Learn Session Training

IBM Watson Expert Services

Lab Exercise

Topic: Expert Services Learn – Watson Tone Analyzer Service

Version: **0.2**

Updated: Novemeber 7th, 2018

For questions about the Watson workshop and materials, contact:
Victor Povar *E-mail: victorp@ca.ibm.com*

Lab Overview

This lab guides you through the configuration steps required to use Watson Tone Analyzer service. Watson Tone Analyzer will be used to monitor customer support conversations. Escalate customer conversations when they turn sour, or find opportunities to improve customer service scripts, dialogs and customer journeys. Tones detected with this endpoint include frustrated, sad, satisfied, excited, polite, impolite and sympathetic.

Change Log

Revision History: Changes to this document are summarized in the following table in reverse chronological order (latest version first).

Revision	Date	Author	Short Description of Changes
0.2	11/07/18	Suj Perepa	Reviewed and provided feedback
0.1	11/01/18	Victor Povar	Wrote initial draft

Table of Contents

Course overview

Credits

This project is as a collaborative effort of many experts within and outside of Watson in the hopes of broadening skills within IBM, our clients, partners and the developer community.

If you have suggestions for how to improve this guide, please send a note to victorp@ca.ibm.com.

Objectives

During the first lab, you will use the Tone Analyzer, Cloud Object Storage and Watson Studio services to analyze and monitor the tone with an email. As part of the second lab, you will integrate Watson Assistant, Tone Analyzer and Cloud Functions to tailor the answers returned by the Watson Assistant service.

The tasks and tools that are covered within the Lab allow you to perform and complete all necessary steps to use Tone Analyzer to derive insights from production conversations as well as extend the Watson Assistant service to provide a tailored user experience.

Lab Structure

The labs include explanations for reference to support the general understanding of the subject matter and to get a better sense of what you are actually doing.

To complete the exercises, you only need to perform numbered steps.

This lab exercise is designed for every student to work independently using a Mac or PC.

Each student requires access to an IBM Cloud account to provision the required services as well as complete the lab instructions.

Preparation

To prepare for the lab exercises, locate the Lab folder provided to you by your instructor. Make sure to unzip the Lab Folder to a directory on your local hard drive.

The lab folder contains the following files:

File name	Description
ToneAnalyzerLearn/	Folder contains the Instruction Guide
ToneAnalyzerLearn/data	Folder containing sample data files

1 Getting Started

This Lab can be performed using a Windows, Mac OS X or Linux system. The current version of this guide has screen shots taken from a Mac OS X computer, your results may vary according to the operating system you use.

1.1 Overview

This lab guides you through the steps required to use Watson Tone Analyzer Service Application Programming Interfaces (APIs). The APIs will be accessed via command line or through a development environment of your choice for all configurations and queries. The training examples used in this tutorial are not tailored to a specific domain.

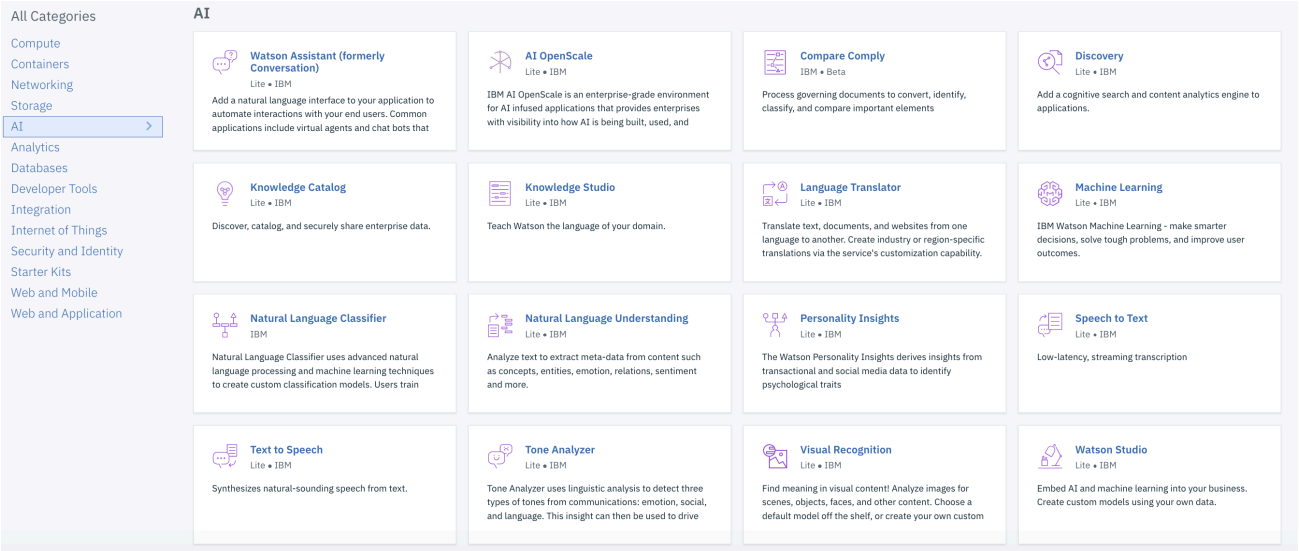
1.2 Working with Watson Developer Cloud

1.2.1 IBM Cloud

IBM Cloud is an implementation of IBM's Open Cloud Architecture that leverages Cloud Foundry to enable developers to rapidly build, deploy, and manage their cloud applications. The purpose of this guide is not to introduce you to IBM Cloud, which you should already be familiar with, at least on a high level.

1.2.2 Watson Developer Cloud (WDC)

IBM Watson Developer Cloud (WDC) is a platform of cognitive services, designed for developers to build solutions that extract insight from their custom data. Cognitive computing systems learn and interact naturally with humans to augment their ability to make better decisions from data. As such, the Watson Developer Cloud services offer a variety of services that cover various aspects of natural interaction including text (Natural Language Understanding, Natural Language Classifier, Assistant and Discovery), images (Visual Recognition) and speech (Speech to Text and Text to Speech). Furthermore, WDC offers services to understand a user's personality (Personality Insights), emotional/social tone (Tone Analyzer) and and translation (Watson Translator) in a scalable manner.



1.2.3 Watson Tone Analyzer

Watson Tone Analyzer uses linguistic analysis to detect three types of tones from communications: emotion, social, and language. This insight can then be used to drive high impact communications. Tone Analyzer service comes with an easy-to-use API to translate content via cURL, Node.js, Java or Python. Use of Tone Analyzer entails the following steps:

- Creating the service instance;
- Choosing the endpoint;
- Integrating with complimentary service;

Tone Analyzer service can also be used in combination with other Watson services to create end-to-end solutions. For instance, combined with Watson Assistant and Watson Speech Services one can build high impact chat bot applications that listen to the end user and provide answers that are tailored to the user's current state of mind.

1.3 Prerequisites

This section provides instructions to help you get started quickly with the IBM Watson Developer Cloud services using Python as your programming runtime environment. You will be guided through examples to let you experiment with the available services.

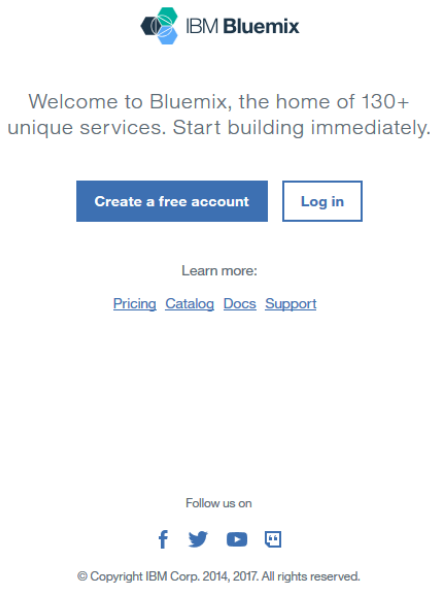
1.3.1 Obtaining an IBM Cloud account

IBM Cloud is a cloud Platform as a Service (PaaS), which allows you to host your application on-line and bind it to a variety of Software as a Service (SasS) offerings from IBM including Watson analytic services. If you are new to IBM Cloud, you can create a trial account at the IBM Cloud site.

1. Direct your browser to the IBM Cloud home page: https://console.ng.bluemix.net/home/ to access your dashboard
2. If you do not yet have an IBM Cloud account, click **Sign Up** on the top right
3. Enter requested information and click **Create Account**
4. If you use your personal email address, you have 30 days to evaluate IBM Cloud. Some services, such as Watson Discovery Service, are free for limited use during the trial period.

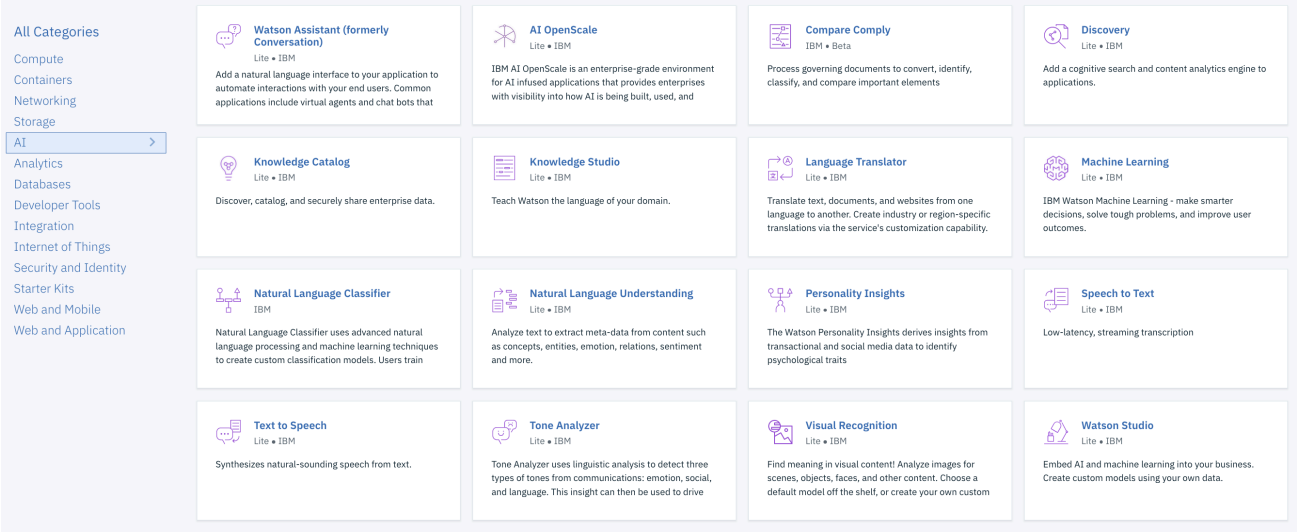
1.4 Creating Your Tone Analyzer Instance

1. Navigate to IBM Cloud - https://console.ng.bluemix.net/ and click on Log In

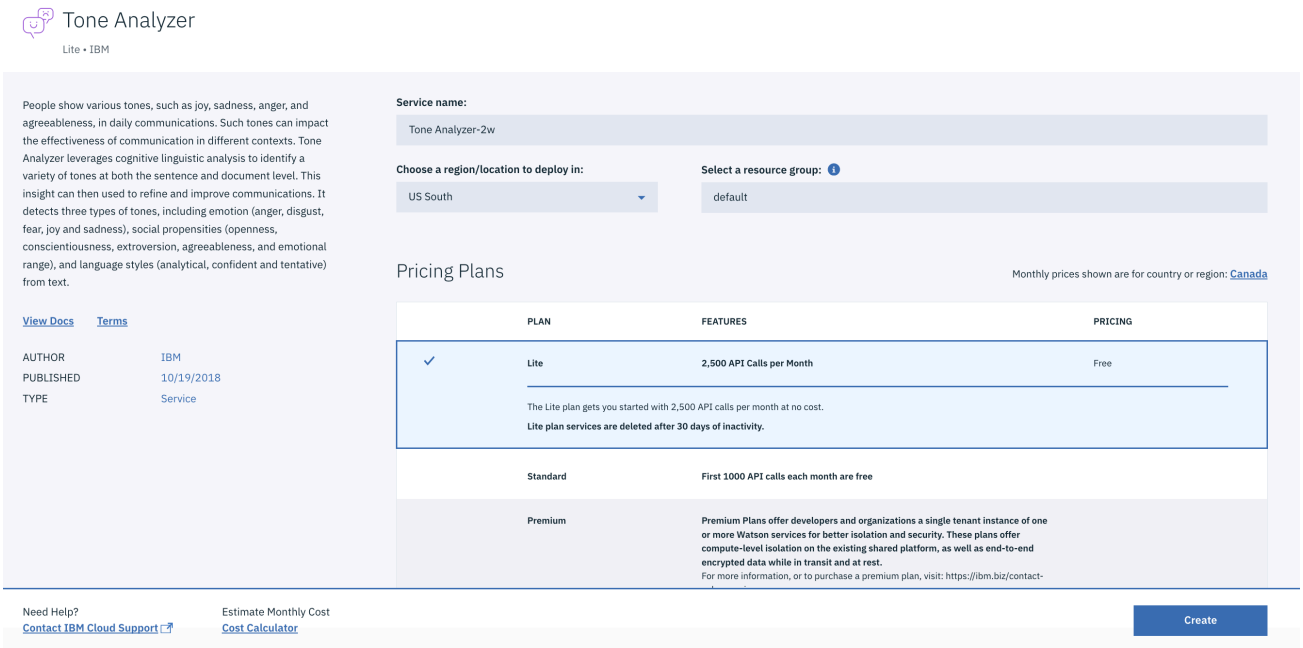


2. Use your organization credentials to Log in to IBM Cloud. These were created and given to you when you signed up to IBM Cloud.
3. Once logged in, navigate to the IBM Cloud Catalog by clicking on Catalog in the top left-hand corner. Next select the AI category from the category navigation. Alternatively, you can follow this link - https://console.bluemix.net/catalog/?category=ai

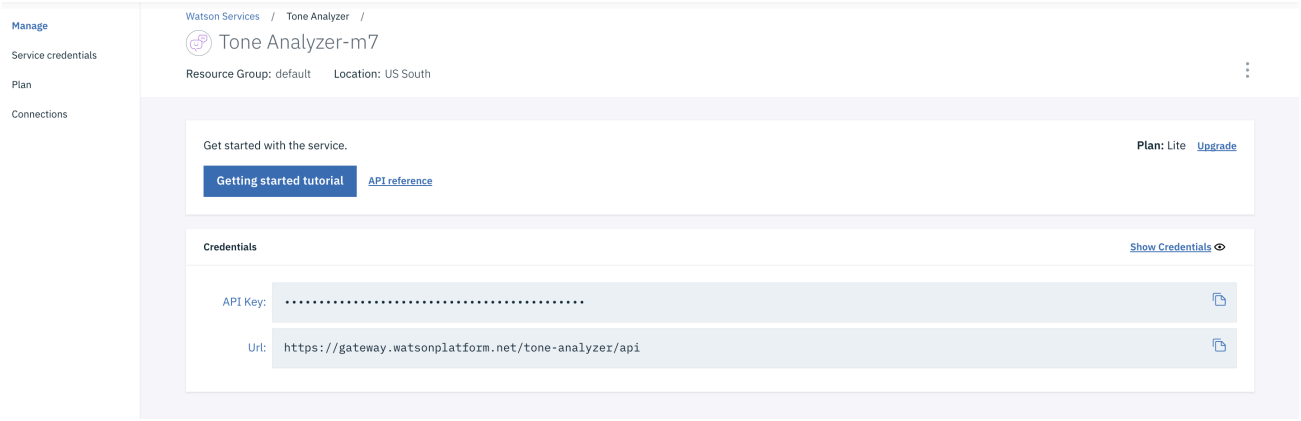
4. Click on the Tone Analyzer tile in the IBM Cloud catalog:



5. Within this window, do not make any changes to the Credential name or Service name. Create a trial service instance by selecting the Lite Plan option (default setting) and by clicking Create.



6. IBM Cloud will now instantiate your personal Tone Analyzer instance. It will automatically guide you to the service dashboard.

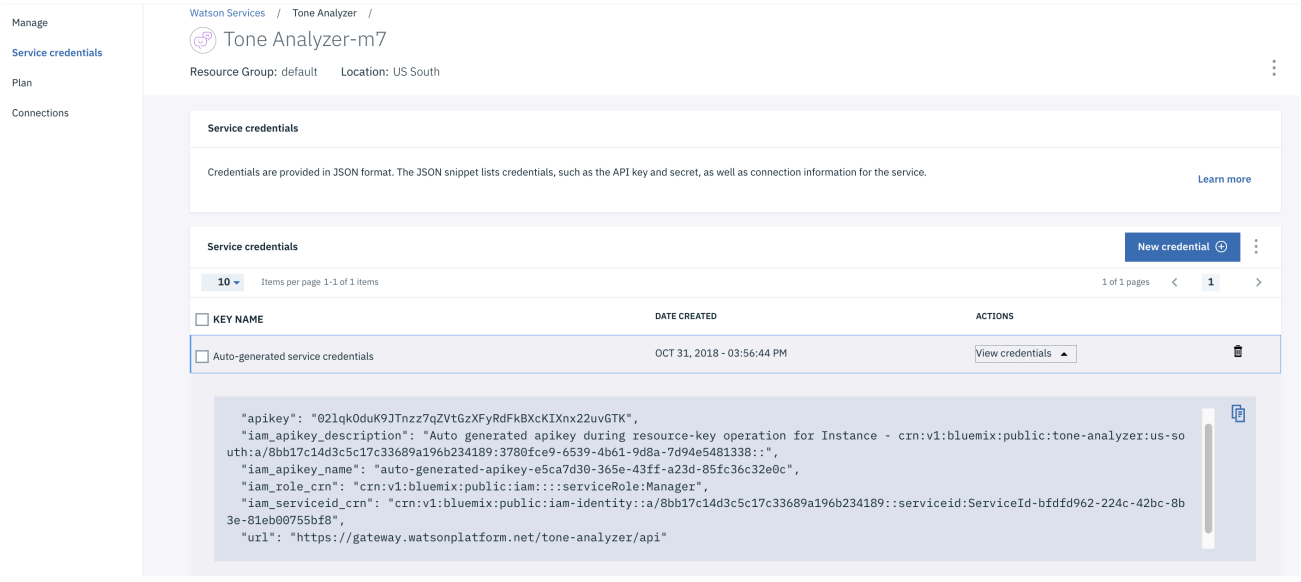


If you close the tab and wish to re-access it, you can navigate to the service dashboard directly by selecting the created Tone Analyzer service from the list of services created in your account. The list of the services you have created can be found at https://console.ng.bluemix.net/dashboard/apps

7. In the service dashboard, click on the "Service Credentials" tab and go to service credentials section. Click on the "View Credentials" dropdown to view API Key and URL.

8. If Credentials do not exist, click the create "New Credentials" button.

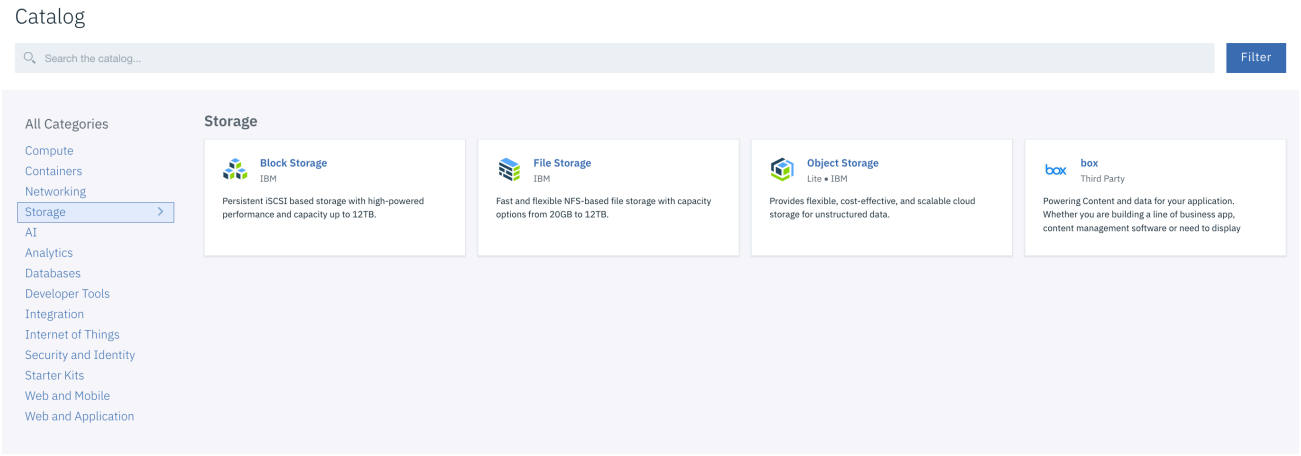
Note: Copy the API KEY information and keep it handy as you will need it throughout this lab.



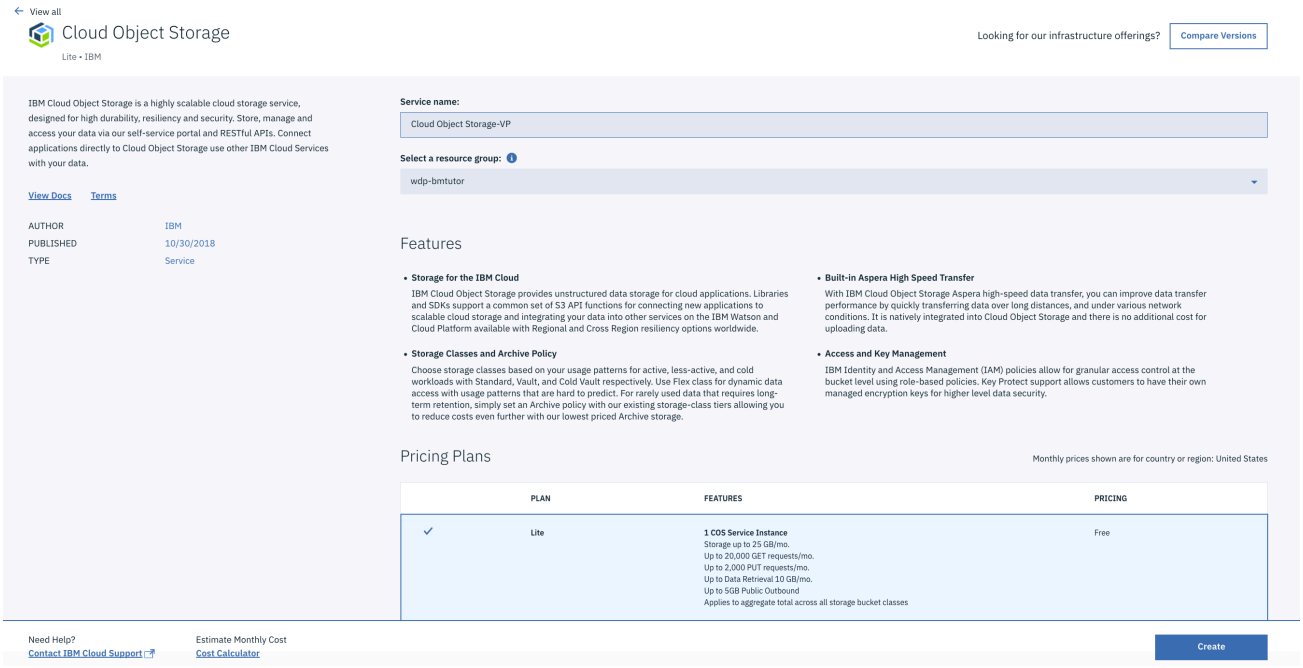
1.5 Creating Your Cloud Object Storage instance

1. Navigate to the IBM Cloud Catalog by clicking on Catalog in the top left-hand corner. Next select the Storage category from the category navigation. Alternatively, you can follow this link - https://console.bluemix.net/catalog/?category=storage

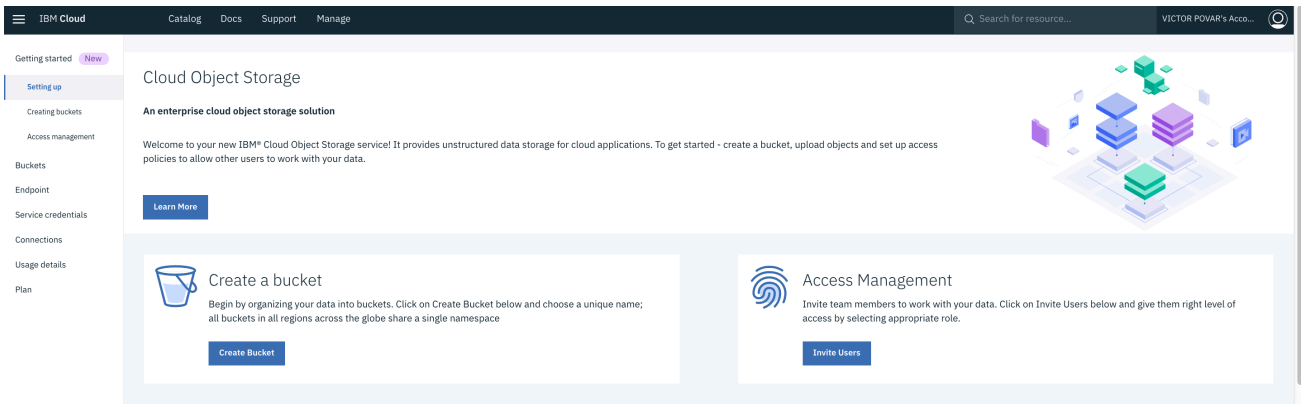
2. Click on the Object Storage tile in the IBM Cloud catalog:



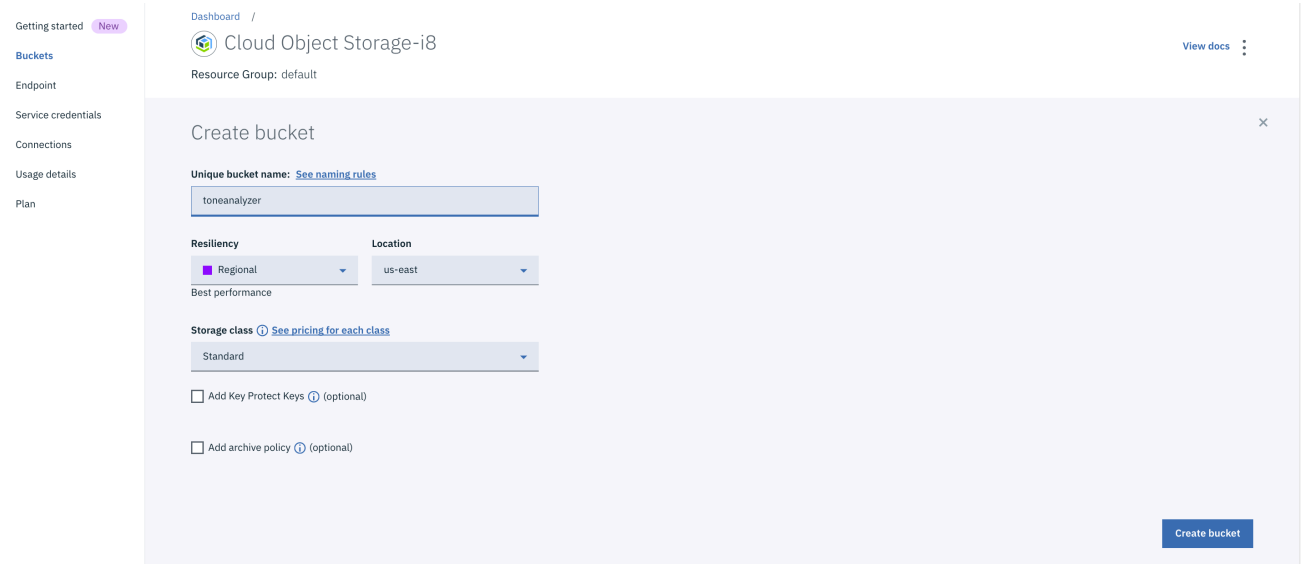
3. Within this window, do not make any changes to the Credential name or Service name. Create a trial service instance by selecting the Lite Plan option (default setting) and by clicking Create.



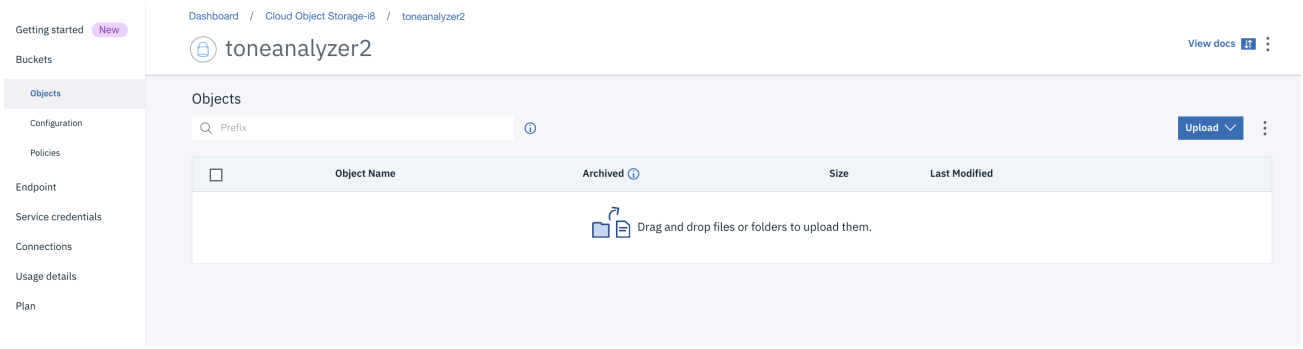
4. Click “Create Bucket” from the Setting up page



5. Give bucket the name “toneanalyzer”, choose the Resiliency level as well as the Location and click “Create bucket”.



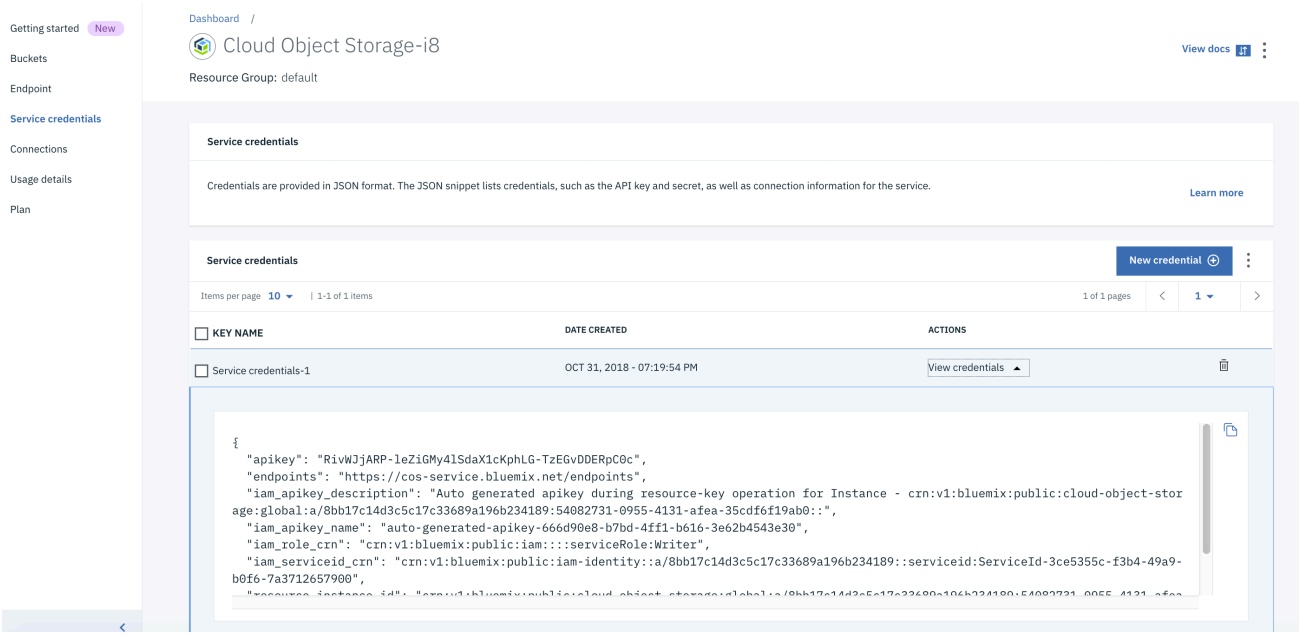
6. Upload the sample files via the “Upload” button or by Dragging and Dropping the files.



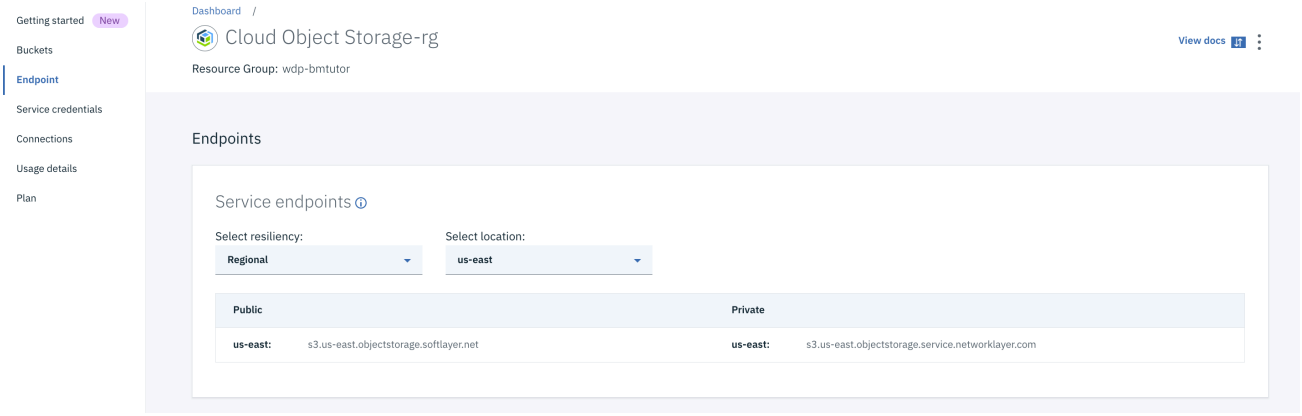
7. Click on the “Service Credentials” left-navigation option and note the following fields:

    a. apikey

    b. resource_instance_id



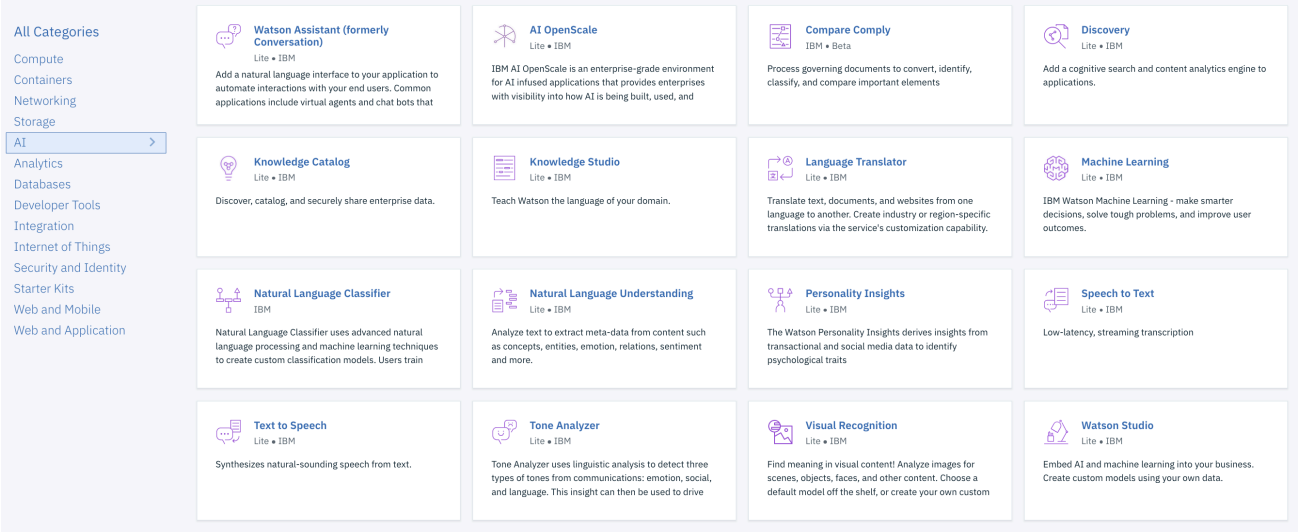
8. Click on the “Endpoint” left-navigation option and note the endpoint for your “Resiliency” and “Location”



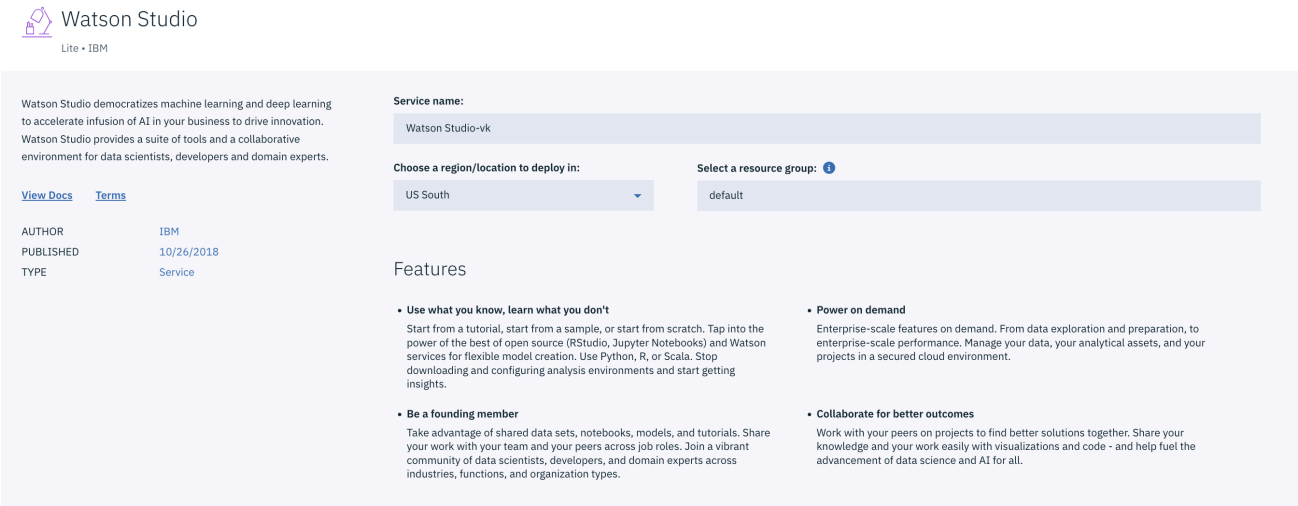
1.6 Creating Your Watson Studio instance

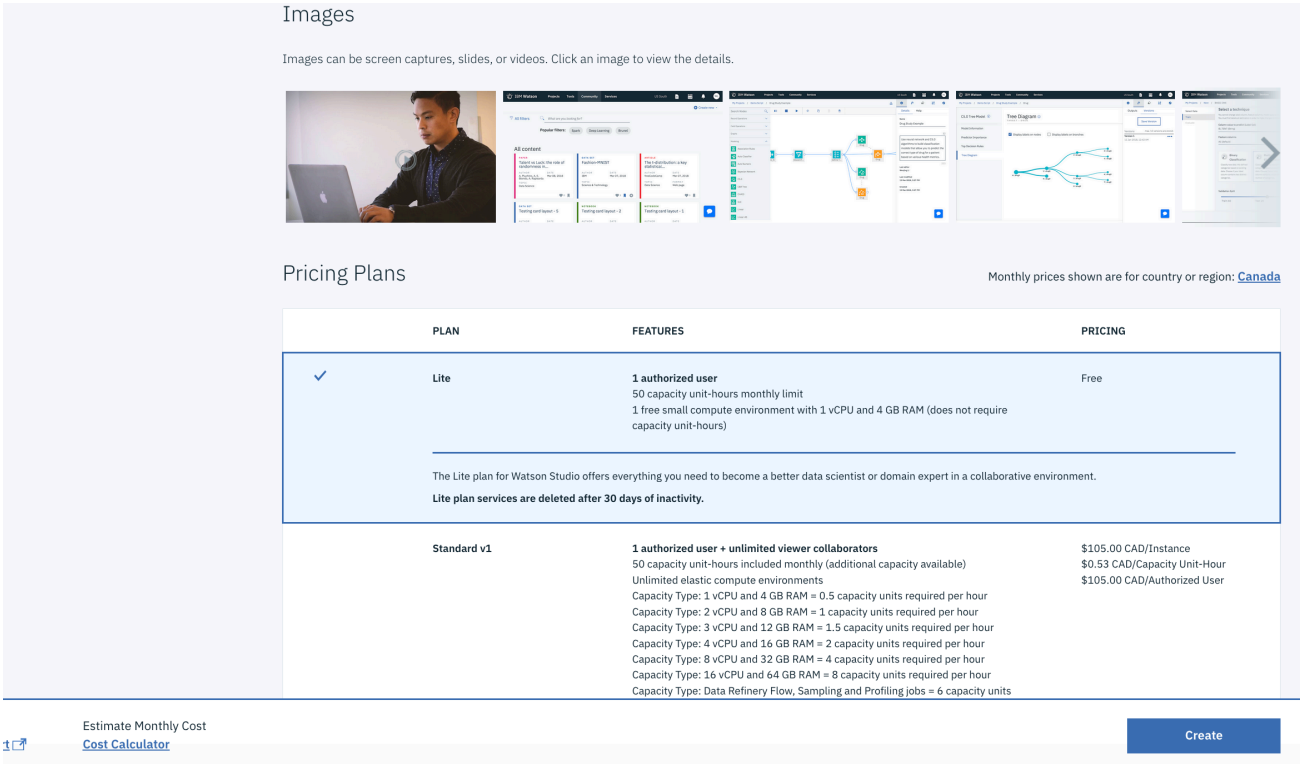
1. Navigate to the IBM Cloud Catalog by clicking on Catalog in the top left-hand corner. Next select the AI category from the category navigation. Alternatively, you can follow this link - https://console.bluemix.net/catalog/?category=ai

2. Click on the Waston Studio tile in the IBM Cloud catalog:

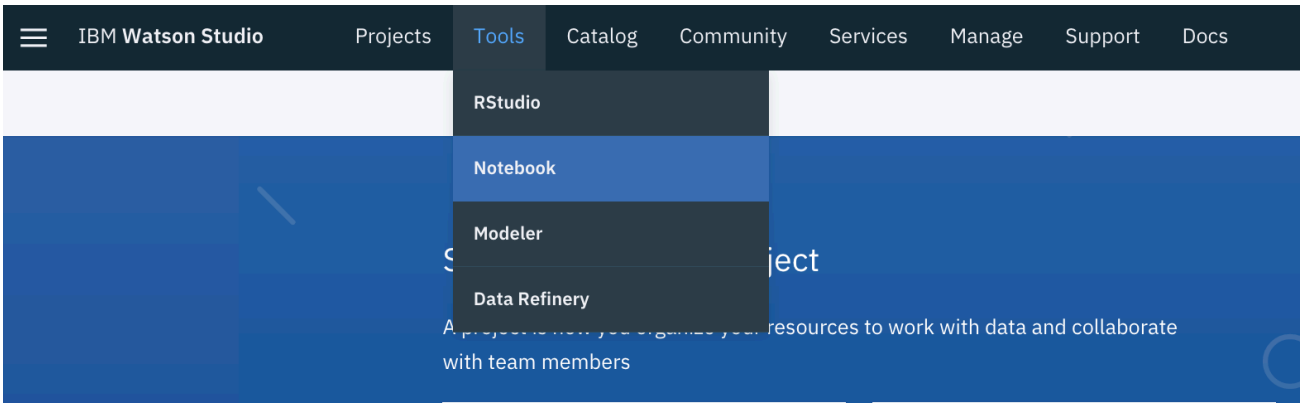


Within this window, do not make any changes to the Credential name or Service name. Create a trial service instance by selecting the Lite Plan option (default setting) and by clicking Create.

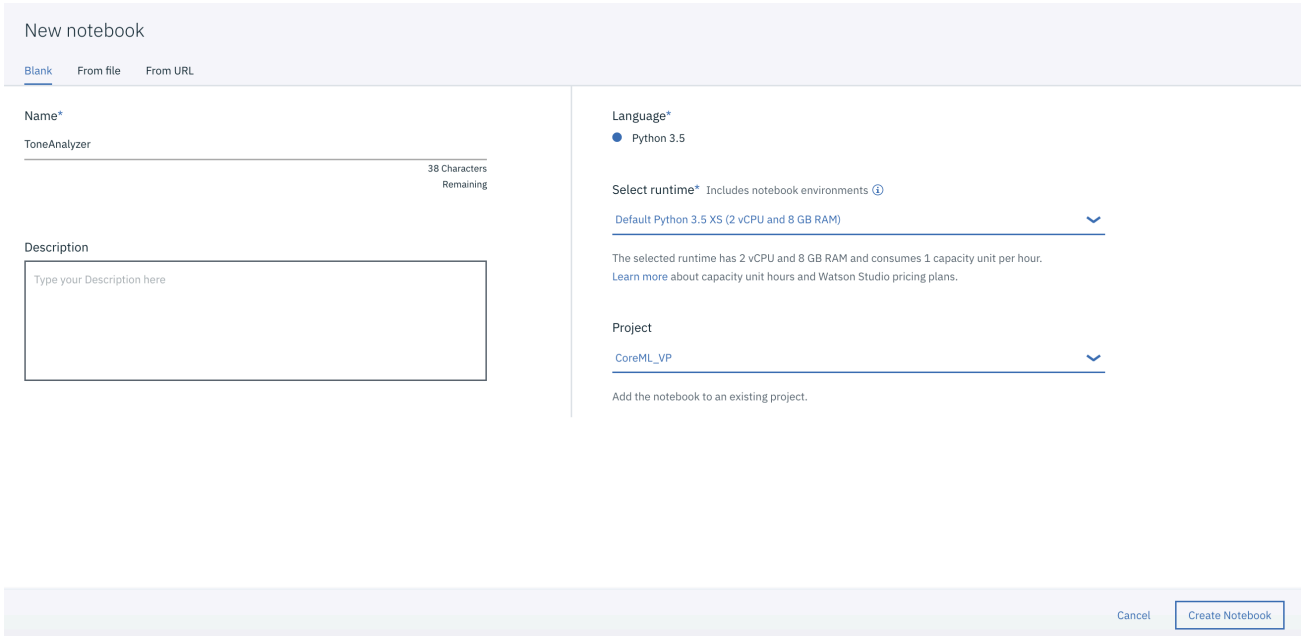




3. Select “Tools” from Top Navigation. Select “Notebook” to create a Notebook.



4. Specify the name of the notebook and click Create Notebook.



2 Tone Analyzer via API

The IBM Watson™ Tone Analyzer service uses linguistic analysis to detect emotional and language tones in written text. The service can analyze tone at both the document and sentence levels. You can use the service to understand how your written communications are perceived and then to improve the tone of your communications. Businesses can use the service to learn the tone of their customers' communications and to respond to each customer appropriately, or to understand and improve their customer conversations.

In this section of the lab we'll consume data stored in COS (Cloud Object Storage) for analysis with the Tone Analyzer service within Watson Studio. COS allows you to securely store production data and access it within Watson Studio for analysis. All objects stored in IBM® Cloud Object Storage are encrypted by default. If you want to encrypt the data using your own keys, you can manage them using IBM® Key Protect and use them to encrypt buckets.

2.1.1 Connect to the Tone Analyzer Service

REFERENCE: https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/python.html?python#authentication

Add the following Python code to your Watson Studio Notebook to establish a connection with the Tone Analyzer Service.

Python:

```
!pip install --upgrade "watson-developer-cloud>=2.2.6"
import json
from watson_developer_cloud import ToneAnalyzerV3

#Instantiate an instance of Tone Analyzer with API keys
tone_analyzer = ToneAnalyzerV3(
    version='2017-09-21',
    iam_apikey='{insert apikey for Tone Analyzer}'
)

#Instantiate an instance of Tone Analyzer with username
password
#tone_analyzer = ToneAnalyzerV3(
#    version='2017-09-21',
#    username='{insert the username}',
#    password='{insert the password}',
#    url = "https://gateway.watsonplatform.net/tone-
analyzer/api"
#)
```

2.1.2 Analyze Text with the General Model

REFERENCE: https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/python.html?python#tone

Use the general purpose endpoint to analyze the tone of your input content. The service analyzes the content for emotional and language tones. The method always analyzes the tone of the full document; by default, it also analyzes the tone of each individual sentence of the content.

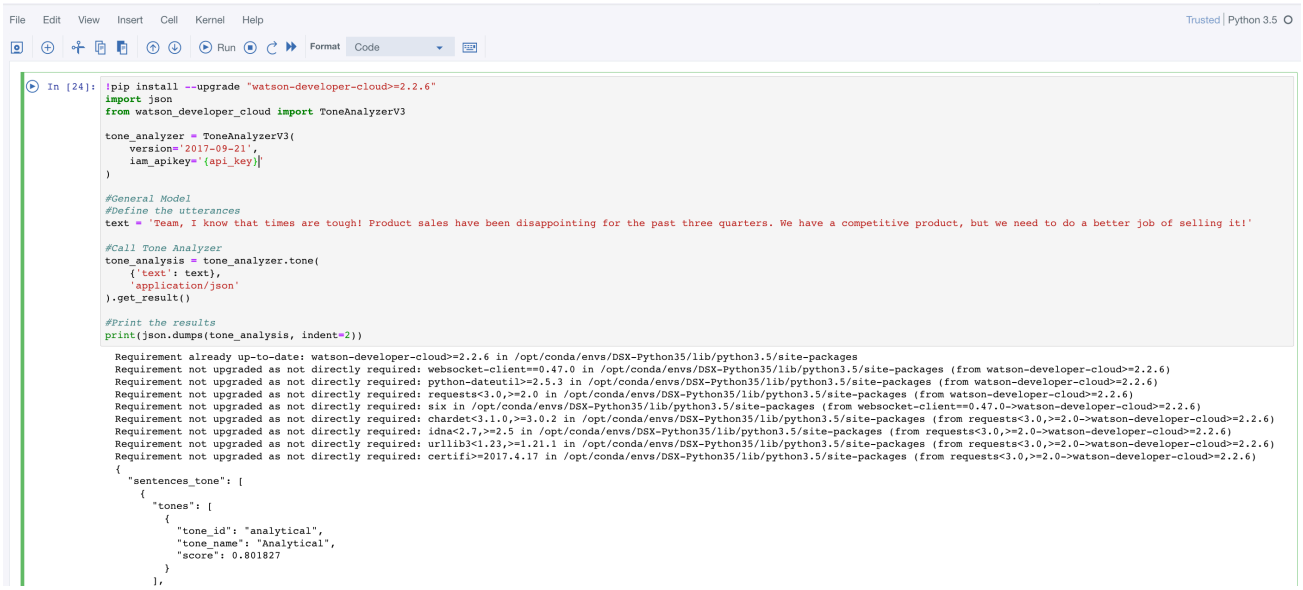
You can submit no more than 128 KB of total input content and no more than 1000 individual sentences in JSON, plain text, or HTML format. The service analyzes the first 1000 sentences for document-level analysis and only the first 100 sentences for sentence-level analysis.

Let's analyze the tone of the following email:

'Team, I know that times are tough! Product sales have been disappointing for the past three quarters. We have a competitive product, but we need to do a better job of selling it!

Add the following code to the Notebook and select Run from Command Panel.

```
#General Model
#Define the text to be analyzed
text = 'Team, I know that times are tough! Product sales have
been disappointing for the past three quarters. We have a
competitive product, but we need to do a better job of selling
it!'

#Call Tone Analyzer
tone_analysis = tone_analyzer.tone(
    {'text': text},
    'application/json'
).get_result()

#Print the results
print(json.dumps(tone_analysis, indent=2))
```



2.1.3 Interpreting results

The service returns the tone for each of the 3 sentences as well as the overall tone of the document.

Note: The score that is returned represents the likelihood that the tone is perceived and not the degree to which the tone is perceived, the tone range is 0.5 to 1. A score greater than 0.75 indicates a high likelihood that the tone is perceived in the content.

The service returnes **emotional** and **language** tones. The list of possible returned tones is:

- Emotional tones
  - anger
  - fear
  - joy
  - sadness
- Language Tones
  - analytical
  - confident
  - tentative

```
{
  "sentences_tone": [
    {
      "tones": [
```

```
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.801827
        }
      ],
      "text": "Team, I know that times are tough!",
      "sentence_id": 0
    },
    {
      "tones": [
        {
          "tone_id": "sadness",
          "tone_name": "Sadness",
          "score": 0.771241
        },
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.687768
        }
      ],
      "text": "Product sales have been disappointing for the
past three quarters.",
      "sentence_id": 1
    },
    {
      "tones": [
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.506763
        }
      ],
      "text": "We have a competitive product, but we need to
do a better job of selling it!",
      "sentence_id": 2
    }
  ],
  "document_tone": {
    "tones": [
      {
        "tone_id": "sadness",
        "tone_name": "Sadness",
        "score": 0.6165
      },
      {
        "tone_id": "analytical",
        "tone_name": "Analytical",
        "score": 0.829888
      }
```

```
    ]
  }
}
```

2.1.4 Analyze Text with the Customer Engagement Model

Use the customer engagement endpoint to analyze the tone of customer service and customer support conversations. For each utterance of a conversation, the method reports the most prevalent subset of the following seven tones: sad, frustrated, satisfied, excited, polite, impolite, and sympathetic.

If you submit more than 50 utterances, the service returns a warning for the overall content and analyzes only the first 50 utterances. If you submit a single utterance that contains more than 500 characters, the service returns an error for that utterance and does not analyze the utterance. The request fails if all utterances have more than 500 characters. Per the JSON specification, the default character encoding for JSON content is effectively always UTF-8.

Add the following code to the Notebook and select Run from Command Panel.

```
#Customer Service Example
#Define the utterances
utterances = [
    {
        "text": "Hello, I'm having a problem with your
product.",
        "user": "customer"
    },
    {
        "text": "OK, let me know what's going on, please.",
        "user": "agent"
    },
    {
        "text": "Well, nothing is working :(",
        "user": "customer"
    },
    {
        "text": "Sorry to hear that.",
        "user": "agent"
    }
]

#Call Tone Analyzer
utterance_analyses =
tone_analyzer.tone_chat(utterances).get_result()

#Print the results
print(json.dumps(utterance_analyses, indent=2))
```

2.1.5 Interpreting results

The service returns the tone for each utterance of the input. The list of possible returned tones is:

- excited
- frustrated
- impolite
- polite
- sad
- satisfied
- sympathetic

Note: The score that is returned represents the likelihood that the tone is perceived and not the degree to which the tone is perceived, the tone range is 0.5 to 1. A score greater than 0.75 indicates a high likelihood that the tone is perceived in the content.


```
{
  "utterances_tone": [
    {
      "utterance_id": 0,
      "tones": [
        {
          "tone_id": "polite",
          "tone_name": "Polite",
          "score": 0.686361
        }
      ],
      "utterance_text": "Hello, I'm having a problem with your
product."
    },
    {
      "utterance_id": 1,
      "tones": [
        {
          "tone_id": "polite",
          "tone_name": "Polite",
          "score": 0.92724
        }
      ],
      "utterance_text": "OK, let me know what's going on,
please."
    },
    {
```

```
      "utterance_id": 2,
      "tones": [
        {
          "tone_id": "sad",
          "tone_name": "Sad",
          "score": 0.997795
        }
      ],
      "utterance_text": "Well, nothing is working :("
    },
    {
      "utterance_id": 3,
      "tones": [
        {
          "tone_id": "polite",
          "tone_name": "Polite",
          "score": 0.730982
        },
        {
          "tone_id": "sympathetic",
          "tone_name": "Sympathetic",
          "score": 0.672499
        }
      ],
      "utterance_text": "Sorry to hear that."
    }
  ]
}
```


2.1.6 Connect to Cloud Object Storage

In this section we'll connect to the Cloud Object Storage and retrieve the documents that we uploaded to our bucket. Once we retrieve the files, we'll read the contents of the file and analyze it with Tone Analyzer. Add the following code in a new cell and specify the apikey, resource_instance_id and service_endpoint noted down in the previous section. Select Run from Command Panel.

```
# provide COS Authentication and verify access
!pip install -U ibm-cos-sdk
import ibm_boto3
from ibm_botocore.client import Config

cos_credentials = {
  "apikey": "{insert the apikey}",
  "resource_instance_id": "{insert the resource instance id}"
}
service_endpoint = 'https://s3-api.us-
geo.objectstorage.softlayer.net'
auth_endpoint = 'https://iam.bluemix.net/oidc/token'
```

```
cos = ibm_boto3.resource('s3',

ibm_api_key_id=cos_credentials['apikey'],

ibm_service_instance_id=cos_credentials['resource_instance_id'
],
                         ibm_auth_endpoint=auth_endpoint,

config=Config(signature_version='oauth'),
                         endpoint_url=service_endpoint)

# Print all buckets in COS
for bucket in cos.buckets.all():
    print("bucket: ", bucket)
```

2.1.7 Interpreting results

If a successful connection is established, the system will return the list buckets that exist with Cloud Object Storage.

```
bucket:  s3.Bucket(name='toneanalyzer')
```

2.1.8 Connect to COS, read the files and analyze with Tone Analyzer

In this section, we'll retrieve the files uploaded to COS and analyze them with Tone Analyzer

```
# Analyze the files within COS
bucket_name = "toneanalyzer"
filetype = ".txt"

bucket_obj = cos.Bucket(bucket_name)
for file_object in bucket_obj.objects.all():
    if (file_object.key.endswith(filetype)):
        file = cos.Object(bucket_name, file_object.key).get()
        filecontent = str(file["Body"].read())

        #Call Tone Analyzer
        tone_analysis = tone_analyzer.tone(
            {'text': filecontent},
            'application/json'
        ).get_result()

        #Print the results
        print(json.dumps(tone_analysis, indent=2))
```

2.1.9 Interpreting results

Analyze the results returned by the Tone Analyzer service.

```
{
  "sentences_tone": [
    {
      "tones": [
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.801827
        }
      ],
      "text": "b'Team, I know that times are tough!",
      "sentence_id": 0
    },
    {
      "tones": [
        {
          "tone_id": "sadness",
          "tone_name": "Sadness",
          "score": 0.771241
        },
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.687768
        }
      ],
      "text": "Product sales have been disappointing for the
past three quarters.",
      "sentence_id": 1
    },
    {
      "tones": [
        {
          "tone_id": "analytical",
          "tone_name": "Analytical",
          "score": 0.506763
        }
      ],
      "text": "We have a competitive product, but we need to
do a better job of selling it!'",
      "sentence_id": 2
    }
  ],
  "document_tone": {
    "tones": [
      {
        "tone_id": "sadness",
        "tone_name": "Sadness",
        "score": 0.6165
      },
      {
```

```
        "tone_id": "analytical",
        "tone_name": "Analytical",
        "score": 0.829888
      }
    ]
  }
}
```

2.1.10 Summary

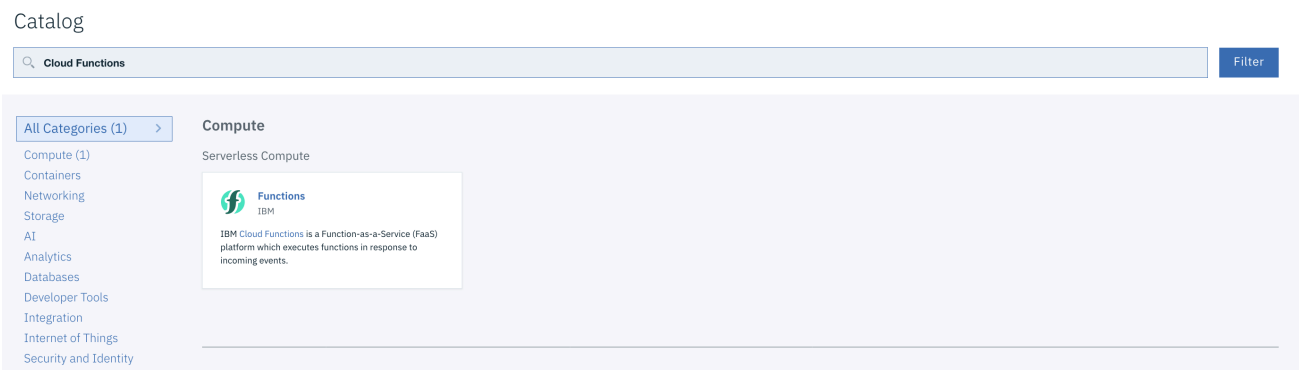
Cloud object provides a secure space to upload documents that need to be consumed and analyzed by other Watson services, such as Tone Analyzer. A powerful use case is using Tone Analyzer to monitor help center conversations to analyze the effectiveness of a customer service agent. Tone Analyzer can also be deployed to monitor Twitter and other platform on user's sentiments related to different products.

3 Combine Tone Analyzer within Watson Assistant via a Cloud Functions

3.1.1 Create a Cloud Function

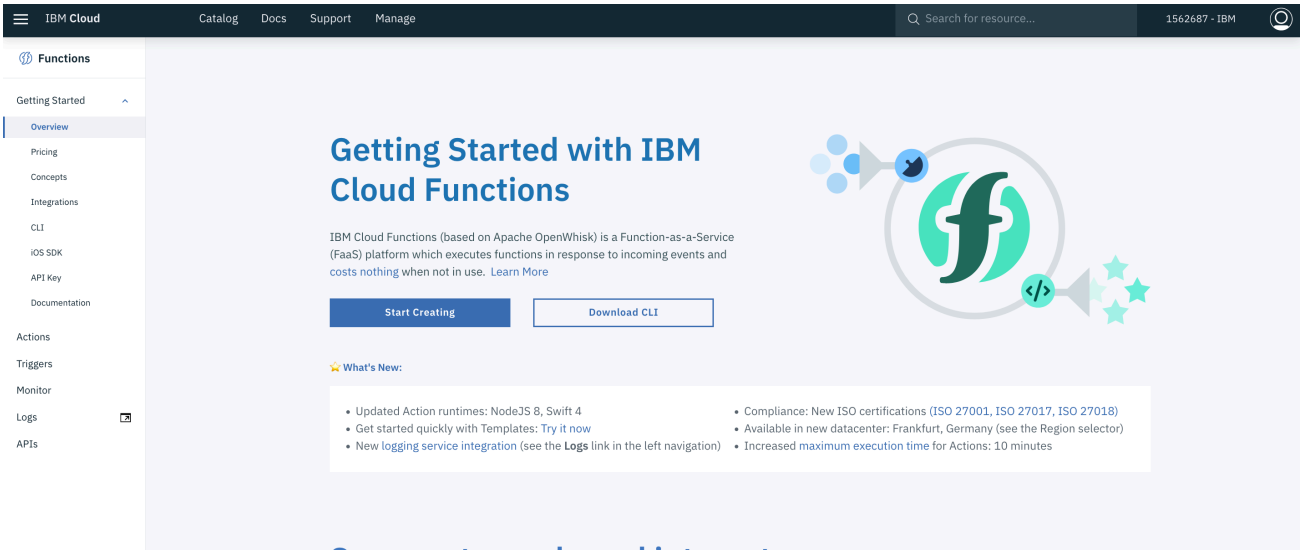
1. Navigate to the IBM Cloud Catalog by clicking on Catalog in the top left-hand corner. Next enter “Cloud Functions” and click Filter.

2. Click on the Functions tile in the IBM Cloud catalog:

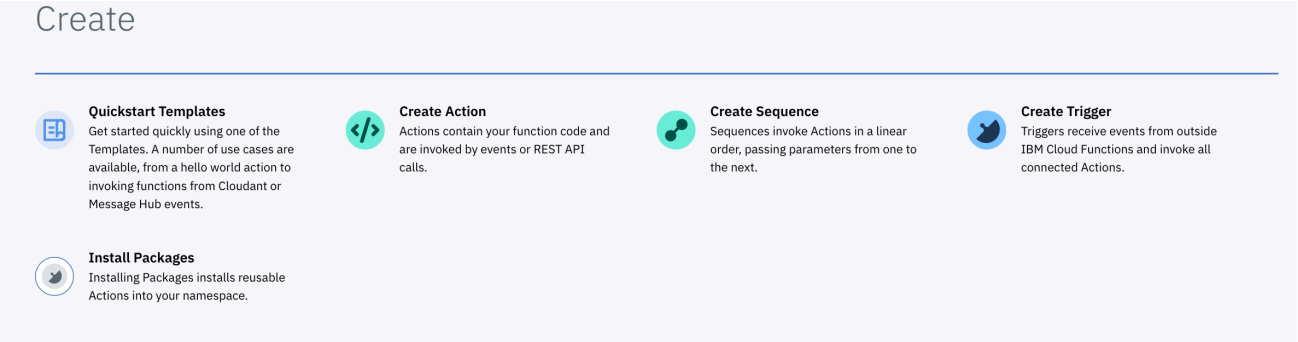


3. Click on “Start Creating”.

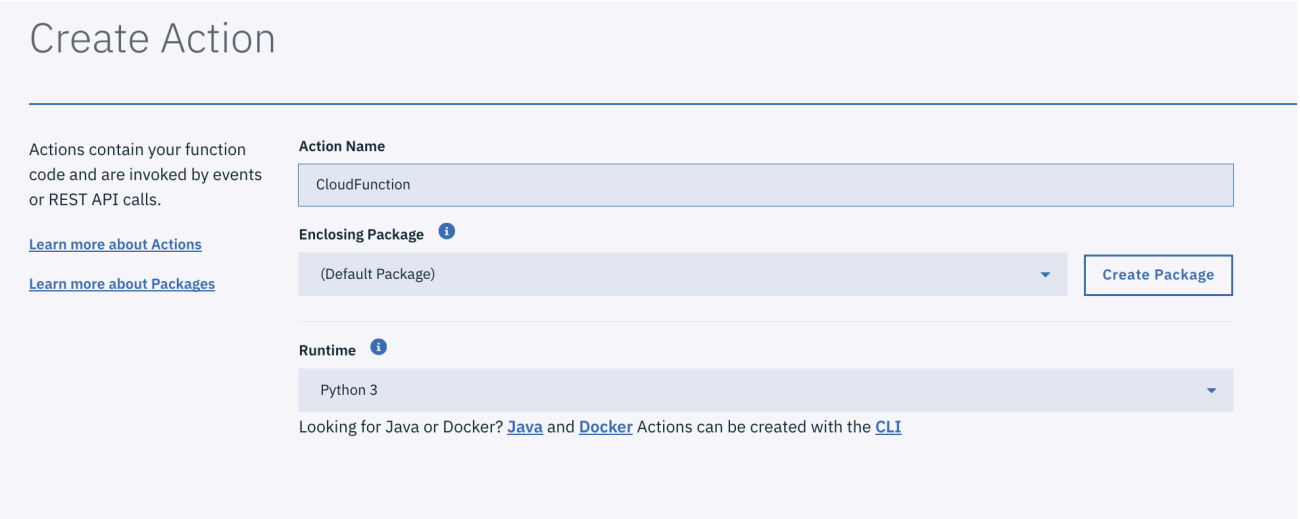
   Note: If a warning message is displayed indicating that a space needs to be provisioned, follow the instructions to create a space.



4. Click on “Create Action”



5. Provide a name for the action in the "Action Name" field and select the runtime as Python 3.



6. A new cloud function is created.

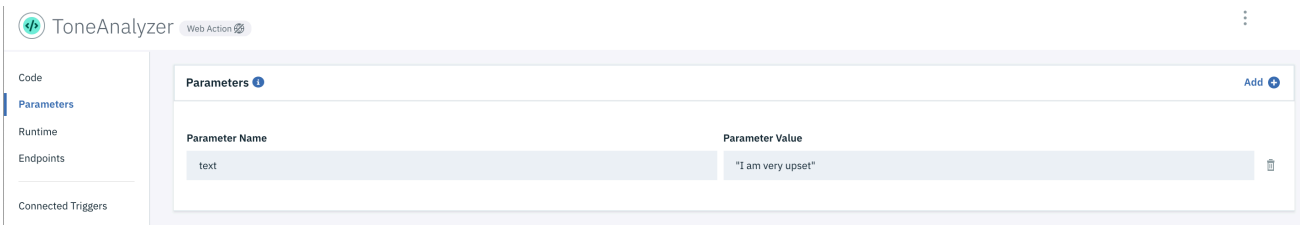
7. Insert the following code snippet that will take the input that is sent to the cloud function and send it to Tone Analyzer for processing and return the first tone identified by the cloud function. This code is almost identical to the code from the first part of the lab.

```
#
#
# main() will be run when you invoke this action
#
# @param Cloud Functions actions accept a single parameter,
which must be a JSON object.
#
# @return The output of this action, which must be a JSON
```

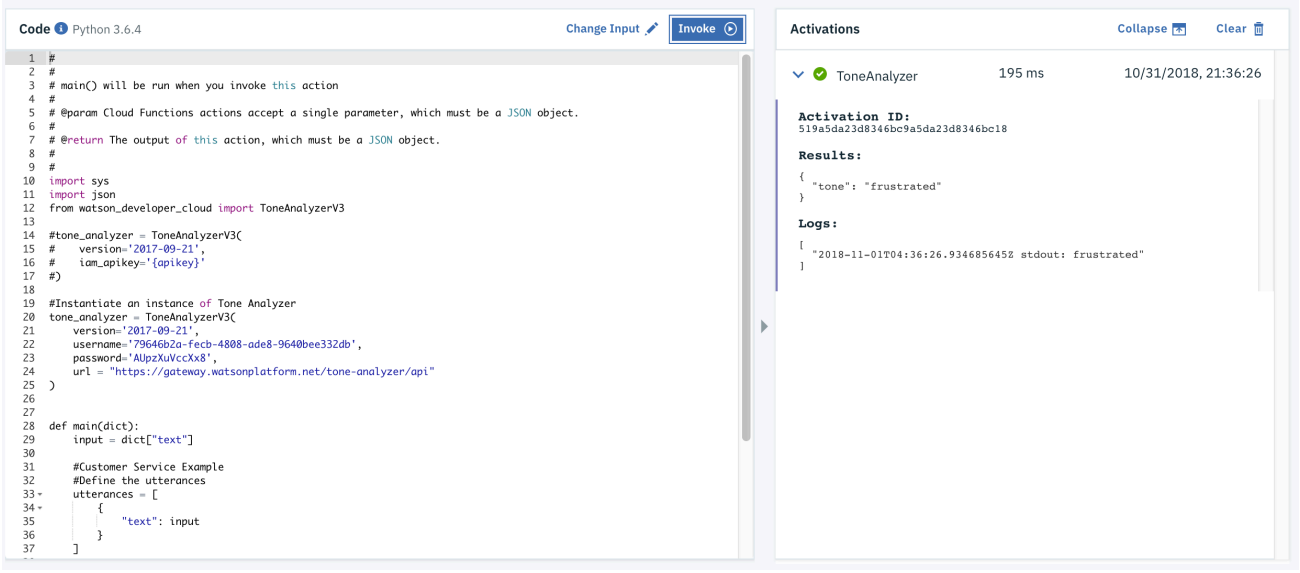
```
object.
#
#
import sys
import json
from watson_developer_cloud import ToneAnalyzerV3

#tone_analyzer = ToneAnalyzerV3(
    version='2017-09-21',
    iam_apikey='{apikey}'
)

def main(dict):
    input = dict["text"]

    #Customer Service Example
    #Define the utterances
    utterances = [
        {
            "text": input
        }
    ]

    #Call Tone Analyzer
    utterance_analyses = tone_analyzer.tone_chat(utterances)

    print
(utterance_analyses["utterances_tone"][0]["tones"][0]["tone_id
"])

    # Select the first tone returned by the service
    first_tone =
utterance_analyses["utterances_tone"][0]["tones"][0]["tone_id"
]
    return {"tone": first_tone}
```

8. Let's add a parameter to test the cloud function. Click on "Parameters" on the left-navigation and add the following parameter:

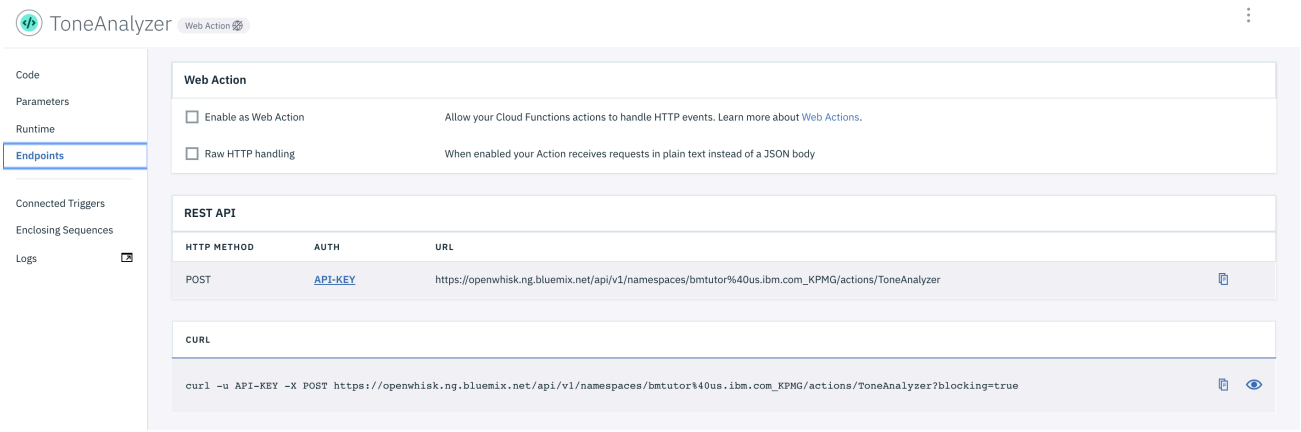
    - Parameter Name: text
    - Parameter Value: "I am very upset!"



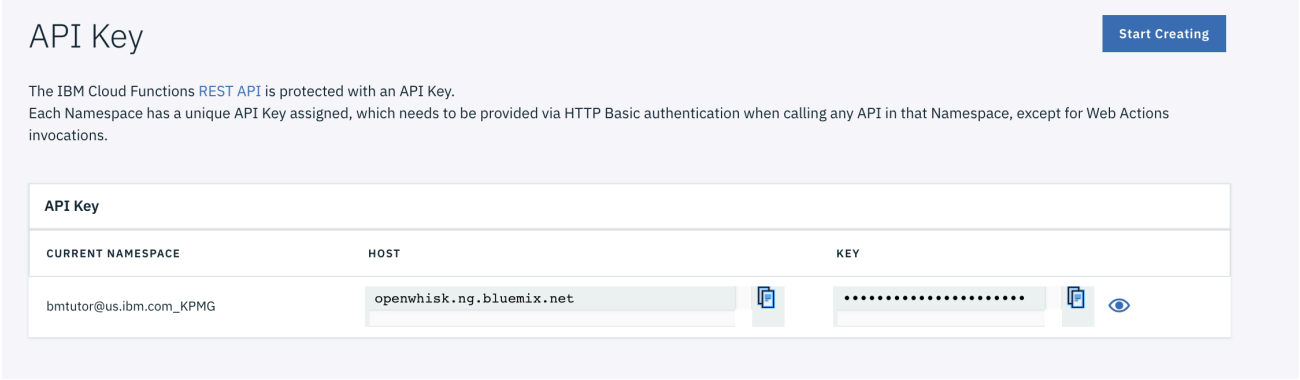
9. Click on “Code” in the left-navigation and click on “Invoke” in the top-right section of the page. The result of the Cloud Function is displayed.



10. Navigate to the “Endpoints” on the left-navigation and note the “URL” under the “REST API” section.



11. Click on “API-KEY” and record the “KEY” on the next page.

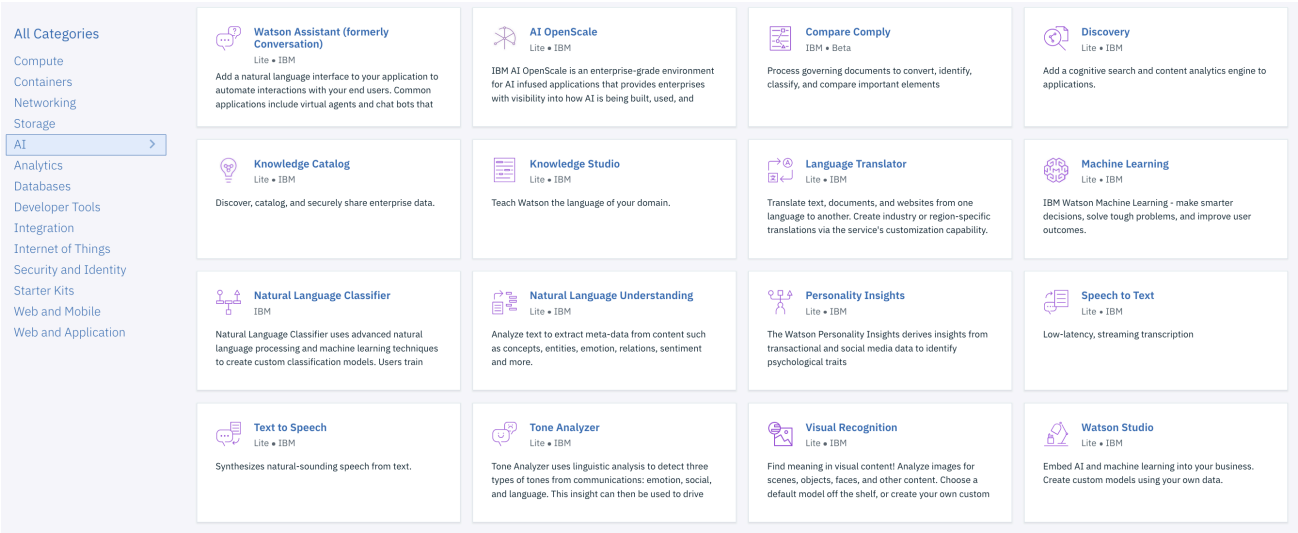


3.1.2 Create the Watson Asistance service

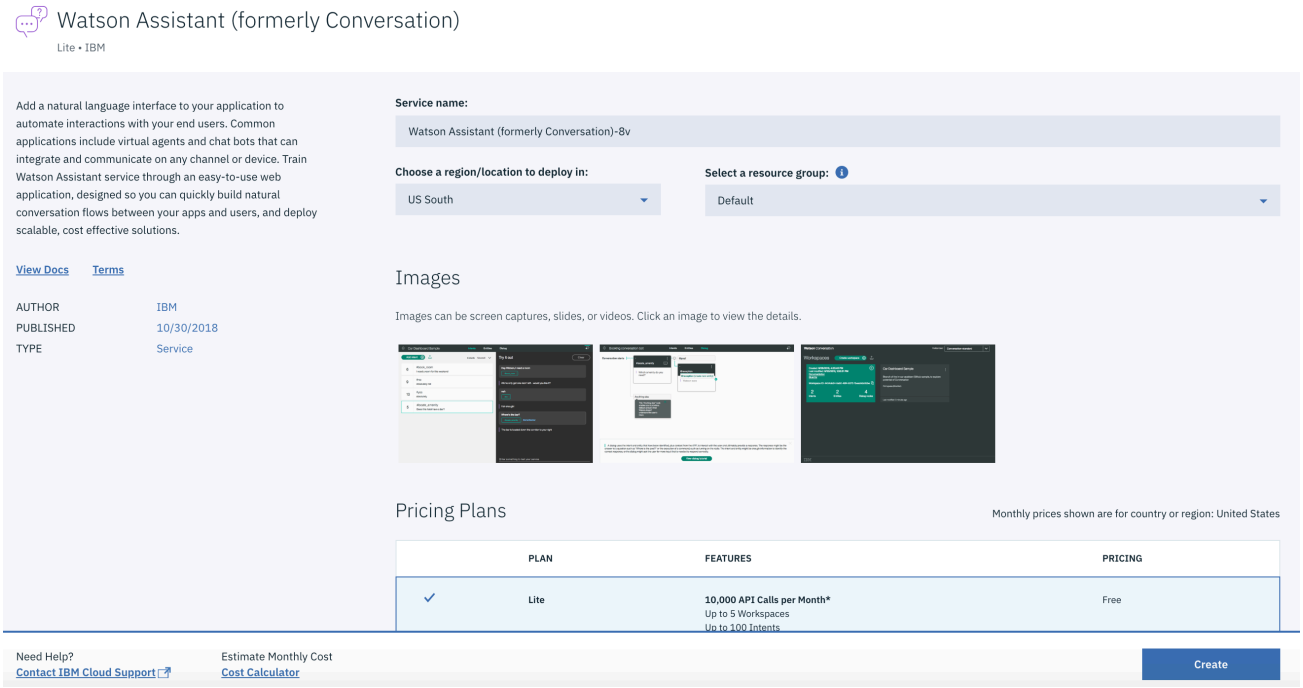
Now let's setup the integration with the Watson Assistant service.

1. Navigate to the IBM Cloud Catalog by clicking on Catalog in the top left-hand corner. Next select the AI category from the category navigation. Alternatively, you can follow this link - https://console.bluemix.net/catalog/?category=ai

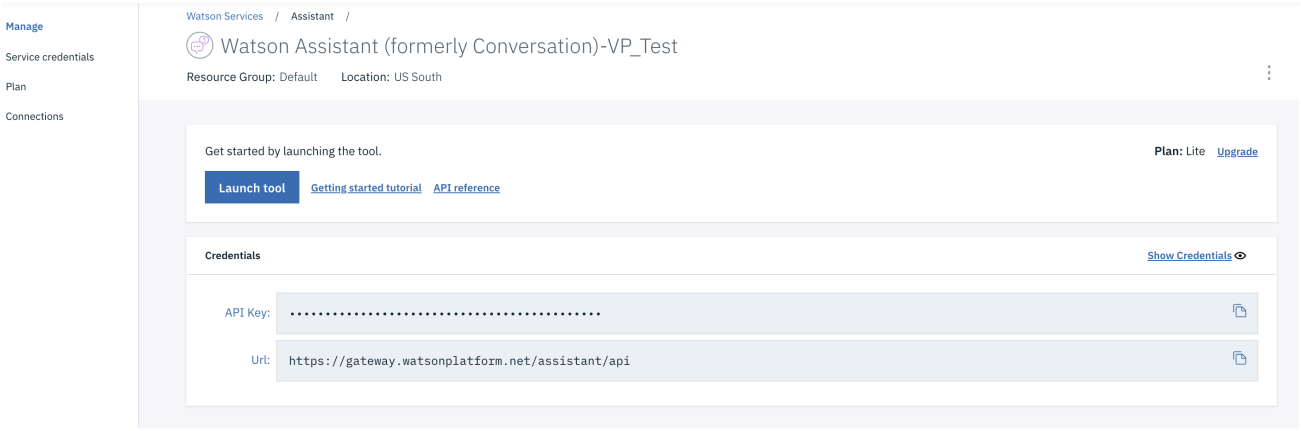
2. Click on the Waston Assistant tile in the IBM Cloud catalog:



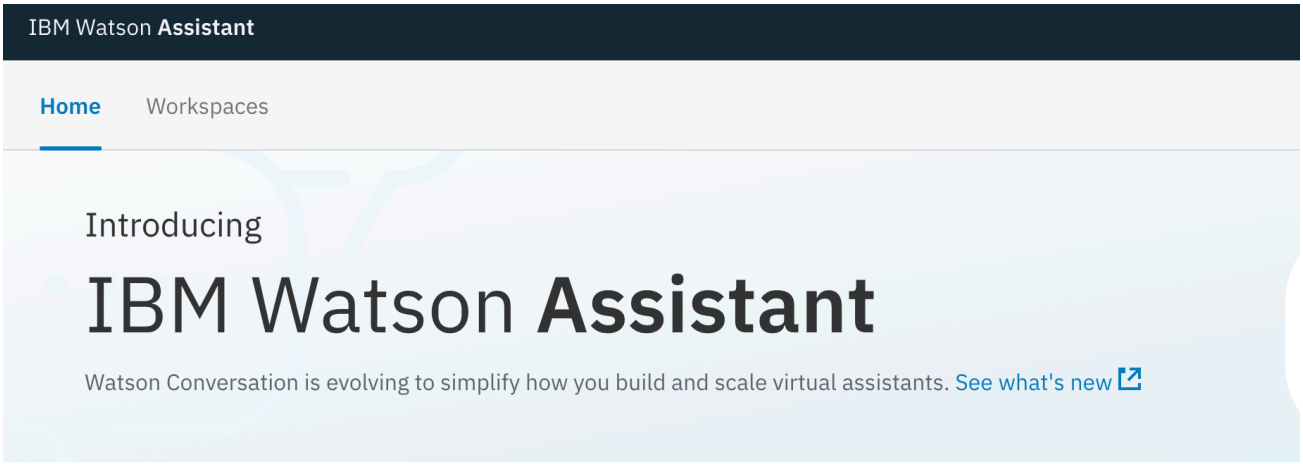
Within this window, do not make any changes to the Credential name or Service name. Create a trial service instance by selecting the Lite Plan option (default setting) and by clicking Create.



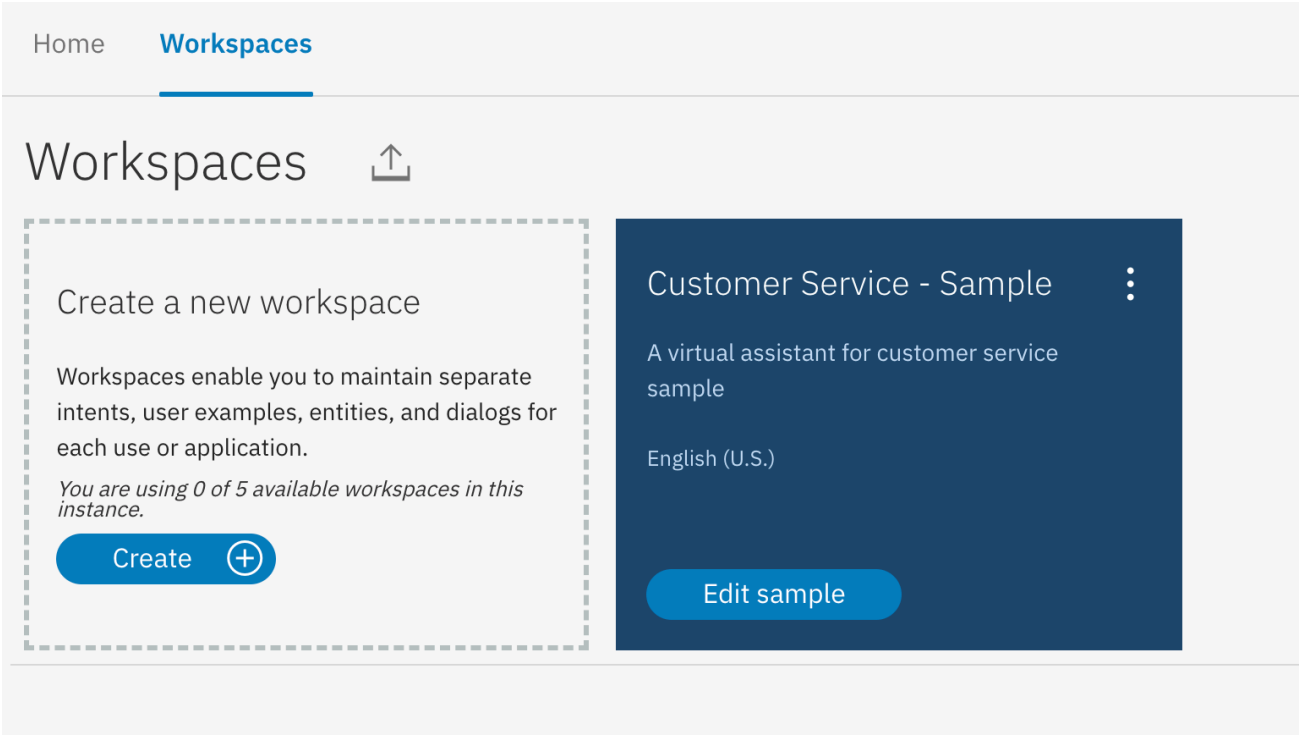
3. Click on "Launch Tool"



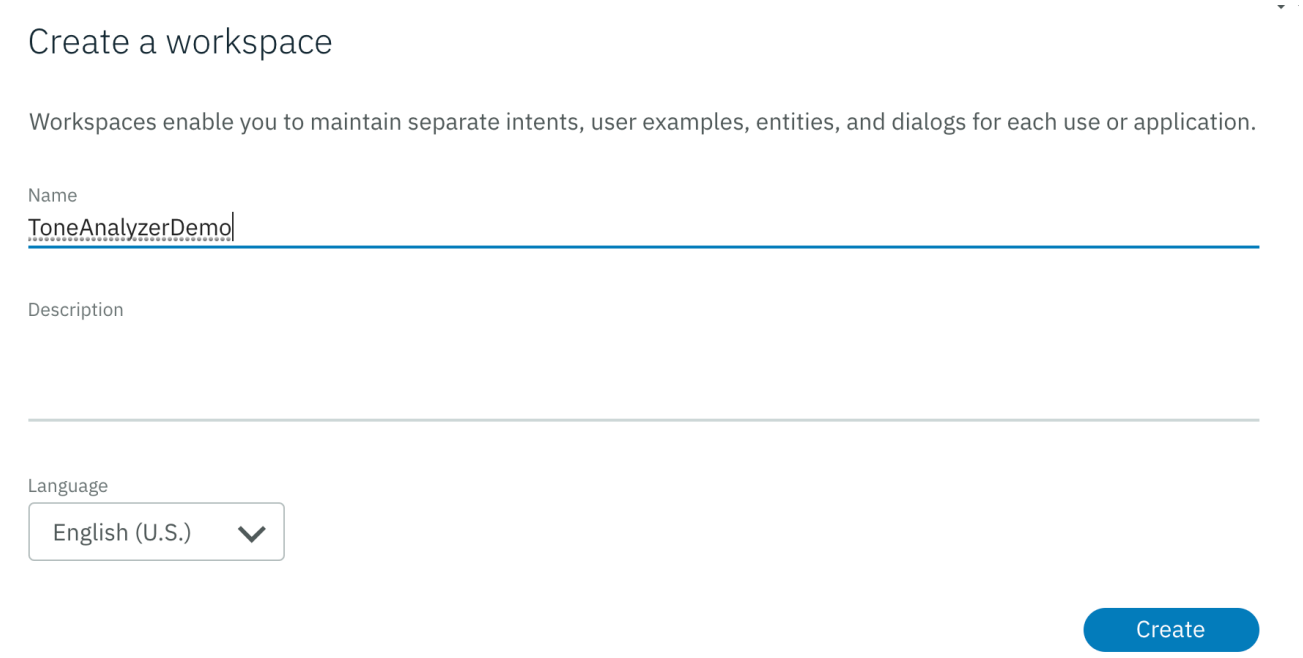
4. Click on “Workspaces” in the top-level navivation



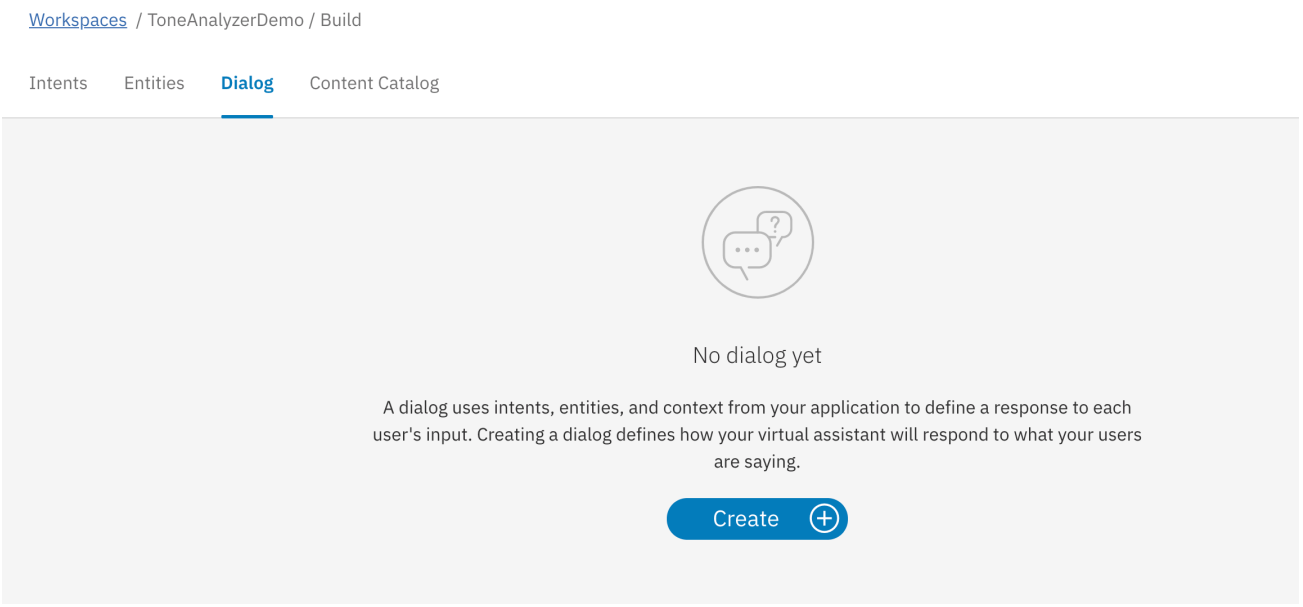
5. Click on “Create”



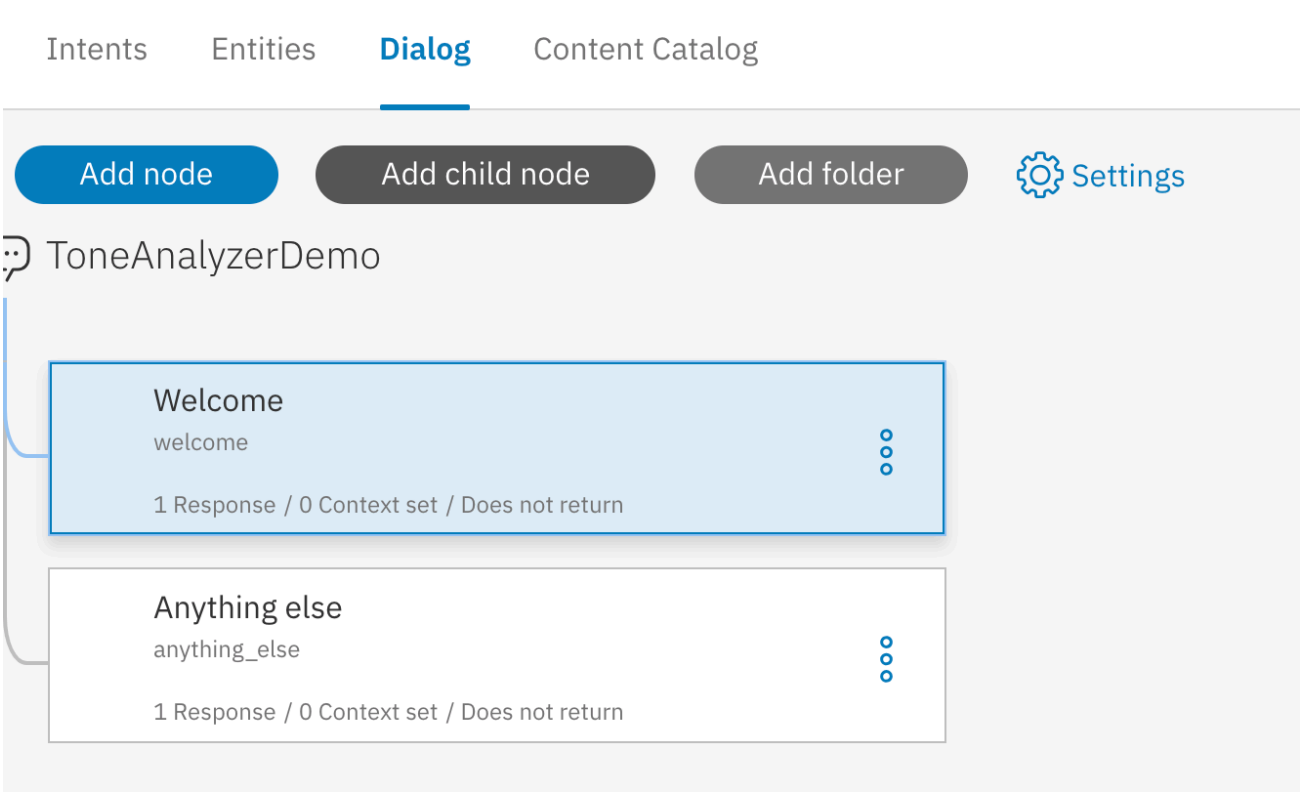
6. Enter the name for the workspace and click “Create”.



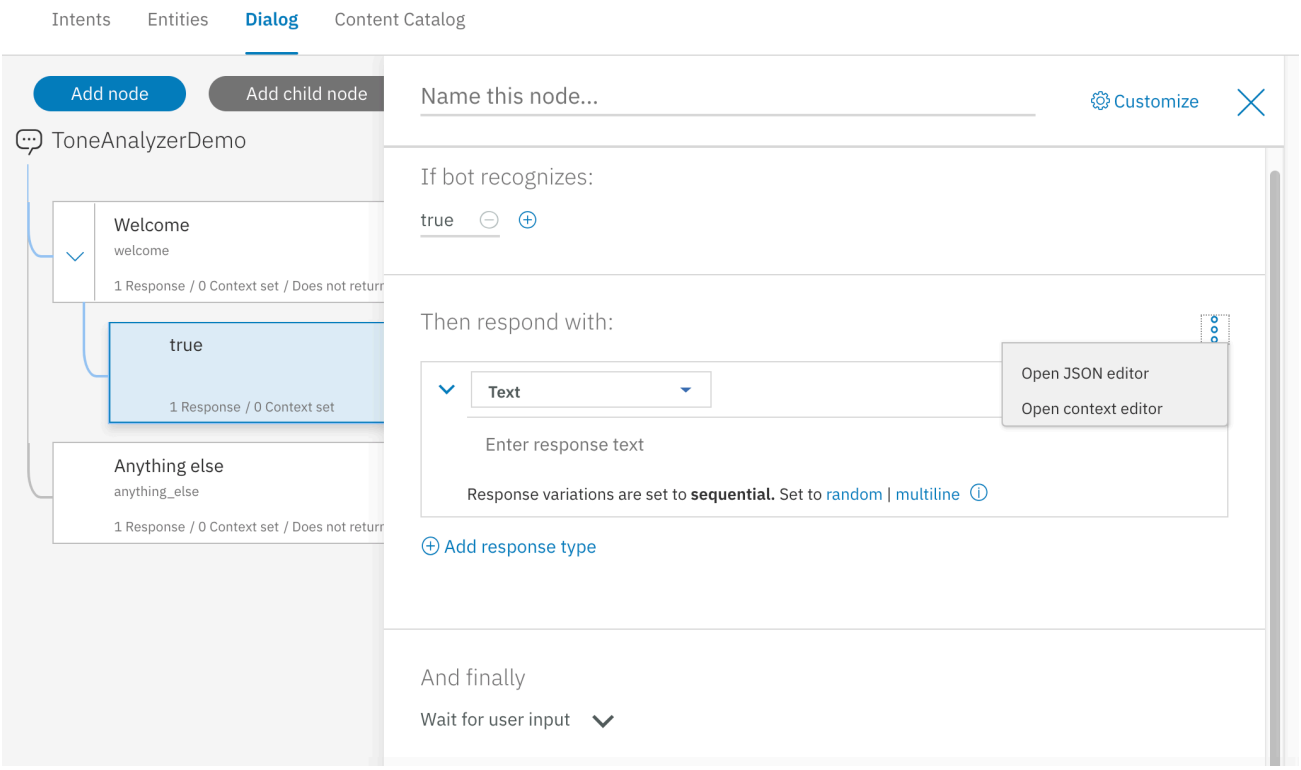
7. Click on “Dialog” in the top-left navigation and click “Create”



8. Click on the “Welcome” to select it. Click on “Add child node”.



9. Enter "true" in the "If bot recognizes" field to ensure that the node is always executed. Click on the "Options" icon and select "Open JSON Editor"



10. Enter the following information to establish a connection to the cloud function and send the user input to the Cloud Function for processing.

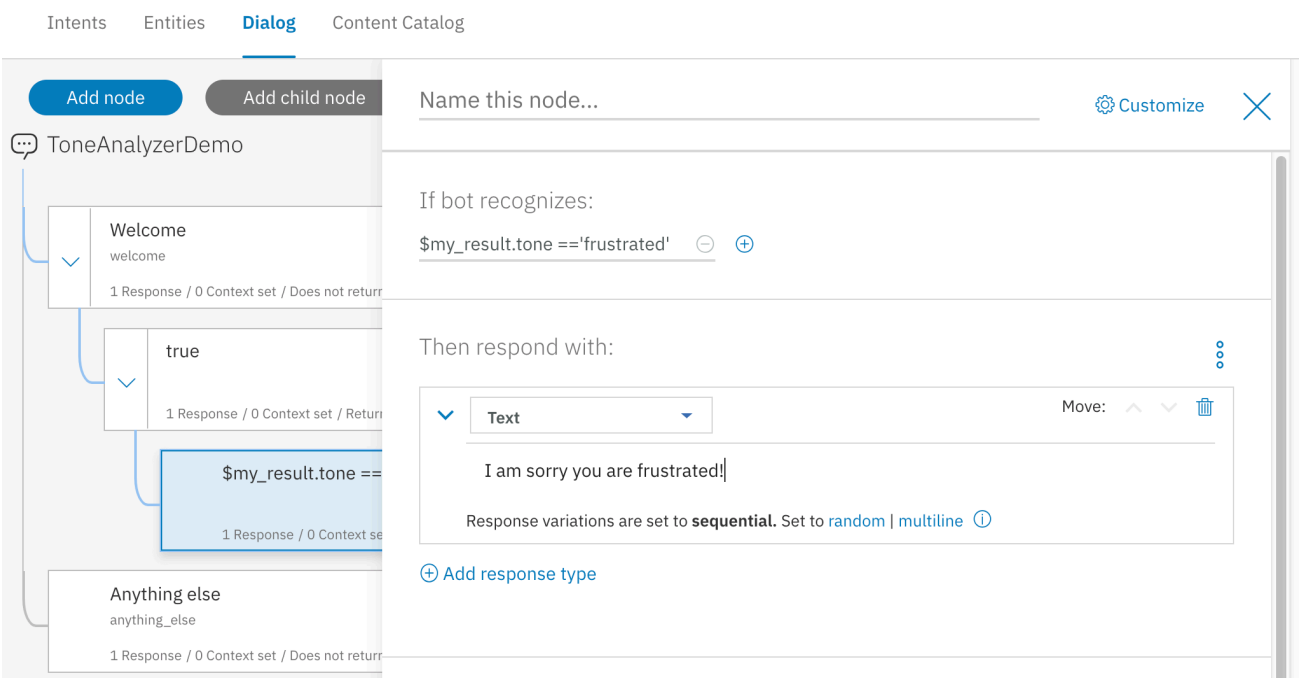
Note: Replace the name value with the corresponding URL value copied in the previous section. Make sure to replace "%40" with the "@" symbol and only keep the data after the "namespaces" part of the URL.

The following bit of code will invoke the Cloud Function we created in the previous section and send the text entered by the user to the Cloud Fuction, which in turn will send it to Tone Analyzer. The credentials to connect to the Cloud Function would typically be injected by the Service Orchestrator into the private.my_credentials context variable. A context variable that starts with "private" will not be logged by the Watson Assistant service. The result returned by the Cloud Function will be stored within the "my_result" context variable.

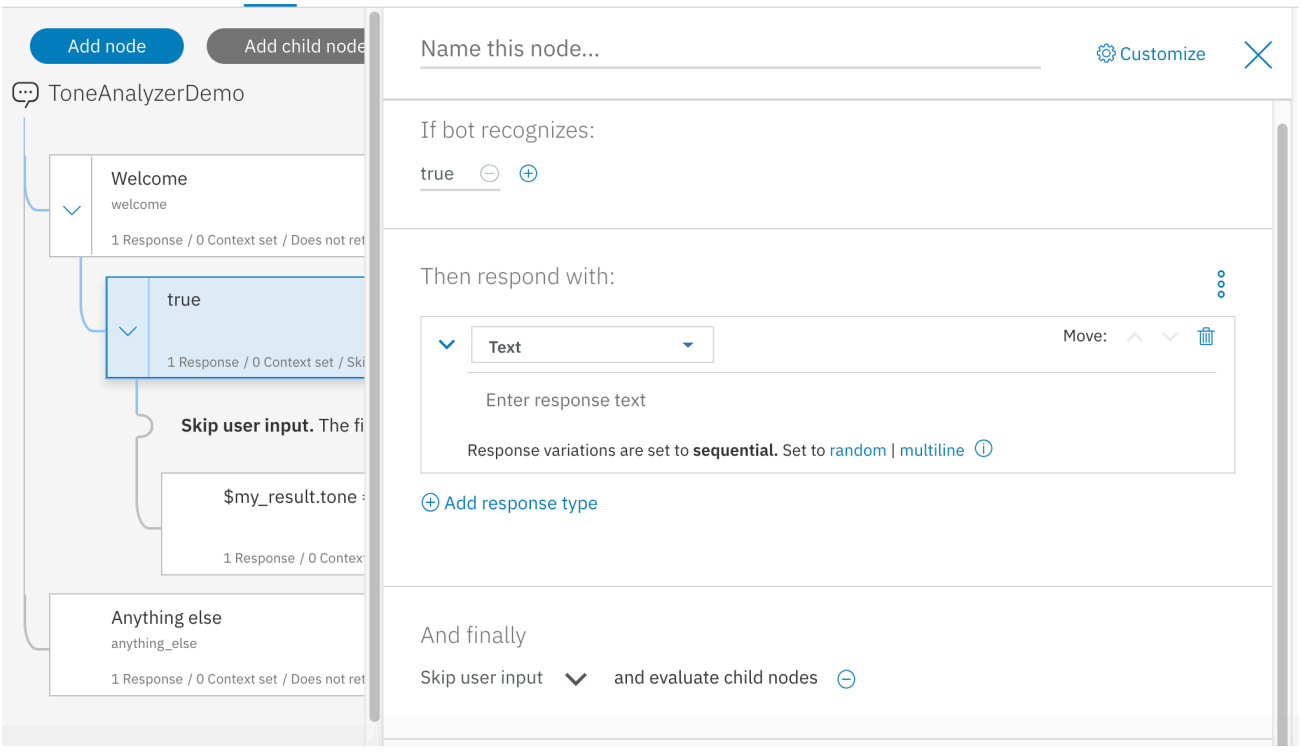
```
{
  "output": {
    "text": {
      "values": [],
      "selection_policy": "sequential"
    }
  },
  "actions": [
    {
      "name": "/bmtutor@us.ibm.com_KPMG/actions/ToneAnalyzer",
      "type": "cloud_function",
      "parameters": {
        "text": "<?input_text?>"
      },
      "credentials": "$private.my_credentials",
      "result_variable": "$my_result"
    }
  ]
}
```

11. In Dialog select the "true" node and click "Add Child Node". Enter the condition "$my_result.tone =='frustrated'" and the answer "I am sorry you are frustrated".

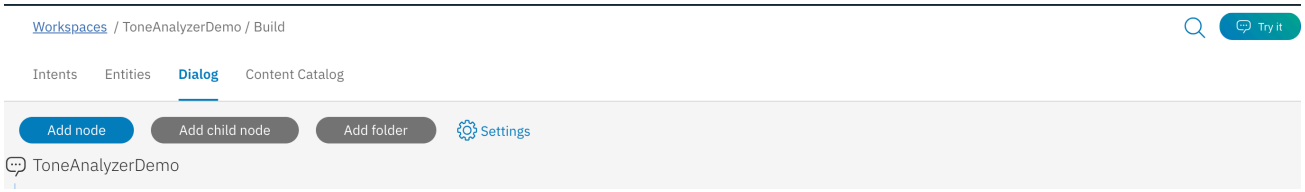
    The above condition examines the "my_result" context variable which would contain the result returned by the Cloud Function and if the value is equal to "frustrated", then the answer will be presented to the user.



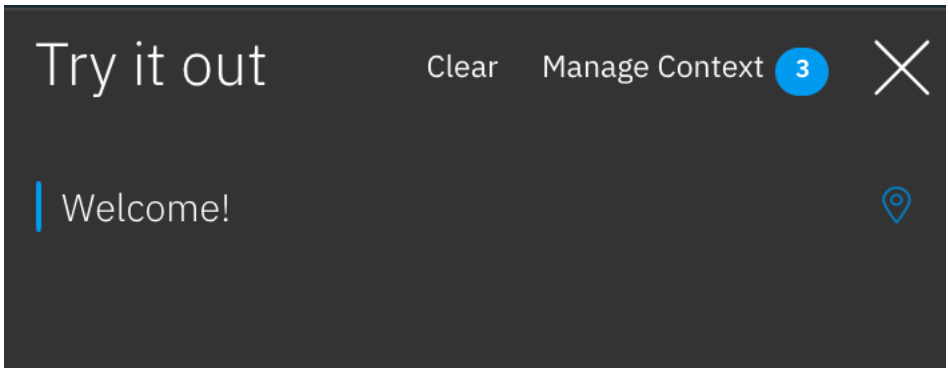
12. Click on the true node and modify the "Finally" condition at the bottom of the page to the value "Skip user input and evaluate child nodes".



13. Click on the "Try it" button in the top-right of the Dialog page.

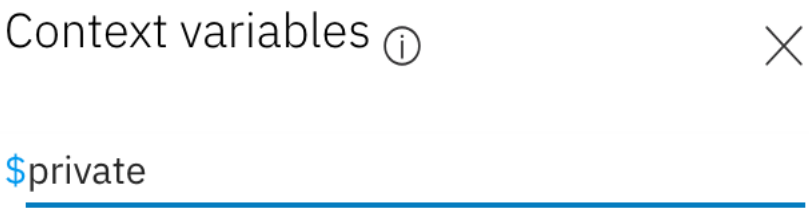


14. Click on “Manage Context” in the top right of the Try it Out panel.



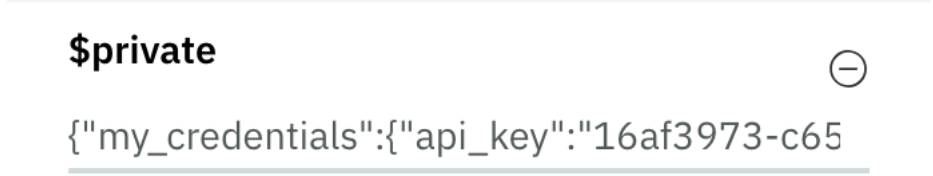
15. As we do not have an integration with a Service Orchestrator, we’ll simulate the credentials being passed via the context variable by adding them directly into the context.

    Enter “private” as the variable name.

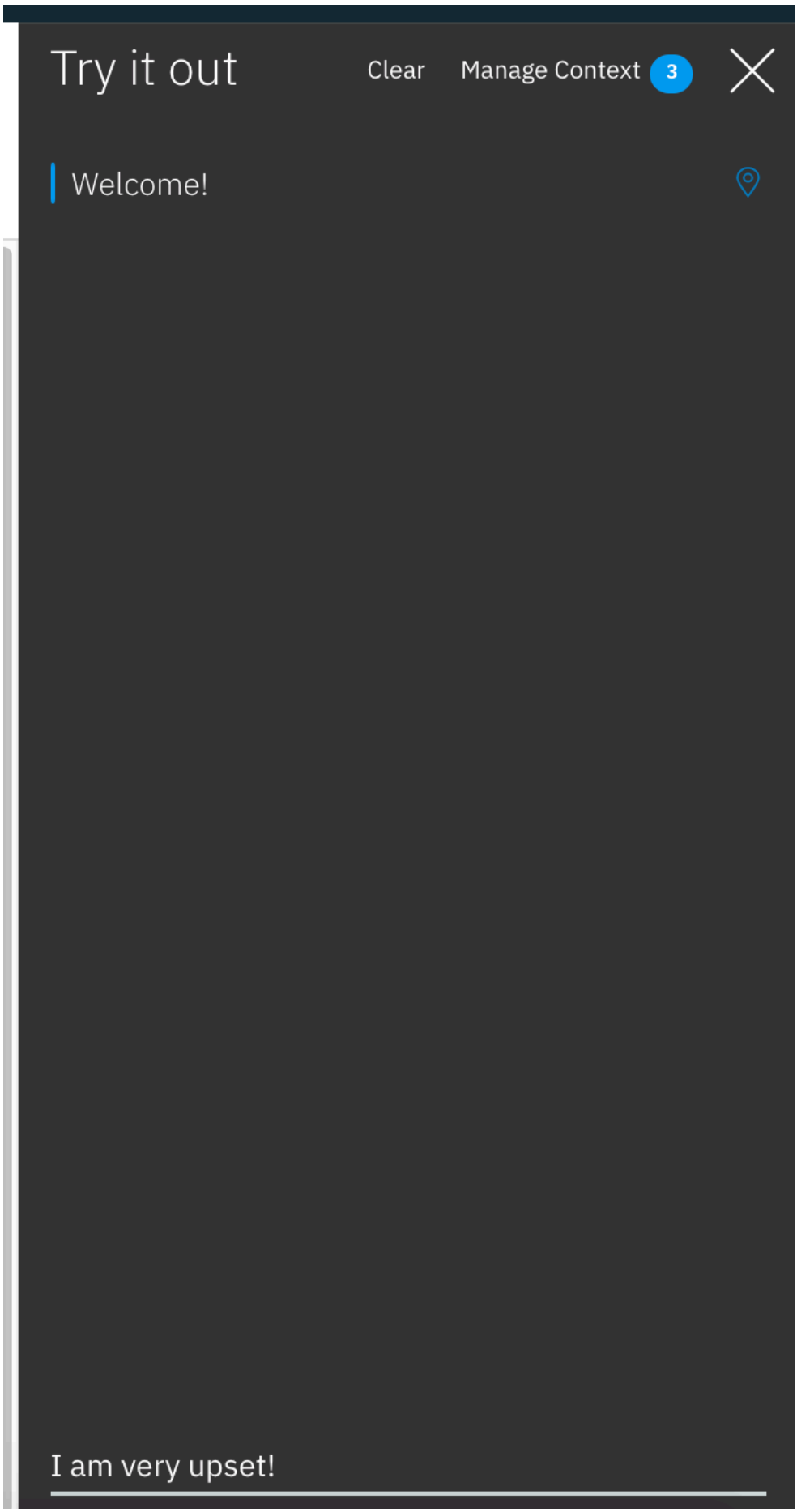


16. Enter the following value for the “private” variable, replace “{api key}” with the API KEY copied for the Cloud Function in the previous section.

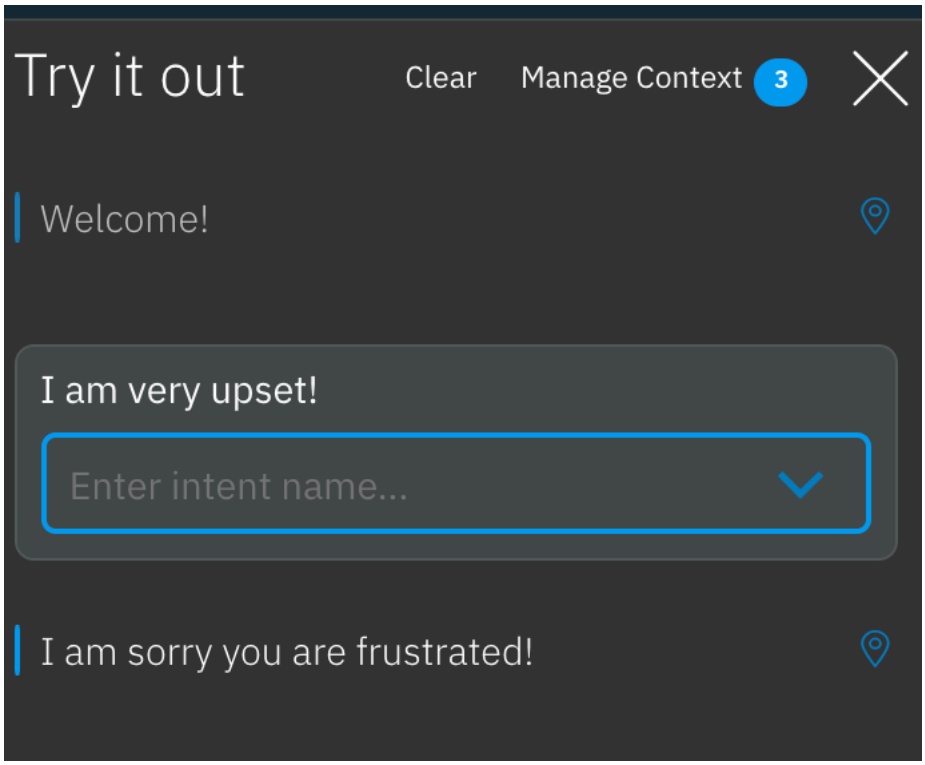
```
private
{"my_credentials":{"api_key":"{api key}"}}
```



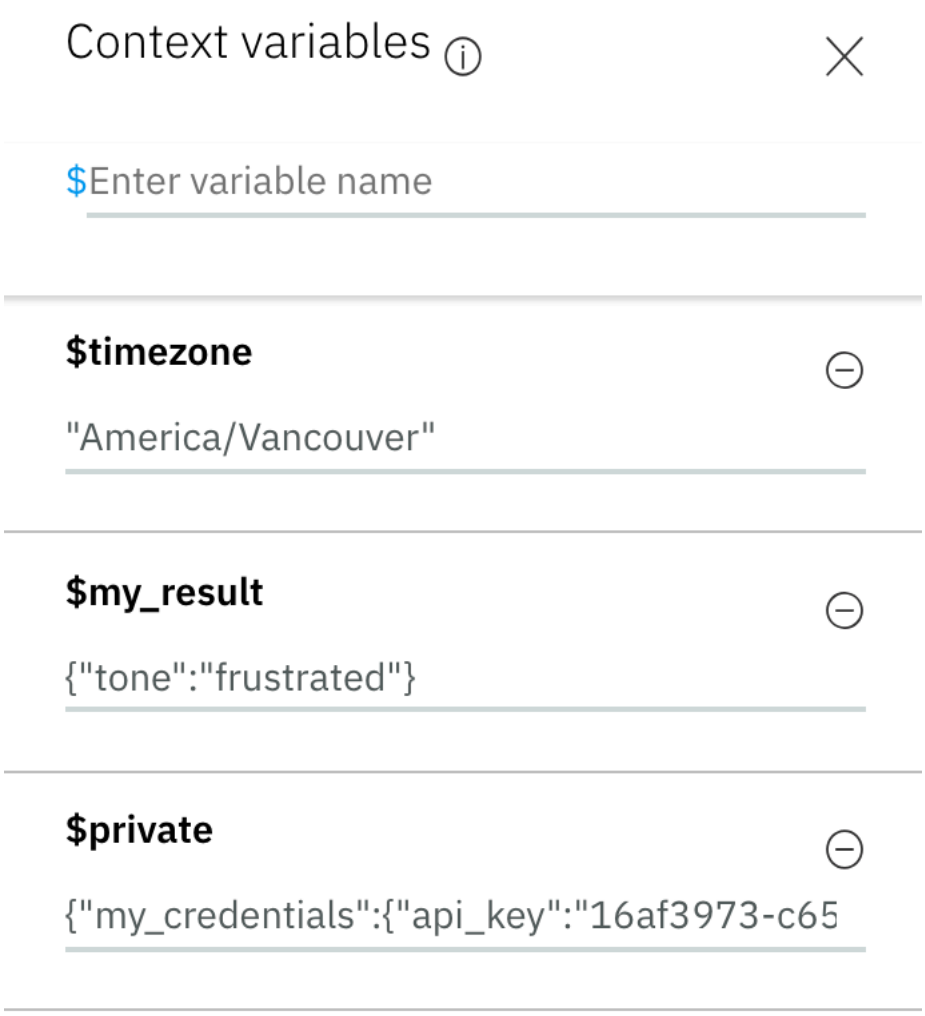
17. Close the “Context Variables” section by click the “X” in the top right of the panel. Enter “I am very upset!” in the “Try It Out” panel.



18. The system wil return the answer “I am sorry you are frustrated!”



19. We can verify the information returned by the Cloud Function by clicking on the “Manage Context” link in the top-right of the “Try It Out” panel and examining the “my_result” context variable.



20. Add additional Dialog logic and try different examples to extend the prototype!

4 Data collection

By default, all Watson services log requests and their results. Logging is done only to improve the services for future users. The logged data is not shared or made public. To prevent IBM from accessing your data for general service improvements, set the X-watson-Learning-Opt-Out header parameter to true when you instantiate the service. (Any value other than false or 0 disables request logging for that call.) You must set the header when you create the service for any call that you do not want IBM to access for general service improvements.

Python:

```
from watson_developer_cloud import ToneAnalyzerV3

tone_analyzer = ToneAnalyzerV3(
    version='2017-09-21',
    iam_apikey='{api_key}'
)

#Opt out of learning
tone_analyzer.set_default_headers({'x-watson-learning-opt-
out': "true"})
```

5 Error Handling

The Tone Analyzer service uses standard HTTP response codes to display whether a method completed successfully. A 200 response always indicates success. A 400 type response is some sort of failure, and a 500 type response usually indicates an internal system error.

STATUS	DESCRIPTION
200 - OK	OK. The request succeeded. If the input is partially correct, the response can include `warning` or `error` fields with appropriate messages.
400 - BAD REQUEST	A required input parameter is null or a specifiec input parameter or header value is invalid or nc supported. The response is also returned if all utterances of the input have more than 500 characters.
401 - UNAUTHORIZED	Access is denied due to invalid service credentials.
404 - NOT FOUND	The requested item or parameter doesn't exist.
429 – TOO MANY REQUESTS	The service is throttling your request because your IBM Cloud ID submitted more than 1200 requests per minute.
500 - SERVER ERRORS	Internal server error.

6 References

- The science behind the service: https://console.bluemix.net/docs/services/tone-analyzer/science.html#the-science-behind-the-service
- Tone Analyzer Sample Applications:
  - General Model: https://tone-analyzer-demo.ng.bluemix.net/
  - Customer Engagement: https://customer-engagement-demo.ng.bluemix.net/
- API Reference: https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/
- Tone Analyzer Documentation: https://console.bluemix.net/docs/services/tone-analyzer/getting-started.html#gettingStarted