InterConnect
2017

March 19–23
MGM Grand &
Mandalay Bay
Las Vegas, NV

# Hands-on Lab
# Session 3305
# Spark for Cloudant Analytics

Holger Kache, IBM

# Table of Contents

# Lab instructions

The lab is running in an Ubuntu 14.4 VMware image. The user id for the image is:

| | |
|---|---|
| user: | **localuser** |
| password: | **passw0rd** |

The instructions are completely web based and require a working internet connection. For browsers we recommend either Firefox (version 47.0 is installed on the lab computer image) or Chrome (version 51.0 is installed on the lab computer image).

You can also opt to execute the lab on your private laptop. The instructions have no local dependencies and all resources are accessible online. There are no specific platform requirements either.

## Draw insights from Twitter data

The lab shows you how to analyze tweets and extract interesting insights from these tweets. You will learn how to find, filter, and sort tweets by tags and sentiment, or location and gender of the person tweeting.

The work is done in Jupyter notebooks running on Bluemix. A shared Spark cluster is running your computations and your results are immediately available in the notebooks. Data is extracted from the Twitter API and staged in a your own Cloudant database instance. The analysis results are written back to another Cloudant database and plotted in graphs inline with the notebook.

You will exercise two languages to run data analysis in Python and Scala and leverage frameworks including Spark Core, Spark SQL, Spark Streaming, Spark MLlib, and Spark GraphX.

Spark is an open source in-memory computing framework for distributed data processing and iterative analysis on massive volumes of data.
Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

## Create a Bluemix account

The services you are about to use in this tutorial are all hosted in the IBM Platform-as-a-Service called ***Bluemix***. If you already have a Bluemix account you can skip this section and proceed to the next section.

To sign up for Bluemix please navigate to

**http://bluemix.net/**

On the signup sheet you have to provide your contact details and create password. For country or region select "UNITED STATES" for the purpose of this lab. You can change your profile later but using the US location will provide you better performance while you are on-site at the convention.

With the successful sign up you get a 30-day trial account at no charge. No credit card information is required to create the account. It will expire automatically after 30 days. You have to provide payment information only if you want to convert to an unlimited account after the 30 days.

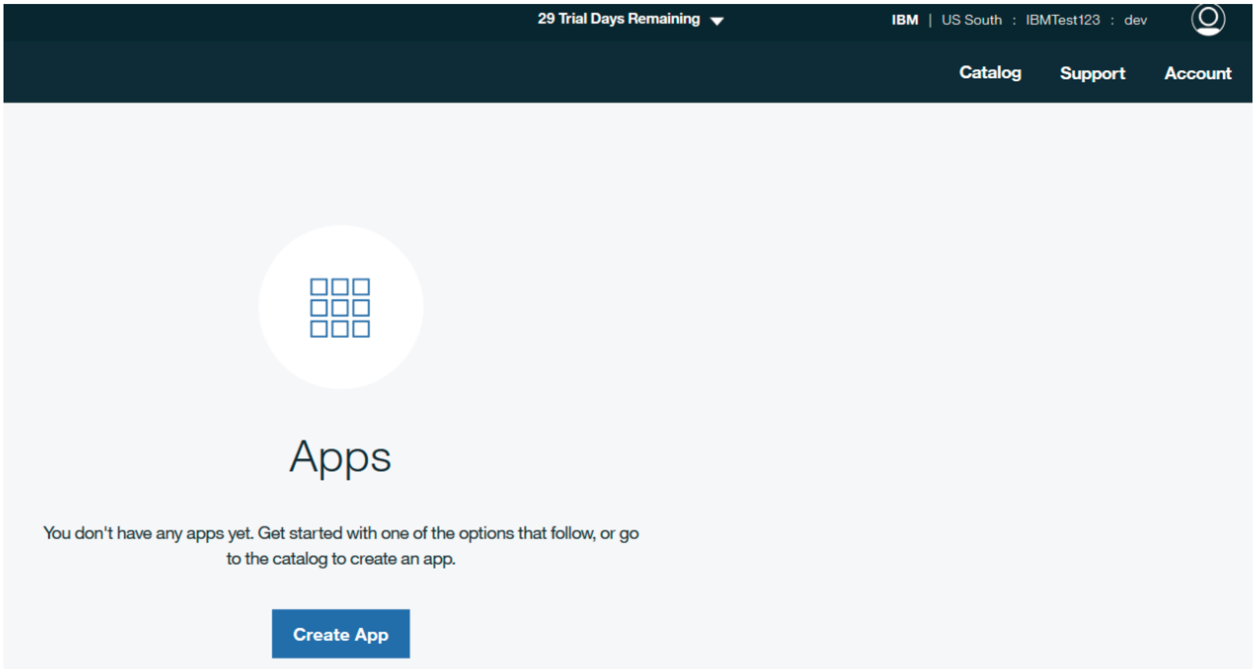
To activate the account, open your email inbox and find a note from "The Bluemix Team" with a subject "Action required: Confirm your Bluemix account". It contains a link to Confirm your account.

> *Note: The activation email should arrive within minutes but can theoretically take up to 24 hours. If you don't have the activation note in your inbox shortly, please ask us. We have a few active Bluemix accounts available we can share.*

Upon activation you should get a Success message with a link to log into your Bluemix dashboard. A three-page wizard opens with a few additional questions.

1 - Your location should already have been set to "US South". If not, please do so.
2 - Name your organization. Feel free to pick any name (including the suggested)
3 - Create a space. You can use the "dev" space for this lab.

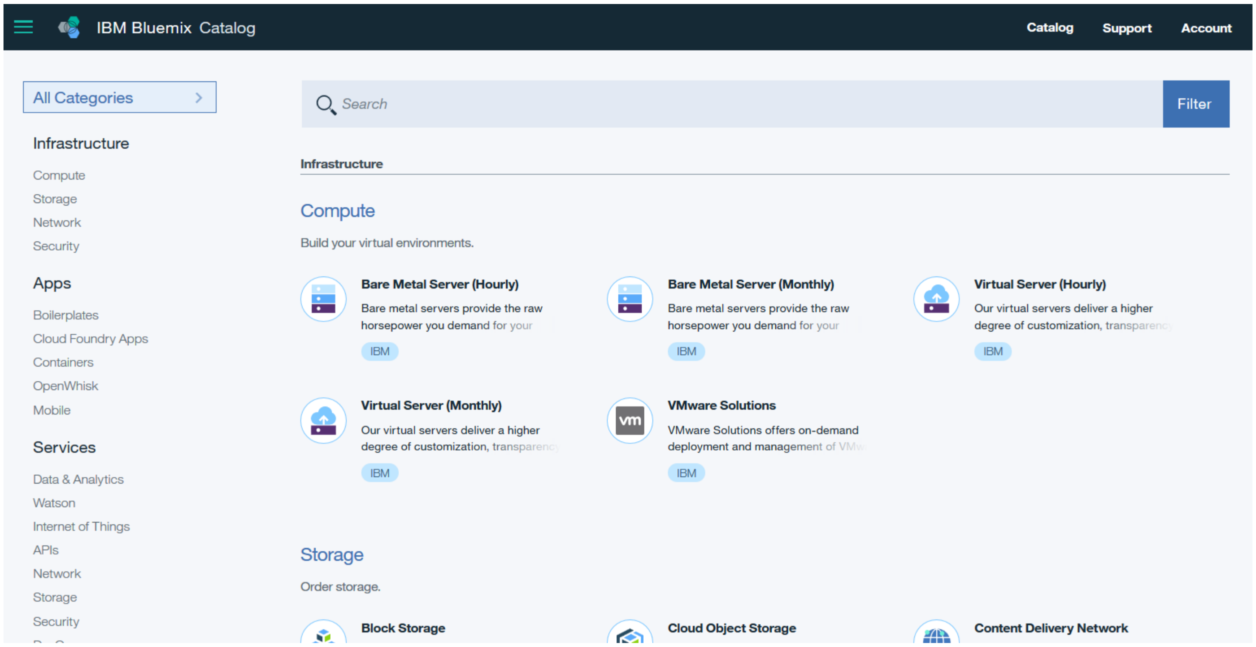
With that you are all set and should be in your Bluemix console looking like the one below.



You don't have any apps yet. Get started with one of the options that follow, or go to the catalog to create an app.

## Prepare the data set and data flow

From the Bluemix overview console you can pick the **Catalog** in the upper right hand menu. It offers a complete catalog of all services hosted on Bluemix.



Compute

Build your virtual environments.

**Bare Metal Server (Hourly)** Bare metal servers provide the raw horsepower you demand for your IBM

**Bare Metal Server (Monthly)** Bare metal servers provide the raw horsepower you demand for your IBM

**Virtual Server (Hourly)** Our virtual servers deliver a higher degree of customization, transparenc IBM

**Virtual Server (Monthly)** Our virtual servers deliver a higher degree of customization, transparenc IBM

**VMware Solutions** VMware Solutions offers on-demand deployment and management of VM IBM

Storage

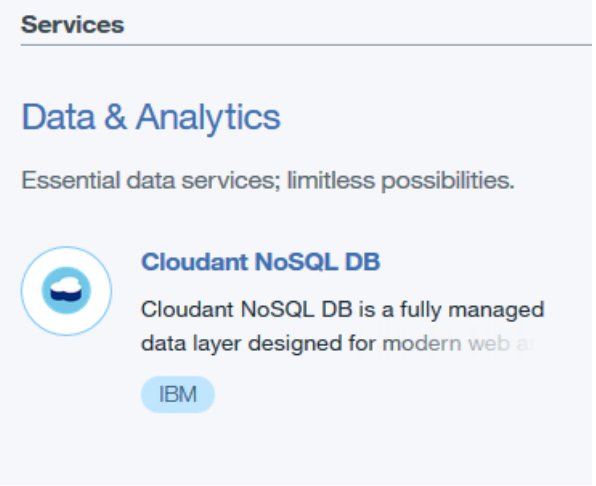
Order storage.

**Block Storage**

**Cloud Object Storage**

**Content Delivery Network**

Search for "Cloudant" with the search bar and pick the one in the "Data & Analytics" section called **Cloudant NoSQL DB**



Services

Data & Analytics

Essential data services; limitless possibilities.

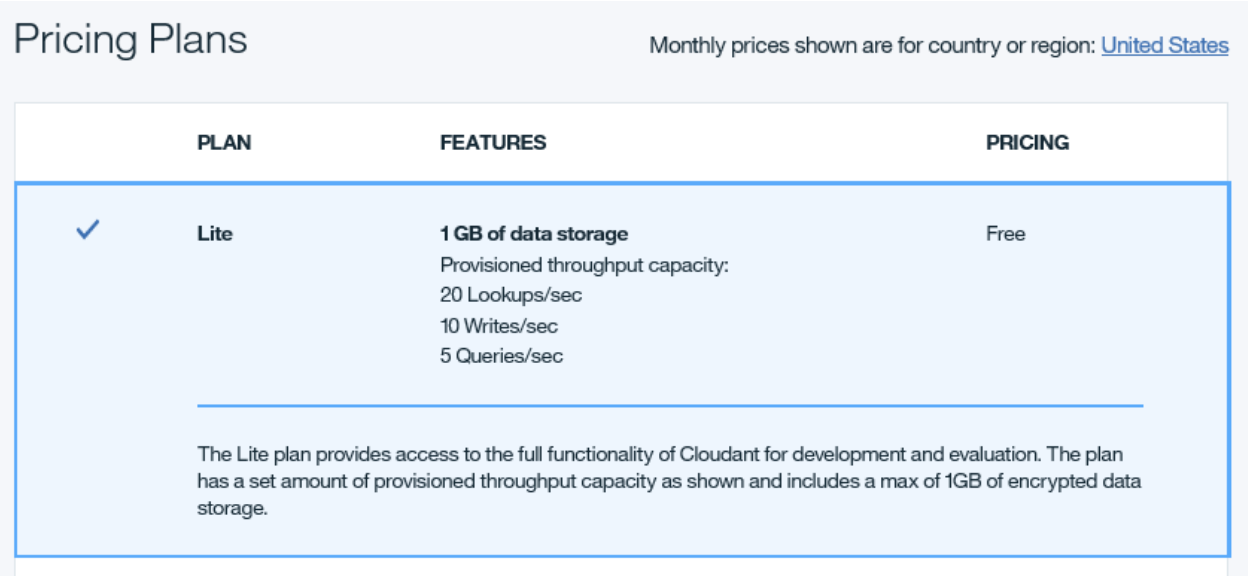
**Cloudant NoSQL DB**

Cloudant NoSQL DB is a fully managed data layer designed for modern web a

IBM

## 1. Provision a new Cloudant account

Click on the Cloudant NoSQL DB service icon to provision a new instance of a Cloudant account. From the list of Pricing Plans you want the Lite plan:



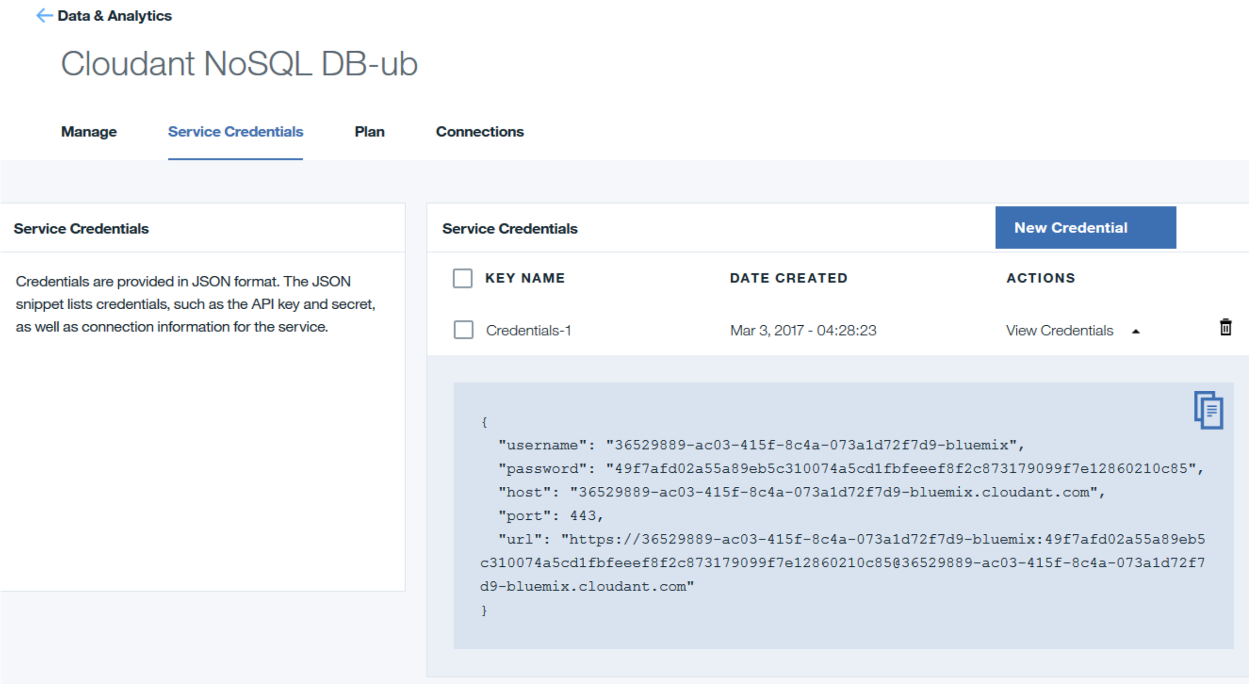
Pricing Plans

Monthly prices shown are for country or region: United States

| | PLAN | FEATURES | PRICING |
|---|---|---|---|
| ✓ | Lite | **1 GB of data storage**<br>Provisioned throughput capacity:<br>20 Lookups/sec<br>10 Writes/sec<br>5 Queries/sec | Free |

The Lite plan provides access to the full functionality of Cloudant for development and evaluation. The plan has a set amount of provisioned throughput capacity as shown and includes a max of 1GB of encrypted data storage.

Create the instance and make note of the user id and password that got created automatically with your Cloudant service instance. You will need those credentials later and can find them in the tab called "Service Credentials".



← Data & Analytics

Cloudant NoSQL DB-ub

Manage | Service Credentials | Plan | Connections

**Service Credentials**

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

**Service Credentials** New Credential

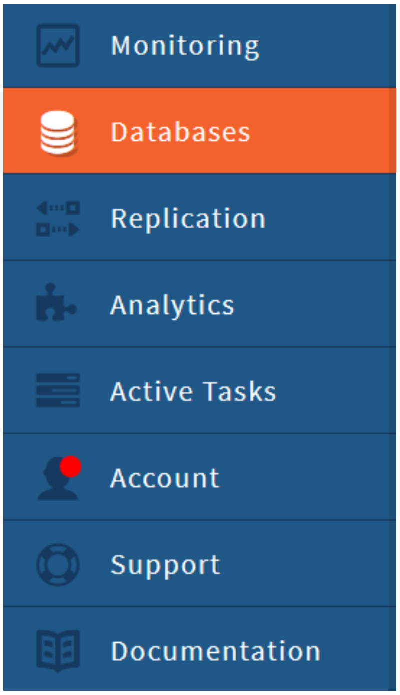
| □ | KEY NAME | DATE CREATED | ACTIONS |
|---|---|---|---|
| □ | Credentials-1 | Mar 3, 2017 - 04:28:23 | View Credentials ▲ |

```
{
  "username": "36529889-ac03-415f-8c4a-073a1d72f7d9-bluemix",
  "password": "49f7afd02a55a89eb5c310074a5cd1fbfeeef8f2c873179099f7e12860210c85",
  "host": "36529889-ac03-415f-8c4a-073a1d72f7d9-bluemix.cloudant.com",
  "port": 443,
  "url": "https://36529889-ac03-415f-8c4a-073a1d72f7d9-bluemix:49f7afd02a55a89eb5
c310074a5cd1fbfeeef8f2c873179099f7e12860210c85@36529889-ac03-415f-8c4a-073a1d72f7
d9-bluemix.cloudant.com"
}
```

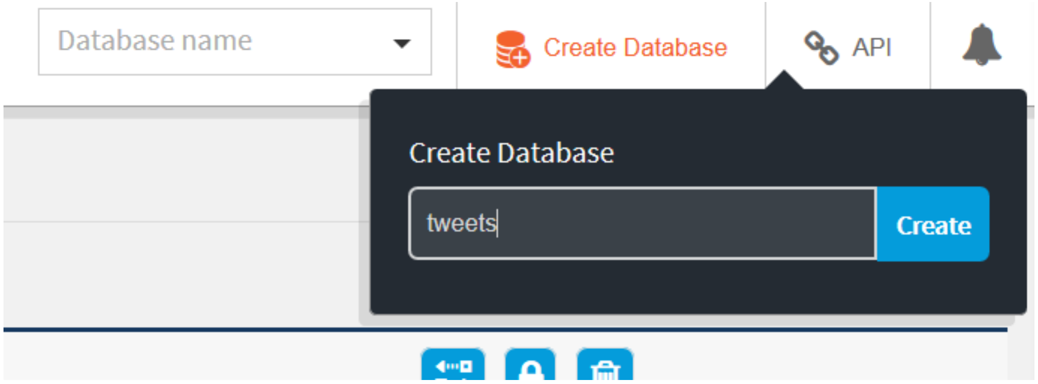
## 2. Create a Cloudant database

Navigate back over to the "Manage" tab and open the Cloudant dashboard with the "Launch" button. The experience will change to a completely different dashboard outside of Bluemix. Here the pages can be navigated on the left hand panel.

To create a database you want to use the Databases page.



Please create a database and note that database name again for later.



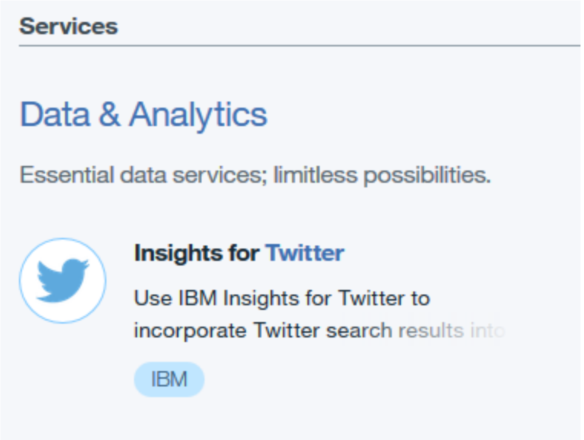
While you are here you should create a second database called “hashtags”. You don’t have to remember that database name but it will be used in one of the scripts later to save analytics results back from a Spark analysis.

## 3. Create an Insights for Twitter service instance

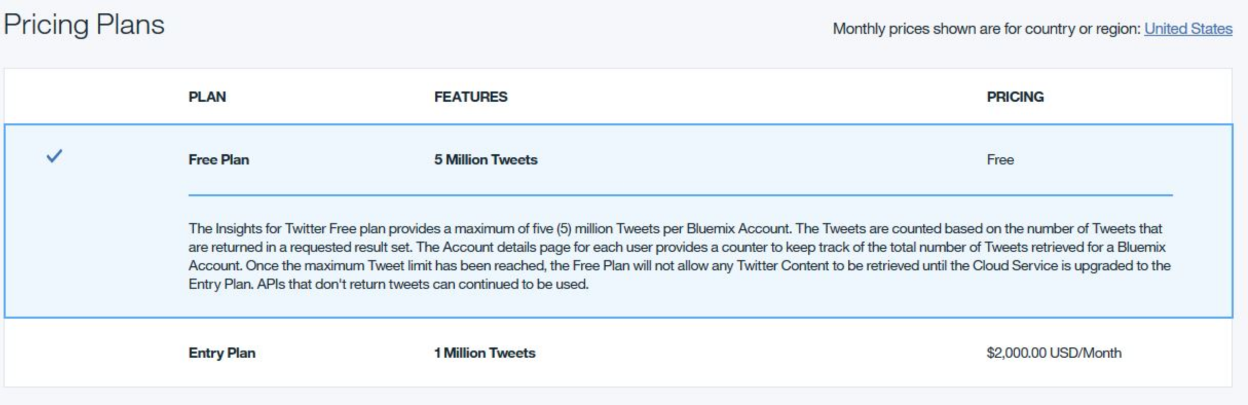
The next step in the analysis process is to harvest the data. You can use the [IBM Insights for Twitter API service]() at

**https://console.ng.bluemix.net/catalog/services/insights-for-twitter/**

to get Twitter data about the election. Back in the Bluemix catalog you can also navigate to the service when you search for the keyword "Twitter"


Services
Data & Analytics
Essential data services; limitless possibilities.
Insights for Twitter
Use IBM Insights for Twitter to incorporate Twitter search results into
IBM

Create a service instance for the “Insights for Twitter” service and accept the default values, including the Free Plan tier (as shown):

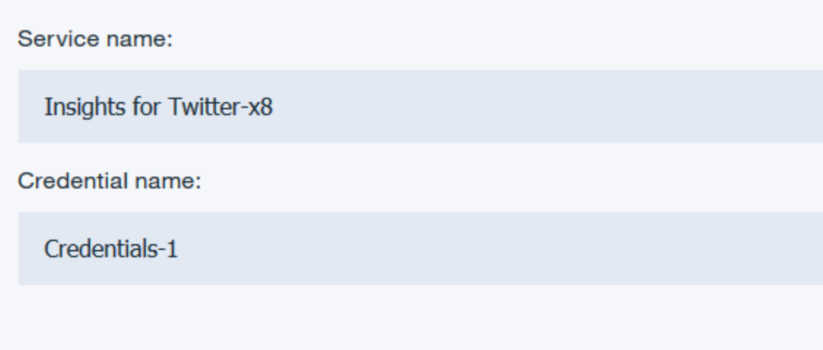


Pricing Plans

Monthly prices shown are for country or region: United States

| | PLAN | FEATURES | PRICING |
|---|---|---|---|
| ✓ | Free Plan | 5 Million Tweets | Free |
| | The Insights for Twitter Free plan provides a maximum of five (5) million Tweets per Bluemix Account. The Tweets are counted based on the number of Tweets that are returned in a requested result set. The Account details page for each user provides a counter to keep track of the total number of Tweets retrieved for a Bluemix Account. Once the maximum Tweet limit has been reached, the Free Plan will not allow any Twitter Content to be retrieved until the Cloud Service is upgraded to the Entry Plan. APIs that don't return tweets can continued to be used. | | |
| | Entry Plan | 1 Million Tweets | $2,000.00 USD/Month |

The service instance will have a generated name and a generated set of credentials.



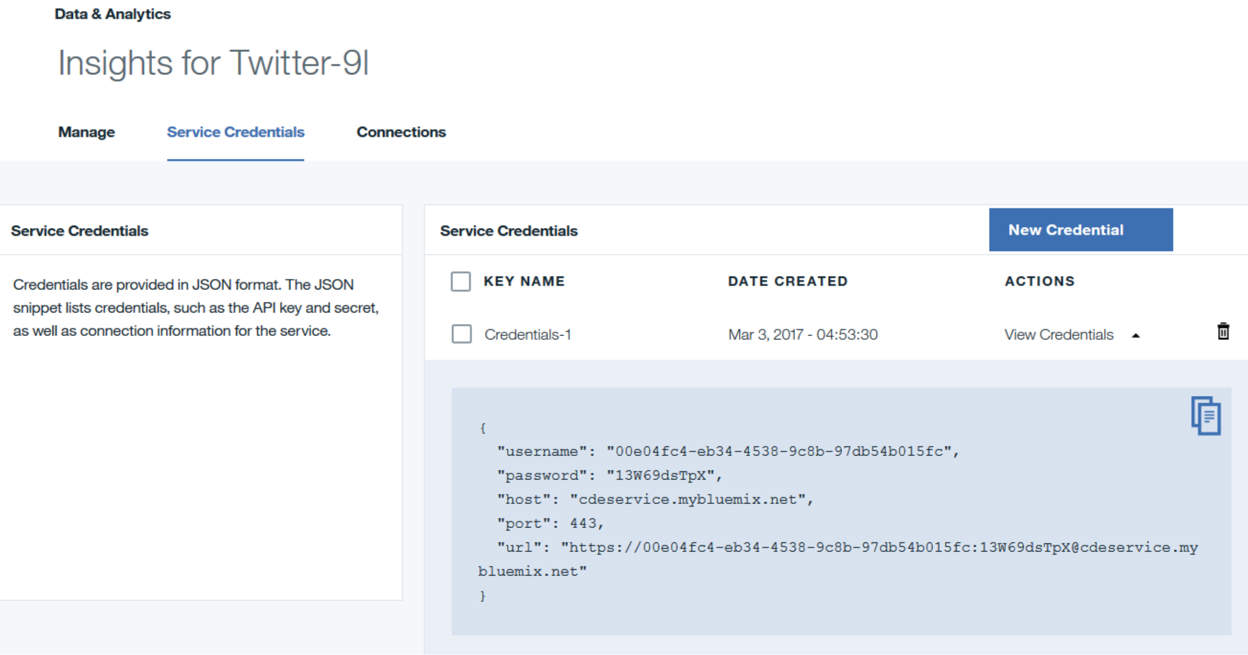
Service name:

Insights for Twitter-x8

Credential name:

Credentials-1

Make note of the Service Credentials in your newly deployed Insights for Twitter service instance. You will need these credentials later in the tutorial. Feel free to copy them to a notepad.



Insights for Twitter-9l

Service Credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Service Credentials

| □ | KEY NAME | DATE CREATED | ACTIONS |
|---|---|---|---|
| □ | Credentials-1 | Mar 3, 2017 - 04:53:30 | View Credentials ▲ |

```
{
  "username": "00e04fc4-eb34-4538-9c8b-97db54b015fc",
  "password": "13W69dsTpX",
  "host": "cdeservice.mybluemix.net",
  "port": 443,
  "url": "https://00e04fc4-eb34-4538-9c8b-97db54b015fc:13W69dsTpX@cdeservice.my
bluemix.net"
}
```

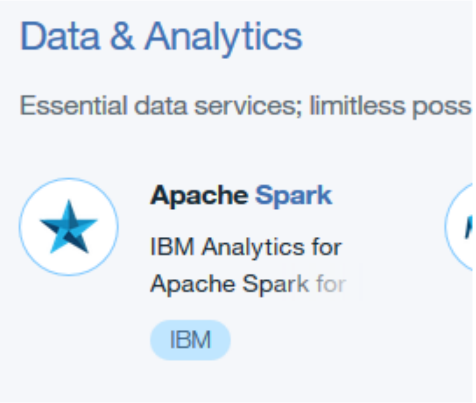
## 4. Create an Apache Spark service instance

In this step you will use the IBM Apache Spark service at

**https://console.ng.bluemix.net/catalog/services/apache-spark/**

in Bluemix to create a Jupyter notebook. The notebook is written in Python and allows you to script the calls to the Twitter service API created above. Results of the Twitter service API calls are persisted into your new Cloudant database.

Again, using the Bluemix catalog you can search for the keyword "Spark" in the catalog to find the same service:



Data & Analytics

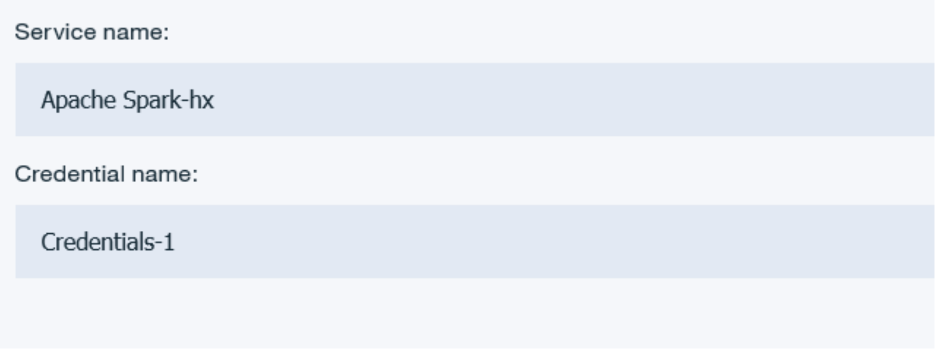
Essential data services; limitless poss

**Apache Spark**

IBM Analytics for Apache Spark for

IBM

Create a new service instance for the IBM Apache Spark service with the default settings.



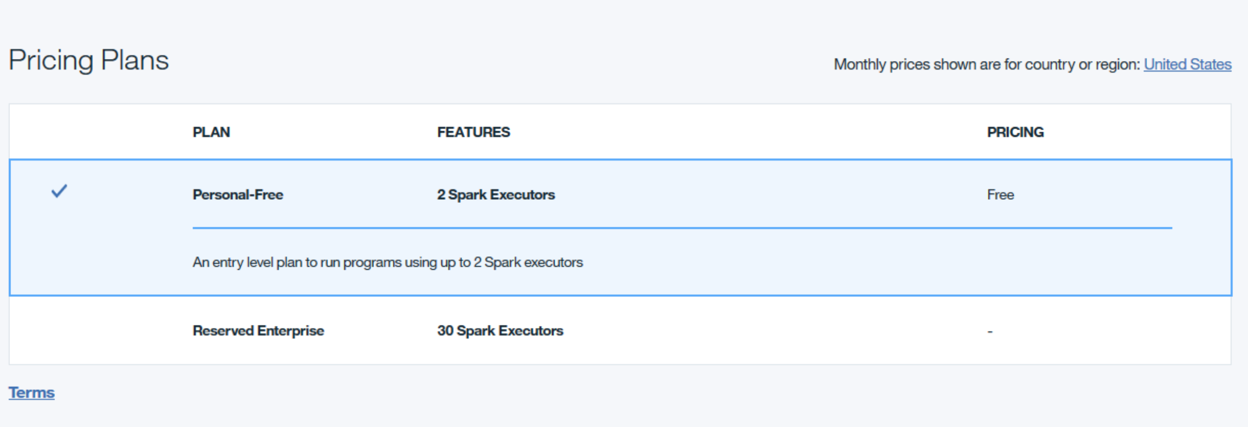
Service name:

Apache Spark-hx

Credential name:

Credentials-1

Select the Personal plan.



Pricing Plans

Monthly prices shown are for country or region: United States

| | PLAN | FEATURES | PRICING |
|---|---|---|---|
| ✓ | Personal-Free | 2 Spark Executors | Free |
| | An entry level plan to run programs using up to 2 Spark executors | | |
| | Reserved Enterprise | 30 Spark Executors | - |

Terms

*Note: The personal plan has a price plan of $0.70 for a 2 node execution engine per hour.* ***With the 30-day Bluemix trial account we don't incur any costs for this lab****.*

## 5. Data Science Experience

To exercise our analytics for tweets we will switch to the IBM Data Science platform called ***Data Science Experience (DSX)***.

To sign up for Data Science Experience please navigate to

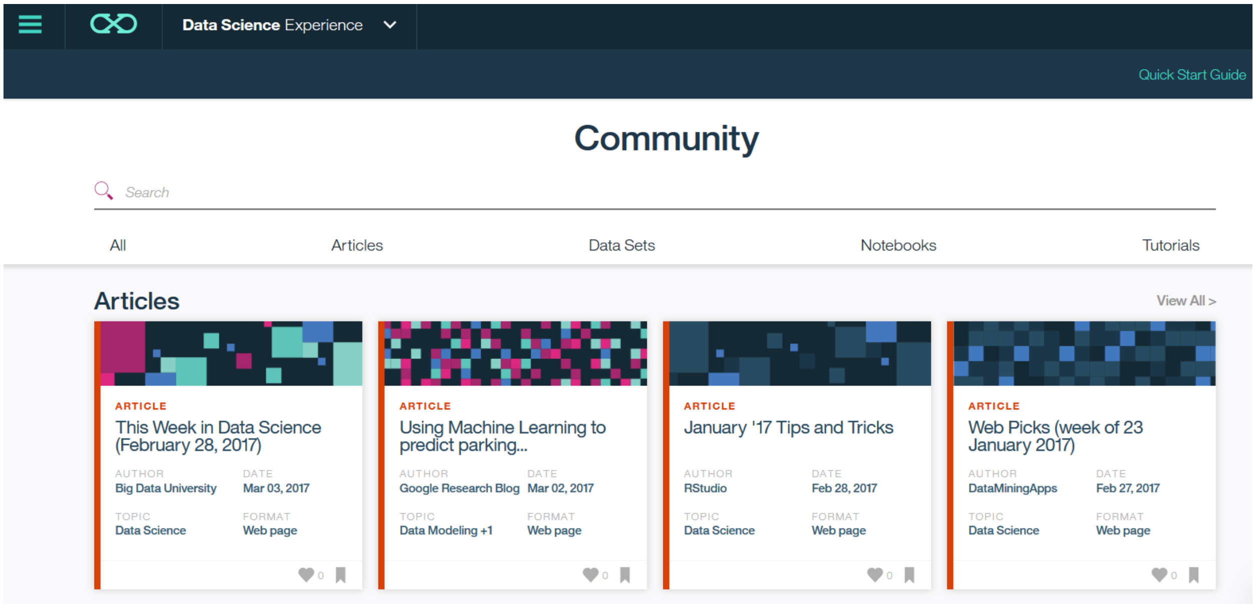
**http://datascience.ibm.com/**

Use the "*Sign Up*" button on the top right of the page.

> *Note*: If you already have a Bluemix account you can "*Sign In*" and advance to the next section immediately.

Please sign in with your IBM ID (the same one you used to register your Bluemix account with). The login process will ask you to validate your Bluemix organization and space and link your DSX account with your Bluemix account.

Finally, you should be able to login to Data Science Experience and see the following screen.



Community

Articles

ARTICLE
This Week in Data Science (February 28, 2017)
AUTHOR Big Data University
DATE Mar 03, 2017
TOPIC Data Science
FORMAT Web page

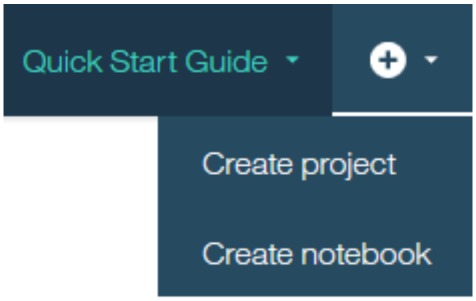
ARTICLE
Using Machine Learning to predict parking...
AUTHOR Google Research Blog
DATE Mar 02, 2017
TOPIC Data Modeling +1
FORMAT Web page

ARTICLE
January '17 Tips and Tricks
AUTHOR RStudio
DATE Feb 28, 2017
TOPIC Data Science
FORMAT Web page

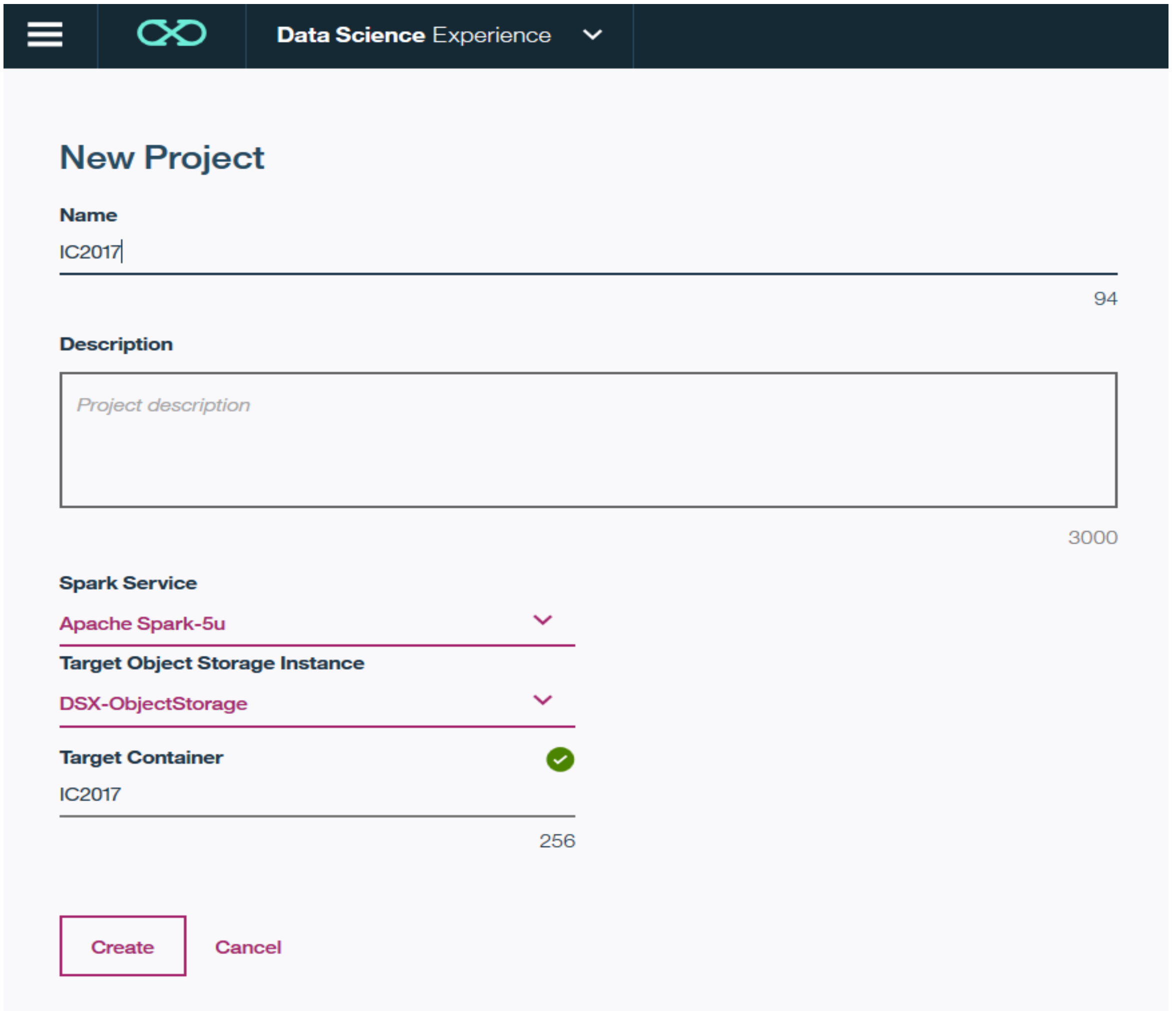
ARTICLE
Web Picks (week of 23 January 2017)
AUTHOR DataMiningApps
DATE Feb 27, 2017
TOPIC Data Science
FORMAT Web page

## Work with a Python notebook

In the Data Science Experience console you will find a + symbol to the right where you first want to create a project.



Provide a name, an optional description, and pick the Spark service instance from the list you created above. For Storage Type you can accept the “Object Storage” option and pick the one created for your Spark Service instance. The Target Container name will be pre-populated with the project name.



New Project

Name

IC2017

94

Description

Project description

3000

Spark Service

Apache Spark-5u

Target Object Storage Instance

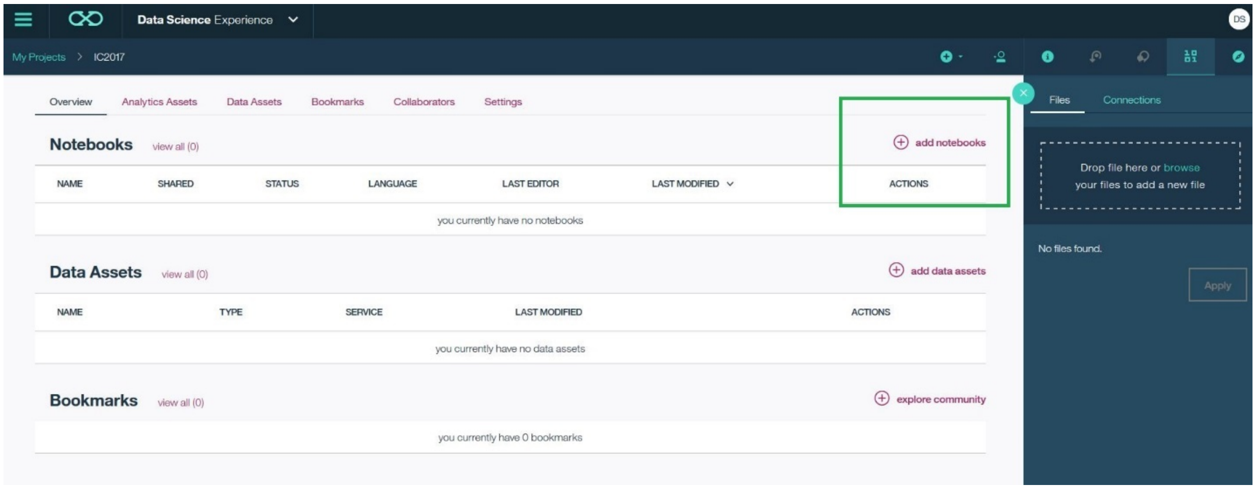
DSX-ObjectStorage

Target Container

IC2017

256

With the project in place you can now add notebooks with the + add notebooks action.



Notebooks view all (0)

⊕ add notebooks

| NAME | SHARED | STATUS | LANGUAGE | LAST EDITOR | LAST MODIFIED | ACTIONS |
|---|---|---|---|---|---|---|
| you currently have no notebooks | | | | | | |

Data Assets view all (0)

⊕ add data assets

| NAME | TYPE | SERVICE | LAST MODIFIED | ACTIONS |
|---|---|---|---|---|
| you currently have no data assets | | | | |

Bookmarks view all (0)

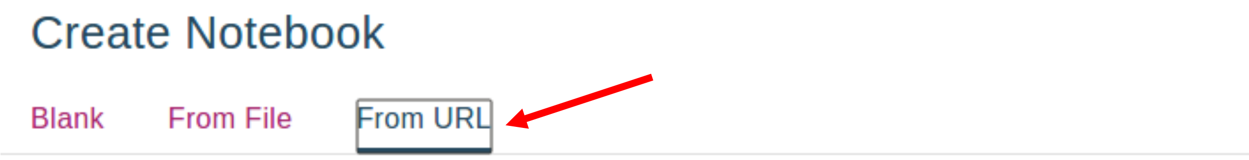
⊕ explore community

you currently have 0 bookmarks

Drop file here or browse your files to add a new file

No files found.

Create the new notebook **From URL** (not as a blank notebook).



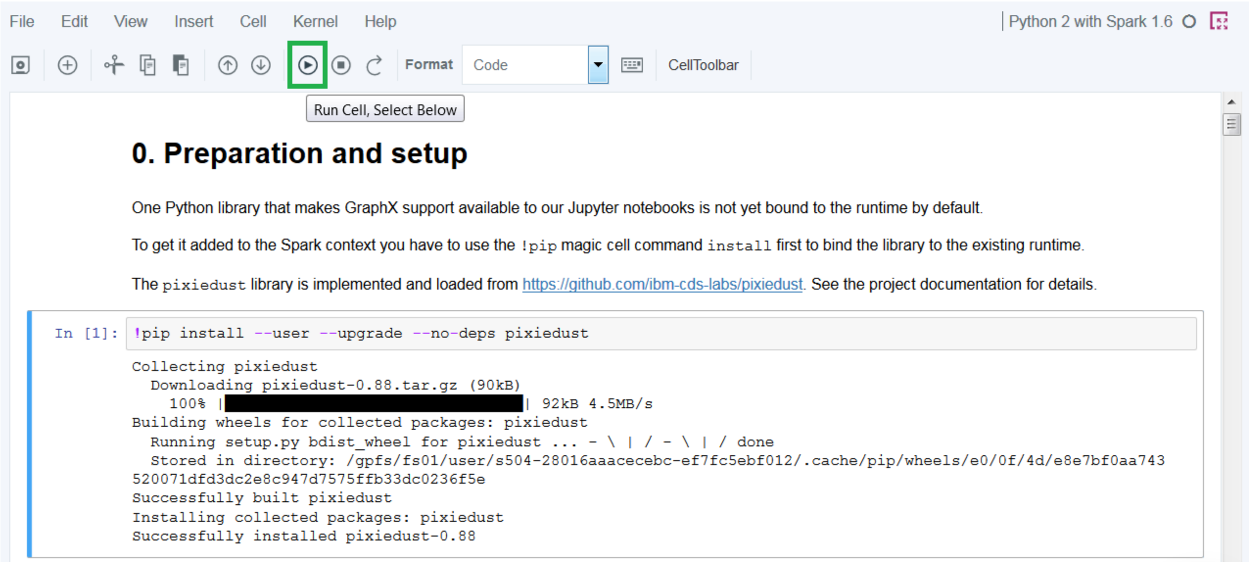
Provide a name, an optional description, and select the following Notebook URL:

> [xx]()
>
> *Note: It will have automatically picked the Spark Service instance we associated with the project earlier.*

The notebook you just loaded contains the actual instructions with Spark as engine and Cloudant as data store. All code in the notebook is written in Python 2 syntax and requires a running Python 2.7 kernel. By default, the kernel should have been started and your notebook be connected to it.

Code is structured into cells where you want to execute cells sequentially, starting at the top. While a cell executes, you should see a [*] next to the cell that indicates the running status. When the cell completed, you see a number like [1]. That number increments with every cell execution. Nothing stops you from running a cell "out of order" instead of sequentially. Just make sure to meet all the conditions for the execution of a cell. We prepared the notebook so that all conditions are met when you run it top-to-bottom.
A successful cell execution will almost always dump some output right below the cell.

For example:



**0. Preparation and setup**

One Python library that makes GraphX support available to our Jupyter notebooks is not yet bound to the runtime by default.

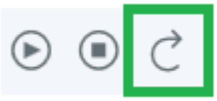
To get it added to the Spark context you have to use the `!pip` magic cell command `install` first to bind the library to the existing runtime.

The `pixiedust` library is implemented and loaded from [https://github.com/ibm-cds-labs/pixiedust](https://github.com/ibm-cds-labs/pixiedust). See the project documentation for details.

```
In [1]: !pip install --user --upgrade --no-deps pixiedust

        Collecting pixiedust
          Downloading pixiedust-0.88.tar.gz (90kB)
            100% |████████████████████████████████| 92kB 4.5MB/s
        Building wheels for collected packages: pixiedust
          Running setup.py bdist_wheel for pixiedust ... - \ | / - \ | / done
          Stored in directory: /gpfs/fs01/user/s504-28016aaacecebc-ef7fc5ebf012/.cache/pip/wheels/e0/0f/4d/e8e7bf0aa743
        520071dfd3dc2e8c947d7575ffb33dc0236f5e
        Successfully built pixiedust
        Installing collected packages: pixiedust
        Successfully installed pixiedust-0.88
```

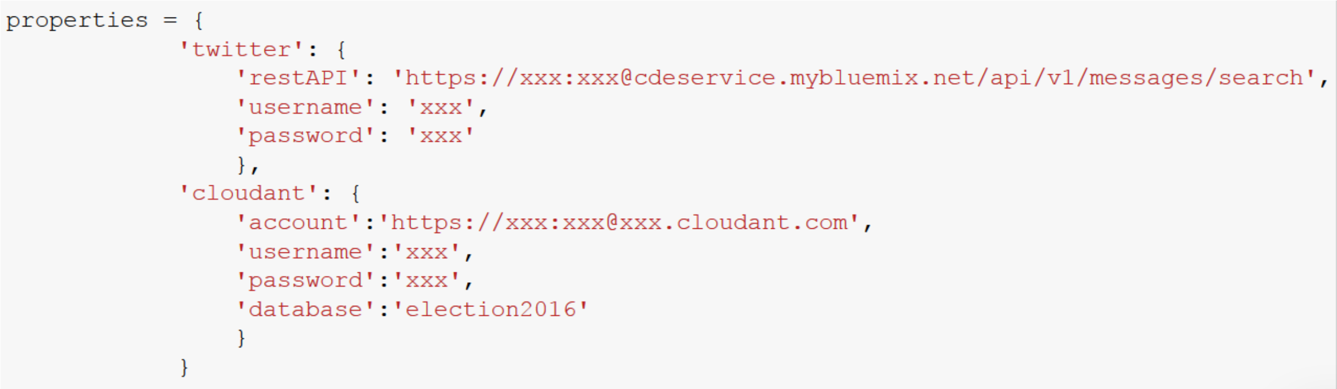
Should there be no output, you probably have a problem with your Python kernel and should perform a kernel restart. Use the menu options or action buttons atop your notebook to interrupt or restart the kernel.



A kernel restart clears your entire session context and you will have to re-run every instruction required up to the point where you want to resume your work. The comments provided in the notebook should make it somewhat obvious what every cell requires for execution. If you are unclear how to proceed after a kernel restart just ask us.

Please go ahead and follow the instructions given in the Python notebook at this point. Pay special attention to the properties map we use in one of the cells to get the connection details for both, your Twitter service and the Cloudant database above. The smallest mistake in any one of these values will cause the script to fail.

Here is a detailed configuration provided to match your own values against:

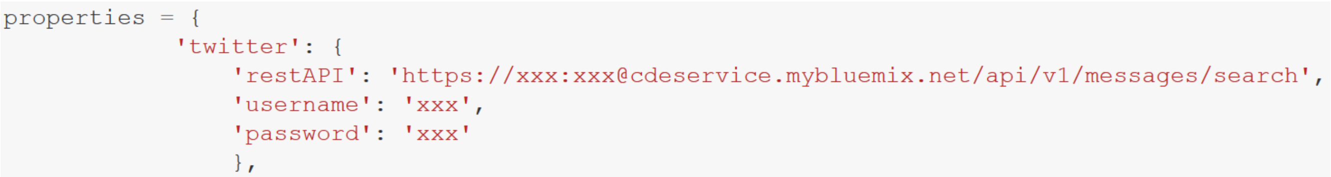


```
properties = {
            'twitter': {
                'restAPI': 'https://xxx:xxx@cdeservice.mybluemix.net/api/v1/messages/search',
                'username': 'xxx',
                'password': 'xxx'
                },
            'cloudant': {
                'account':'https://xxx:xxx@xxx.cloudant.com',
                'username':'xxx',
                'password':'xxx',
                'database':'election2016'
                }
            }
```

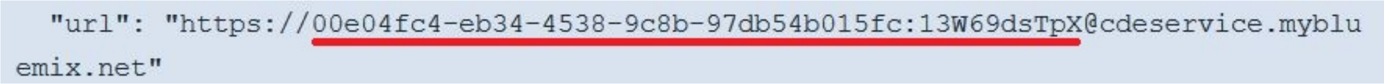
Please compare the service credentials you have noted before. All ‘xxx’ parts of the properties above have to be replaced with your credentials. If you experience any trouble freel free to ask us.

Have a look at the following example, how your Twitter service credentials should look like.

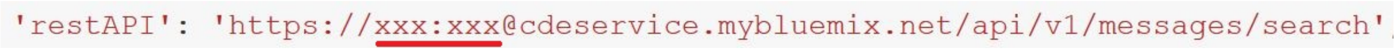
### 1. Enter Twitter credentials



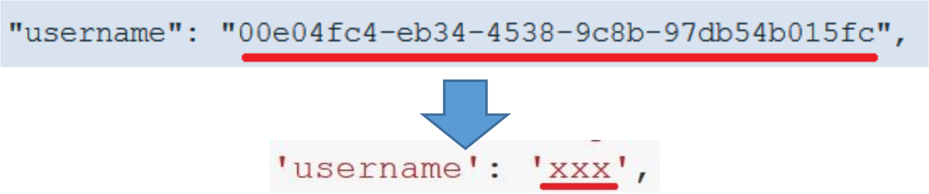
```
properties = {
            'twitter': {
                'restAPI': 'https://xxx:xxx@cdeservice.mybluemix.net/api/v1/messages/search',
                'username': 'xxx',
                'password': 'xxx'
                },
```



```
   "url": "https://00e04fc4-eb34-4538-9c8b-97db54b015fc:13W69dsTpX@cdeservice.myblu
emix.net"
```

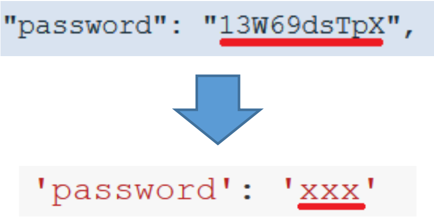


```
'restAPI': 'https://xxx:xxx@cdeservice.mybluemix.net/api/v1/messages/search',
```



```
"username": "00e04fc4-eb34-4538-9c8b-97db54b015fc",
```

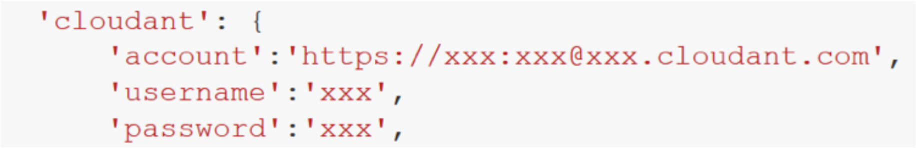
```
'username': 'xxx',
```



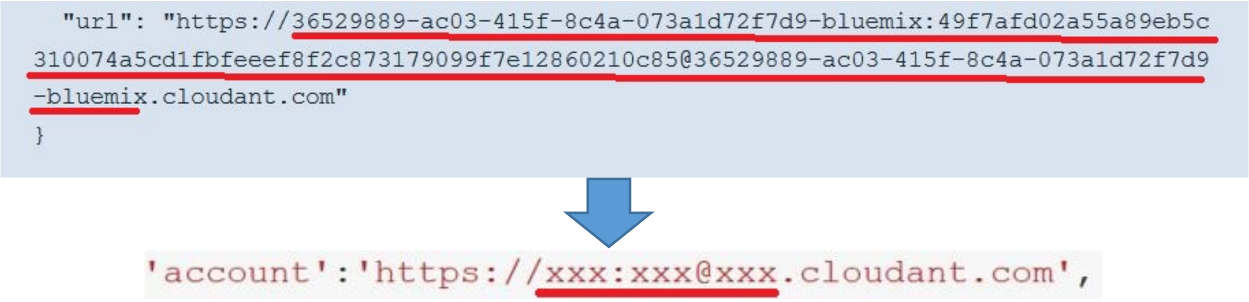
```
"password": "13W69dsTpX",
```

```
'password': 'xxx'
```

## 2. Enter Cloudant credentials

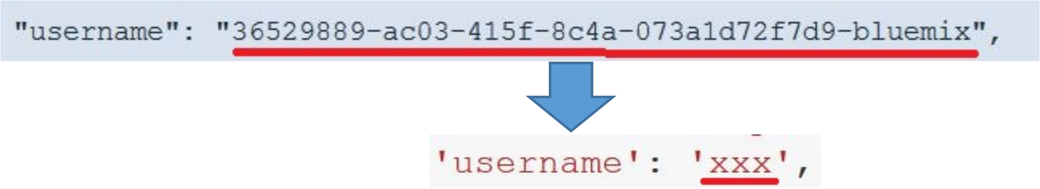


```
 'cloudant': {
     'account':'https://xxx:xxx@xxx.cloudant.com',
     'username':'xxx',
     'password':'xxx',
```



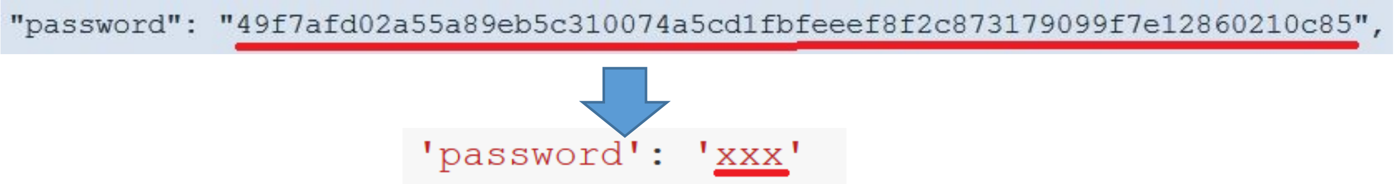
```
   "url": "https://36529889-ac03-415f-8c4a-073a1d72f7d9-bluemix:49f7afd02a55a89eb5c
 310074a5cd1fbfeeef8f2c873179099f7e12860210c85@36529889-ac03-415f-8c4a-073a1d72f7d9
 -bluemix.cloudant.com"
 }
```

```
'account':'https://xxx:xxx@xxx.cloudant.com',
```



```
"username": "36529889-ac03-415f-8c4a-073a1d72f7d9-bluemix",
```

```
'username': 'xxx',
```



```
"password": "49f7afd02a55a89eb5c310074a5cd1fbfeeef8f2c873179099f7e12860210c85",
```

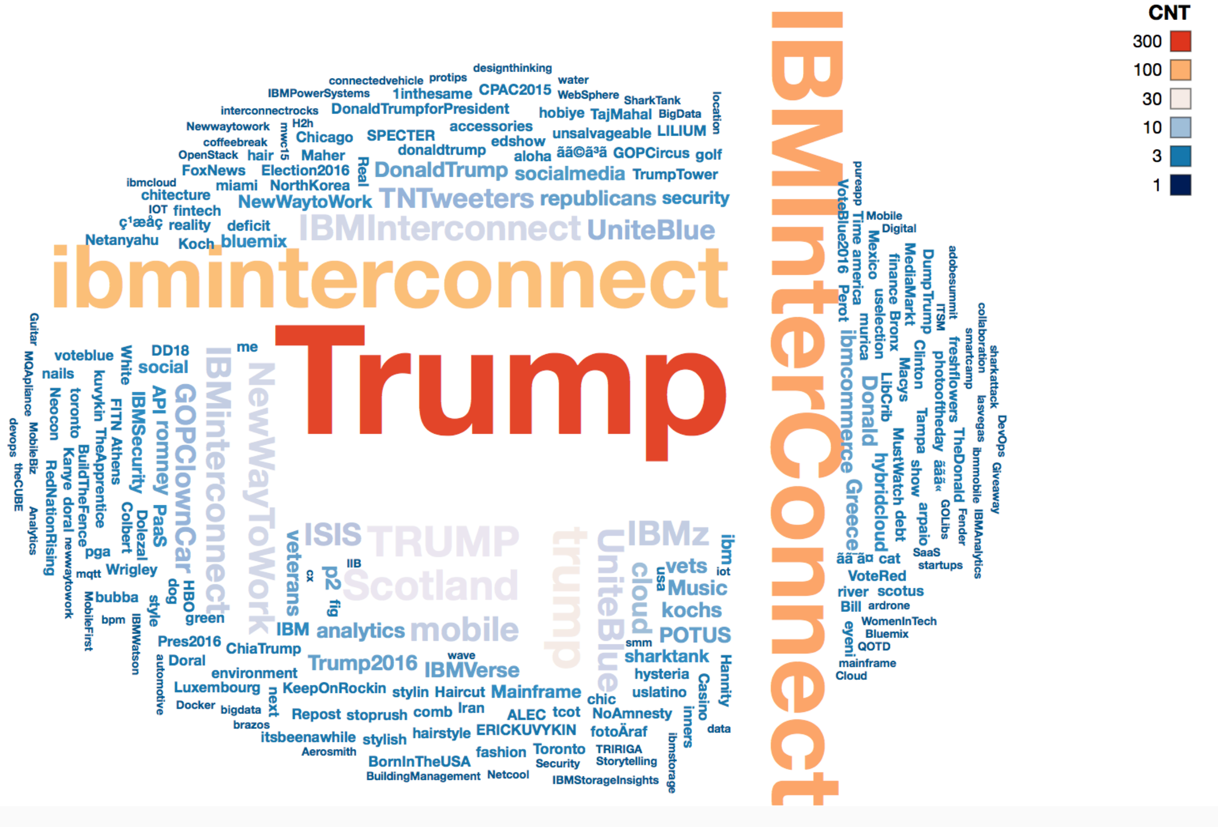
```
'password': 'xxx'
```

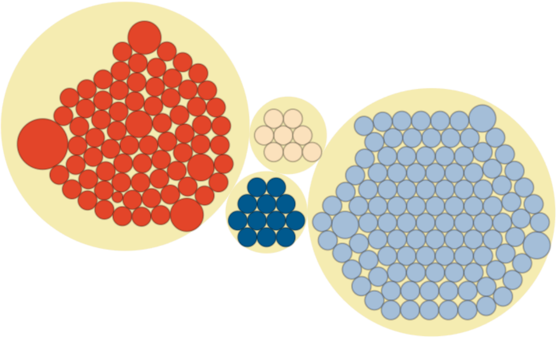
At this point you have everything prepared to work through the notebook itself. Execute all the cells in order and inspect the results. There are query outputs, visualizations, and other statistics you will gather with the execution of the Python notebook.

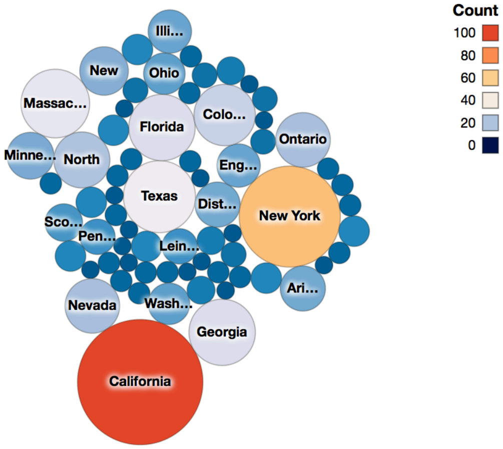
To validate the success of your executions, you can compare with the RESULT output we provided at

**https://github.com/cloudant-labs/spark-cloudant/blob/master/tutorials/InterConnect2017_Python_RESULT.ipynb**

Your cell execution output should look very similar to the output in that HTML page and even have some nice graphs that don't render in the output above.







This concludes your hands-on lab “Spark for Cloudant Analytics”. Thank you!