



Turbo Mobile Bot

Enterprise Bot Framework

Lex

Translate

Rekognition

Comprehend

Slack

Kinesis

Elastic Search

Kibana

Lambda

S3

What is Turbo Mobile Bot

Turbo Mobile Bot is a Amazon Lex based bot framework for enterprise solution.

Translation and Image Analysis

- Translate Text
- Analyze images uploaded on organization's repository and upload the data on Elastic Search
- Find Image
- Recognize Image
- Search Face
- Detect Celebrity Info

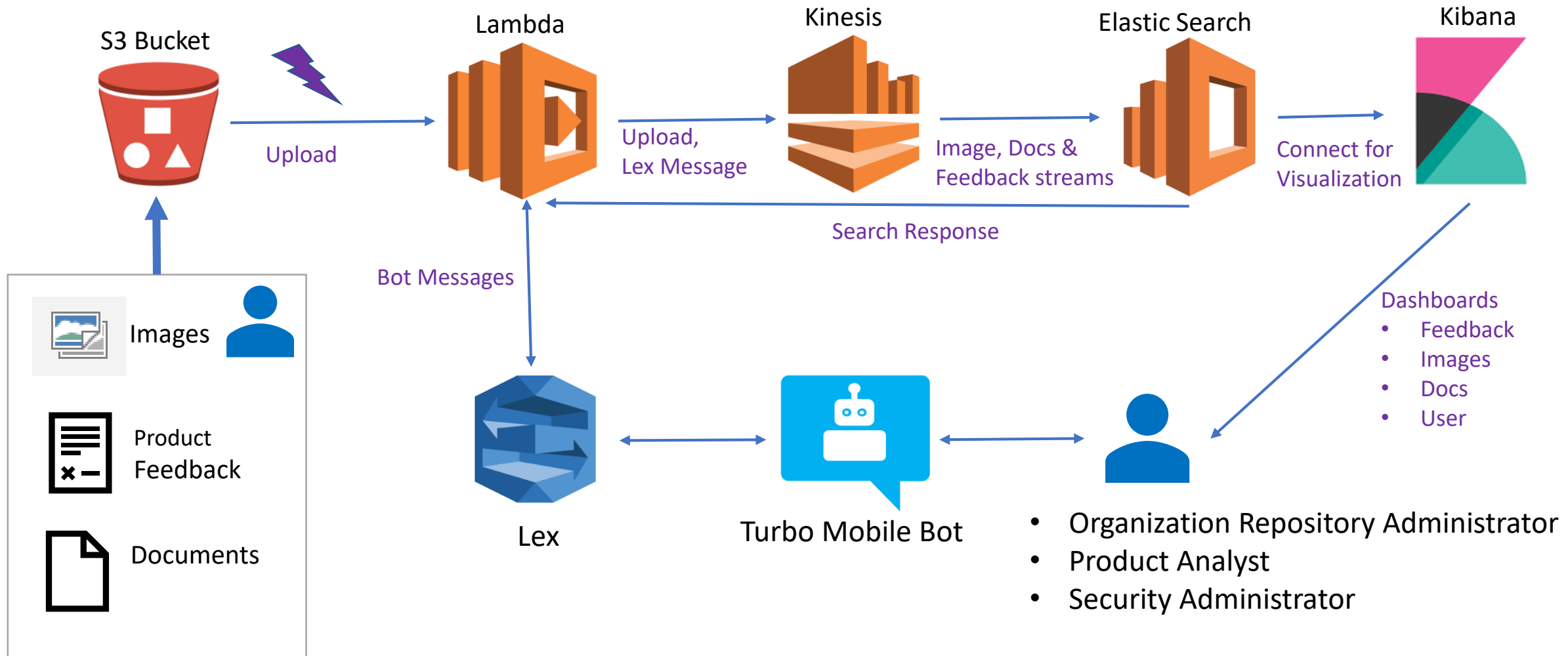
Employee Management

- Create User Image Collection with Name , User ID
- Search User to match the image collected by any device e.g. Attendance system, security system etc)

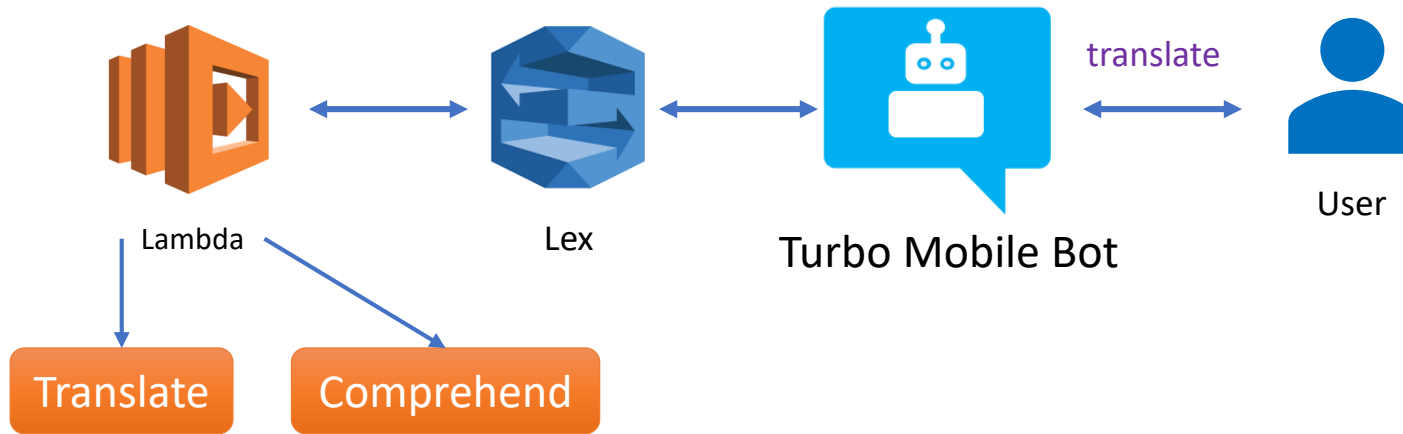
Product Review Analysis

- Detects sentiments from user feedback
- Uploads feedback on Elastic Search for analysis
- Creates Dashboard for live feedback analysis on multiple products of the organization

Turbo Mobile Bot: Data Flow



Use Case: Translate Text



Input Language is detected using 'comprehend' API
Text is translated using 'translate' API

Sample Utterances:

- Translate
- Translate text into German

The screenshot shows a chat interface with the following messages:

- Saleem Javed** 11:58 AM: translate
- TurboMobileBot** APP 11:58 AM: Please specify output language. Supported languages are english, french, german, italian, portugese, russian, spanish and turkish
Specify output language
german
- TurboMobileBot** APP 11:58 AM: please enter text
- Saleem Javed** 11:58 AM: please give me a cup of coffee
- TurboMobileBot** APP 11:58 AM: Bitte geben Sie mir eine Tasse Kaffee
- Saleem Javed** 11:58 AM: translate
- TurboMobileBot** APP 11:58 AM: Please specify output language. Supported languages are english, french, german, italian, portugese, russian, spanish and turkish
Specify output language
english
- TurboMobileBot** APP 11:58 AM: please enter text
- Saleem Javed** 11:58 AM: Bitte geben Sie mir eine Tasse Kaffee
- TurboMobileBot** APP 11:59 AM: Please give me a cup of coffee

Use Case: S3 Repository

S3 bucket is used as repository for all items with following folders

Folder names can vary and they can have subfolders also except 'Feedback' which is used for Product Feedback.

- Images: folder for all images.
- Feedback: for customer feedback files
- Docs: For all documents in .txt or .text format
- General : folder for other documents

When any item is uploaded on S3Bucket and event is triggered and Lambda Function is executed. This lambda functions sort the image in three categories i.e. Image, Docs, Feedback

Image:

Image is processed using 'rekognition' API and following items are detected and Uploaded on Elastic Search domain with index *Images*:: Faces, Texts, Celebrity Information, Labels, face is listed in the face collection

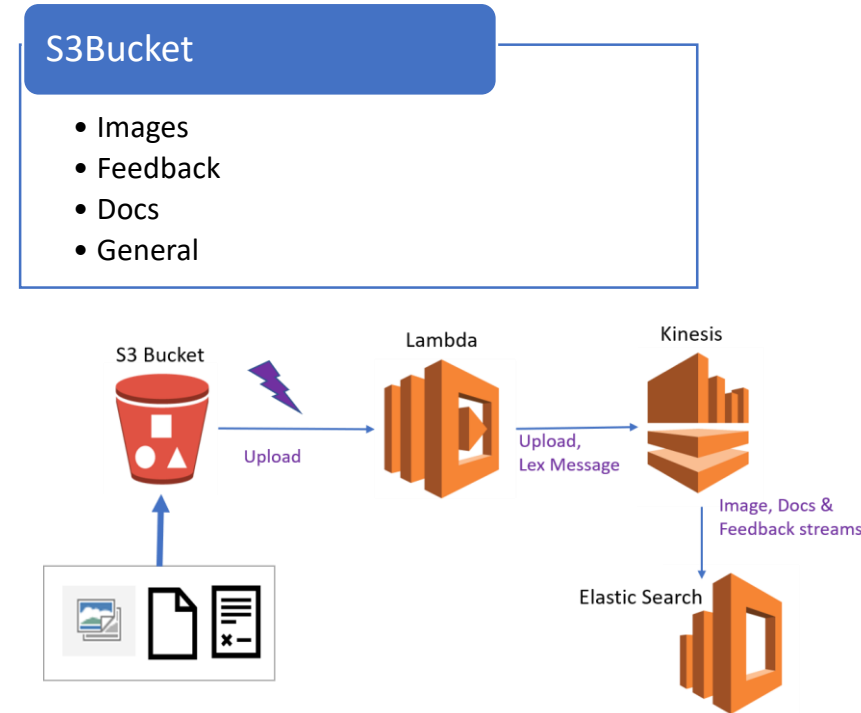
Feedback:

Feedback is processed using 'comprehend' API and feedback's sentiment , entities found and other meta-data is uploaded

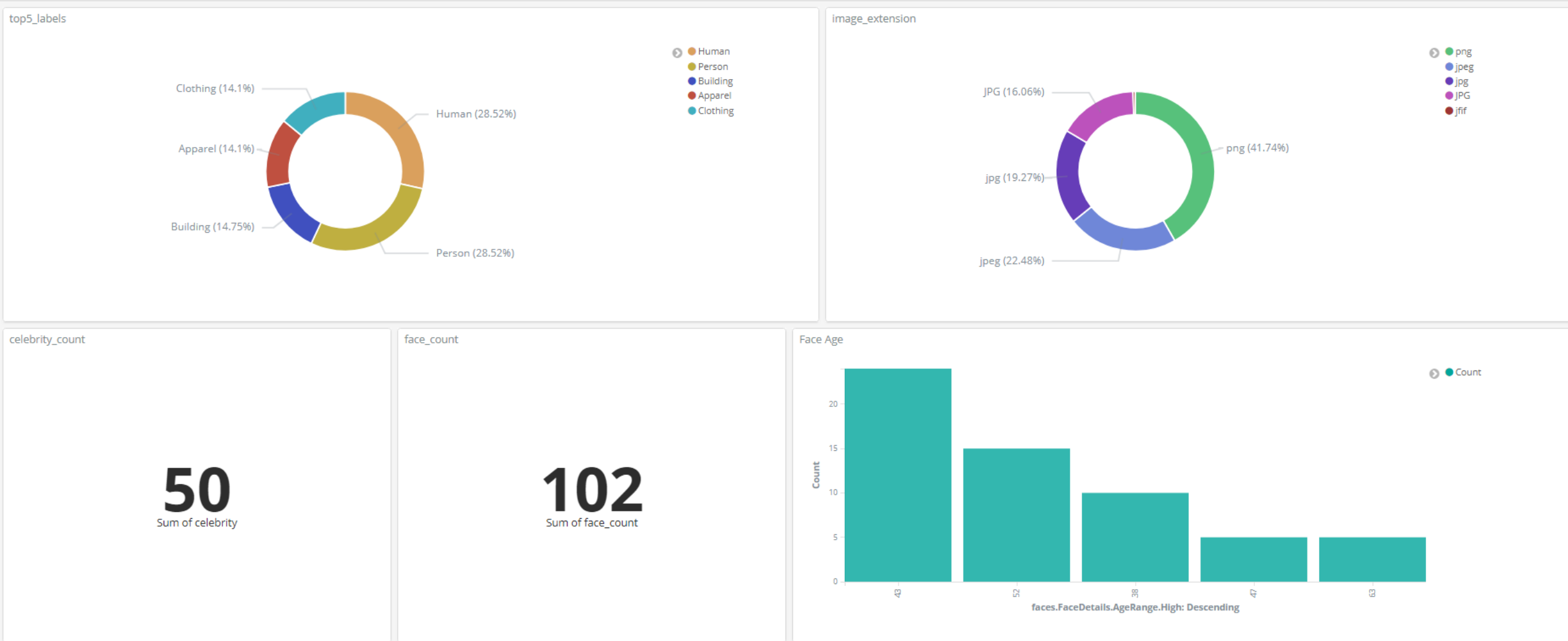
On Elastic Search with ' *Feedback* '

Documents:

Other documents are processed using 'comprehend' API and entities are uploaded on Elastic Search with index '*Docs*'

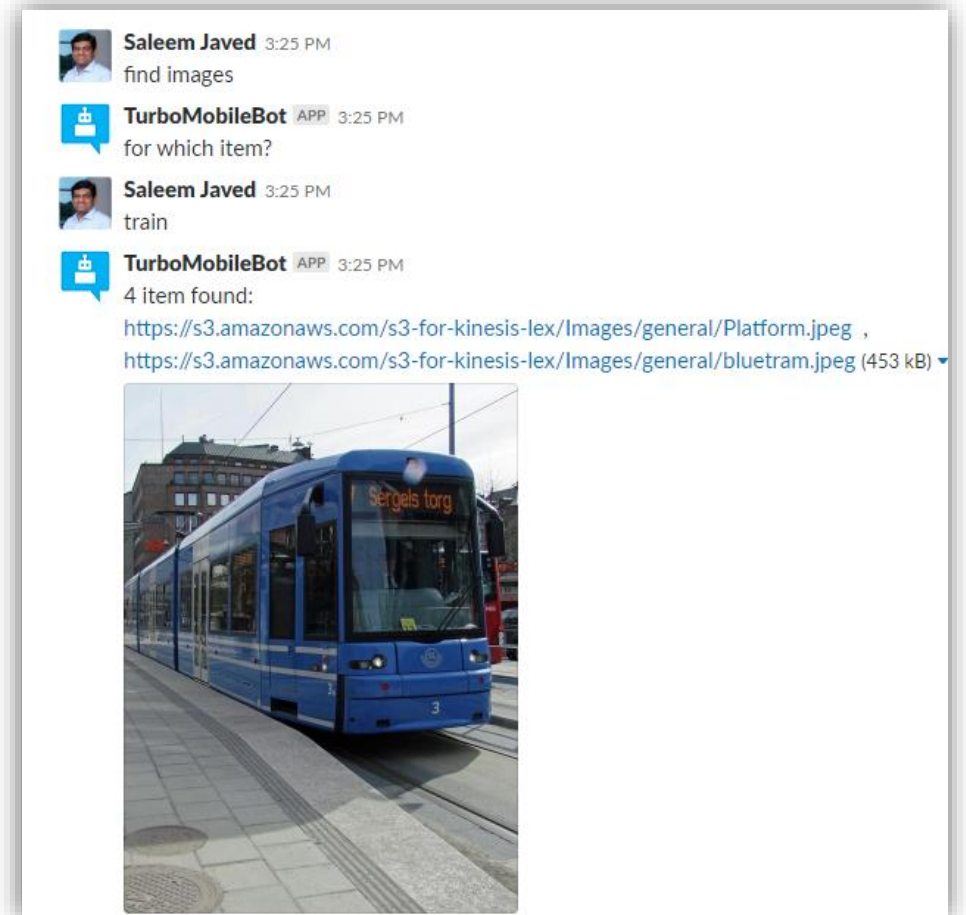
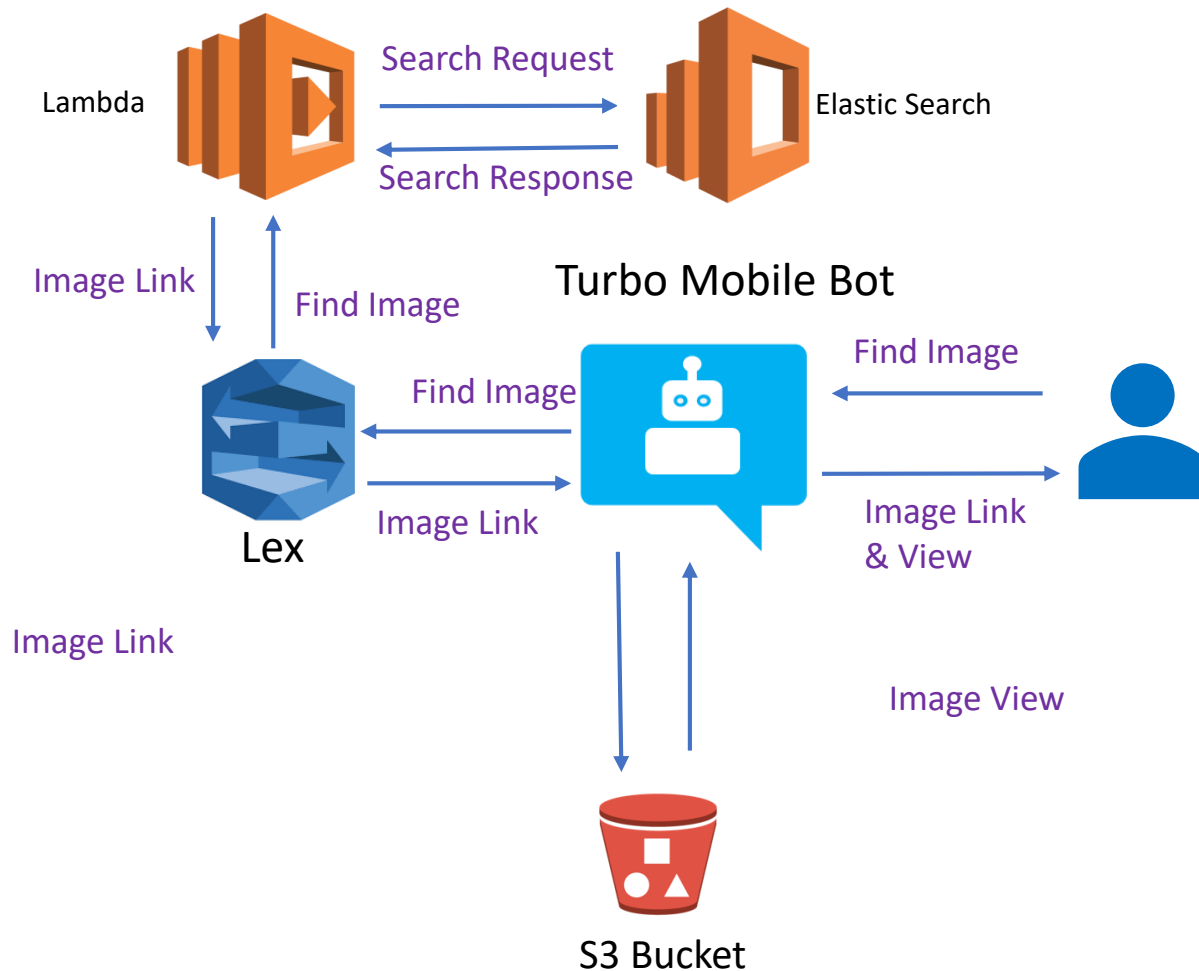


Elastic Search Data Visualization in Kibana



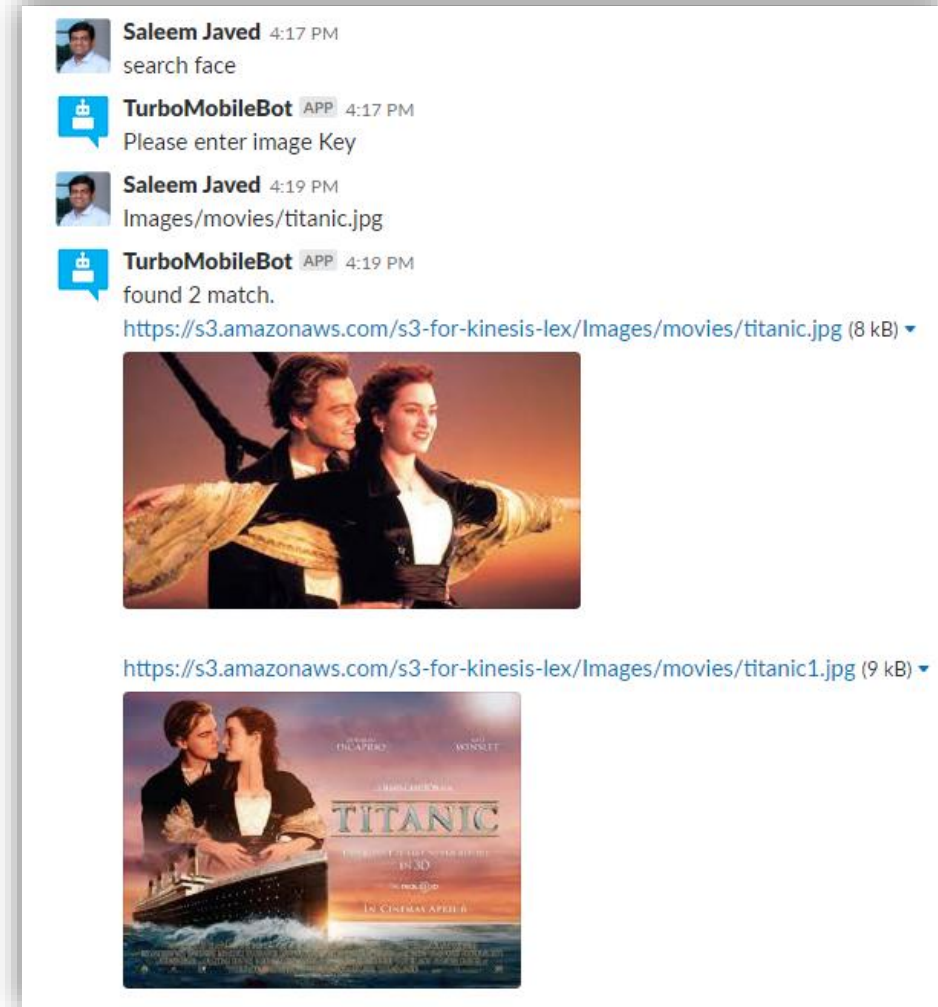
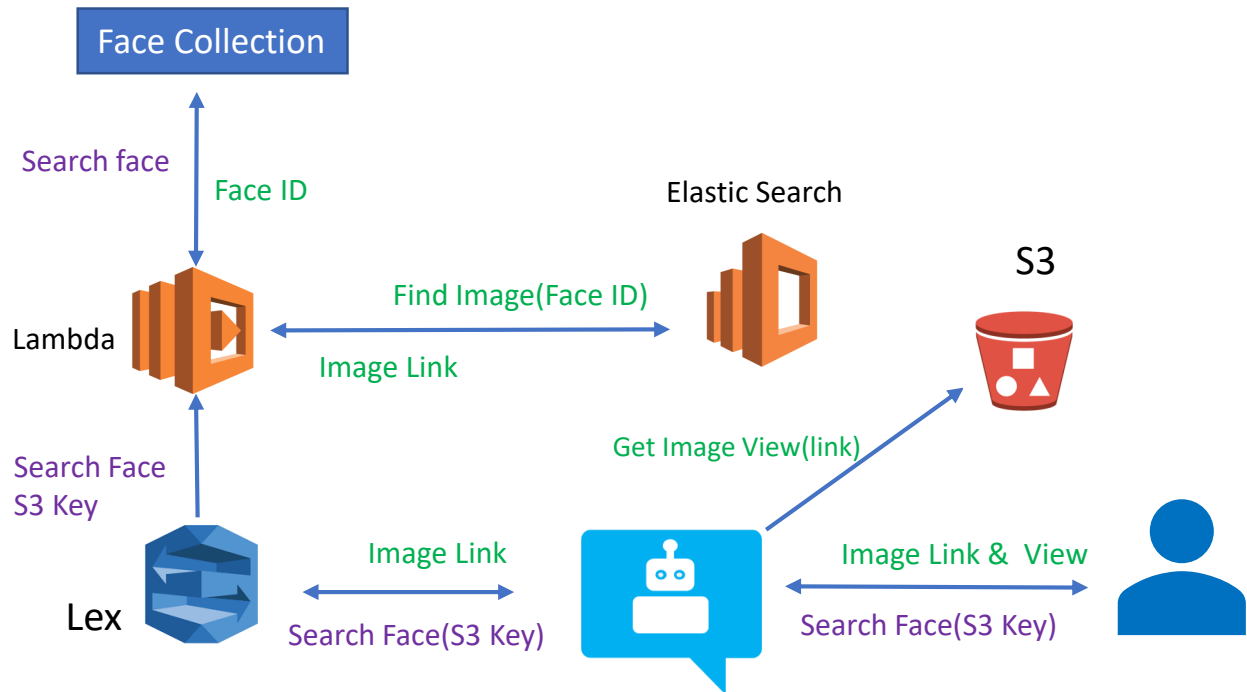
Use Case: Find Image in Repository

For all images uploaded, information detected by 'recognition' API is stored on 'Elastic Search' This information is used to search images available in the repository. Search request is sent to 'Elastic Search' if any matching image is found S3 link is returned. This link is provided to the user. Also Bot fetches image from S3 and provides image view to the user



Use Case: Search Face in Repository

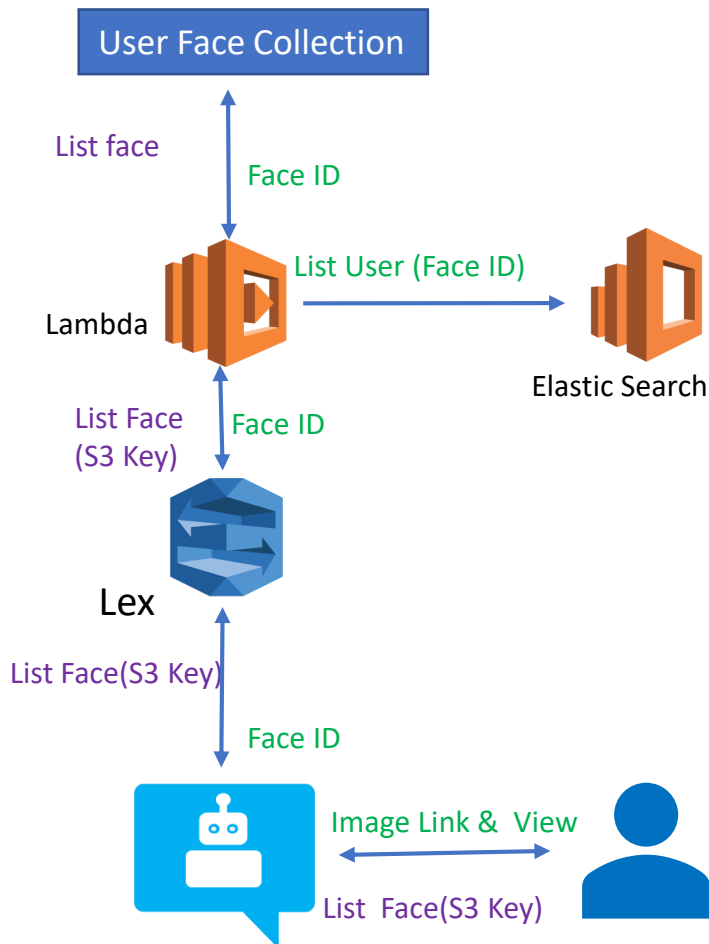
- When any image is uploaded, lambda function is triggered
- Lambda function detects the faces in the image using 'recognition'
- Faces are listed in the collections and saved.
- In an image, face can be searched in the repository using Turbo mobile bot.
- On given S3 key Lambda detects face and then find that in Face Collection.
- Then for matched face id , S3 Bucket link is searched in Elastic Search
- Once link is found , same is returned to the user



Utterance : ' Search Face'

Use Case: List User's Face in Repository

- Turbo Mobile Framework is used for Employee Data Management.
- Using List User Intent users are listed on Elastic Search and User's Face is listed on 'UserFaceCollection', this collection is used later to identify user by attendance system or security system.
- User Data collected: Name, User ID, Picture



UserName	Face.FaceId	FaceDetail.Gender.Value	ExternalImageId
▶ Sourav Ganguly	2603338b-ea16-45dd-9baf-c22db63a8782	Male	1005
▶ Emma Stone	60afdbcd-18da-40eb-9bc1-dadcb2c7ee43	Female	1007
▶ MS Dhoni	a2e0c75d-f1f6-43b7-93b6-65bd449cf0f2	Male	1004
▶ Scarlet Johnson	df7d495d-c6b6-4f0d-8a36-a3a25b24221e	Female	1008
▶ Lewis Hamilton	5aa12c54-4c7b-4b71-834d-5c54232637f2	Male	1005
▶ Vinay CR	18a6f8a5-6503-48a7-b475-b14fbf993ca3	Male	1005
▶ Saleem	dce9f59c-a73b-46d4-942b-f3eb25b35c40	Male	1001
▶ Saleem	dce9f59c-a73b-46d4-942b-f3eb25b35c40	Male	1001
▶ saleem javed	dce9f59c-a73b-46d4-942b-f3eb25b35c40	Male	1001
▶ Brad Pitt	fa240cfa-6eaf-41dc-867e-cd9d7ac8f634	Male	1002

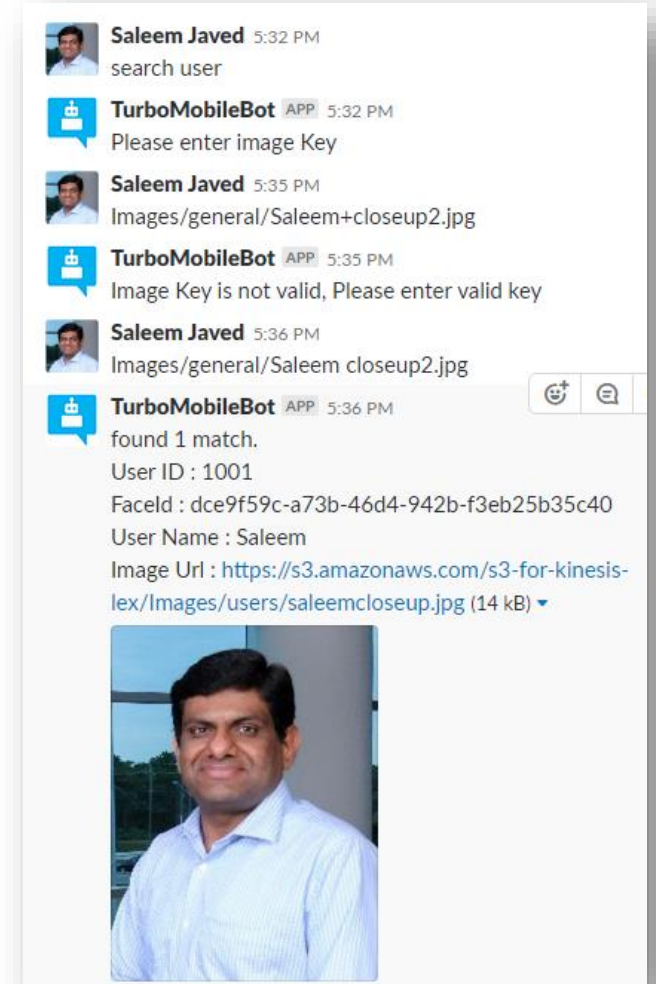
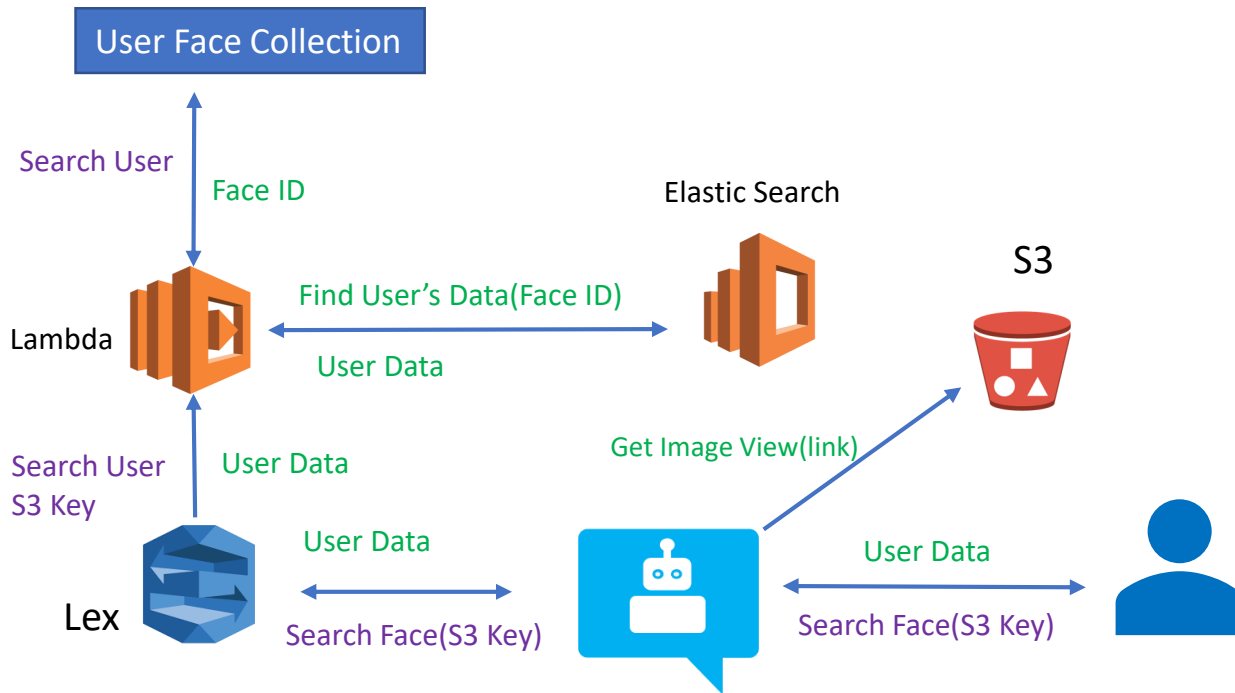
Elastic Search Data in Kibana

The screenshot shows a chat conversation between a user and the Turbo Mobile Bot. The user's intent is 'list face'. The bot responds by asking for the user's name, which is 'George Clooney'. Then, the bot asks for the user's ID, which is '1011'. Finally, the bot asks for the image key, which is 'Images/users/georgeclooney.jpg'. The bot then confirms that the image of user 'George Clooney' is listed successfully with the FaceId 'bde05a37-925a-4e72-a24d-d1dadab0a242'.

Utterance : ' list Face'

Use Case: Search User in Repository

- A user's face captured by any system e.g. attendance system is uploaded on S3 bucket
- Now this image can be matched in User Face Collection to detect that supplied image is of an identified user (employee)



Utterance : ' Search User'

Use Case: Product Feedback Analysis

- Turbo Mobile Framework can create live dashboards for Product Feedback Analysis
- In the application 'Sample Product Feedback' taken from a third party data set has been used only for demonstration purpose
data source: <http://jmcauley.ucsd.edu/data/amazon/>

Feedback Data for four items have been uploaded on S3 and Elastic Search.

(i) Cell Phones (ii) Automotive Products (iii) Digital Music (iv) Musical Instruments

1- Feedback Database downloaded

2 - Each feedback is saved as individual feedback JSON file

3- JSON file uploaded on respective S3 folder e.g. for mobile -> 'Feedback\mobile'

4 -Sentiments of feedback is determined using 'comprehend' API

5- Feedback data is parsed from JSON

6 - Every File upload triggers Lambda function

7- Entities are also listed from feedback text using 'comprehend' API

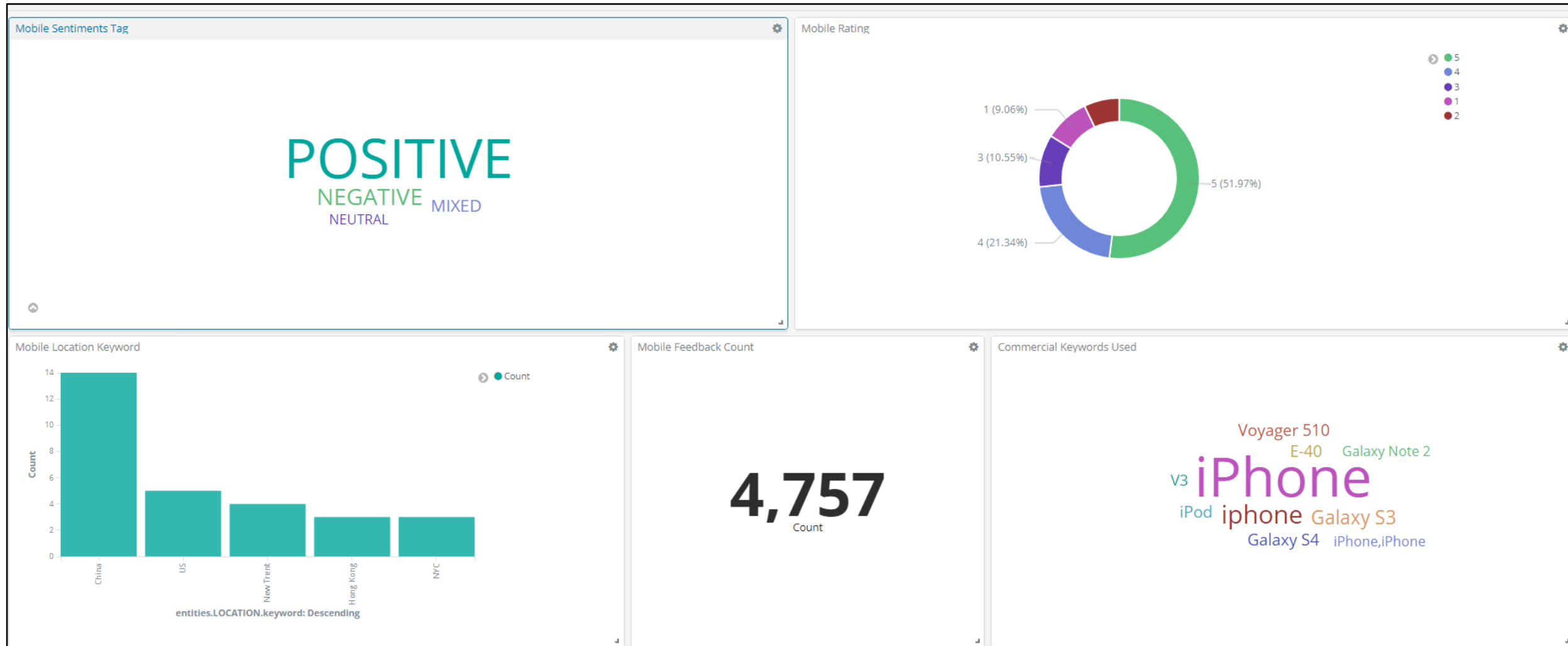
8 - Sentiment and entities are uploaded on Elastic Search with index – Feedback and type – product name

9- Visualizations and Dashboards are created on Kibana

Sample Dashboard: Mobile Feedback Analysis

Feedback Sentiment , Overall Rating, Keywords used in feedback, Total Feedback, Commercial keywords used in feedback

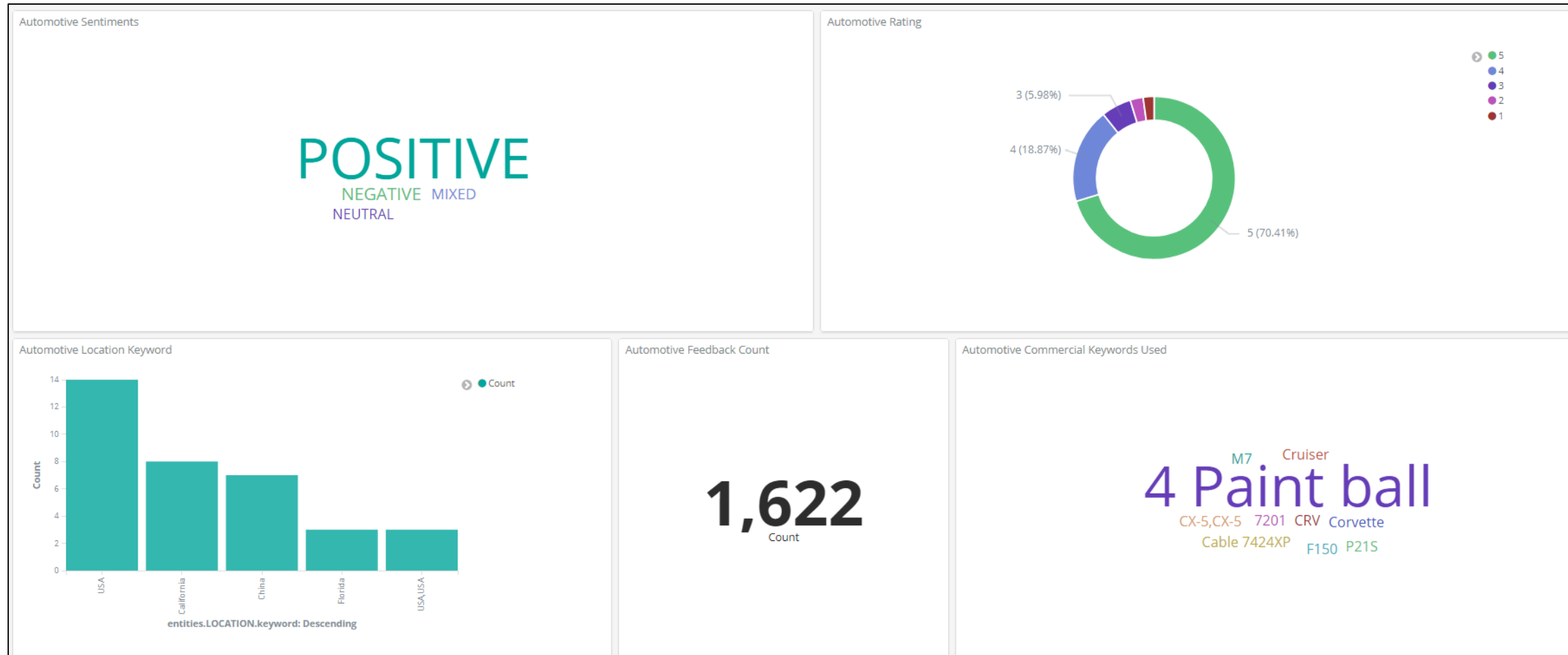
Kibana Link : [Kibana Page](#), Dashboard: [Mobile Feedback Dash Board](#)



Sample Dashboard: Automotive Feedback Analysis

Feedback Sentiment , Overall Rating, Keywords used in feedback, Total Feedback, Commercial keywords used in feedback

Kibana Dashboard Link : [Click here](#)



Thank You