Enriching your applications with IBM Cloud services

Unit objectives

- Define your business problem and goals
- Identify functional and non-functional requirements
- Select the technical components that best fit your solution
- Design a simple architecture for a cloud application
- Identify services in the IBM Cloud catalog that you can use to enrich your cloud apps
- Describe App ID, Watson Natural Language Understanding, Watson Tone Analyzer, LogDNA, and IBM Cloud Monitoring services and their integration in the sample use case

Business problem and requirements

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

Define your business problem and goals

- Define your problem domain and establish common goals
  - You cannot solve a problem without first understanding it.
- Define your business objectives
  - Define the “who”, “what”, and “why”.
  - A business objective must center on a measurable outcome, or the "why.“
  - State the benefits that the stakeholders expect from the system.
- Identify potential problems and bottlenecks
  - Surface the big issues that might negatively impact or prevent the desire outcomes.

Cognitive Tweets Analyzer: Business problem and objectives

- Business problem
  - A movie production company wants to know how the audience reacts to an episode of their popular movie series. They also want to identify the type of scenes that cause the most excitement and positive reaction from the audience.
  - Business objectives
    - Who: Screenplay writers, movie trailer editors
    - What: Guidelines for screenplays and movie trailers
    - Why: Attract larger audiences and new advertisers, sell future episodes
  - Potential problems and bottlenecks
    - Cannot identify duplicate tweets that are retrieved with tweets replies or retweets.
    - Users post emoticons, GIFs, or videos to express their opinions and emotions, which cannot be analyzed

Functional and Non-functional requirements

Functional requirements (FRs)	Non-functional requirements (NFRs)
Specify what the system should do or accomplish	Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.
Describe the functions and services that should be provided by the system	Guide how the system should fulfill the functional requirements
Provide information on how business needs and goals will be delivered through a specific project	Cover requirements that are not covered by the functional requirements. They specify the criteria that judge the working of a system.
Are implemented by the application capabilities, functions and features	Apply to a system as a whole

Cognitive Tweets Analyzer : Functional requirements

Functional requirements (FRs)	Selection
Use a social media platform to capture viewers' reactions to the movie in real time.	**Twitter**.
Retrieve the data from the social media platform, orchestrate the services integration, and provide the logic of the solution.	**Node.js app** **Twitter API** **REST APIs**
The web UI can be accessed from mobile devices and desktop. The analyzed data results are displayed in the web UI with graphics and charts.	**Angular app**
Analyze the social media data to classify the information, detect the viewers' sentiment over time, extract the viewers' emotions, and the keywords that best describe the topics that they are talking about.	**Watson Natural Language Understanding** **Watson Tone Analyzer**
Store the data that is retrieved from the social media platform and the metadata (analyzed data) in a database.	**Cloudant**

Cognitive Tweets Analyzer : Non-functional requirements

Non-functional requirements (NFRs)	Selection
Users must authenticate to the app with their Google account to protect back-end resources and simplify the management of user access.	**App ID**
Monitor the health of the app to ensure its availability.	**IBM Cloud Monitoring**
Manage system and application logs in the cloud. Provide capabilities to view, monitor, and manage logs.	**LogDNA**

Solution architecture

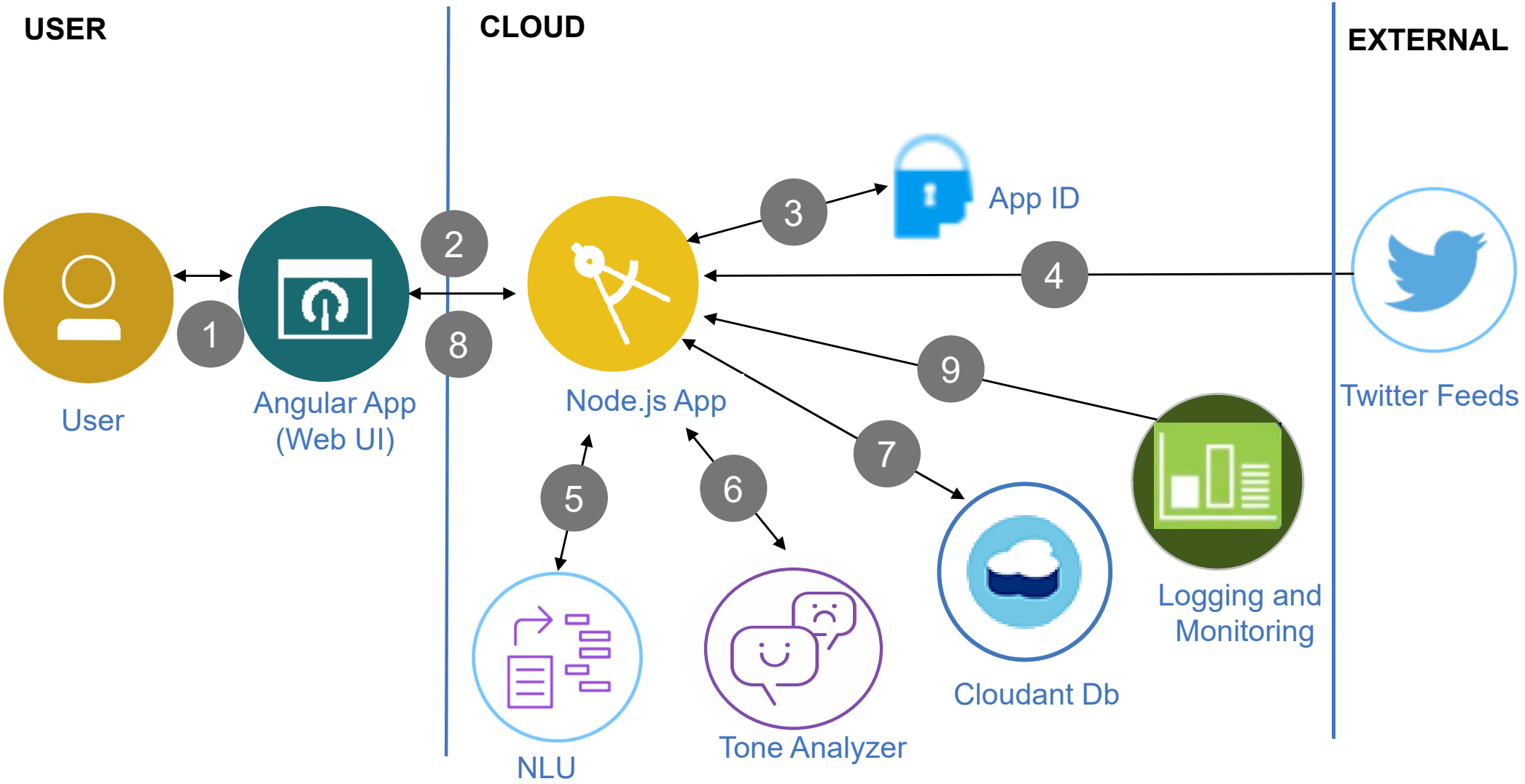
Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

IBM Cloud Architecture Center

- Architectures provide a roadmap to build, extend, and deploy an application.
- IBM Cloud Architecture Center provides:
  - Practices and reference architectures for building apps on the cloud
  - Editable architecture diagram templates
  - Solutions, templates, case studies, and samples in each architecture
- Examples of architecture types in IBM Cloud Architecture Center:
  - Multicloud
  - Hybrid
  - Built for AI
  - Popular application styles
  - Aspects for scaling applications
  - Industry solutions
- Access the IBM Cloud Architecture Center at https://www.ibm.com/cloud/garage/architectures

Cognitive Tweets Analyzer: Architecture



Cognitive Tweets Analyzer code: https://github.com/IBMRedbooks/Cloud-Application-Developer/tree/master/UseCase/cognitive-social-crm

Components

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

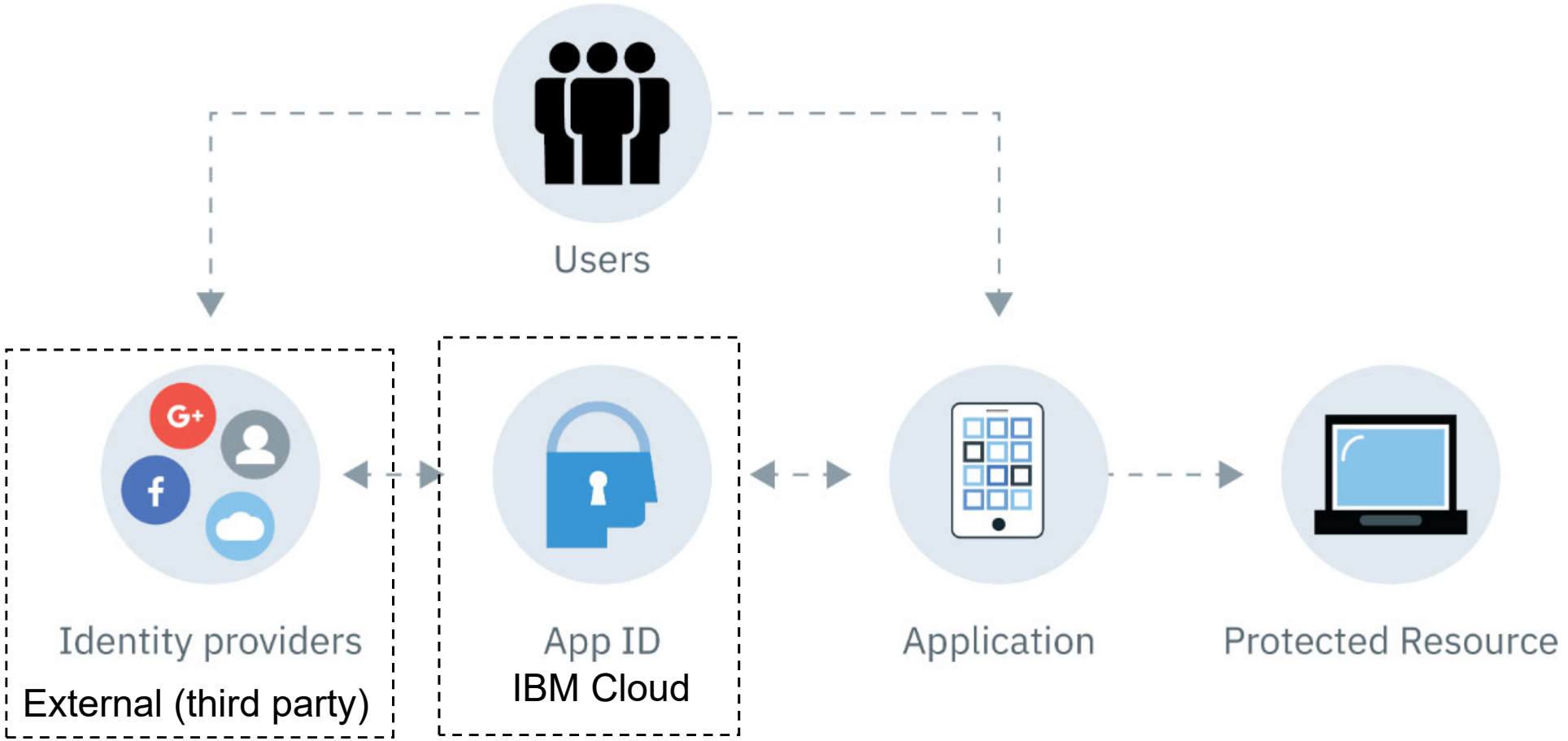
Node.js server app and Angular web UI

- Node.js server application
  - Retrieves the tweets, orchestrates the integration with the other services in the system, and provides the logic of the solution.
  - Uses the Twitter API to access and filter public tweets.
  - The integration of cloud services is implemented through REST API calls.
  - Implements standardized logging and monitoring.
  - Node.js is a popular Javascript runtime for server-side apps.
- Angular web UI
  - Provides a web UI that can be accessed from mobile devices and desktop.
  - Angular is a JavaScript framework that makes it easy to build front-end apps for the web. Some features of the Angular framework are:
    - Cross platform: Progressive web apps, native, and desktop
    - Speed and performance: Code generation, universal, code splitting
    - Productivity: Templates, Angular CLI, IDEs
    - Full development story: Testing, animation, accessibility

App ID service: Overview

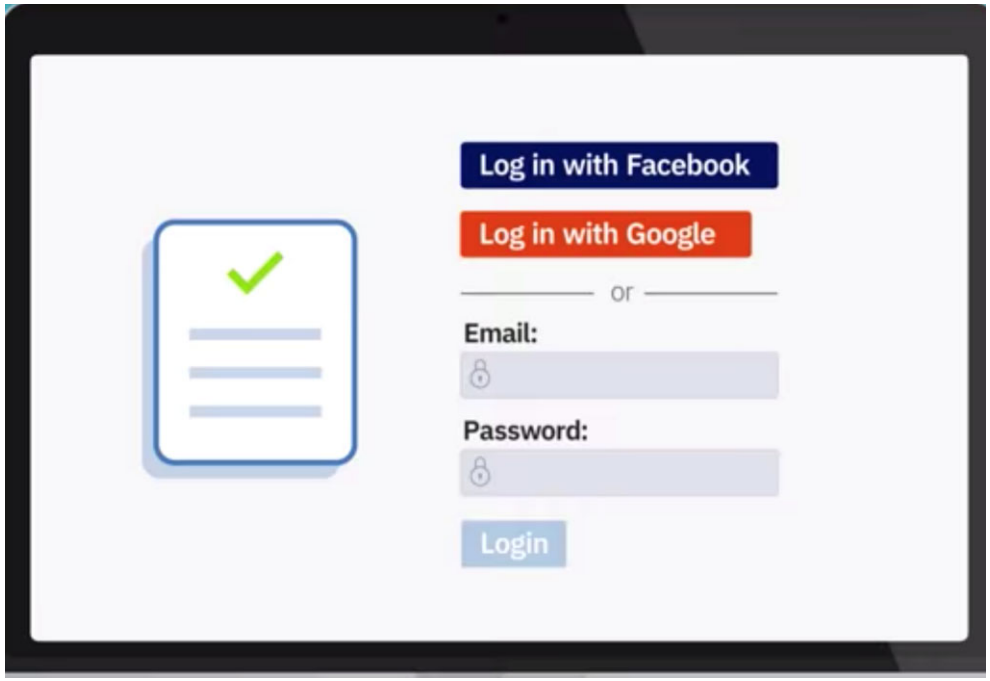
- Secures resources and adds authentication to mobile or web applications with few lines of code.
- By requiring users to sign in to your app, you can:
  - Store user data such as app preferences.
  - Store information from public social profiles.
  - Use the user’s data to customize each user's experience within the app.
- Provides a log-in framework, but you can also bring your own branded screens to use with Cloud Directory.
- Based on industry standard protocols and specifications:
  - OAuth 2.0 Authorization Framework
  - Open ID Connect.
  - OAuth 2.0
- App ID successfully completed several certifications, audits, and standards.

App ID service: How it works



App ID service: Authentication options

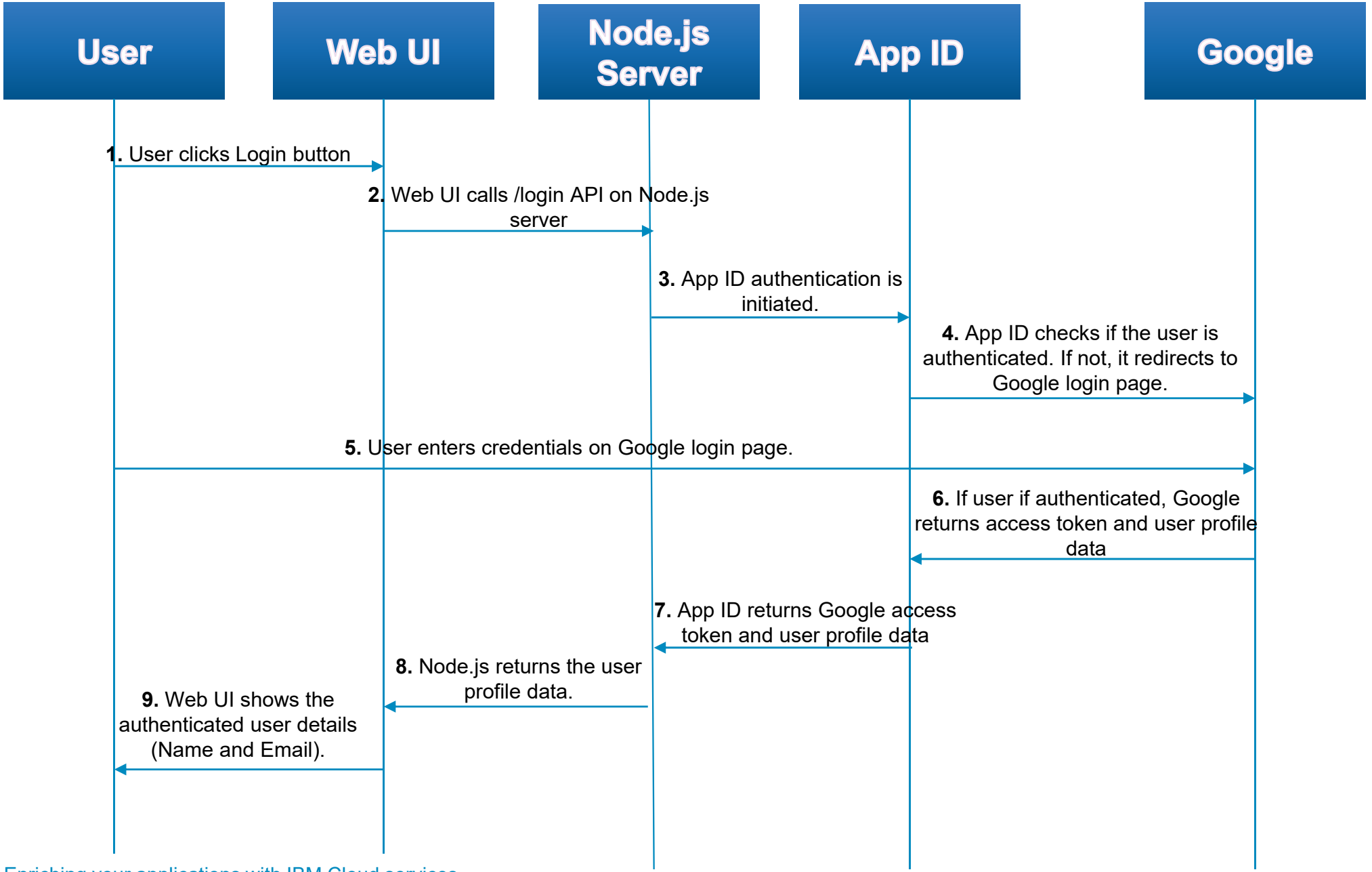
Identity providers	Type
Cloud Directory	Managed registry
SAML	Enterprise
Facebook	Social
Google+	Social
Custom	



App ID service: Scenarios and solutions (optional)

Scenario	Solution
You need to add authorization and authentication to your mobile and web apps but do not have a background in security.	App ID makes it easy to add an authentication step to your apps. You can add email or username sign in, social sign in, or enterprise sign in to your apps with APIs, SDKs, prebuilt UIs, or your own branded UIs.
You want to limit access to your apps and back-end resources.	You can secure your apps, back-end resources, and APIs easily by using the standards based authentication provided by App ID.
You want to build personalized app experiences for your users.	With App ID, you can store user data such as app preferences or information from their public social profiles, and then use that data to customize each experience of your app.
You want to manage users in a scalable way.	App ID allows you to create a Cloud Directory, which makes it possible for you to add user sign-up and sign-in to your apps. Cloud Directory provides you with the framework to maintain a user registry that can scale with your user base. With the pre-built functionality for self service, such as email verification and password resets, you can be sure that your app is authenticating users securely.

How is App ID used in Cognitive Tweets Analyzer



Retrieving and processing tweets

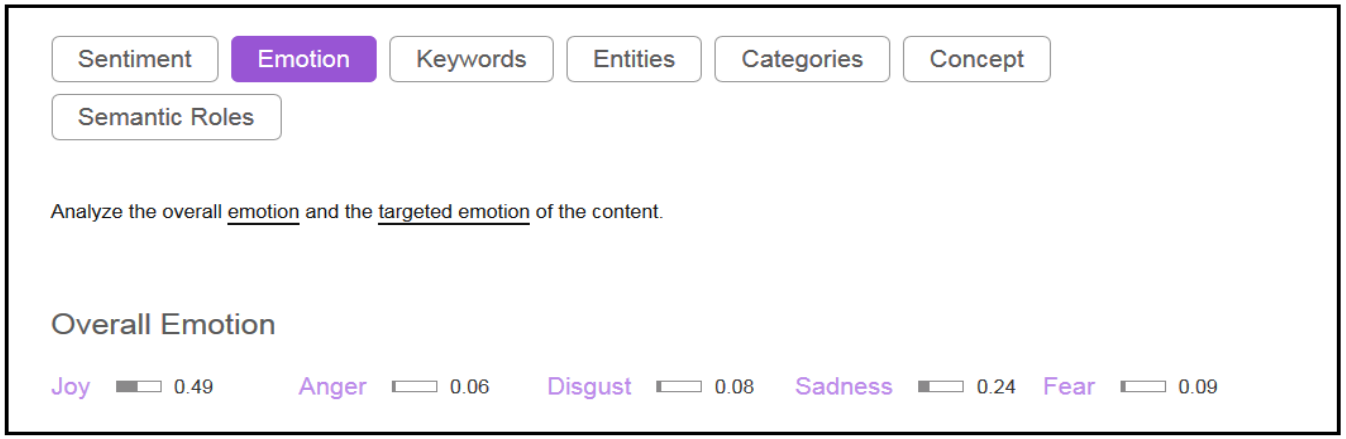
1. Get Twitter API keys
   - API key, API secret, Access token, and Access token secret
2. Install the Twitter library
   - **twit (**a Twitter API client for Node.js) is used in the Cognitive Tweets Analyzer app
3. Connect to Twitter Streaming APIs

```
1.  //initializing Twit Library
2.  const twitOptions = {};
3.   twitOptions.consumer_key = config.consumer_key || '';
4.   twitOptions.consumer_secret = config.consumer_secret || '';
5.   twitOptions.access_token = config.access_token;
6.   twitOptions.access_token_secret = config.access_token_secret;
7.   twitOptions.timeout_ms = 60 * 1000; // optional HTTP request timeout to apply to all
    requests.
8.   this.twitterClient = new Twit(twitOptions);

9.  //Calling Twit stream API
10.  const twitParams = {};
11.  twitParams.lang = 'en';
12.  twitParams.follow = this.options.userIds;
13.  this.status.listening = this.options.listenTo;
14.  this.stream = this.twitterClient.stream('statuses/filter', twitParams);
```

Watson Natural Language Understanding service

- NLU extracts meaning from unstructured data.
- NLU processes input provided as text, HTML, or a public URL.
- NLU analyzes semantic features of input text and provides an output that includes:
  - Categories
  - Emotion
  - Entities and relationships
  - Sentiment analysis
  - Keywords



- It can be trained by creating a custom model by using IBM Watson Knowledge Studio.
- Example applications: Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

How is Natural Language Understanding used in Cognitive Tweets Analyzer

1. Configure Natural Language Understanding API with the required features:

```
1.  //configuring nlu parameters to be sent
    to the nlu analyze api
2.    nluParams = {
3.      features: {
4.        emotion: {},
5.        sentiment: {},
6.        entities: {
7.          emotion: false,
8.          sentiment: false,
9.          limit: 2
10.       },
11.       keywords: {
12.         emotion: false,
13.         sentiment: false,
14.         limit: 2
15.       }
16.     }
17.   };
```

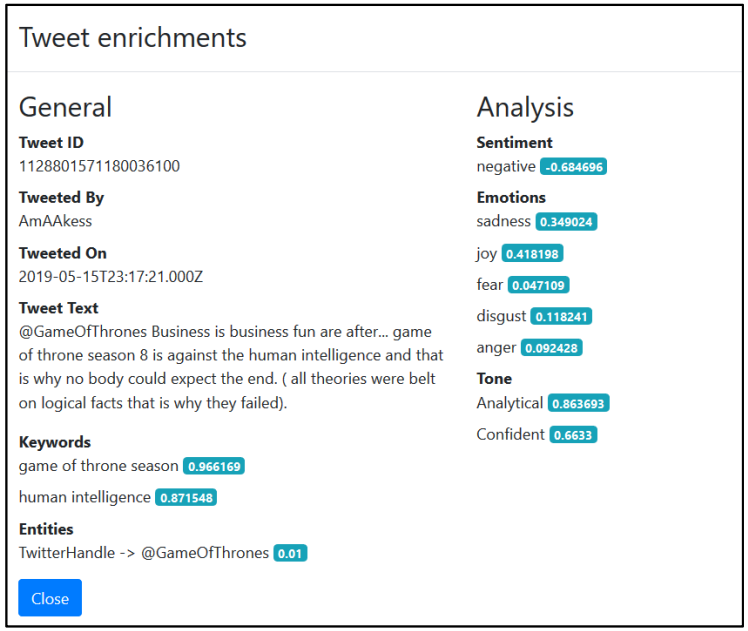
2. Call Natural Language Understanding **analyze** API

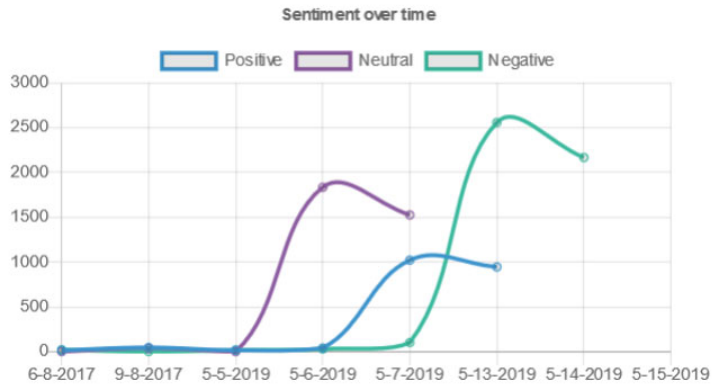
```
1.  try {
2.      this.nluParams.text = text;
3.      this.nluParams.language = 'en';
4.
    this.nlu.analyze(this.nluParams, (err,
    success) => {
5.   if (err) {
6.      this.LOGGER.error('NLU: ' + err);
7.      return reject('NLU: ' + err);
8.   }
9.   resolve({ nlu: success });
10.  });
11.  } catch (err) {
12.     reject(err);
13.  }
```

How is Natural Language Understanding used in Cognitive Tweets Analyzer

- Sample output:

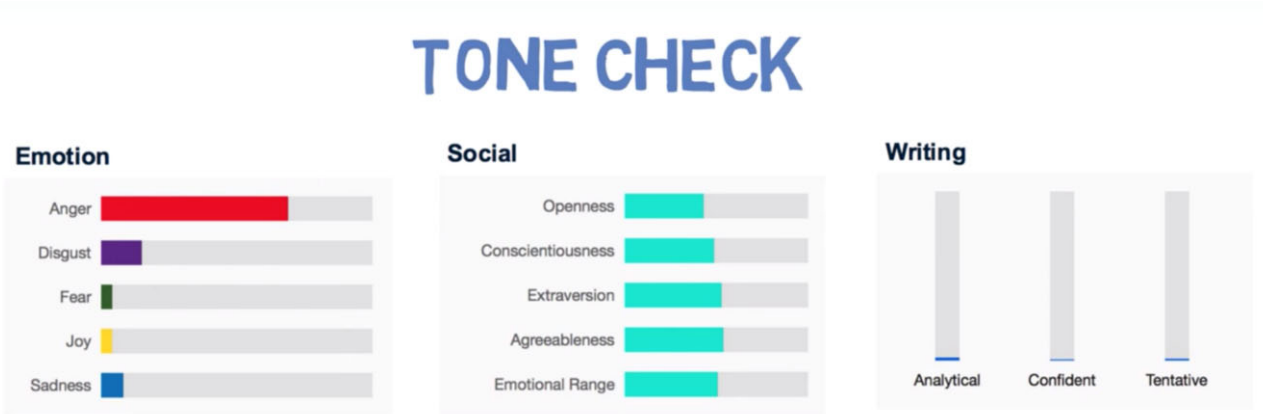
- usage
  - text_units : 1
  - text_characters : 159
  - features : 4
- sentiment
  - document
    - score : 0.970327
    - label : "positive"
  - language : "en"
- keywords
  - 0
    - text : "incredible episode"
    - relevance : 0.989512
    - count : 1
  - 1
    - text : "last night of #GameOfThrones Check"
    - relevance : 0.789245
    - count : 1
- entities
  - 0
    - type : "Hashtag"
    - text : "#TheLastoftheStarks"
    - relevance : 0.01
    - count : 1
  - 1
    - type : "Hashtag"
    - text : "#GameOfThrones"
    - relevance : 0.01
    - count : 1
- emotion
  - document
    - emotion
      - sadness : 0.110331
      - joy : 0.76558
      - fear : 0.01976
      - disgust : 0.065466
      - anger : 0.026278





Watson Tone Analyzer service

- Watson Tone Analyzer uses linguistic analysis to identify various tones. It detects three types of tones from text:
  - Emotion (anger, disgust, fear, joy, and sadness)
  - Social tendencies (openness, conscientiousness, extroversion and introversion, agreeableness, and emotional range)
  - Language/Writing styles (analytical, confident, and tentative)
- The service offers two endpoints:
  - General-purpose endpoint: Used to analyze shorter web data, such as email messages or tweets, or longer documents, such as articles or blog posts.
  - Customer-engagement endpoint: Used to monitor customer service and support conversations.



How is Tone Analyzer used in Cognitive Tweets Analyzer

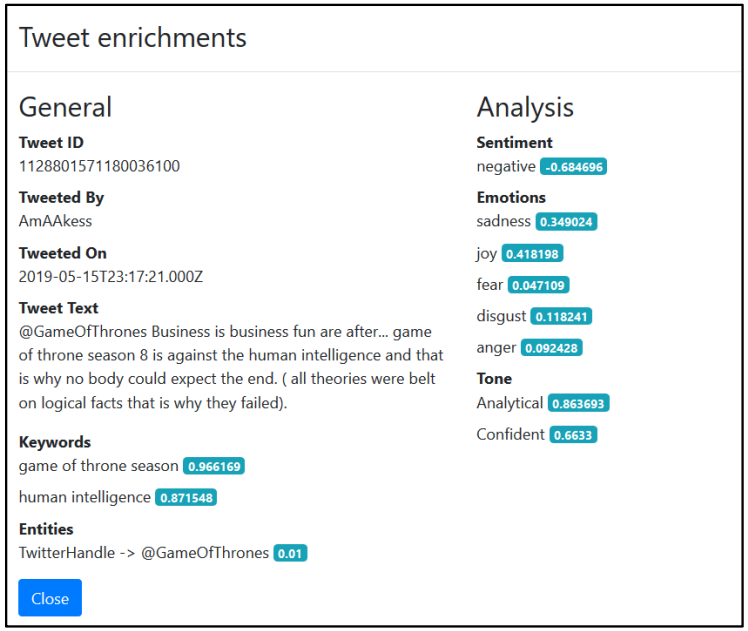
- Call Tone Analyzer **tone** API passing the text to be analyzed

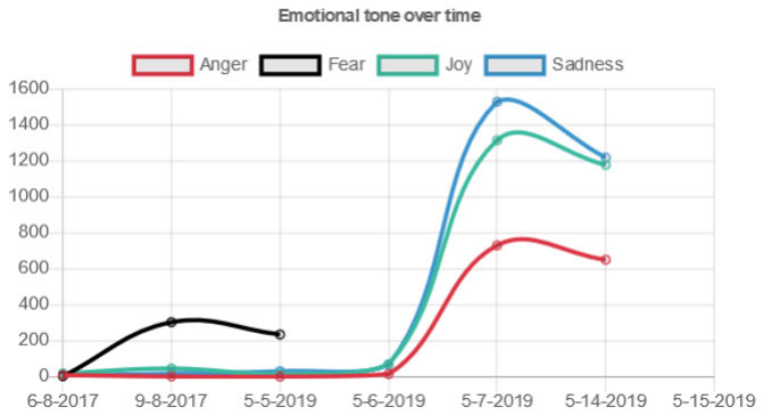
```
1.   try {
2.           this.toneParams.text = text;
3.           this.toneParams.sentences = false;
4.           this.toneAnalyzer.tone(this.toneParams, (err, success) => {
5.             if (err) {
6.               this.LOGGER.error('Tone: ' + err);
7.               return reject('Tone: ' + err);
8.             }
9.             resolve({ tone: success });
10.          });
11. } catch (err) {
12.   reject(err);
13. }
```

How is Tone Analyzer used in Cognitive Tweets Analyzer

- Sample output:

- document_tone
  - tones
    - 0
      - score : 0.76558
      - tone_id : "joy"
      - tone_name : "Joy"
    - 1
      - score : 0.687768
      - tone_id : "analytical"
      - tone_name : "Analytical"





Cloudant database

- Fully managed
  - NoSQL database that is conceived as a native JSON database.
    - The Twitter API returns tweets encoded using JSON.
  - Provides a fully managed, distributed JSON document database.
  - Easy to deploy and create databases on IBM Cloud.
  - Scale throughput capacity and data storage to meet application's requirements.
- Secure
  - ISO27001, SOC 2 Type 2 compliant, and HIPAA ready. All data is encrypted over the wire and at rest.
- Globally available
  - Available in all IBM Cloud regions and 55+ data centers across the world.
  - Easy to set up for disaster recovery between continents.
  - Easy to set up for scaling an app through a horizontal scaling architecture that can handle millions of users and terabyte.
- Data flexibility
  - Flexible JSON schema and powerful API that is compatible with Apache CouchDB
- Durable replication
  - Move application data closer to all the places it needs to be, for uninterrupted data access, offline or on.

How is Cloudant used in Cognitive Tweets Analyzer

1. Initialize Cloudant instance by using Cloudant credentials.

```
1.  const cloudant = Cloudant({
2.     account: config.cloudant_username,
3.     password: config.cloudant_password,
4.     plugins: { retry: { retryErrors:
    false, retryStatusCodes: [429] } }
5.      });
6.  this.cloudant = cloudant;
```

2. Call Bulk Save to Cloudant

```
1.  this.LOGGER.debug('Saving to
    Cloudant...');
2.  this.cloudantDB.bulk(this.bulkSaveBuffer,
    (err, result) => {
3.     if (err) {
4.      this.LOGGER.error('Error while saving
    to database::' + err);
5.     reject(err);
6.  } else {
7.    this.LOGGER.debug('Successfully saved '
    +
8.    this.bulkSaveBuffer.docs.length +
9.   ' docs to Cloudant.');
10. this.bulkSaveBuffer.docs = [];
11.   resolve();
12.  }
13. });
```

LogDNA service

- Centralized cloud log management software
  - Aggregates all system and application logs in one centralized logging system
  - Third-Party service from LogDNA https://logdna.com/
- Features:
  - Troubleshoot logs in real time to diagnose issues and identify problems
  - Automatic parsing and indexing of log sources
  - Keyword-based log search and graphing
  - Get alert notifications of important events and errors
  - Provides easy and fast integration of various log sources

IBM Cloud Log Analysis with LogDNA Service

- Based on the LogDNA Third-Party product
- Operated by LogDNA in partnership with IBM
- Offers administrators, DevOps teams, and developers advanced features to filter, search, and tail log data, define alerts, and design custom views to monitor application and system logs.
- Does *not* support Cloud Foundry

Note: In the Cognitive Tweets Analyzer application, LogDNA (not IBM Cloud Log Analysis with LogDNA) is used due to the lack of Cloud Foundry support in the IBM Cloud offering

Integrating LogDNA with Cognitive Tweets Analyzer

- The Cognitive Tweets Analyzer Cloud Foundry application integrates LogDNA through a user-provided service instance.
- User-provided service instances enable developers to use services that are not available in IBM Cloud with their apps running on Cloud Foundry.
  - You can bind your application to services outside of IBM Cloud.
  - Once created, user-provided service instances behave like service instances that are created through the IBM Cloud catalog.
- To use a user-provided service with your Cloud Foundry application on IBM Cloud, you need:
  - A service outside of IBM Cloud (LogDNA in this use case)
  - Host and port to the external service (endpoint URL)
  - Credentials to access the user-provided service (if required by the service)
  - Create the user-provided service instance
  - Your application binds to the user-provided service

Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

- Summary of integration steps. Ensure that you have the IBM Cloud CLI installed in your workstation:

1. Login to LogDNA at https://app.logdna.com
2. Create your organization
3. Choose to connect **Via Platform** > **Cloud Foundry**
4. Select **Provision a syslog port for Cloud Foundry.**
5. You will receive a new LogDNA syslog **endpoint**, which includes the host and port that you need to connect to the LogDNA service. Example: **syslog-a.logdna.com:15177**

6. Run the following ibmcloud CLI command to create the user-provided services, name it my-logs and link it to the syslog url:
   ibmcloud cf cups my-logs -l syslog://syslog.logdna.com:<syslog-port>
   where <syslog-port> is the syslog port you provisioned in step 5.

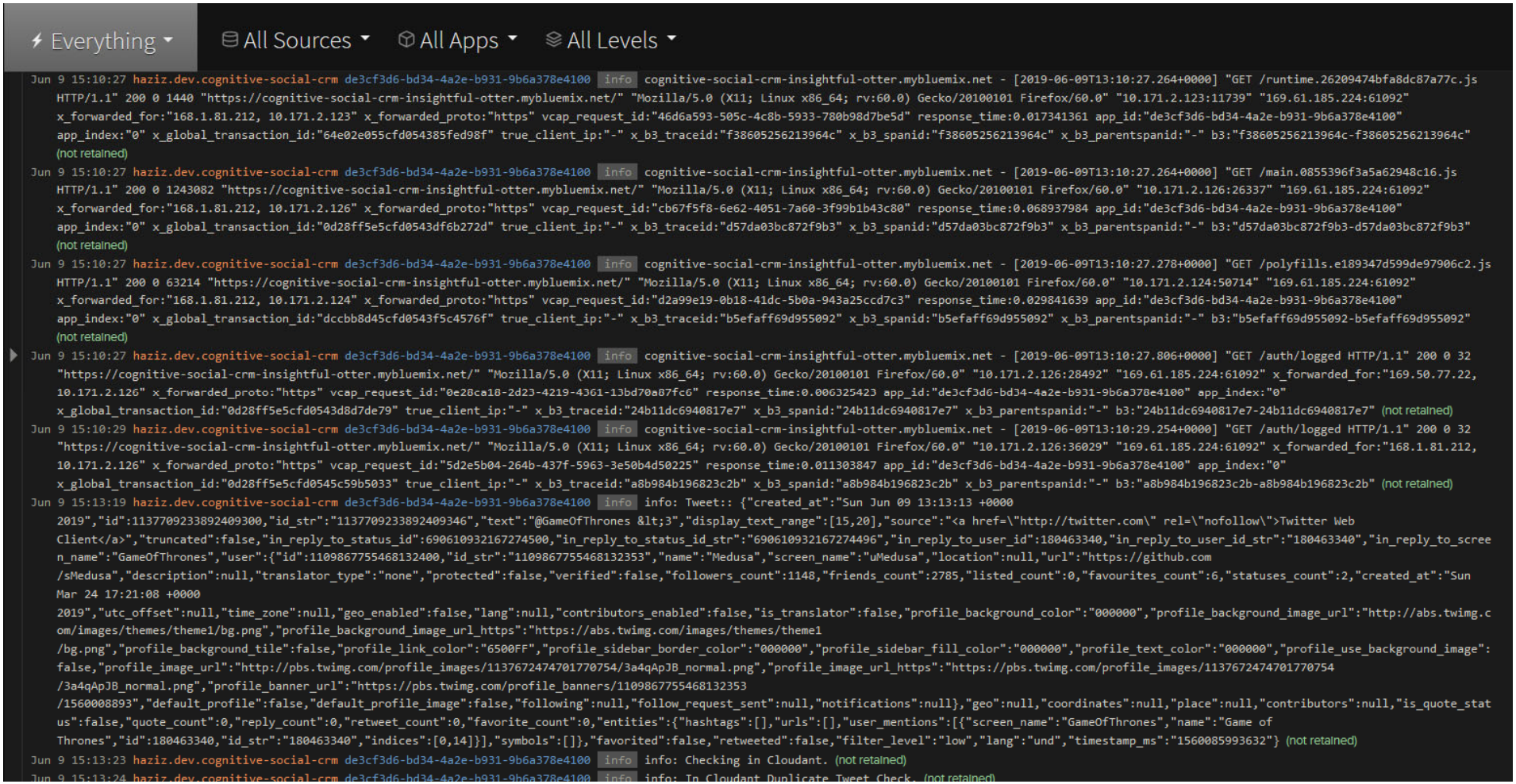
7. Run the following command to bind the user-provided service to your Cloud Foundry application:

   ibmcloud cf bind-service <CF APP NAME> my-logs
   where <CF APP NAME>, is the Cloud Foundry application name.

8. Run the following command, to restart the application after binding it to the LogDNA service.
   ibmcloud cf restart <CF APP NAME>
   where <CF APP NAME>, is the Cloud Foundry application name

Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

- Sample logs on LogDNA Dashboard



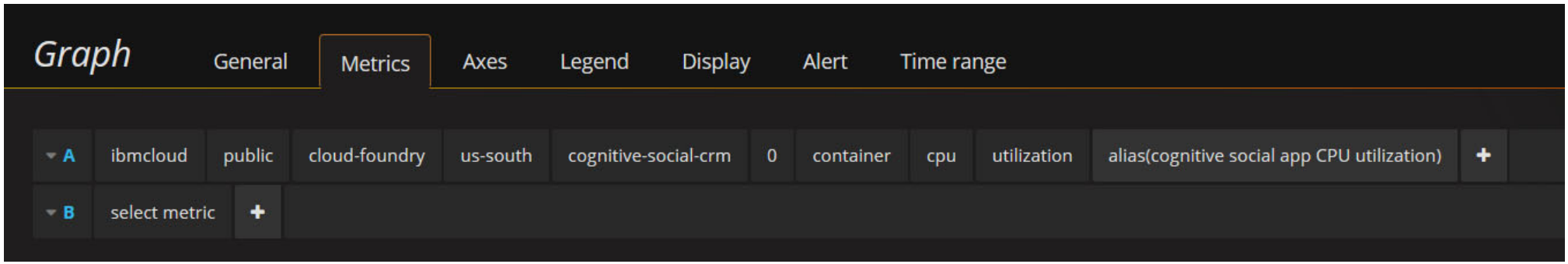
IBM Cloud Monitoring service

- Expand your collection and retention capabilities when working with metrics.
- Define rules and alerts that notify you of conditions that require attention.
- Gain insight into how your apps are performing and consuming resources.
- Identify trends, detect and diagnose problems.
- Integrate your monitoring data into your applications and operations through the Monitoring service APIs.
- Send metrics for your Cloud Foundry applications and Virtual Machines (VMs) into the Monitoring service.
- Provision the Monitoring service through the IBM Cloud catalog.
- View and analyze metrics collected by the Monitoring service through the IBM Cloud dashboard.

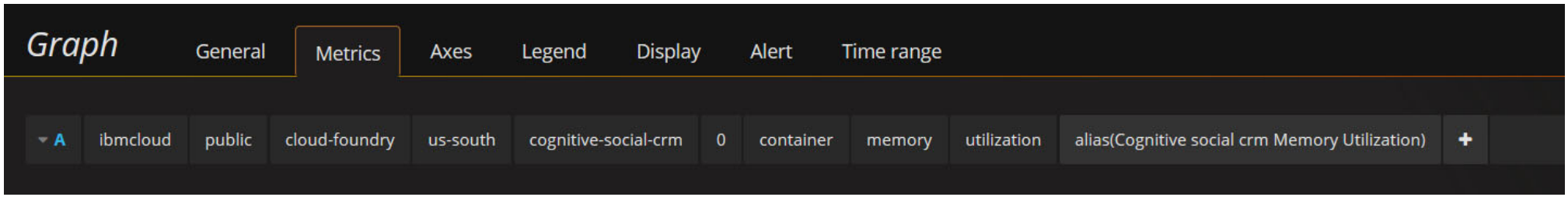
Integrating Monitoring with Cognitive Tweets Analyzer

- On the Monitoring service dashboard, create a Grafana dashboard to monitor the metrics that you want to display.
- On the Monitoring service dashboard, create a Grafana dashboard to monitor the metrics that you want to display.

- CPU utilization metric

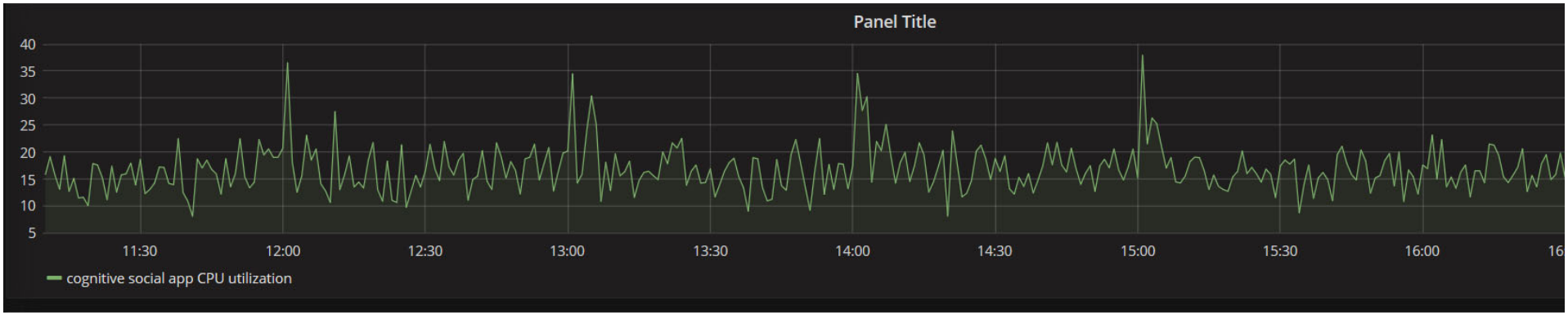


- Memory utilization metric

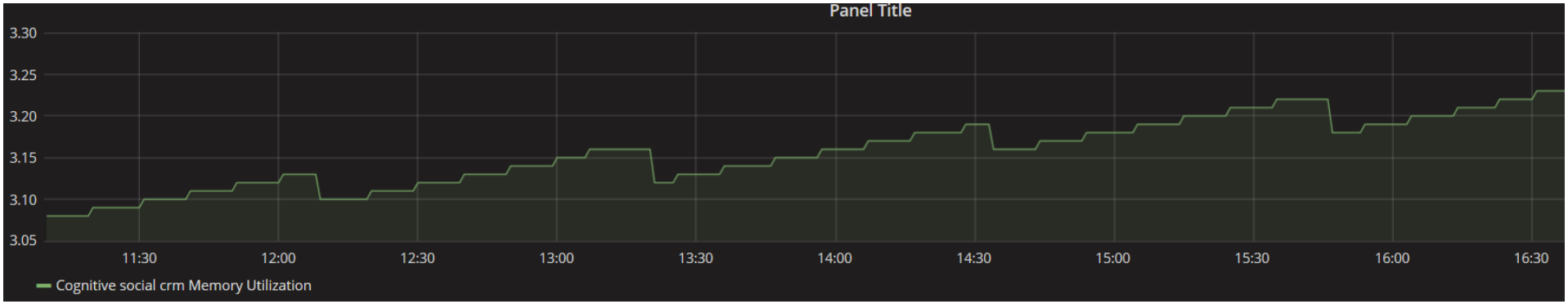


Integrating Monitoring with Cognitive Tweets Analyzer

- CPU utilization metric: Sample output



- Memory utilization metric: Sample output



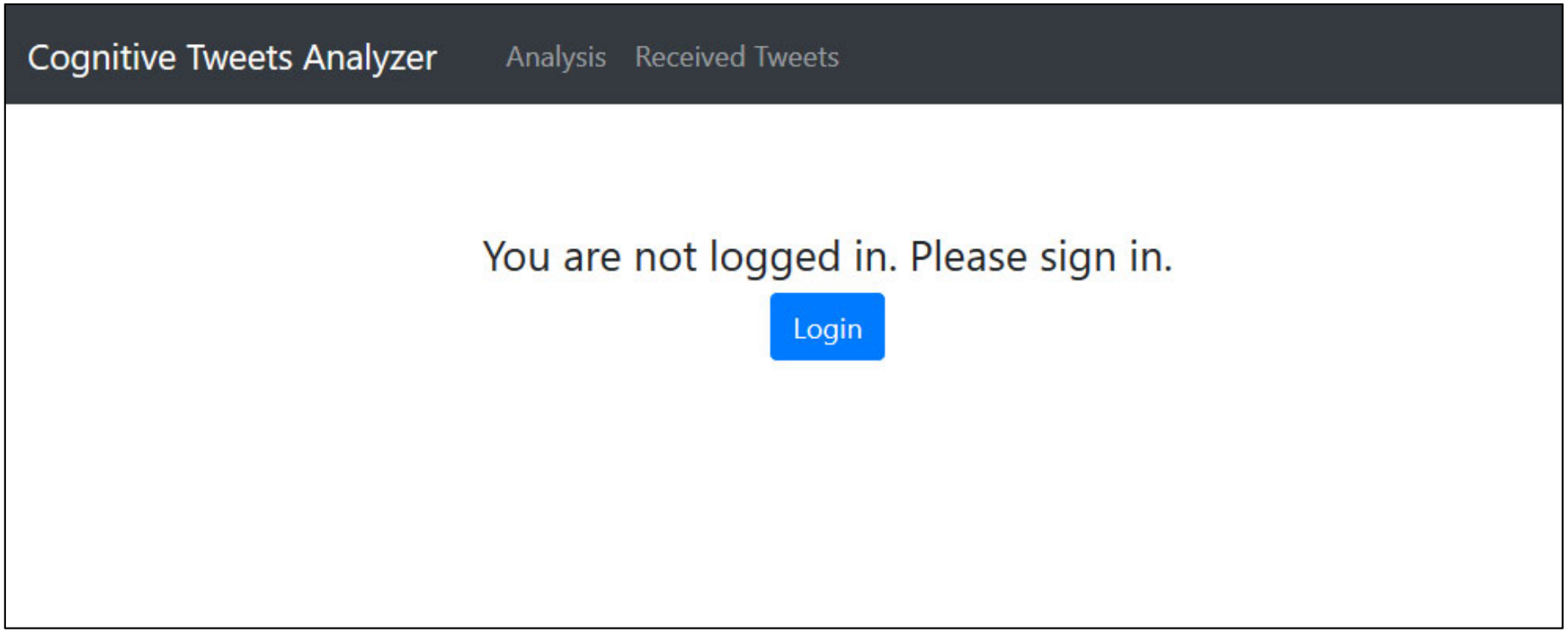
Cognitive Tweets Analyzer: Demonstration

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

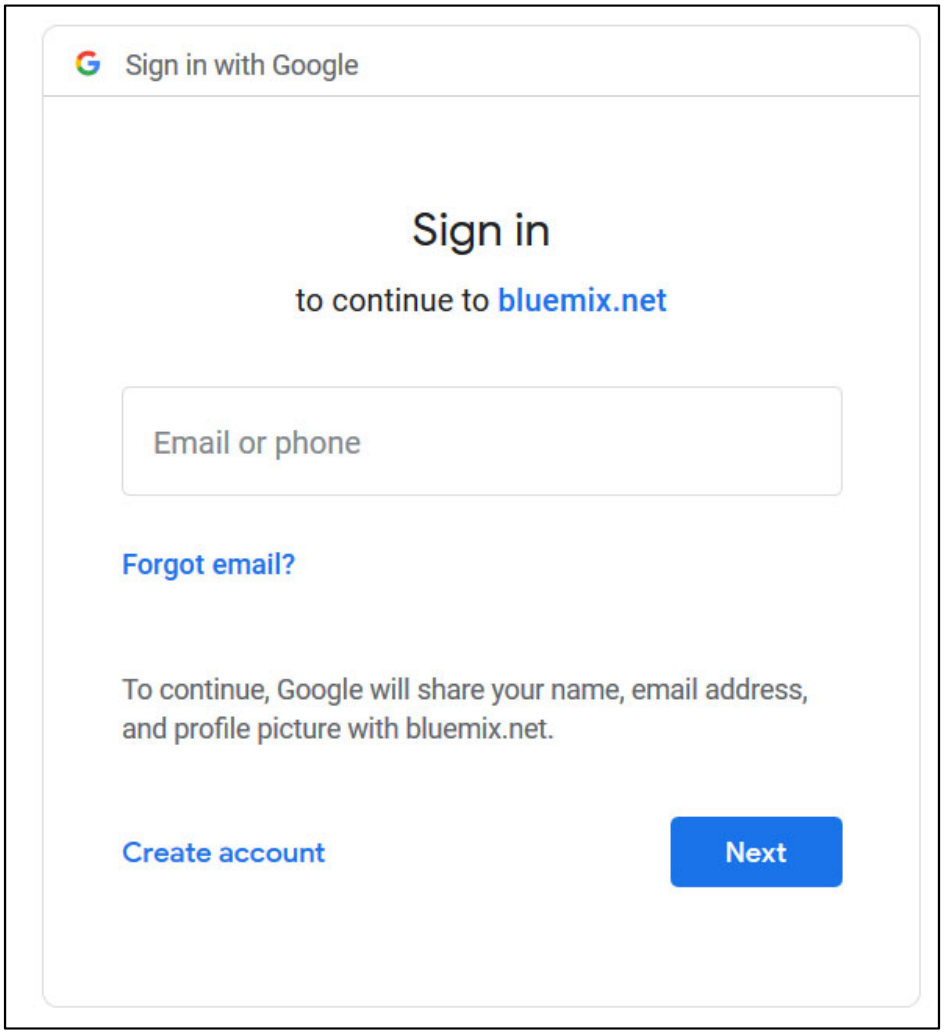
Cognitive Tweets Analyzer App: Demo

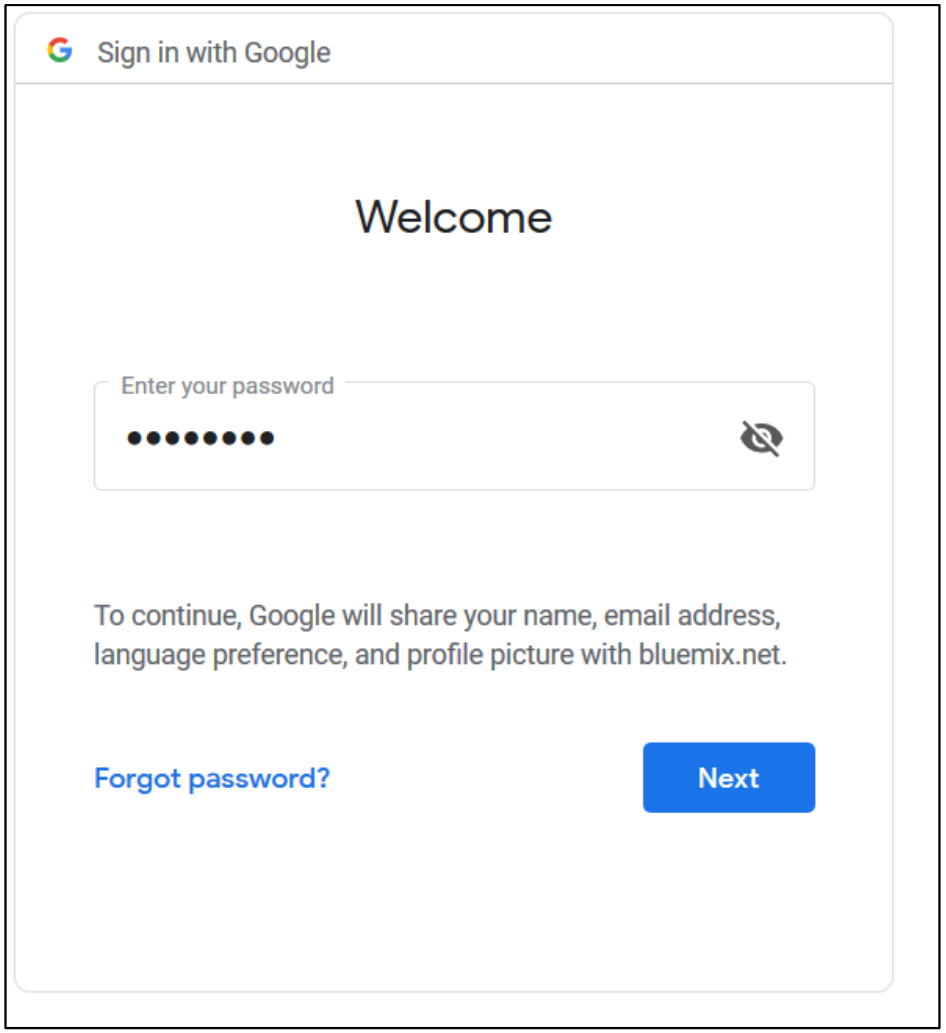
1. User opens the Cognitive Tweet Analyzer app in his browser.



Cognitive Tweets Analyzer App: Demo (cont.)

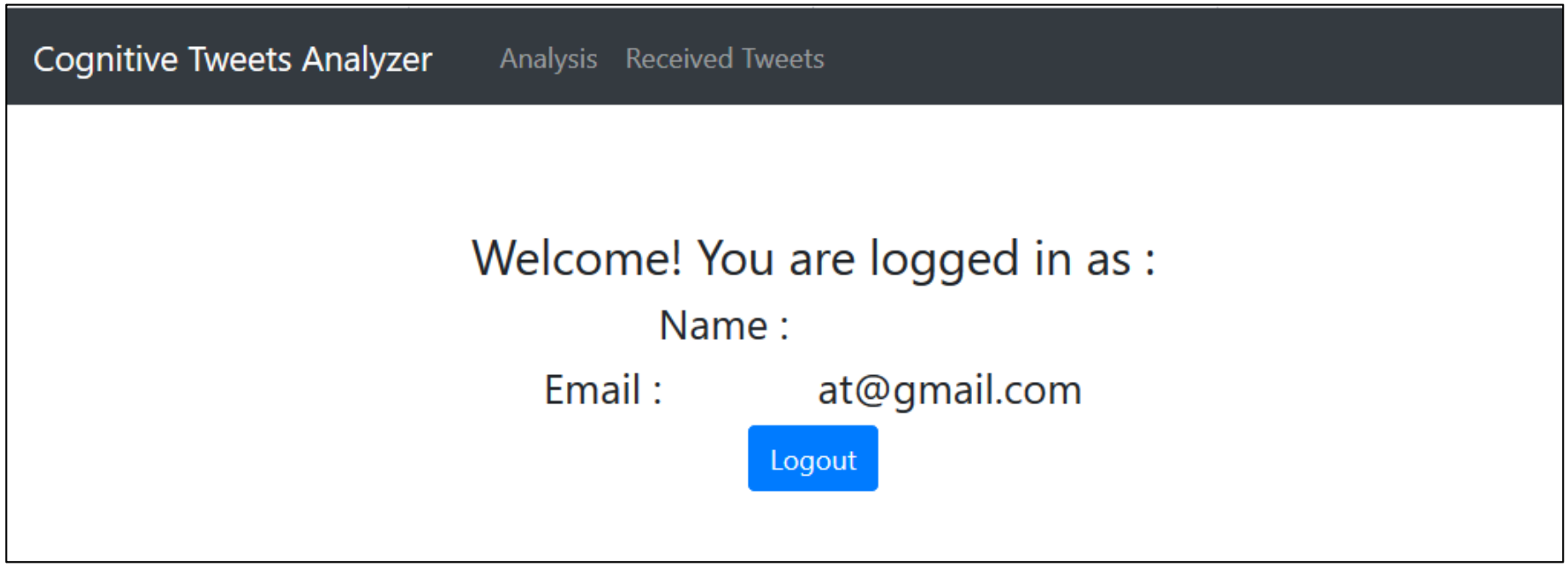
2. User logs in with a Google account





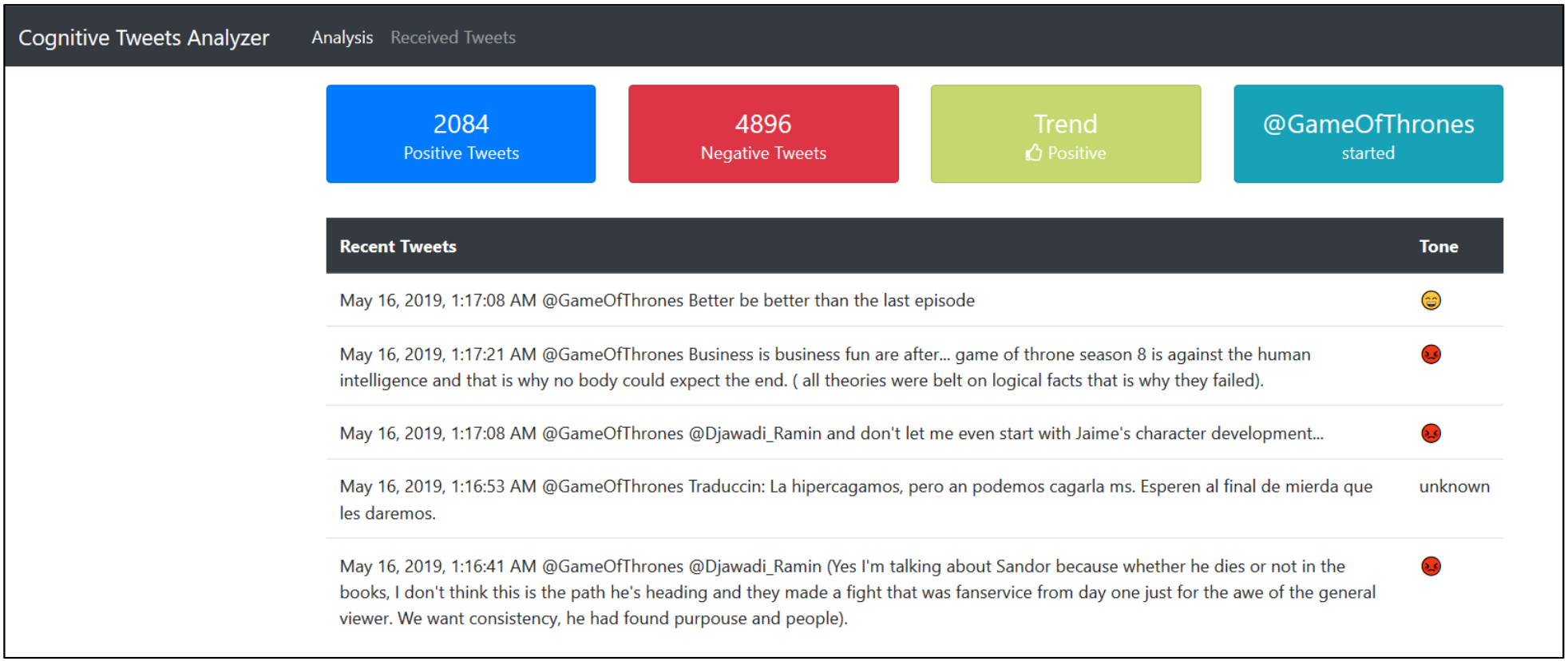
Cognitive Tweets Analyzer App: Demo (cont.)

3. After successful log in, the user's name and email are displayed on the page.



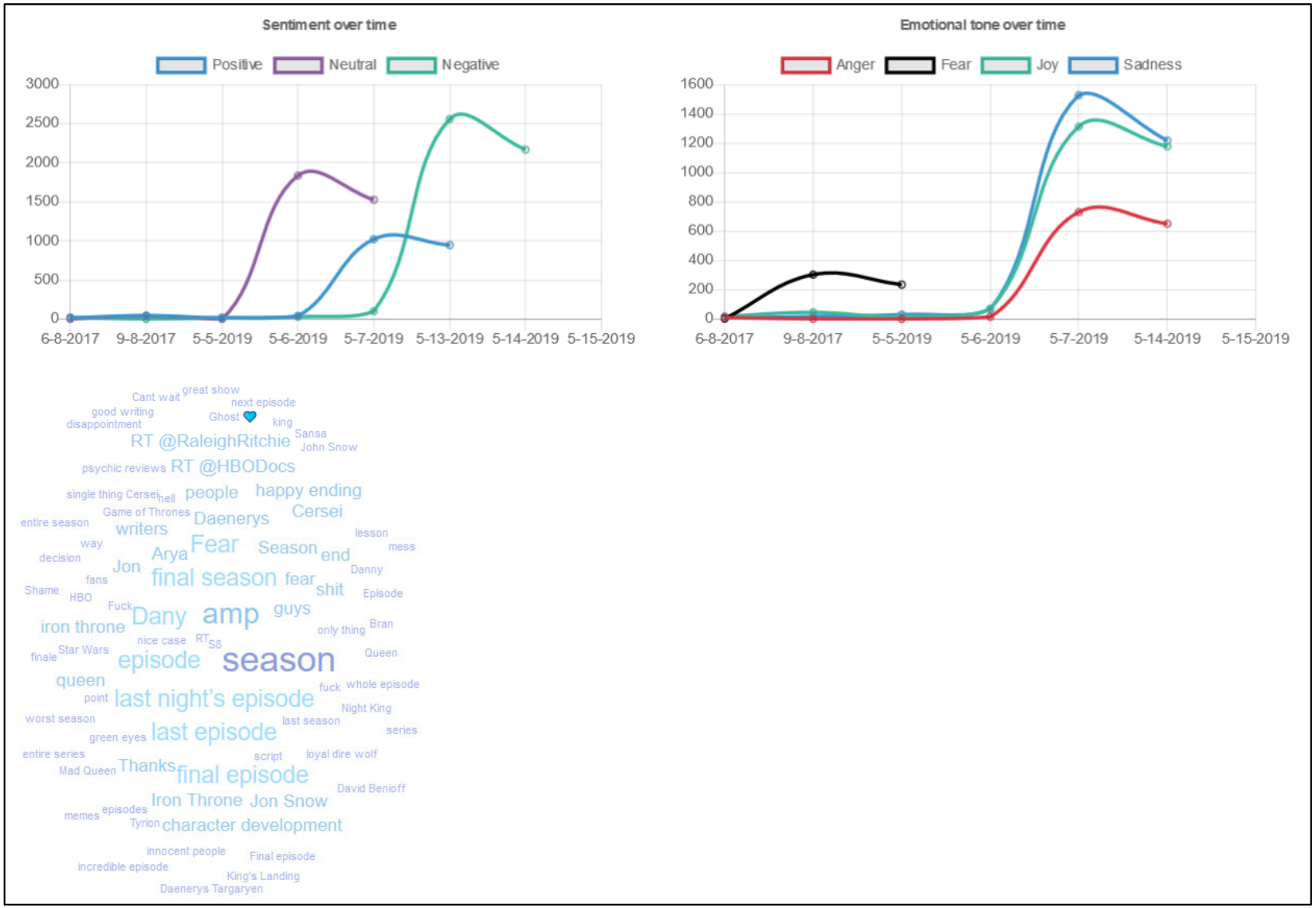
Cognitive Tweets Analyzer App: Demo (cont.)

4. The user opens the Analysis tab, to visualize the analysis data on “GameOfThrones” hashtag.



Cognitive Tweets Analyzer App: Demo (cont.)

5. Analyze data graphs in the Analysis tab.



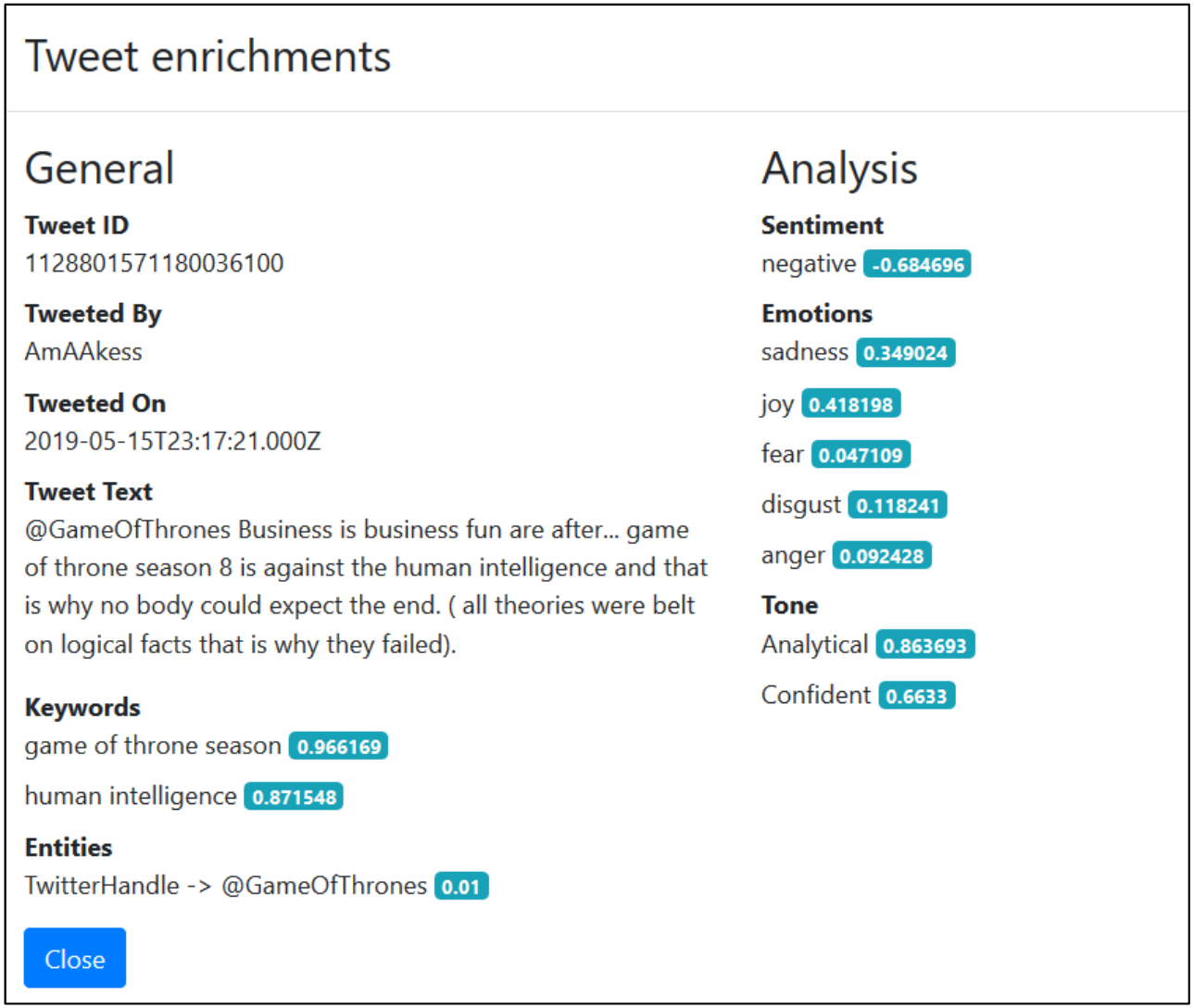
Cognitive Tweets Analyzer App: Demo (cont.)

6. The user opens the Tweets tab to display the list of tweets that are pushed to the application, which is sorted by posted date.

Received	Text	Tone
May 16, 2019, 1:17:08 AM	@GameOfThrones Better be better than the last episode	joy
May 16, 2019, 1:17:21 AM	@GameOfThrones Business is business fun are after... game of throne season 8 is against the human intelligence and that is why no body could expect the end. (all theories were belt on logical facts that is why they failed).	joy
May 16, 2019, 1:17:08 AM	@GameOfThrones @Djawadi_Ramin and don't let me even start with Jaime's character development...	sadness
May 16, 2019, 1:16:53 AM	@GameOfThrones Traduccin: La hipercagamos, pero an podemos cagarla ms. Esperen al final de mierda que les daremos.	joy
May 16, 2019, 1:16:41 AM	@GameOfThrones @Djawadi_Ramin (Yes I'm talking about Sandor because whether he dies or not in the books, I don't think this is the path he's heading and they made a fight that was fanservice from day one just for the awe of the general viewer. We want consistency, he had found purpouse and people).	sadness
May 16, 2019, 1:16:36 AM	@GameOfThrones No need to watch. Jon will off Daenerys and go north. Arya will wander off hither and yon. Sansa will rule the North. And Bran will become the ruler of the seven kingdoms -- which is absurd and disappointing beyond belief.	sadness
May 16, 2019, 1:15:06 AM	@GameOfThrones @Djawadi_Ramin The worst season ever	sadness
May 16, 2019, 1:15:21 AM	@GameOfThrones This season was wack as hell, I hope to God someone was dreaming or something please! Make it make sense	anger
May 16, 2019, 1:15:19 AM	@GameOfThrones @Djawadi_Ramin Redo this. Nonsensical character arcs, nonsensical character outcomes and even some nonsensical deaths and a fanservice fight made exclusively for the general viewer's awe without respecting his char	sadness

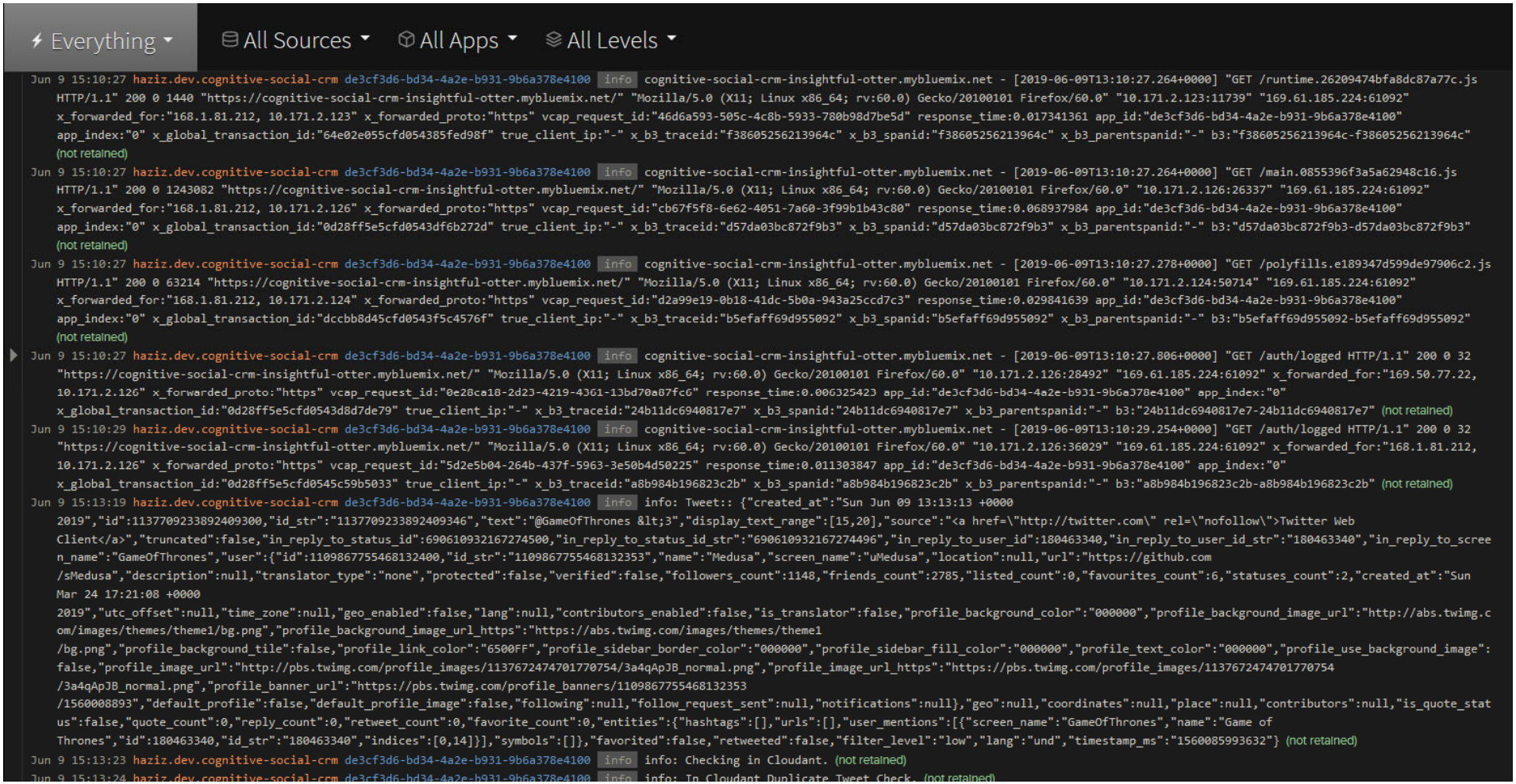
Cognitive Tweets Analyzer App: Demo (cont.)

7. When the user clicks one of the tweets that are listed in the Tweets tab, the sentiment and emotional tone analysis data is shown in the details view.



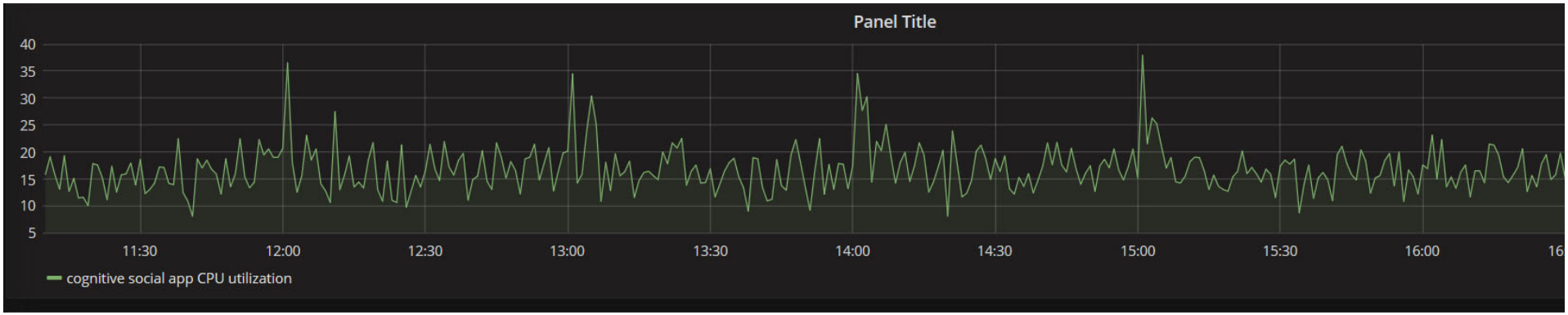
Cognitive Tweets Analyzer App: Demo (cont.)

- Sample app logs displayed in the LogDNA dashboard

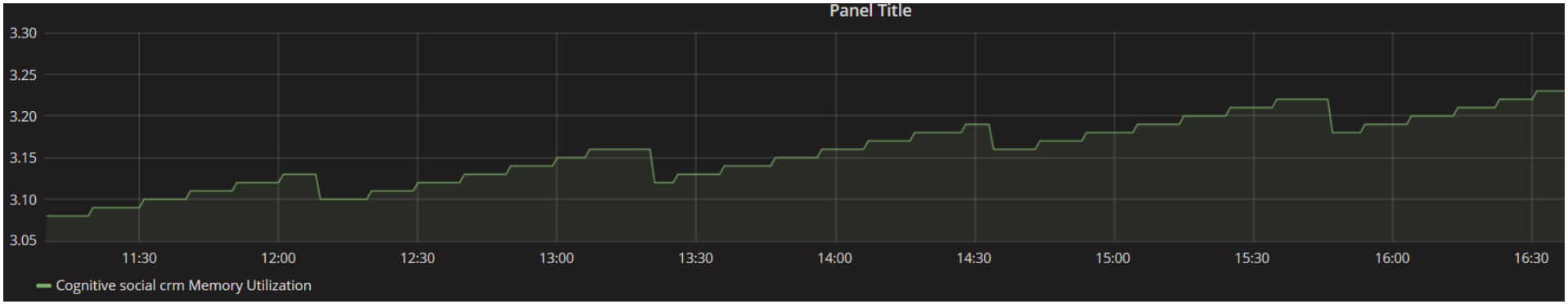


Cognitive Tweets Analyzer App: Demo (cont.)

- CPU utilization metric: Sample output



- Memory utilization metric: Sample output



Unit summary

- Define your business problem and goals
- Identify functional and non-functional requirements
- Select the technical components that best fit your solution
- Design a simple architecture for a cloud application
- Identify services in the IBM Cloud catalog that you can use to enrich your cloud apps
- Describe App ID, Watson Natural Language Understanding, Watson Tone Analyzer, LogDNA, and IBM Cloud Monitoring services and their integration in the sample use case

Exercise 4: Securing a web application with single sign-on (optional)

Exercise objectives

- In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.
- After completing this exercise, you should be able to:
  - Create an App ID service.
  - Bind the App ID service to an application to add single sign-on capability to the application.
  - Describe different configurations in the App ID service.