq_analytics_question_replies (
  user_handle,
  thread_ts,
  text)
from 's3://guide-bot-export/test-run/ftq-question-replies'
iam_role 'arn:aws:iam::625139634126:role/redshifts3read'
CSV
DELIMITER AS '~'
REGION as 'eu-central-1'
TIMEFORMAT as 'epochmillisecs';
```

Top 3 participants (Sample Query)

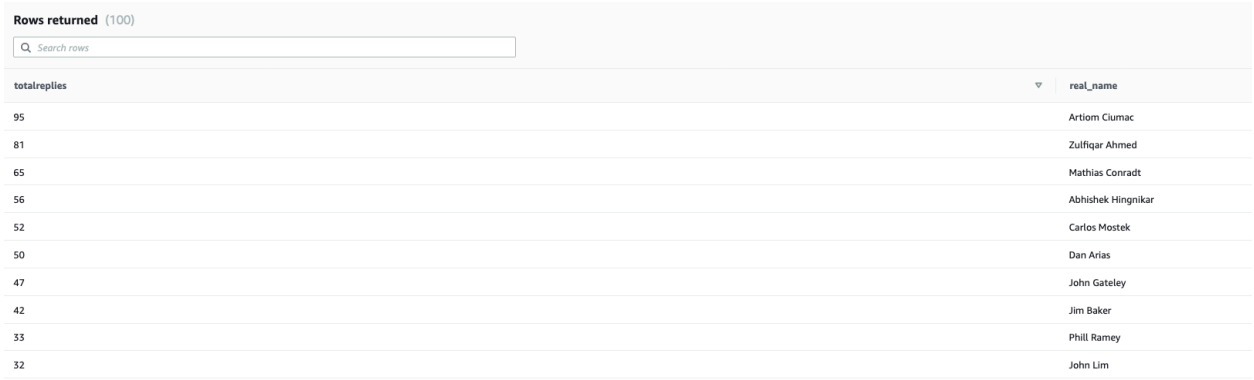
select count(*) as totalReplies, up.real_name from question_replies as qr inner join user_profiles as up on up.user_handle=qr.user_handle group by up.real_name order by totalReplies desc

Unique Threads

select count(distinct thread_ts) as threads_participated, up.real_name as name from question_replies as qr inner join user_profiles as up on up.user_handle=qr.user_handle group by up.real_name order by threads_participated desc

September distinct by thread:
select count(distinct thread_ts) as threads_participated, up.real_name as name from question_replies as qr inner join user_profiles as up on up.user_handle=qr.user_handle WHERE thread_ts >= 1598918400 and thread_ts < 1601510400 group by up.real_name order by threads_participated desc

October distinct by thread:
select count(distinct thread_ts) as threads_participated, up.real_name as name from question_replies as qr inner join user_profiles as up on up.user_handle=qr.user_handle WHERE thread_ts >= 1601510400 and thread_ts < 1604188800 group by up.real_name order by threads_participated desc



Rows returned (100)

Search rows

totalreplies	real_name
95	Artiom Ciumac
81	Zulfiqar Ahmed
65	Mathias Conradt
56	Abhishek Hingnikar
52	Carlos Mostek
50	Dan Arias
47	John Gateley
42	Jim Baker
33	Phill Ramey
32	John Lim

Fresh Export

1. Resume the RedShift cluster in PS account (918886955095)

Truncate existing tables from Redshift console and rerun the export from CURL or Slack

TRUNCATE user_profiles;
TRUNCATE question_replies;
TRUNCATE initial_questions

2. Trigger export either from slack (#field-team-questions) or shell

@guide export 180

curl --header 'Content-Type: application/json' --data '{"channel":"C9YBNE6T0","days":"30","token":"xoxb-***"}' 'https://tfl-bot.herokuapp.com/pump'

This will export the data to RedShift, where you can explore/verify the data. You can get CSV by unloading to s3 as shown below.

Unloading RedShift Data to S3

UNLOAD('select iq.text, (select listagg(qr.text, "|+|") from question_replies as qr where qr.thread_ts=iq.thread_ts group by qr.thread_ts) from initial_questions as iq') TO 's3://redshift-load-store/2020/07/30/16/q-rep00001.csv' iam_role 'arn:aws:iam::918886955095:role/myRedshiftRole' delimiter '~' `parallel off;`

`Unload September`
UNLOAD('select iq.text, (select listagg(qr.text, "|+|") from question_replies as qr where qr.thread_ts=iq.thread_ts group by qr.thread_ts) from initial_questions as iq WHERE iq.thread_ts >= 1598918400 and iq.thread_ts < 1601510400 ') TO 's3://redshift-load-store/2020/11/10/16/September20-FTQ-Data.csv' iam_role 'arn:aws:iam::918886955095:role/myRedshiftRole' delimiter '~' parallel off;

`Unload September`
UNLOAD('select iq.text, (select listagg(qr.text, "|+|") from question_replies as qr where qr.thread_ts=iq.thread_ts group by qr.thread_ts) from initial_questions as iq WHERE iq.thread_ts >= 1601510400 and iq.thread_ts < 1604188800 ') TO 's3://redshift-load-store/2020/11/10/16/October20-FTQ-Data.csv' iam_role 'arn:aws:iam::918886955095:role/myRedshiftRole' delimiter '~' parallel off;

Word Cloud

Upload the unloads here: https://monkeylearn.com/word-cloud