

A **new world** awaits.

IBM World of Watson 2016

#ibmwow

IBM

Mandalay Bay, Las Vegas  
October 24-27, 2016

## Lab Center – Hands-on Lab

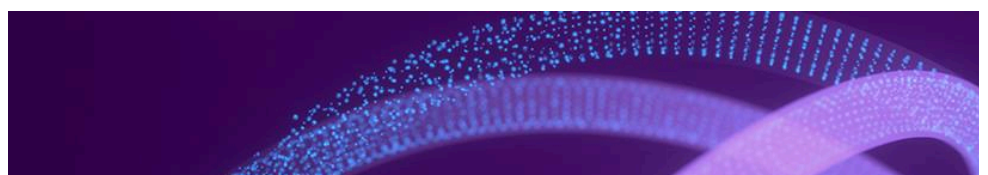
### Session 1865

### Spark for Cloudbant Analytics

Holger Kache, IBM, [kache@de.ibm.com](mailto:kache@de.ibm.com)

Mayya Sharipova, IBM, [mayyas@ca.ibm.com](mailto:mayyas@ca.ibm.com)

Tony Sun, IBM, [tonysun@us.ibm.com](mailto:tonysun@us.ibm.com)



## Table of Contents

<b>Lab instructions</b> .....	3
Draw insights from Twitter data about the upcoming US election.....	3
Create a Bluemix account .....	4
Prepare the data set and data flow.....	6
1. Provision a new Cloudant account .....	7
2. Create a Cloudant database.....	8
3. Create an Insights for Twitter service instance.....	9
4. Create an Apache Spark service instance .....	11
Work with a Python notebook .....	13
Work with a Scala notebook .....	15
<b>We Value Your Feedback!</b> .....	17



## Lab instructions

The instructions in this lab are completely web based and require a working internet connection. For browser we recommend Firefox (version 45.2 is installed on the lab computer).

You can go ahead and 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 about the upcoming US election

The lab shows you how to analyze tweets about the upcoming US election and extract interesting insights from these tweets. You will learn how to find, filter, and sort tweets by location, sentiment, party affiliation, and candidate.

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 Cloudfant database instance. The analysis results are written back to another Cloudfant 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.



## 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 with security questions to recover it. 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.



# Welcome, Antje.

Overview All Items (0)

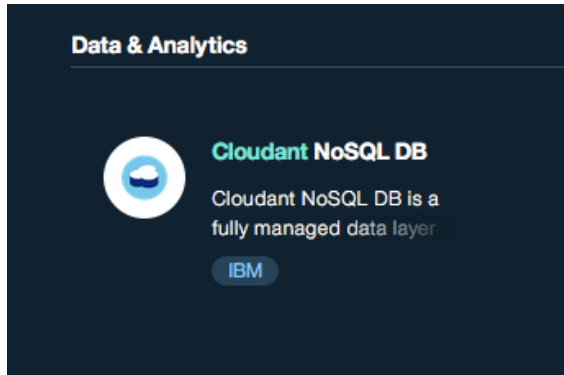
View Your Items by Category:

 <p><b>Compute</b> Use Cloud Foundry applications, containers, OpenWhisk, and virtual servers.</p>	 <p><b>Network</b> Develop on robust networking services.</p>	 <p><b>Storage</b> Store data with secure, durable, and highly scalable services.</p>	 <p><b>Data &amp; Analytics</b> Use these services to get data, build data-driven apps, and analyze data.</p>	 <p><b>Watson</b> Build cognitive apps to enhance, scale, and accelerate expertise.</p>
 <p><b>Internet of Things</b> Take advantage of data and analytics from your connected devices and sensors.</p>	 <p><b>APIs</b> Use APIs shared in your org or from API Connect.</p>	 <p><b>DevOps</b> Accelerate app delivery with automation, tracking, and monitoring tools.</p>	 <p><b>Security</b> Secure your cloud platform and applications.</p>	 <p><b>Application Services</b> Add features to supplement your Compute resources.</p>
 <p><b>Mobile</b> Quickly get started with your next mobile app.</p>	 <p><b>Integrate</b> Extend existing investments and infrastructure.</p>			



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



# 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:

The screenshot shows the 'Pricing Plans' page for Cloudant. At the top right, it says 'Monthly prices shown are for country or region: [United States](#)'. Below this is a table with three columns: 'Plan', 'Features', and 'Pricing'. The 'Lite' plan is selected, indicated by a checkmark in the 'Plan' column. The 'Features' column for the Lite plan lists: '1 GB of data storage', 'Provisioned throughput capacity: 20 Lookups/sec, 10 Writes/sec, 5 Queries/sec'. The 'Pricing' column for the Lite plan is 'Free'. Below the table, there is a note: '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".

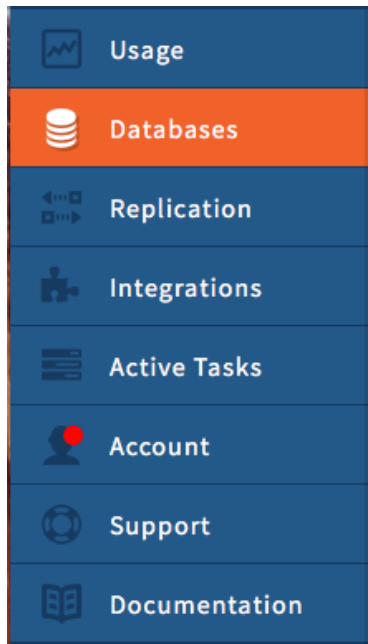
The screenshot shows the Cloudant console interface. At the top, it says 'Data & Analytics' and 'Cloudant NoSQL DB-hx' with a status indicator 'Service available'. Below this are tabs for 'Manage', 'Service Credentials', 'Plan', and 'Connections'. The 'Service Credentials' tab is active. On the left, there is a 'Service Credentials' panel with a description: '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.' On the right, there is a table of 'Service Credentials' with columns 'KEY NAME', 'DATE CREATED', and 'ACTIONS'. One credential is listed: 'Credentials-1' created on 'Sep 14, 2016 - 03:48:30'. Below the table, a JSON snippet is displayed, containing the username, password, host, port, and url for the service.



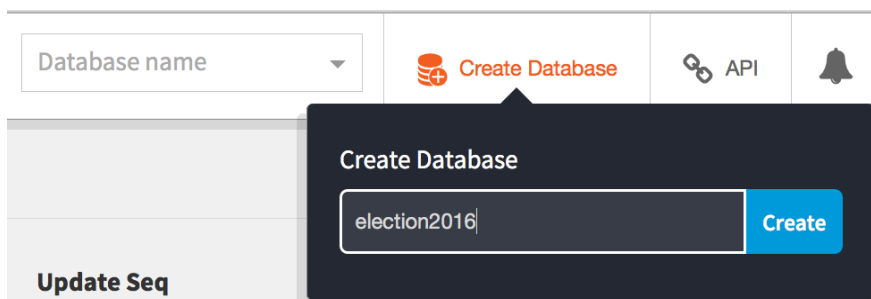
## 2. Create a Cloudant database

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.





### 3. Create an Insights for Twitter service instance

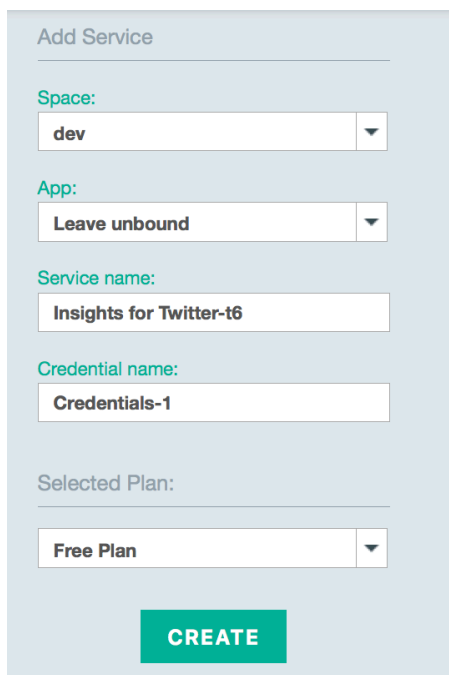
The next step in the analysis process is to harvest the data. You can use the [IBM Insights for Twitter API service](#) at

<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"



Create a service instance and accept the default values, including the Free Plan tier:

A screenshot of the "Add Service" form in the Bluemix console. The form is light blue and contains several fields: "Space:" with a dropdown menu set to "dev"; "App:" with a dropdown menu set to "Leave unbound"; "Service name:" with a text input field containing "Insights for Twitter-t6"; "Credential name:" with a text input field containing "Credentials-1"; and "Selected Plan:" with a dropdown menu set to "Free Plan". At the bottom of the form is a green "CREATE" button.

Make note of the Service Credential in your newly deployed Insights for Twitter service instance.

The screenshot shows the Cloud Foundry console interface for a service instance named "Insights for Twitter-t6". The left sidebar contains navigation options: "Back to Dashboard...", "Insights for Twitter-t6", "Manage", "Service Credentials" (selected), "Service Access Authorization", and "APPS USING SERVICE". The main content area is titled "Service Credentials" and includes an "ADD CREDENTIALS" button. Below this, a table lists a single credential named "Credentials-1" with a "DELETE" button. The "SERVICE CREDENTIALS" section displays a JSON snippet:

```
{
  "credentials": {
    "username": "5e2d04c1-cbcd-4159-901d-229e5a8d7054",
    "password": "JoOpVsIDMq",
    "host": "cdeservice.mybluemix.net",
    "port": 443,
    "url": "https://5e2d04c1-cbcd-4159-901d-229e5a8d7054:JoOpVsIDMq@cdeservice.mybluemix.net"
  }
}
```



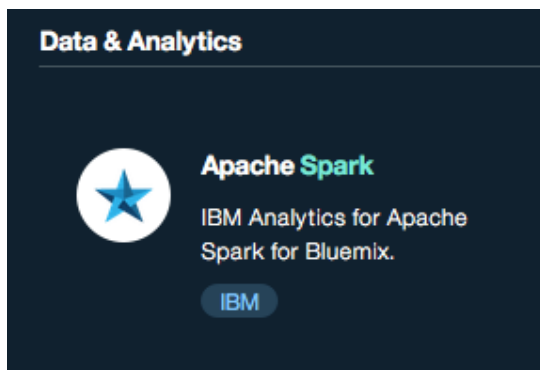
## 4. Create an Apache Spark service instance

In this step you will use the [IBM Apache Spark service](#) at

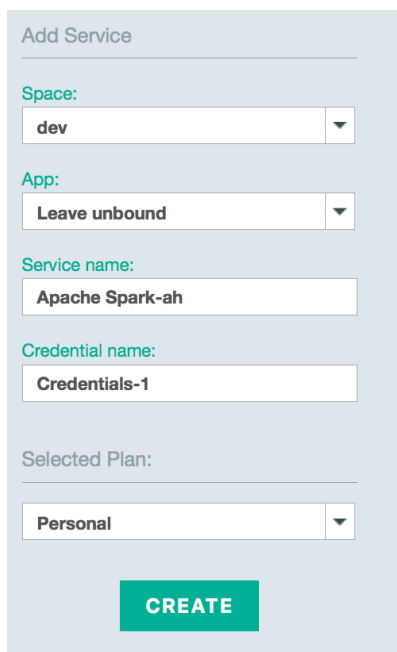
<https://console.ng.bluemix.net/catalog/services/apache-spark/>

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:



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

A screenshot of the "Add Service" form in the Bluemix console. The form is light blue and contains several fields: "Space:" with a dropdown menu showing "dev"; "App:" with a dropdown menu showing "Leave unbound"; "Service name:" with a text input field containing "Apache Spark-ah"; "Credential name:" with a text input field containing "Credentials-1"; and "Selected Plan:" with a dropdown menu showing "Personal". At the bottom of the form, there is a green button with the text "CREATE" in white.

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. Should you use your personal Bluemix account and don't want to see any charge on your credit card for this lab, please ask us to give you a trial account instead.



## Work with a Python notebook

In the new service instance you are presented with the console. Create a new Notebook and select the "Create Notebook" From URL option. Provide a name, an optional description, and select the following Notebook URL:

<https://raw.githubusercontent.com/cloudant-labs/spark-cloudant/master/tutorials/wowPython.ipynb>

Note: In case you get an error like "The service is not responding, try again later" use the back button and try the load again.

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:

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

Collecting pixiedust
  Downloading pixiedust-0.32.tar.gz (44kB)
    100% |#####| 51kB 1.2MB/s
Installing collected packages: pixiedust
  Found existing installation: pixiedust 0.28
    Uninstalling pixiedust-0.28:
      Successfully uninstalled pixiedust-0.28
  Running setup.py install for pixiedust ... - \ | done
  Successfully installed pixiedust-0.32
```

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.

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

[https://github.com/cloudant-labs/spark-cloudant/master/tutorials/wowPython\\_RESULT.ipynb](https://github.com/cloudant-labs/spark-cloudant/master/tutorials/wowPython_RESULT.ipynb)

Your cell execution output should look very similar to the output in that HTML page.



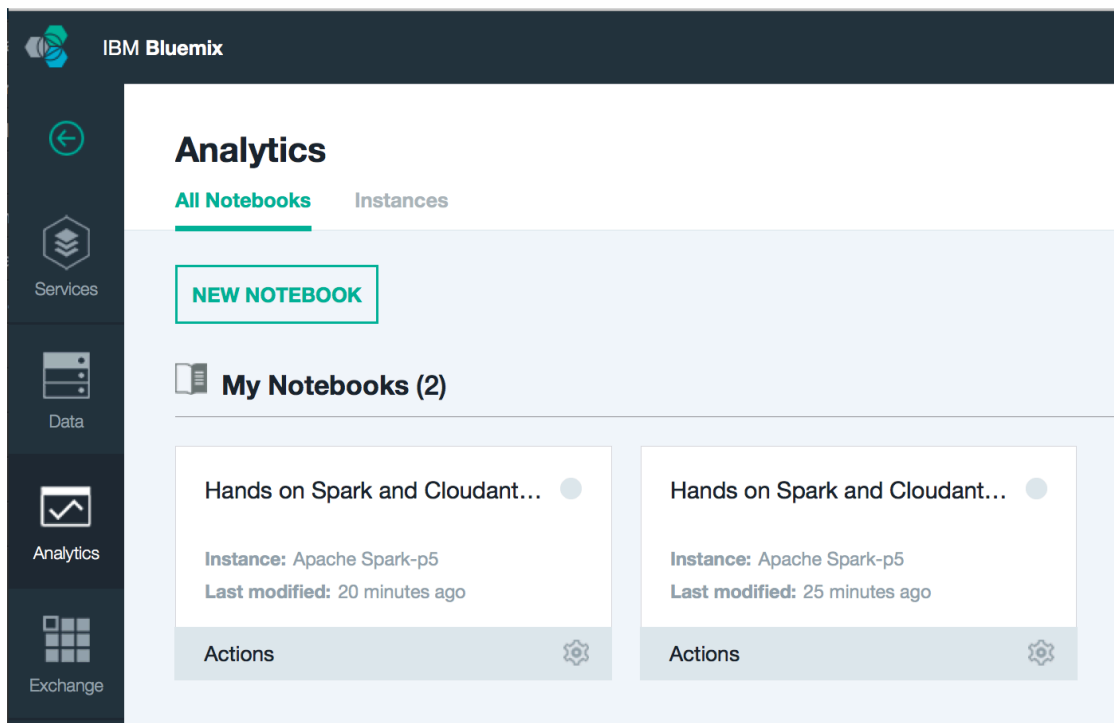
## Work with a Scala notebook

With the Python notebook above you have already seen a lot of Spark core and graphing technology at play. There is, however, one powerful framework in Spark to support analysis of data streams with Spark Streaming. Spark Streaming is currently supported only in Scala notebooks. We prepared a second notebook written in Scala you can use to exercise Spark Streaming technology.

Note: Don't worry if you are not familiar with Scala as language. The notebook can again be executed in a point-and-click fashion and does not require any actual coding.

To load the Scala notebook, open a new browser tab and navigate to the Spark service instance you already created in Bluemix earlier.

Note: If you need a quick way to navigate back to the Spark service, use the Bluemix menu on the left hand side and click the "Analytics" tab. There you will immediately see all your notebooks.



The screenshot shows the IBM Bluemix Analytics dashboard. On the left is a dark sidebar with navigation icons for Services, Data, Analytics, and Exchange. The main content area is titled "Analytics" and has two tabs: "All Notebooks" (selected) and "Instances". A prominent "NEW NOTEBOOK" button is at the top. Below it, a section titled "My Notebooks (2)" displays two notebook cards. Each card shows the title "Hands on Spark and Cloudant...", the instance "Instance: Apache Spark-p5", and the last modified time ("Last modified: 20 minutes ago" and "Last modified: 25 minutes ago" respectively). Each card also has an "Actions" button with a gear icon.

Create a new Notebook and select the "Create Notebook" From URL option again. Provide a name, an optional description, and select the following Notebook URL:

<https://raw.githubusercontent.com/cloudant-labs/spark-cloudant/master/tutorials/wowScala.ipynb>



Please go ahead and follow the instructions given in the Scala notebook at this point.

At one point in the notebook you are asked to start the Spark Streaming Context (ssc). It will subscribe to changes in your Cloudbant database and process any changes in a window of 120 sec. During these 120 sec, you should initiate changes in your Cloudbant database.

A simple way to initiate changes to the Cloudbant database is to simply add new tweets. Go back to your Python notebook and re-execute the cell that calls the `TwitterToCloudbant()` class.

Note: If you closed the tab with the Python notebook accidentally, you will have to re-open it and re-execute the cells that load libraries and create context. Don't forget to adjust your connection details either.

Here you can optionally change the query you want to use for the Twitter API and/or the count of tweets to process.

```
query = "#election2016"  
count = 300
```

```
TtC = TwitterToCloudbant()  
TtC.count = count  
  
TtC.query_twitter(properties, None, query, 0)
```

Now you simulate a stream of events (loading new documents with tweets) in your Cloudbant database and see how the Spark Streaming receiver can process these events.

After the 120 sec you can always restart your Streaming Context or increase the duration.

```
ssc.start()  
Thread.sleep(120000L)  
ssc.stop(true)
```

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

[https://github.com/cloudbant-labs/spark-cloudbant/master/tutorials/wowScala\\_RESULT.ipynb](https://github.com/cloudbant-labs/spark-cloudbant/master/tutorials/wowScala_RESULT.ipynb)

Your cell execution output should look very similar to the output in that HTML page.





## We Value Your Feedback!

- Don't forget to submit your World of Watson session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.
- Access the World of Watson Conference Connect tool to quickly submit your surveys from your smartphone, laptop or conference kiosk.

