Chatbot using Azure Services

|  |  |  |
| --- | --- | --- |
| References | Pre-requisites | Tools, Frameworks, API’s & Services used |
| * <https://home.openweathermap.org/> * <https://towardsdatascience.com/nlp-vs-nlu-vs-nlg-know-what-you-are-trying-to-achieve-nlp-engine-part-1-1487a2c8b696> * <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1> | * Microsoft account for Azure * Luis Account (You can leverage Microsoft account) * Weather API account sign-up * Telegram account key for integration * Github Account | * Python 3+ * Flask * Weather API * Microsoft Bot Emulator * Any python IDE :PyCharm * Azure App Service Web Apps * Git local setup |

|  |  |  |
| --- | --- | --- |
| Use cases |  |  |
| * Customer Experience Management * Customer Support * Scheduling Appointments * FAQ Buddies * Workflow automations * Product Enquires |  |  |

A chatbot is a computer application that creates human chat conversation experiences with text & voice. One of AI based feature that can be integrated through the messaging services provided by cloud providers. Each provider has their own services wrapped around their services which does lot of heavy lifting with NLP, NLU & NLG.

Some of the major NLP powered virtual assistants that are available in market are :

* Google Now
* Apple Siri
* Microsoft Cortana

Image from how to chatbots work? Call workflow is as shown below (1 to 3 and vice-versa)

|  |  |  |
| --- | --- | --- |
| Channels (1) | Bot Instance (2) | NLP Engine (3) |
| * Web Apps * Mobile Apps * Messaging Apps * Etc…. | * Custom Bot App deployed * App Logic * Database * Integrations if any | * AI/Neural Network * NLU, NLU, NLG * Speech to Text * Text to Speech * Intents, Entities |

Some simple glossary to understand before jumping into our implementation journey. We are going to deal with first 2 of the below rest you can reference from links shared.

* Intent
* Entity
* Tokenization
* Stemming
* Lemmatization

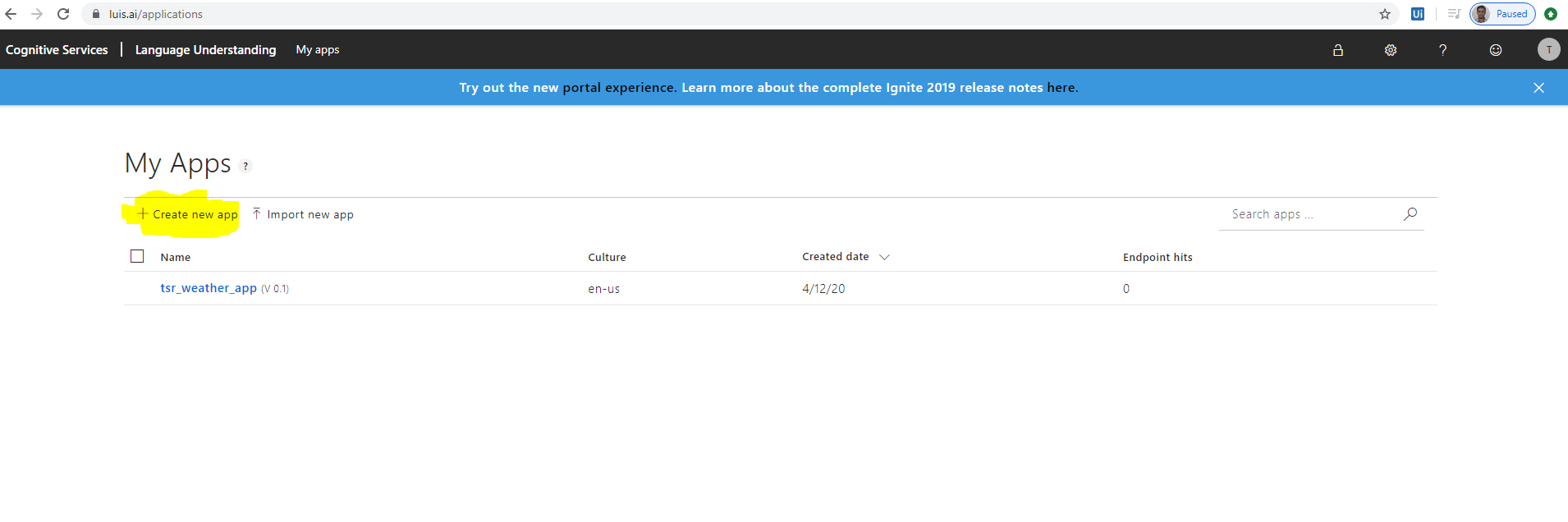
Types of chatbots:

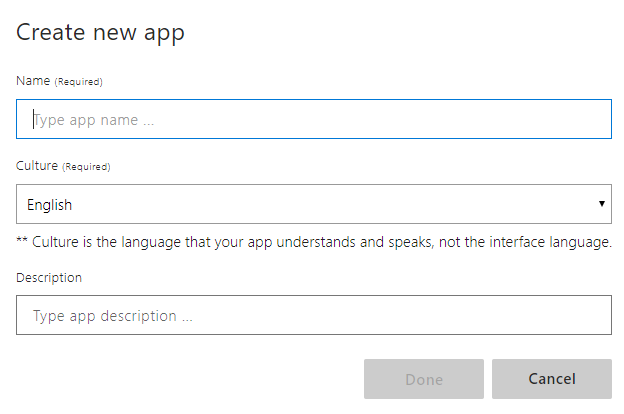
* Rule based chatbots : A bunch of if-else statements programmed to handle the conditions, used by technical folks
* NLU based chatbots : A more generic chatbots which can used by common people, they have ability to understand the intent & entities using NLU. Powered by AI NLP engines.

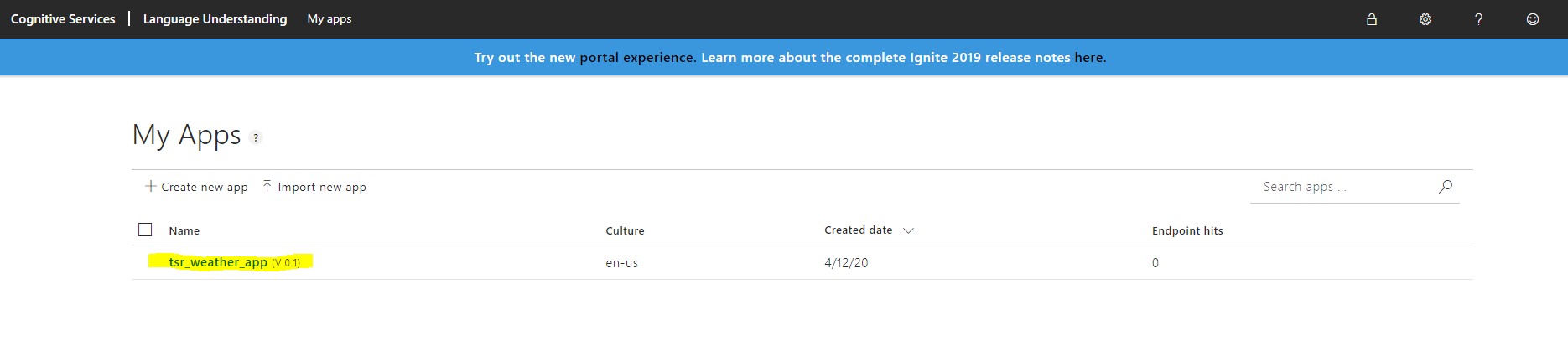
Traditional way of developing the NLP application which requires lot of data, data cleaning, feature setup, model selection, training model & deployment. No of iterations to test each of the model for its performance will only increase time and the complexity setting major barriers for the business usecase to be deployed in production. Hence the new approach is to leverage the NLP engine services provided by Google, AWS & Azure.

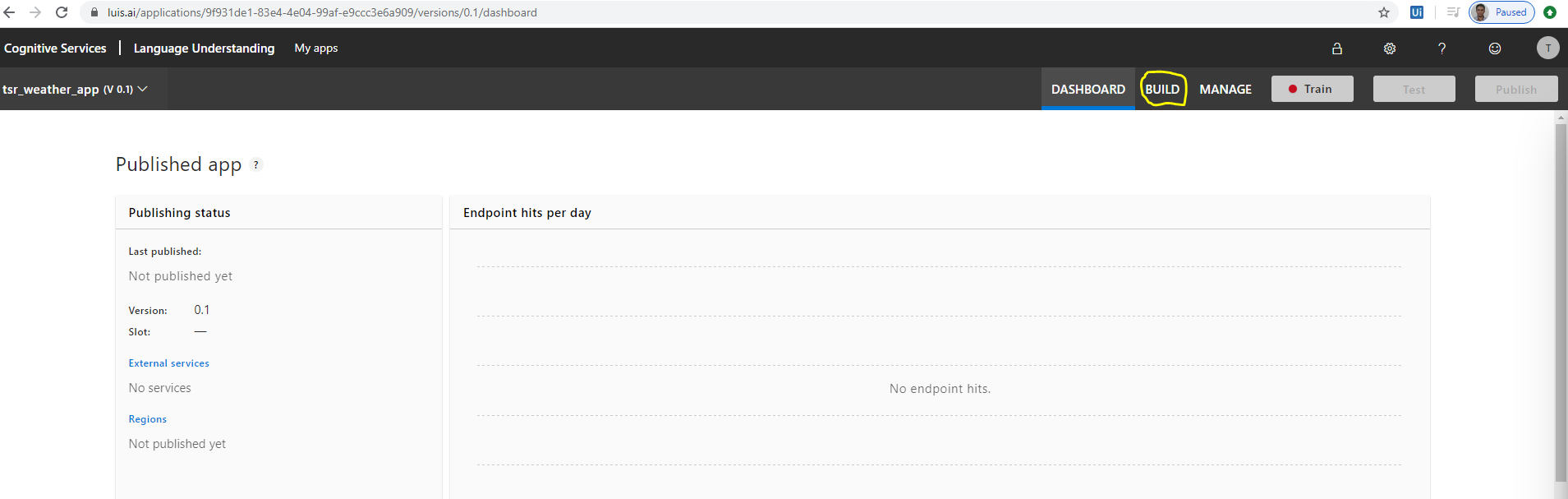
**Step 1: LUIS app creation** (LUIS : Language Understanding Intelligent Service by Microsoft)

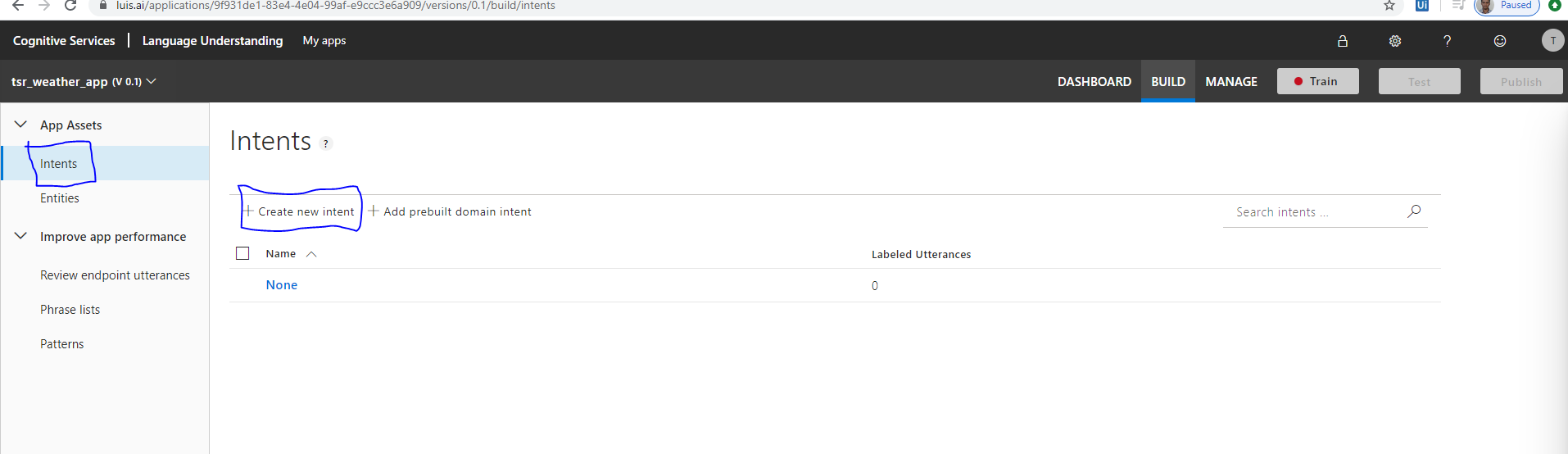
1. Go to url <https://www.luis.ai>
2. Click on Sign-up for the account creation [if you already have Microsoft account that can be leveraged as well]
3. After login if it is your first time LUIS will ask for couple of things like region, connect to Azure Microsoft account etc.
4. Click on +Create new app
5. Pop-up requests for the Name, Culture, Description ( For example : test\_app, English, test\_app) submit the form by clicking “Done”
6. Click on the new app created as shown below
7. Click on the “BUILD” option on the top right
8. Click on the left menu option “Intent” to create new intent
9. Provide the new intent name and click “Done”
10. Create the new intent (weather)
11. Create the new entity (city\_name)& add utterances (How is the weather in [Bangalore]? -> select Bangalore to map to city\_name entity created)
12. Click on the “Train” button on top right corner of the window (wait to check the red dot turning to green)
13. Cick on the “Test” button and type the chat string
14. Click on the “Manage” tab on the top right corner and select “Azure resources” option to view the endpoints, key etc (We will use these things in python app)

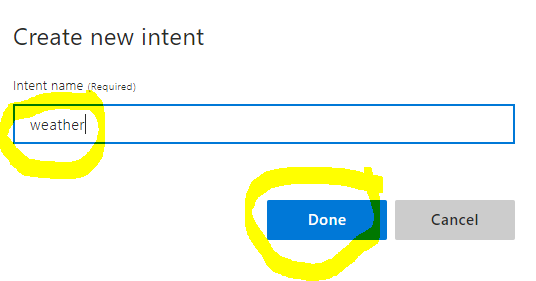


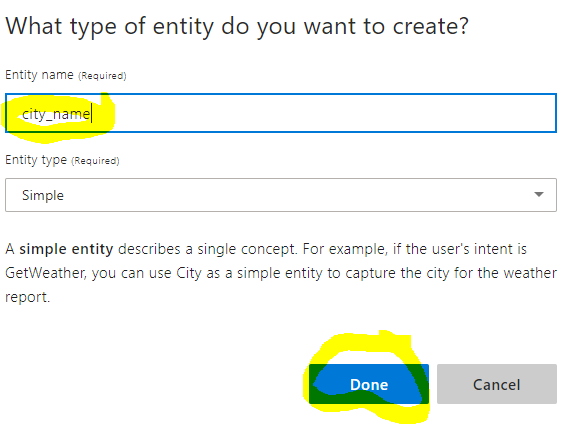


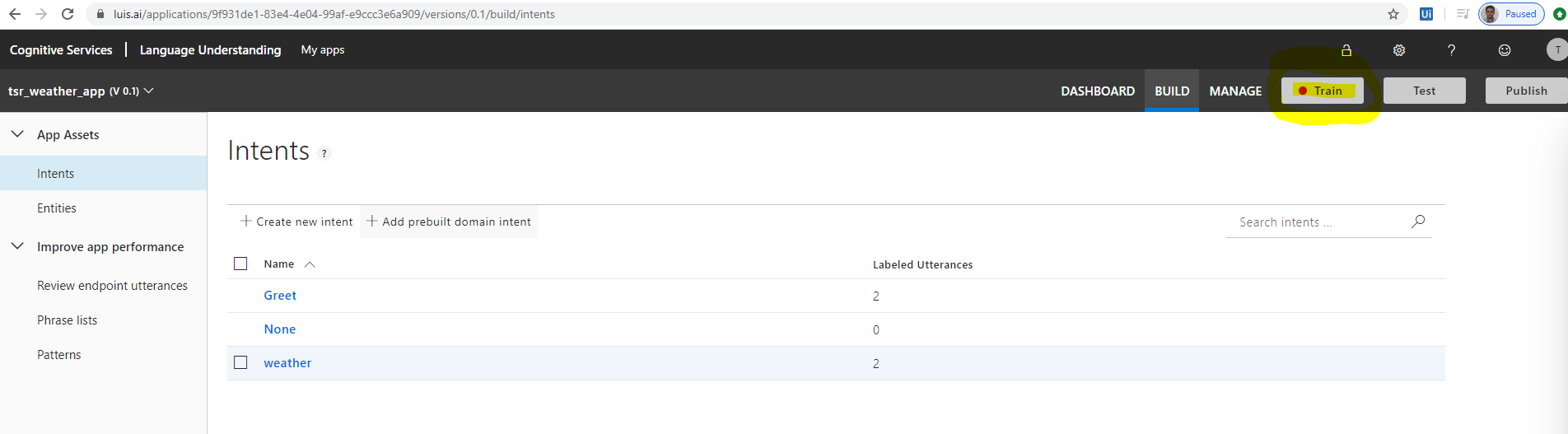


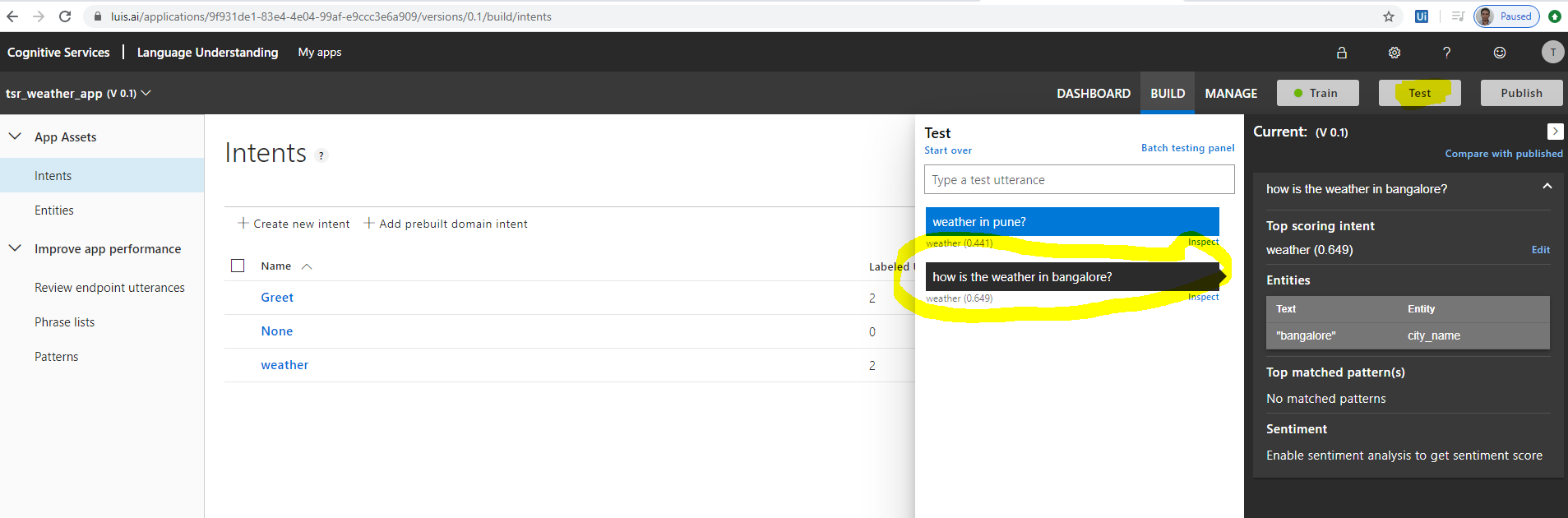


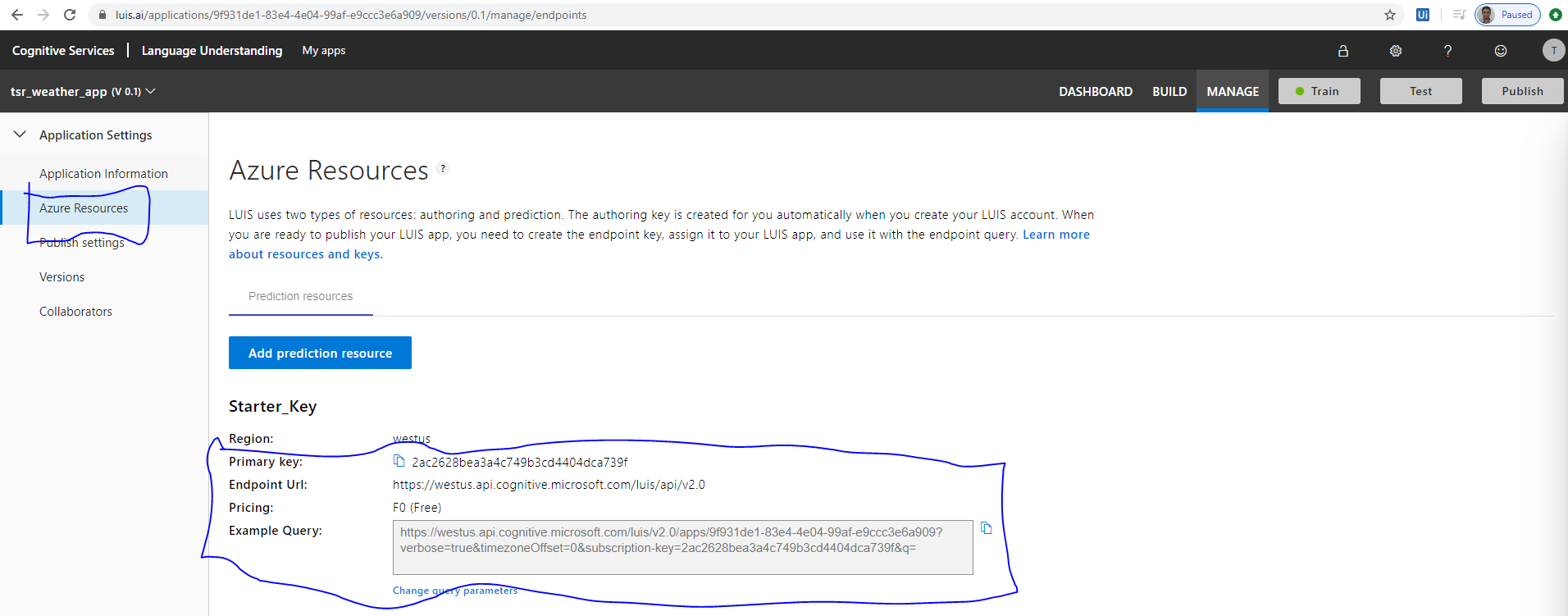


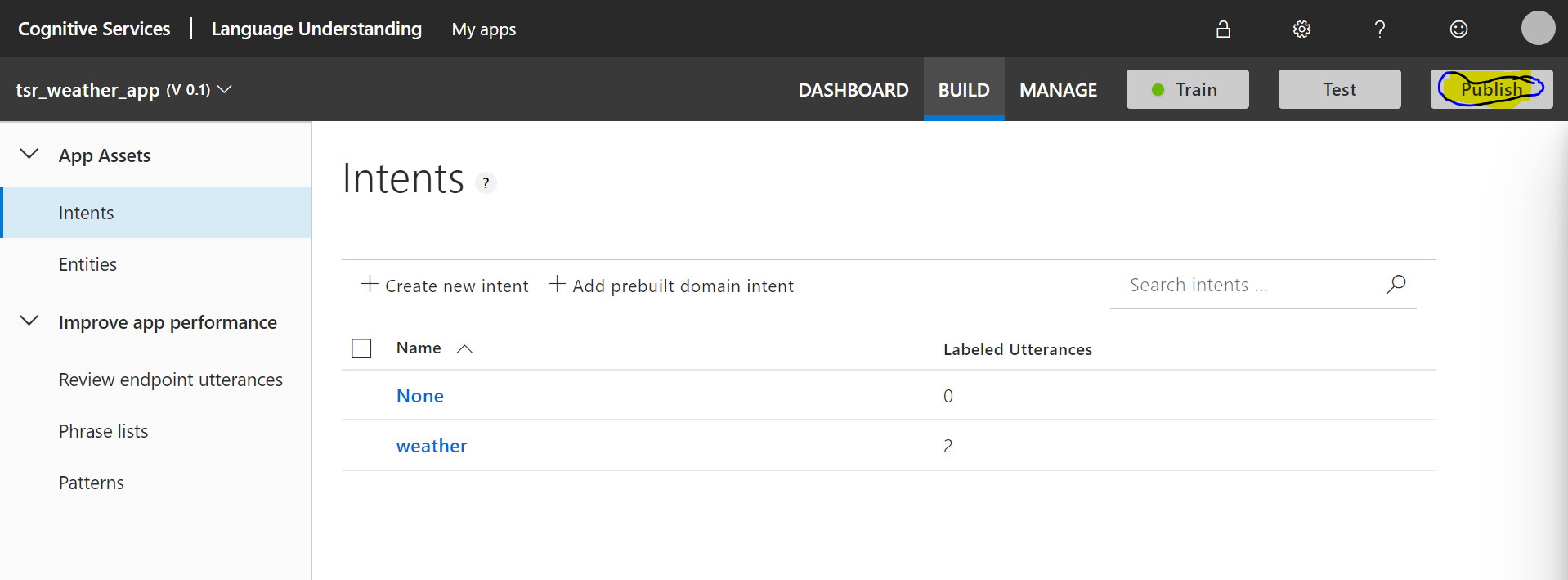


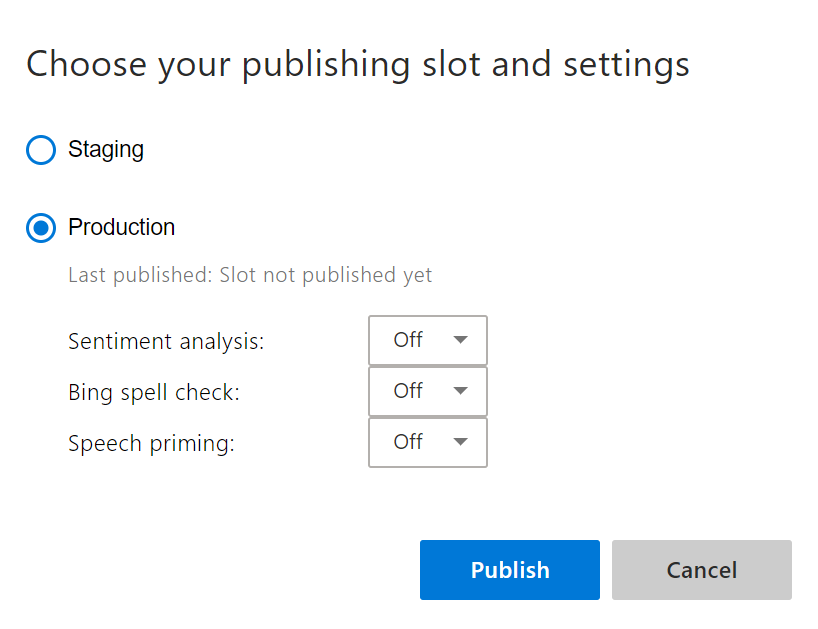












**Step 2: Python app creation**

1. Go to <https://home.openweathermap.org/>
2. Sign-up and create API Key for calling from python app (*weather\_api\_key*)
3. Open Pycharm IDE
4. Create a folder “AzurePythonBot”
5. Create folder “Config”, write the code in pycharm IDE by creation the python file for config reader i.e. config\_reader.py
6. Create folder ”conversationLogs”, create the file “TSRsession.txt” for logging the session info
7. Create folder ”logger”, write the code in pycharm IDE by creation the python file for config reader i.e. logger.py
8. Create folder ” luis”, write the code in pycharm IDE by creation the python file for config reader i.e. luisApp.py
9. Create folder ” weather”, write the code in pycharm IDE by creation the python file for config reader i.e. weather.py
10. Write the python code in the files

pip install botbuilder-ai

pip install botbuilder-applicationinsights

pip install botbuilder-azure

pip install botbuilder-core

pip install botbuilder-dialogs

pip install botbuilder-schema

pip install botframework-connector

pip install flask

Bot Emulator for Windows 64bit setup.exe

<https://github.com/Microsoft/BotFramework-Emulator/releases>

emulator runs on http://localhost:60528

openweatherAPI Key

