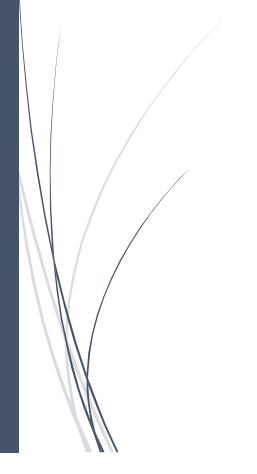
# Playbook for Watson Assistant Metrics Notebook

Reporting key performance metrics from the logs of your virtual assistant



Author: Preeth Muthusamy LAST REVISION DATE: JUNE 17, 2020

# Contents

1	Intro	Introduction			
2	Create resources				
3	3 Configure Db2 tables				
4	Create Project in Watson Studio				
5	Import the Jupyter notebook				
6	Conf	igure the notebook	. 5		
	6.1	Configure & Establish DB2 connection	. 5		
	6.2	Configure notebook with Watson Assistant credentials	. 6		
	6.3	Define the dialog nodes for Coverage and Escalation	. 7		
	6.4	Save a version of the notebook	. 8		
7	Auto	matic notebook jobs	. 8		
	7.1	Setup job schedule	. 8		
	7.2	Validate job execution	. 9		
8	Creat	te a Dashboard in Watson Studio	. 9		
	8.1	Add a Db2 Connection to your project	. 9		
	8.2	Add a dashboard to your project	10		
	8.3	Share the dashboard link	11		

## 1 Introduction

This playbook demonstrates how to analyze Watson Assistant user logs and create a dashboard that can be used to share key performance metrics with stakeholders.

These four steps will be described:

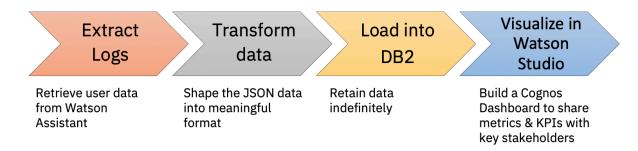


Figure 1: Steps described in this playbook

The first three steps (ETL) will be performed by a Jupyter notebook and will be run inside Watson Studio. Knowledge of python is not necessary in order to complete this playbook; however, any custom metrics or notebook adjustments may require it.

The fourth step is performed by Cognos Embedded Dashboard within Watson Studio for visualizations and reporting on key metrics; instructions and a tutorial will be provided in this playbook.

The notebook will output the shaped data in two forms:

- 1. Data written to Db2 tables for long term data retention and trending analysis purposes.
- 2. CSV files saved to Watson Studio's project assets. These files can be downloaded and analyzed in order to identify areas for improvement.

A <u>GitHub repository</u> is used for hosting the necessary tools and artifacts. It is managed by the author and is not a formally supported product by IBM. You can <u>create an issue in Github</u> if you want to report a bug, suggest an enhancement or submit other requests.

This playbook is NOT intended to act as a primary guide to <u>improve</u> the performance of your assistant. For that, check out the <u>Watson Assistant Continuous Improvement</u> best practices guidebook (shortcut link: <a href="http://ibm.biz/wa-improve-best">http://ibm.biz/wa-improve-best</a>) that includes links to other tools/notebooks.

## 2 Create resources

This analytics pattern requires the following IBM Cloud resources. Login to cloud.ibm.com/catalog and create the resources under the specific account and resource group. The following table describes each of the required services and recommended plan types. Lite plans may be used for testing purposes.

Cloud Service Name	Plan Type
Watson Studio	Standard
Cloud Object Storage	Lite
Db2	Flex
Cognos Embedded Dashboard	Pay as you go

# 3 Configure Db2 tables

The Jupyter notebook will save the transformed log data into Db2. Db2 is a relational database with tables that require a defined schema. For this notebook we will setup two tables:

- 1. WA FULL LOGS
- 2. WA\_LAST\_RUN\_LOG

The first table will contain all of the important fields from each of the user log events. These fields are transformed and defined throughout the notebook. The second table will contain information about the last Db2 load from the last notebook run.

In order to setup the tables, we will run two SQL commands inside Db2.

- First, navigate to the Db2 instance created previously in Section 2. Go to cloud.ibm.com/resources — Click on your provisioned Db2 instance — Click 'Manage' — 'Open Console'
- 2. On the top left hand side click the hamburger menu Run SQL click 'Blank' copy/paste in the entire SQL command from the <a href="https://dbe-sql.txt">db2-tables-sql.txt</a> file in the Github repository. Select 'Run All'
- Confirm that the two (2) tables were setup properly. Click the hamburger menu Explore —
  Tables Watson. The two tables should be listed on the left, the columns should be listed on
  the right.

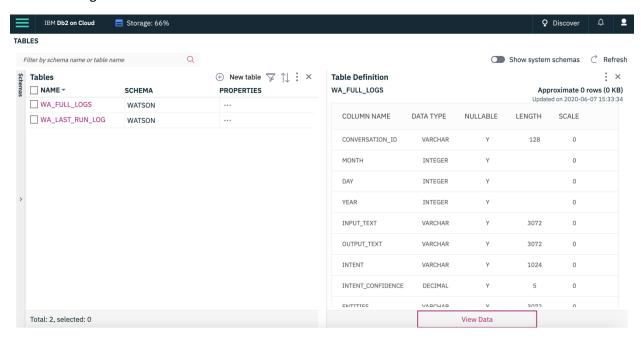


Figure 2 - Successful setup of Db2 tables

You've finished setting up your Db2 tables.

# 4 Create Project in Watson Studio

A Watson Studio project holds all of the data assets (CSV files) and services (Notebook, Jobs, Dashboard) for a given topic.

Open up Watson Studio (https://dataplatform.cloud.ibm.com/)

Create a project — Create an empty project – set a name e.g. [Client] Watson Project.

Choose the Cloud Object Storage as storage location for the project assets. Click Create.



Create a project token.

Under Project/Settings/Access tokens, click New token and give a name and select Editor role. Click Create.

Note: If you see any issues with creating the project, one can delete the cloud storage, try creating the storage as part of creation of the project itself.

# 5 Import the Jupyter notebook

Open up the project that you created in the prior section.

Click the 'Add to project' — Notebook — From URL — Under Notebook URL, copy/paste in the URL:

https://github.com/preethm/watson-assistant-metrics-notebook/blob/master/Watson%20Assistant%20Metrics%20Notebook.ipynb

Name the notebook — Optional description — Select runtime (Python 3.6 XS is fine) — Create

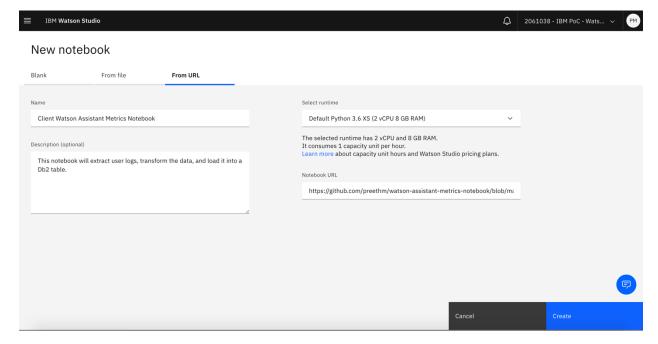


Figure 3: Example demonstrating New Notebook setup

The notebook has now been added into your Watson Studio project.

# 6 Configure the notebook

The notebook requires some initial configuration.

Throughout the notebook, you will see markup comments like this indicating an action is required. Carefully complete the step and run through the cells, paying attention to any errors in the output.

**Action Required:** Update the project\_id and project\_access\_token in order to access you https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/token.html

Run through the cells of the notebook. Sections 6.1, 6.2 & 6.3 below will provide guidance.

## 6.1 Configure & Establish DB2 connection

You will need to enter the values for your database in Section 1.1 of the Notebook. These values are listed in the Service Credentials for the Db2 instance.

Go to cloud.ibm.com/resources — open up the db2 instance created — Service credentials from left menu. If there are no service credentials listed, click 'New Credential +'.

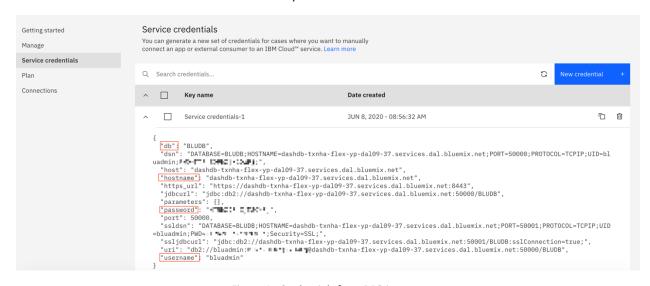


Figure 4 - Credentials from DB2 instance

Replace the X's below in the notebook with the values from your credentials above.

```
# Enter the values for you database connection. This can be found in DB2's Service Credentials from the tooling.

dsn_database = "XXXXXXXX"  # e.g. "MORTGAGE"

dsn_uid = "XXXXXXXX"  # e.g. "dash104434"

dsn_pwd = "XXXXXXXX"  # e.g. "7dBZ3jWt9xN6$00JiX!m"

dsn_hostname = "XXXXXXXXX"  # e.g. "Use the same IP as Web Console"

dsn_port = "50000"  # e.g. "50000"

dsn_protocol = "TCPIP"  # i.e. "TCPIP"

dsn_driver = "IBM DB2 ODBC DRIVER"  # Don't change
```

Figure 5 - Insert Db2 credentials into this notebook cell

## 6.2 Configure notebook with Watson Assistant credentials

An important step in the configuration is defining the credentials for the assistant. This needs to be defined in Notebook Section 1.2. Ensure that you do not provide both a workspace\_id and assistant\_id.

- For **chatbots** leveraging the V2 API and an Assistant layer (e.g. using Web Chat), you should define workspace\_id=None and provide an assistant\_id filter.
- For **Voice Interaction** solutions (V1 API), set a workspace\_id and comment out the assistant\_id definitions and filter.

```
# Extract logs from your assistant. Complete this information.
iam_apikey = 'XXXXXXXX'
url = "https://api.us-east.assistant.watson.cloud.ibm.com" # Set the URL to the region
assistant_id = 'XXXXXXXX'
workspace_id = None

# If not using assistant_id, comment out the 2nd line below.
log_filter="language::en,response_timestamp>=" + log_fetch_start \
+",request.context.system.assistant_id::" + assistant_id
```

Figure 6 - Update these values from your Watson Assistant credentials

The iam\_apikey, assistant\_id and workspace\_id can be obtained by going to the Assistant and click the on the right and select **setting** and then **API details** 

Assistant name:	Banking Service Demo	
Assistant ID:	11c1db44-4598-4d0a-90da-"-" - "-" "-" "-"	<u></u>
Assistant URL:	https://gateway.watsonplatform.net/assistant/api/v2/assist nts/11c1db44-4598-4d0a-	
Service credentials		
Credentials name:	Auto-generated service credentials	
API key:	7PkfAill   Adk-InowkuyhiCSOx SKPoTVySF-	<u>-</u>

The `url` can be found by going to <a href="https://cloud.ibm.com/resources">https://cloud.ibm.com/resources</a> — Select your Watson Assistant instance — copy/paste the URL (exclude everything to the right of ibm.com as shown below).

URL:

Run the cells up until Notebook Section 2.1.2 Coverage Metrics. That section is described below.

### 6.3 Define the dialog nodes for Coverage and Escalation

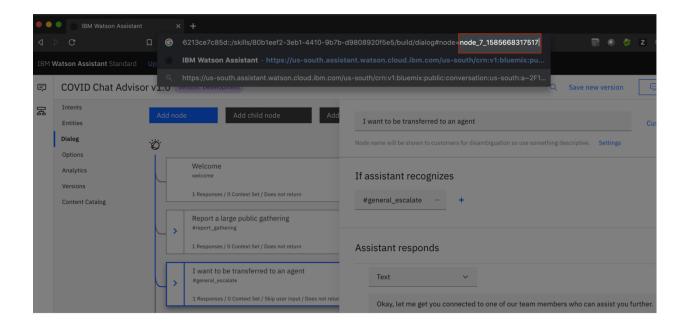
Coverage and escalation are two important metrics for a virtual assistant; they are calculated in the notebook.

- **Coverage** is the measurement of the portion of total user messages that your assistant is attempting to respond to. For most use cases, uncovered messages represent visits to anything\_else node ("Sorry I didn't understand").
- **Escalation** refers to the visits to a dialog node that represent a user's request to escalate. Typically, a user requests help, an intent is recognized (e.g. #General-Agent-Escalation), and a response is provided including reference to customer service.

The notebook requires definition of these two metrics based upon above information. You will need to define the specific dialog nodes that pertain to each of the two metrics. **Note** this is NOT the dialog node name found in the tooling. There are two options to identify this dialog\_node "ID" as shown below.

#### **Option 1: Watson Assistant Tooling URL**

Log into the Watson Assistant tooling — Open up the skill — Click on the Dialog Node (e.g. Anything Else) — copy the end of the URL for everything to the right of **#node**= (shown below).



#### **Option 2: Skill JSON**

You can also look inside the raw JSON of the Skill to find the dialog ID.

Log into the Watson Assistant tooling — Open up the skill — Go to Versions — Export the version that is relevant to the logs.

Open up the JSON into a text editor and search for the dialog node name in the raw data. This can often be done by doing a control-F search (e.g. looking for "anything\_else" or "escalate"). Once you find the dialog, look for the **dialog\_node** field in the raw data. This is the ID that you need to define in the notebook.

```
5669 "conditions": "anything_else",

5670 "dialog_node": "node_5_1584327171291",

5671 "previous_sibling": "node_2_1584325821879"

5672 }.
```

Figure 7 - Using Skill JSON to find dialog\_node (ID)

There may be multiple anything\_else nodes and the list in the notebook is a comma separated array containing all of the dialog\_node IDs.

Once the variables are defined, run the cell and check the output to make sure it looks accurate. If you notice either metric print out **0** visits, then you likely defined the wrong dialog node ID.

Perform a similar process for defining the Escalation Metric in Notebook Section 2.1.4.

Note: The two steps here in 6.3 are very important, ensure you take a few minutes to do this properly.

#### 6.4 Save a version of the notebook

Once you have run all of the cells of the notebook successfully, save as new version. Jobs will pick up and execute only the latest version as saved with 'save version' only.

Under File – save version.

# 7 Automatic notebook jobs

Jobs can be setup to automatically run the Jupyter notebook that you configured. This can be done hourly, daily, etc, and is up to the customer requirements. It is recommended to run it hourly if you want your stakeholders to see the most up-to-date information.

#### 7.1 Setup job schedule

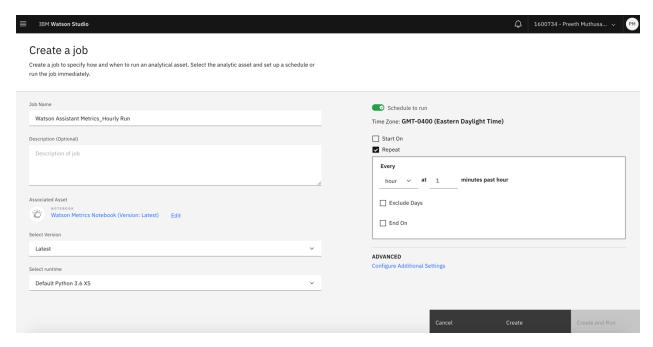
Go to Projects — Jobs tab — Click 'New Job'

Give a name for the Job.

Click on Select asset, select the notebook you have created.

Defaults are sufficient: Select Version, Latest — Select runtime – Default Python 3.6 XS

On right side, turn on schedule to run. Fill out the details appropriately. See below for example.

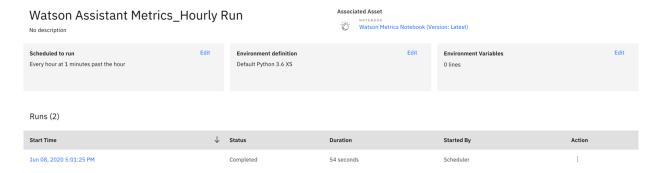


#### Click Create.

# 7.2 Validate job execution

Once the job is executed, you will see the job execution log under project – jobs tab.

The data will be saved in your Db2 tables and all the exported CSV files could be seen under Assets (and within the Cloud Object Storage bucket).



## 8 Create a Dashboard in Watson Studio

Create a dashboard inside Watson Studio using Cognos Embedded Dashboard service. This will enable you to visualize your performance metrics and share it with your key stakeholders.

## 8.1 Add a Db2 Connection to your project

You need to setup a connection between your Watson Studio project and Db2 database.

Add to Project — Connection — Db2 — Fill in details from Step 6.1 as shown below

Name the connection something useful, Connection to WA Metrics DB — Do not check the two boxes for SSL or Secure Gateway — Click Create

# ♦ New connection to Db2 Enter information for the selected data source

Connection overview	Connection Details		
Name	Database ①	SSL Certificate (optional)	Secure Gateway 🛈
Connection to WA Metrics DB	BLUDB		Use a secure gateway
Description	Hostname or IP Address 🛈		
IBM Db2 database	dashdb-txnha-flex-yp-dal09-37.services.dal.bluemix.ne		
	Port ①		<u> </u>
	50000	Username ①	
	Port is SSL-enabled (optional) ①  The port is configured to accept SSL connections	bluadmin	3
			_
		Password ①	

# 8.2 Add a dashboard to your project

Go to your Project — Add to Project — Dashboard — Blank

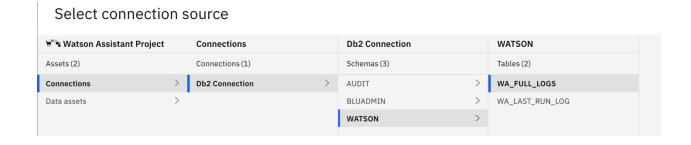
Name your dashboard (e.g. Watson Dashboard)

Associate a Cognos Dashboard Embedded service instance with your project

Reload — Select Cognos instance — Create

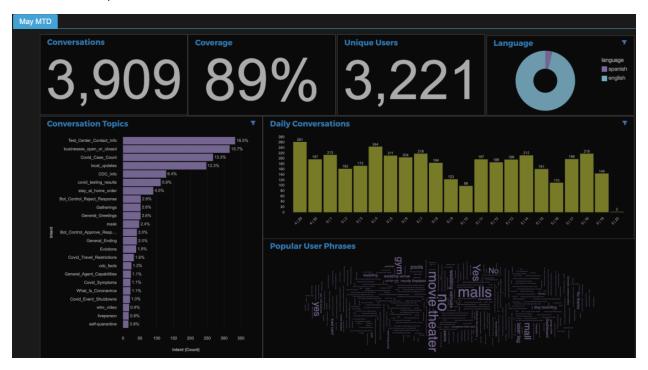
Follow the instructions in the tutorial to create dashboard visualizations.

Note: When you get to <u>Define the source of the Data</u>, add the Db2 Connection source shown below.



Be creative with your visualizations – sky's the limit!

Here is an example dashboard created for a customer with a chatbot solution:



## 8.3 Share the dashboard link

Once you are finished with your dashboard, you can publish and share the link. Select from the top right. Select

It is recommended that you shorten this URL by logging into <a href="https://snip.innovate.ibm.com/">https://snip.innovate.ibm.com/</a> URL. Follow the on screen notes to and send out the shortened, customized URL's for easy understanding. (e.g. https://ibm.biz/WatsonAcmeDashboard).