

# IBM Watson

## **What is Watson?**

The platform of cognitive business which is a system that reasons about problems like we humans does.

➤ The following flow is followed in cognitive processes:

1. Visible phenomenon
2. Interpret: to generate hypothesis about what it means
3. We evaluate which is right or wrong
4. Finally, we decide which hypothesis.

## **Let's Get Started with Building the Apps in our account**

### **But First:**

### **Getting started with Watson Developer Cloud and Bluemix**

- The IBM Watson™ Developer Cloud (WDC) offers a variety of services for developing cognitive applications.
- Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service.
- Some services, such as the Speech to Text service, provide additional interfaces.

### **Getting started with Bluemix**

- IBM Bluemix® is the cloud platform in which you deploy applications that you develop with Watson Developer Cloud services.
- The Watson Developer Cloud documentation provides information for developing applications with Watson services in [Bluemix.Community](#)

## Developing a Watson application in Node.js

1. Satisfy the prerequisites to prepare for working with the sample Node.js application.
2. Prepare the application to obtain the source code for the application and get it ready for deployment in Bluemix.
3. Deploy and run the application in Bluemix to upload the application to the Bluemix environment and run it from there.
4. Run the application locally to run the application from your local machine after deploying and running it in Bluemix.

### Satisfy the prerequisites

Before developing an application, make sure you satisfy any of the following prerequisites you do not already meet:

- **Obtain a Bluemix ID:** Register for Bluemix to obtain a Bluemix ID by clicking the **SIGN UP** button at [console.ng.bluemix.net](https://console.ng.bluemix.net).
- **Install the Cloud Foundry command-line tool**

[https://github.com/cloudfoundry/cli?cm\\_mc\\_uid=70500961251414634213050&cm\\_mc\\_sid\\_50200000=1465589467#downloads](https://github.com/cloudfoundry/cli?cm_mc_uid=70500961251414634213050&cm_mc_sid_50200000=1465589467#downloads)

OR

### commands to install

Step 1: brew tap cloudfoundry/tap

Step 2: brew install cf-cli

- **Install the Node.js runtime:**

[https://nodejs.org/en/?cm\\_mc\\_uid=70500961251414634213050&cm\\_mc\\_sid\\_50200000=1465589467](https://nodejs.org/en/?cm_mc_uid=70500961251414634213050&cm_mc_sid_50200000=1465589467)

## GitHub Repo IBM Watson API

[https://github.com/watson-developer-cloud?cm\\_mc\\_uid=70500961251414634213050&cm\\_mc\\_sid\\_50200000=1465589467](https://github.com/watson-developer-cloud?cm_mc_uid=70500961251414634213050&cm_mc_sid_50200000=1465589467)

## Prepare the application

1. Obtain the source code
2. Modify the manifest file.
3. Note service and plan information.

## Deploy and run the application in Bluemix

1. Set the target URL for the API. Use the `cf api` command to specify the endpoint for contacting the Watson services APIs:

➤ `cf api https://api.ng.bluemix.net`

2. Log in to Bluemix. Use the `cf login` command to authenticate with your Bluemix ID and password:

➤ `cf login -u username -p password`

3. Find the name of the service and plan.

4. Create an instance of the service. The third argument is the name of the service instance specified by the `services` attribute of the `manifest.yml` file.

➤ `cf create-service service-name service-plan service-instance-name`

For example, you would issue the following command to create the service instance for the Personality Insights service:

➤ `cf create-service personality_insights standard personality-insights-service-standard`

5. Push the application to Bluemix. The sole argument to the command is the name of your application specified in the manifest.yml file.

➤ `cf push application-name`

The `cf push` command obtains the file to be pushed from the pathname specified in the manifest.yml file. The command includes an optional `-p` option that lets you specify the pathname of the application, for example, the .zip file for a Node.js application.

6. Access the application in a browser.

### Run the application locally

1. Run your application in Bluemix.
2. Log in to Bluemix. Use the `cf login` command to authenticate with your Bluemix ID and password:

➤ `cf login -u username -p password`

3. Obtain your Bluemix credentials and URL. Use the `cf env` command to view the environment variables for your running application:

➤ `cf env application-name`

4. Use the URL and your credentials with the application.

5. url: '<url>',

6. username: '<username>',

password: '<password>'

7. Install required node packages.

➤ `npm install`

8. Start the application. Enter the node command with the argument `app.js` to start your application:

➤ `node app.js`

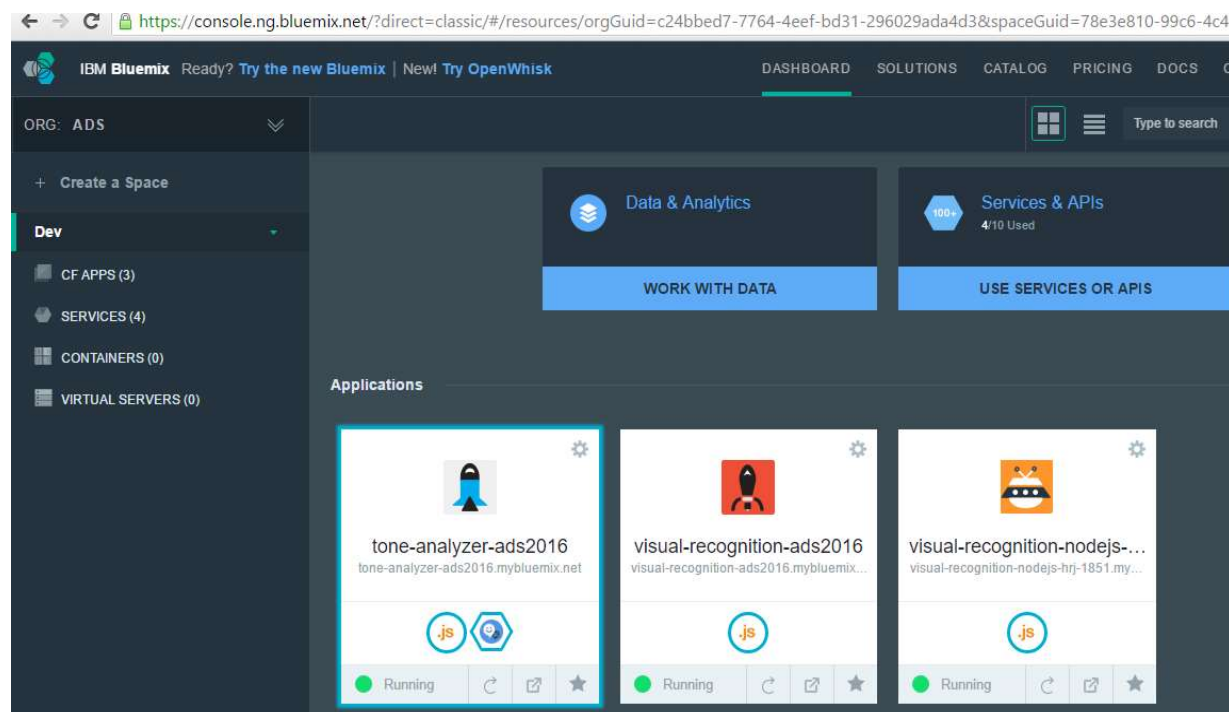
9. Access the application in a browser (<http://localhost:3000>)

OR

10. Access your app by entering the following URL into your browser:

➤ [tone-analyzer-ads2016.mybluemix.net](http://tone-analyzer-ads2016.mybluemix.net)

## 11. Home Page after Building the Apps:



12.

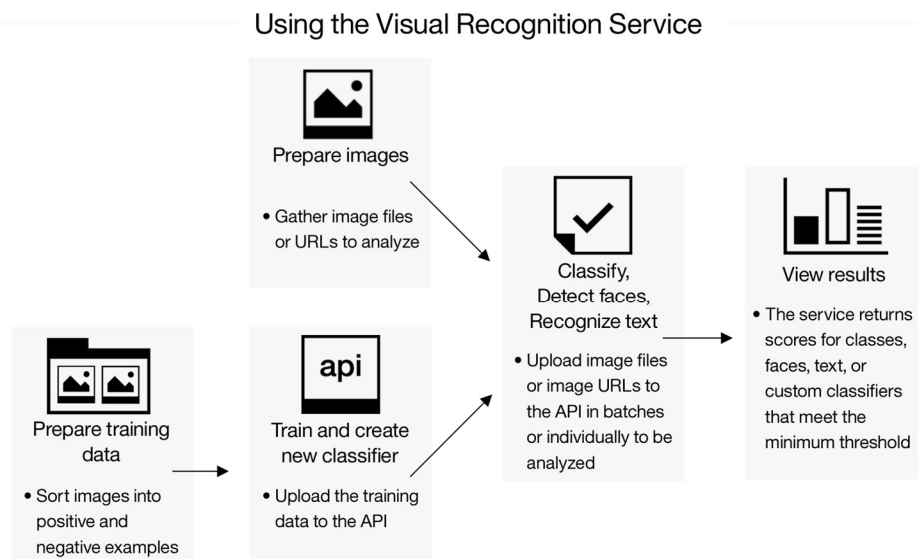
## Demo 1: Visual Recognition:

- The IBM Watson™ Visual Recognition service uses deep learning algorithms to analyze images (.jpg, .gif, or .png) for scenes, objects, faces, text, and other content, and return keywords that provide information about that content.
- In classify calls these keywords are called **classes**, and there are hundreds built in to the service and available for use immediately.
- You can use these classes to identify relationships between images and create applications that can utilize those relationships.

- When you classify an image, each identified class receives a **score**. Scores can range from 0 - 1, with higher scores indicating greater correlation.
- You can also train a group of new classes to create your own custom classifier with the Visual Recognition service.
- During training, you can supply images to create two or more classes that are trained against each other.
- These classes are grouped into a **classifier** and maintain their relationship later, when you use the custom classifier to analyze an image.
- This allows you to create a multi-faceted classifier that is specific to your needs.

### Flow of the service

The Visual Recognition services comes with a set of built-in classes so that you can analyze images with high accuracy right out of the box. You can also train custom classifiers to create specialized classes.



## Use cases

The Visual Recognition service can be used for diverse applications and industries, such as:

- **Manufacturing:** Use images from a manufacturing setting to make sure products are being positioned correctly on an assembly line
- **Visual Auditing:** Look for visual compliance or deterioration in a fleet of trucks, planes, or windmills out in the field, train custom classifiers to understand what defects look like
- **Insurance:** Rapidly process claims by using images to classify claims into different categories.
- **Social listening:** Use images from your product line or your logo to track buzz about your company on social media
- **Social commerce:** Use an image of a plated dish to find out which restaurant serves it and find reviews, use a travel photo to find vacation suggestions based on similar experiences, use a house image to find similar homes that are for sale
- **Retail:** Take a photo of a favorite outfit to find stores with those clothes in stock or on sale, use a travel image to find retail suggestions in that area
- **Education:** Create image-based applications to educate about taxonomies, use pictures to find educational material on similar subjects

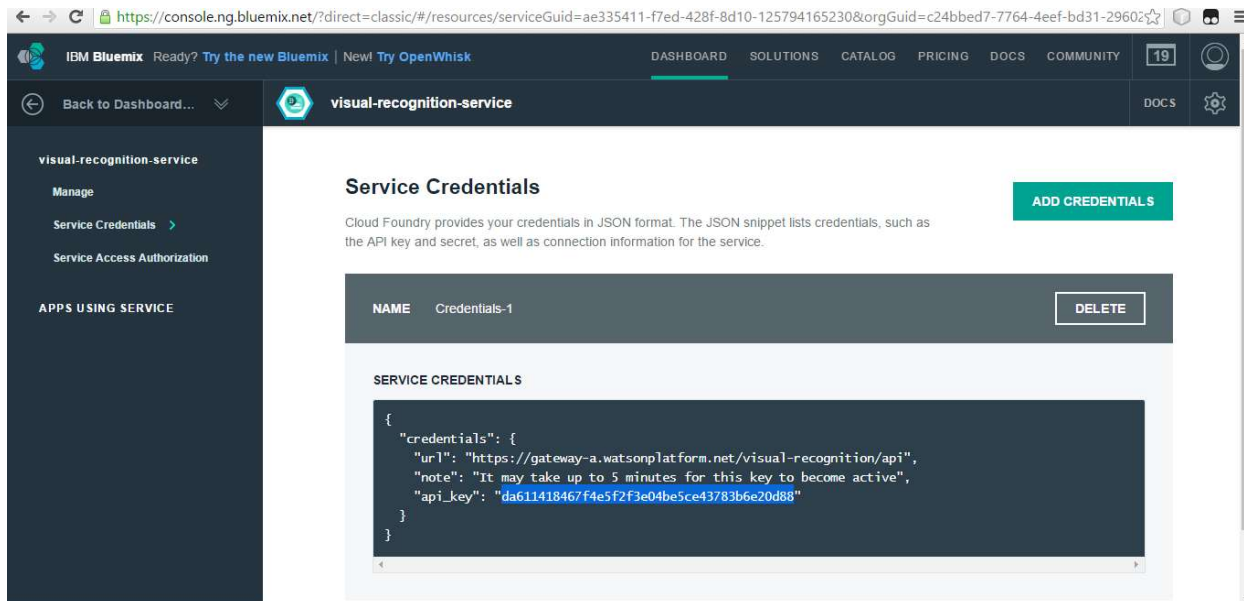
With the demo, you can analyze images for subject matter, faces, and text, as well as train a custom classifier.

Reference to analyze a quick demo (Ctrl+Click):

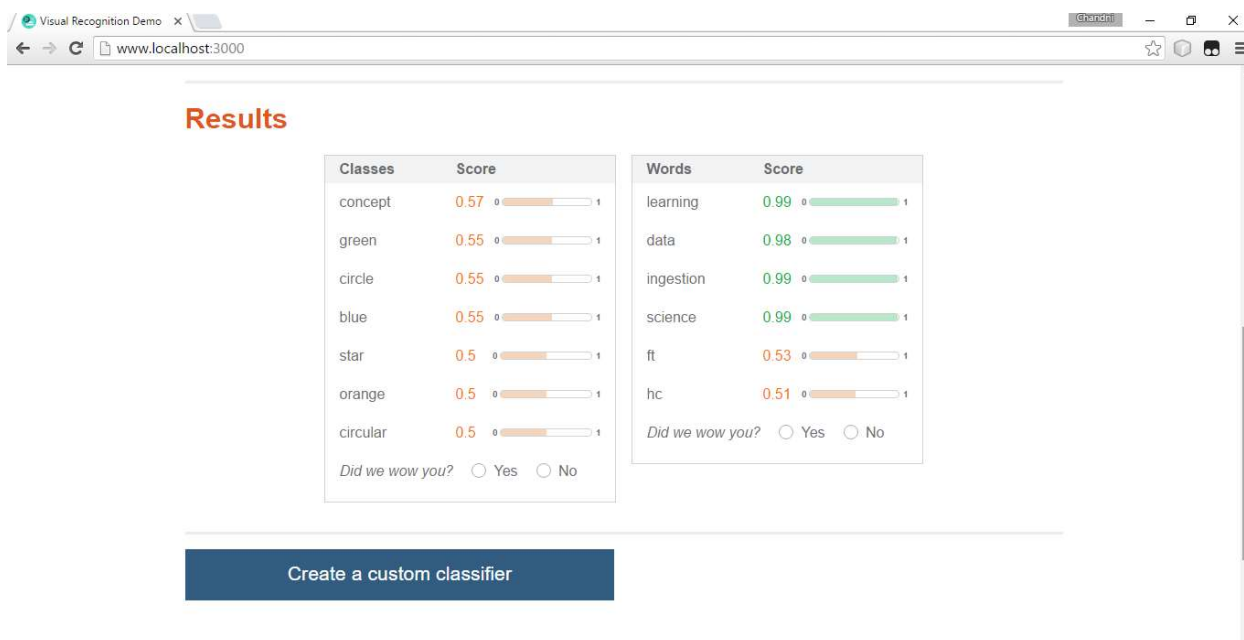
<http://visual-recognition-demo.mybluemix.net/>

Reference of the code:

<https://github.com/watson-developer-cloud/visual-recognition-nodejs>



The above is the service API key you would need.



The above is the final output of the Visual Recognition that you would see.

## Demo 2: Tone Analyzer

GitHub Repo:

<https://github.com/watson-developer-cloud/tone-analyzer-nodejs>



# Overview of the Watson™ Tone Analyzer Service

- The IBM Watson™ Tone Analyzer Service uses linguistic analysis to detect three types of tones from written text: emotions, social tendencies, and writing style. Emotions identified include things like anger, fear, joy, sadness, and disgust.
- Identified social tendencies include things from the Big Five personality traits used by some psychologists.
- These include openness, conscientiousness, extraversion, agreeableness, and emotional range. Identified writing styles include confident, analytical, and tentative.

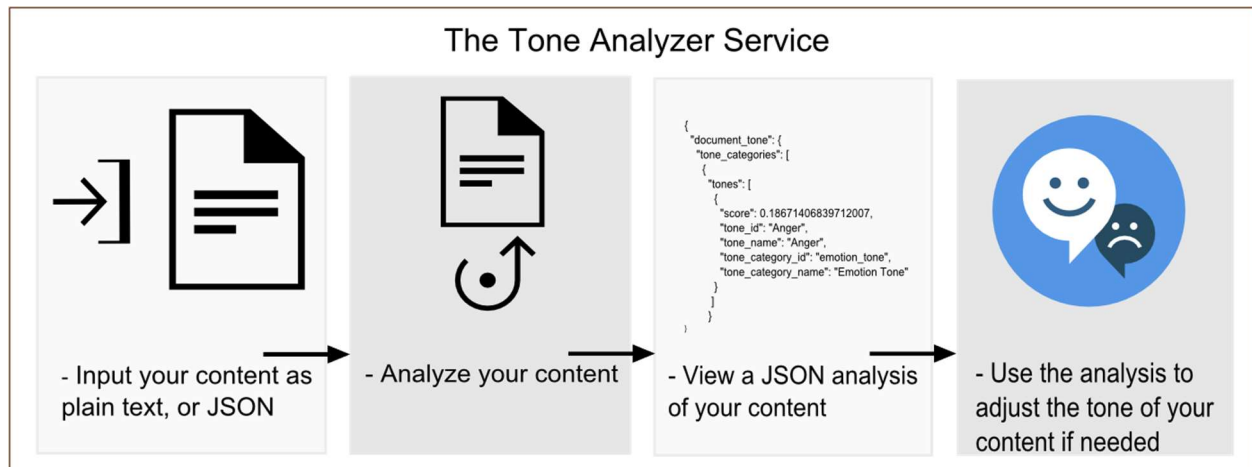
Input email and other written media into the Tone Analyzer service, and use the results to determine if your writing comes across with the emotional impact, social tone, and writing style that you want your intended audience to see.

## Tone Analyzer Service Use Cases

- **Personal and business communications**
- **Market research** - Financial advisors and investors.
- **Self-branding** - Bloggers and journalists.
- **Automated contact-center agent**

## Tone Analyzer Basic Flow

- Input JSON or plain text that contains your written content into the service, and then get back results in JSON that analyzes the tone for each sentence of your input.
- Use these results to adjust the tone of your content to change how people will perceive your online identity and to increase the effectiveness of your communication.



- **Steps We performed:**

## Deploying your app with the command line interface

You can use the command line interface to deploy and modify applications and service instances.

Before you begin, install the Cloud Foundry and IBM® Bluemix® command line interfaces.

**Restriction:** The command line tools are not supported by Cygwin. Use the tools in a command line window other than the Cygwin command line window.

After the command line interfaces are installed, you can get started:

1. Extract the package to a new directory to set up your development environment.
2. Change to your new directory.
  - `cd your_new_directory`
1. Make changes to your app code as you see fit. We suggest making sure the app runs locally before you deploy it back to IBM® Bluemix®.

One file you should take note of is the manifest.yml file. When deploying your app back to IBM® Bluemix®, this file is used to determine your application's URL, memory allocation, number of instances, and other crucial parameters.

3. Connect to IBM® Bluemix®.
  - `bluemix api https://api.ng.bluemix.net`

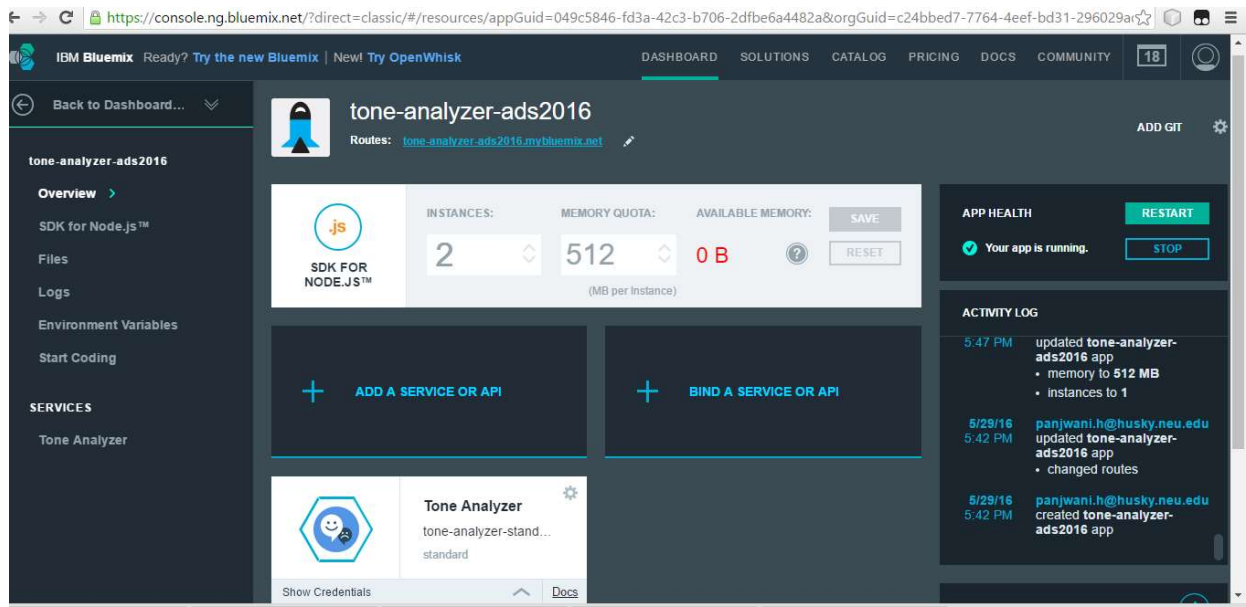
4. Log in to Bluemix.
  - `bluemix login -u panjwani.h@husky.neu.edu -o ADS -s Dev`
5. Deploy your app to Bluemix. For more information about `cf push` command, see [Uploading your application](#).
  - `cf push tone-analyzer-ads2016`
6. Access your app by entering the following URL into your browser:
  - `tone-analyzer-ads2016.mybluemix.net`

- **Service Code Details:**

The screenshot shows the IBM Bluemix console interface. The left sidebar contains navigation links for 'tone-analyzer-ads2016', including Overview, SDK for Node.js (selected), Files, Logs, Environment Variables, Start Coding, and SERVICES. The main content area displays the 'SDK for Node.js' details. At the top, there are controls for INSTANCES (set to 2), MEMORY QUOTA (512 MB), and AVAILABLE MEMORY (0 B), with SAVE and RESET buttons. Below this is the 'Instance Details' section, which includes a message 'ACTIONS: No actions available for this runtime' and a table of running instances.

Instance	State	CPU	Memory Usage	Disk Usage	Start Time
0	Running	0.0%	71.6 MB / 512 MB	87.4 MB / 1 GB	May 29, 2016, 6:49:36 ...
1	Running	0.0%	65.0 MB / 512 MB	87.4 MB / 1 GB	May 29, 2016, 6:49:43 ...

**Click on tone-analyzer-ads2016**



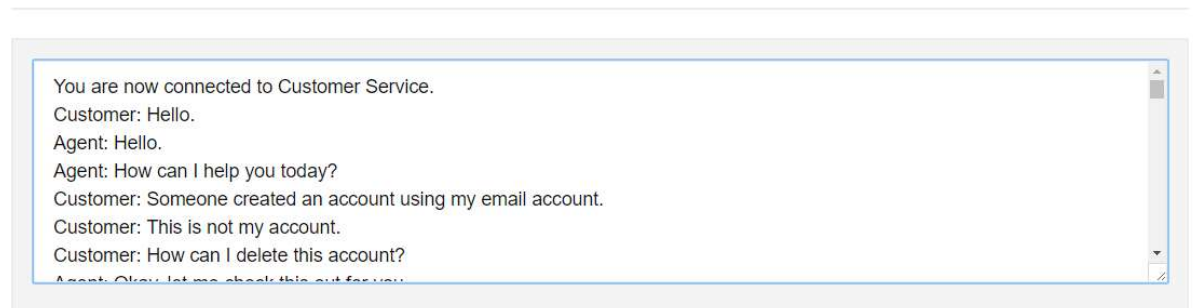
**Click on Demo and then Click on Demo again.**



## Sample use cases

Choose an example to learn how you can adjust the tone of your content to change people's perceptions, or improve its effectiveness. [Learn more.](#)

☒ Customer service chat ☐ Email message ☐ Corporate announcement ☐ Your own text

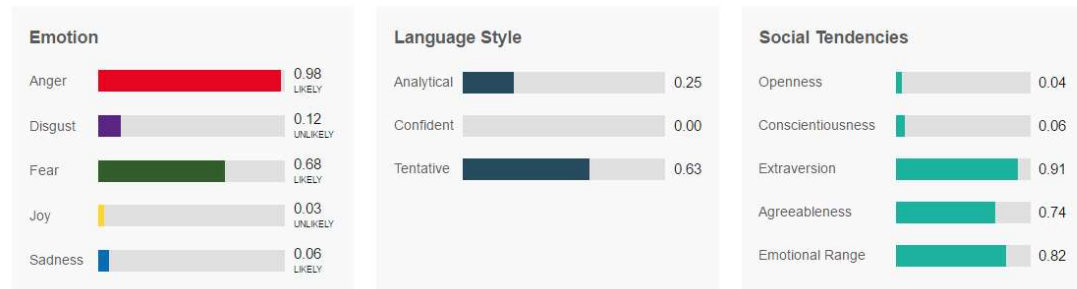


**Click on Analyze**

## Output

The Tone Analyzer Service analyzes text at the document level and the sentence level. Use the document level analysis to get a sense of the overall tone of the document, and use the sentence level analysis to identify specific areas of your content where tones are the strongest.

### Document-level



The Emotion graph represents the likelihood that an emotion tone is present in the text.

The Language Style and Social Tendencies graphs represent the amount of a language or social tone present in the text. [Learn more.](#)


The above is the output.

## Deploying the App in Our Space (DevOps - IBM Bluemix)

### Visual Recognition App Home Page:

Deploy this application to Bluemix

Deploying this app will create a private DevOps Services project for you. [Learn more.](#)

**VISUAL-RECOGNITION-NODEJS**  
GIT URL: <https://github.com/watson-developer-cloud/visual-recognition-nodejs>  
GIT BRANCH: master

APP NAME  
**visual-recognition-nodejs-hrj-1851**

REGION ORGANIZATION SPACE

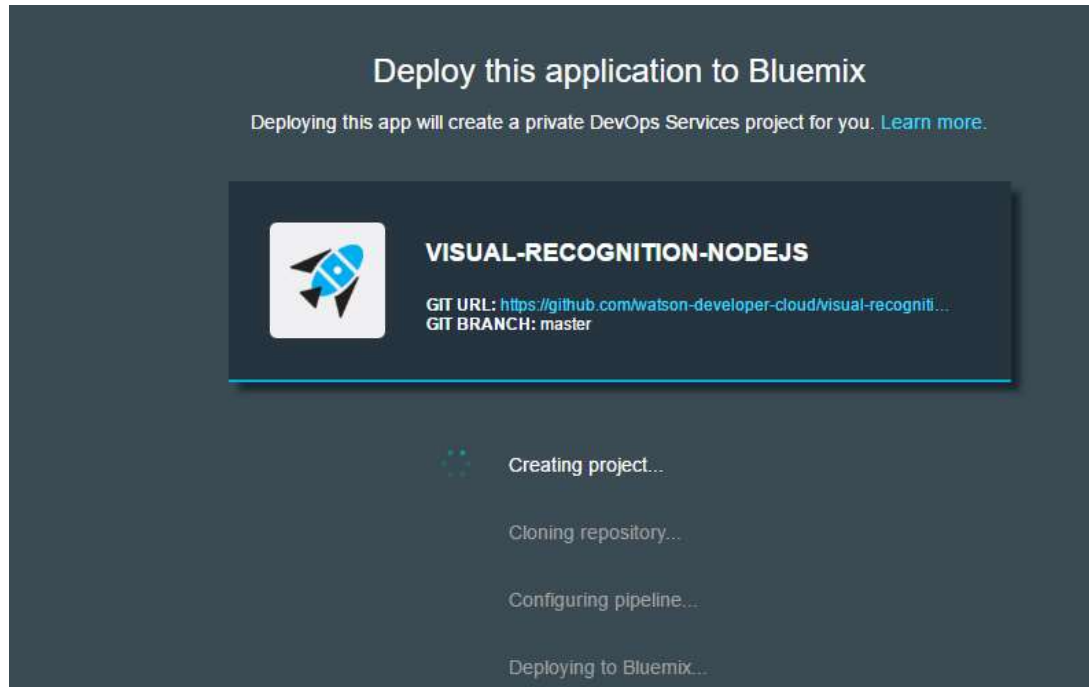
IBM Bluemix US Sot ADS Dev

DEPLOY

- App Name: visual-recognition-nodejs-hrj-1851
- Organization: ADS

- SPACE: Dev
- Alias chosen to deploy: hrj

### Interoperable steps in Deployment:



## Deploy this application to Bluemix

Deploying this app will create a private DevOps Services project for you. [Learn more.](#)



### VISUAL-RECOGNITION-NODEJS

GIT URL: <https://github.com/watson-developer-cloud/visual-recognition-nodejs>  
GIT BRANCH: master



Created project successfully



Cloned repository successfully



Configured pipeline successfully



Deploying to Bluemix...

After you deploy the App successfully in your account, you see the following message:



## VISUAL-RECOGNITION-NODEJS

GIT URL: [https://github.com/watson-developer-cloud/visual-recognition...](https://github.com/watson-developer-cloud/visual-recognition-nodejs)  
GIT BRANCH: master

- ✓ Created project successfully
- ✓ Cloned repository successfully
- ✓ Configured pipeline successfully
- ✓ Deployed to Bluemix successfully

### Success!


You've added an instance of this app to your organization in Bluemix.

[VIEW YOUR APP](#)

[EDIT CODE](#)

When we click on 'View Your App'





## Visual Recognition








Visual Recognition uses deep learning algorithms to analyze images for scenes, objects, faces, text, and other subjects that can give you insights into your visual content. You can organize image libraries, understand an individual image, and create custom classifiers for specific results that are tailored to your needs.


**Resources:**  
[API Reference](#)  
[Documentation](#)  
[Fork on Github](#)  
[Deploy on Bluemix](#)

[Try](#) [Train](#) [Test](#)

### Try the service

Choose a sample image or upload your own image (max 2mb) to try out Visual Recognition.



  
[Select](#) or drag your own image

We can try it out.

### **End of the IBM Watson part:**

- You can submit comments or ask questions about **Visual Recognition** in the [Watson forum](#).
- You can also read posts about Watson services that are written by IBM researchers, developers, and other experts on the [Watson blog](#).