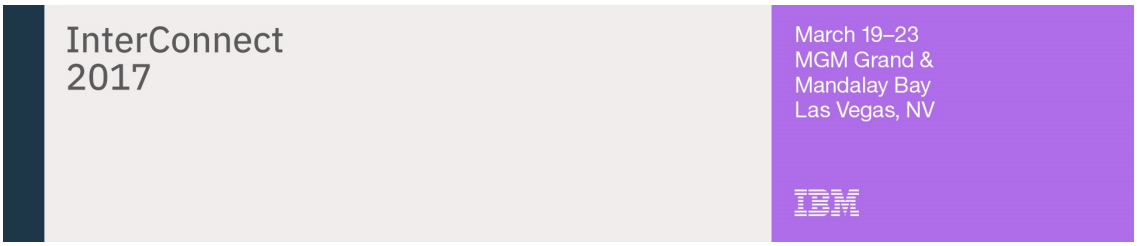


Hands-on Lab
Session 2450
Working with OpenWhisk in IBM Bluemix

Budi Darmawan, Bluemix Enablement

Pam Geiger, Bluemix Enablement

Social Review Microservice implemented with OpenWhisk actions

Introduction

This project is built to demonstrate how to build a Microservices application implemented as OpenWhisk actions to access an IBM Cloudant NoSQL database. It provides basic operations of saving and querying reviews from a database as part of a Social Review function. Additionally, the review text is analyzed using the Watson Tone Analyzer service to determine whether to flag negative reviews for further inspection. The project covers the following technical areas:

- Leverage OpenWhisk actions and REST API gateway to build a Serverless Microservices application.
- Use IBM Cloudant NodeJS library to access a Cloudant database.
- Use OpenWhisk triggers to fire an OpenWhisk action on Cloudant database changes.
- Use Watson Tone Analyzer REST API to analyze text.

About Watson Tone Analyzer

Tone Analyzer uses linguistic analysis to detect three types of tones in written text: emotions, social tendencies, and writing style. The Tone Analyzer service can be to understand the emotional context of conversations and communications. You can then use this insight to respond in an appropriate manner. Tone Analyzer is used when you need to understand text more deeply than simple positive and negative sentiment. You can use the service to deeply understand written text. Common uses for the Tone Analyzer service include:

- Analyze communications and improve message effectiveness
- Optimize the tones in your communication to increase the impact on your audience
- Understand and route call center customers based on their tone
- For bloggers and journalists, fine-tuning of writing to reflect a specific personality or style

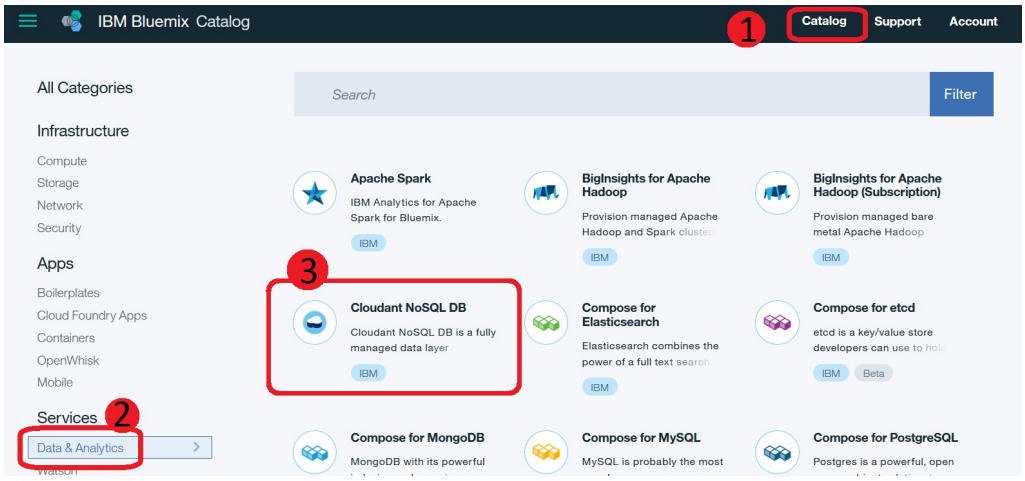
It takes any text as input and outputs a hierarchical representation of the analysis of the terms in the input message in JSON format.

In this lab exercise, you will provision a cloudant database to store the reviews for a social review application. The OpenWhisk cloudant package is used to configure Cloudant to fire a trigger when a change is made to the social review database. You create OpenWhisk actions to invoke the Watson Tone Analyzer to analyze the text when a new review is stored in the database, or when a change is made to an existing review. The analysis results are stored into the socialreview database.

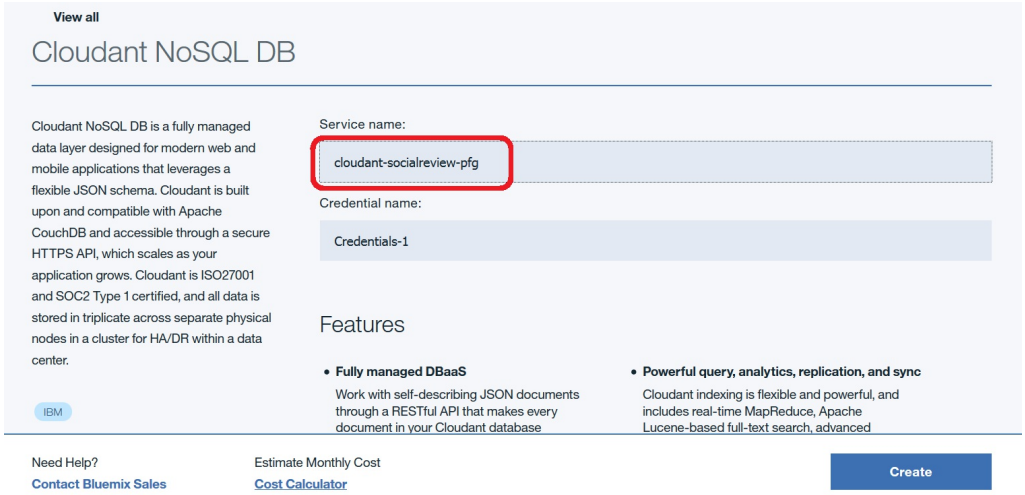
Provision Cloudant Database in Bluemix

1. Login to the virtual machine as `bmxuser` and password of `passw0rd`.
2. Open a Firefox browser by clicking **Activities** and then clicking the Firefox icon.

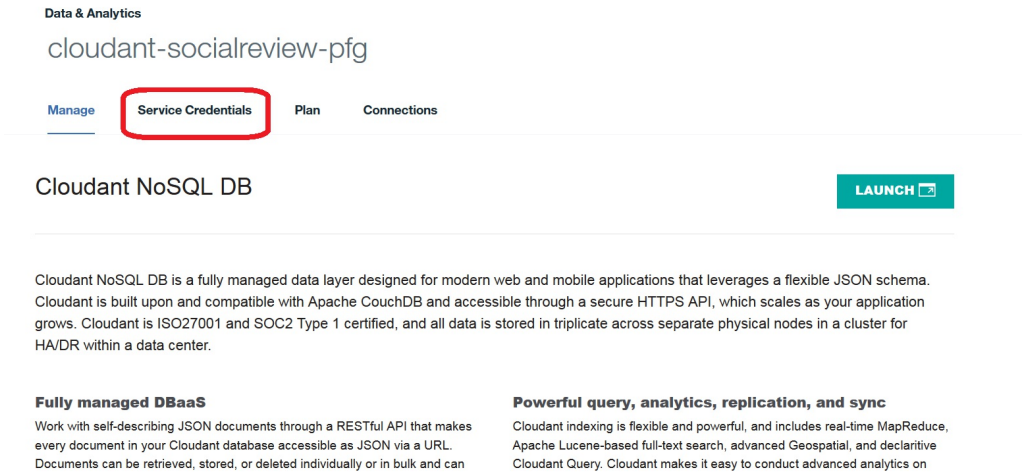
3. Login to your Bluemix console at **console.ng.bluemix.net** for the US region. You may typically use a different region, but for these exercises, please use the US region.
4. Click **Catalog**, then under Services, **Data and Analytics**.
5. Click on **Cloudant NoSQL DB**.



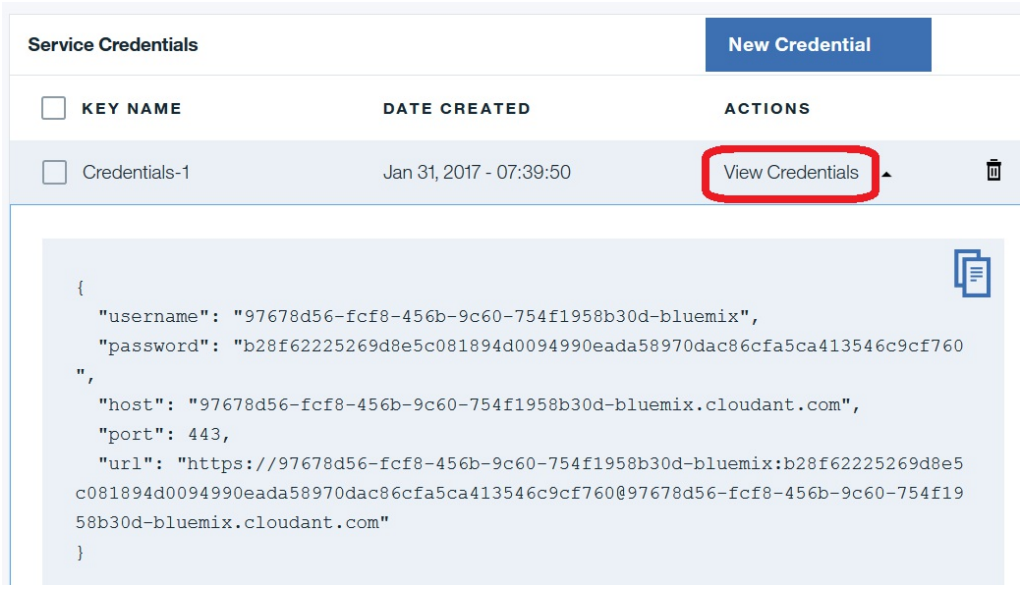
6. Give your Cloudant service a unique name like `cloudantdb-socialreview-[yourinitials]`, replacing *[yourinitials]* with your own initials, such as pfg in this example.



7. For testing, use the default **Lite** plan and click **Create**
8. Once the service has been created, click **Service Credentials** to view the service credentials.



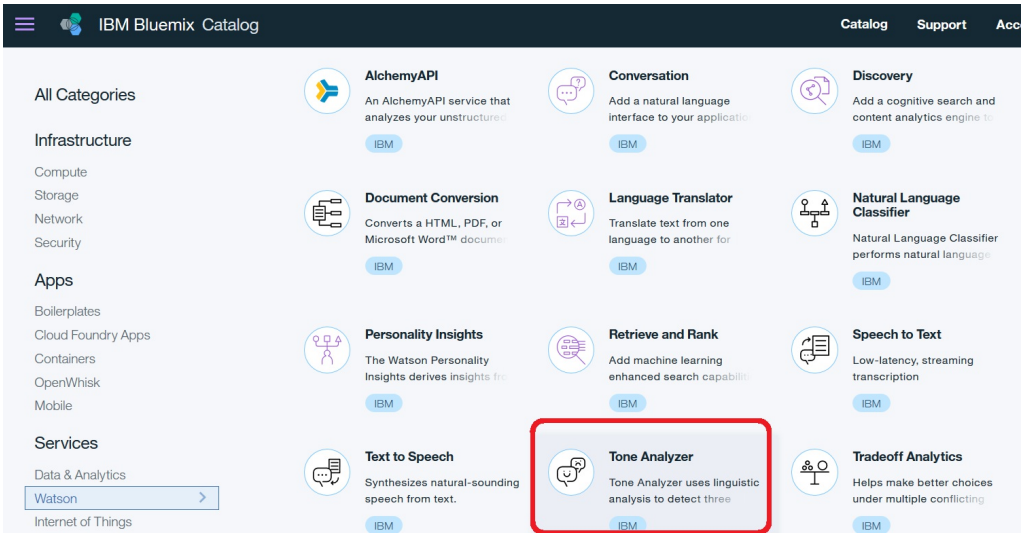
9. Click **View Credentials** to display the credentials for the Cloudant NoSQl DB Service. The Social Review microservice requires the `url` property.



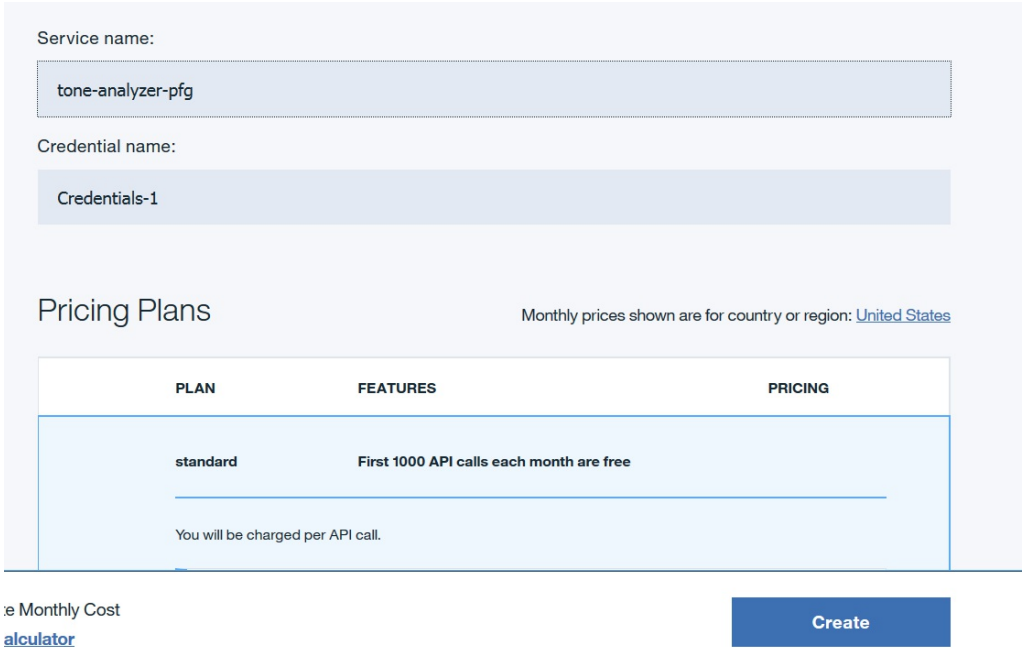
Provision Watson Tone Analyzer in Bluemix

Next you provision an instance of the Tone Analyzer service to analyze the reviews that are posted.

1. In your Bluemix console, Click **Catalog**, then under **Services** click **Watson**.
2. Click **Tone Analyzer**.



3. Name your Watson Tone Analyzer service `tone-analyzer-[yourinitials]`, replacing *[yourinitials]* with your own initials.
4. For testing, use the default **standard** plan and click **Create**.



5. Once the service has been created, note the service credentials under **Service Credentials**. In particular, the Social Review microservice requires the `username`, `password`, and `url` properties.

Download and configure the OpenWhisk CLI

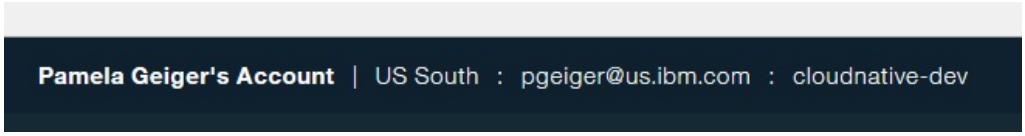
1. Open a terminal Window by selecting **Activities** and **Show Applications** from the panel and typing **XTerm** into the search field. Click the Xterm icon to open the terminal.

2. Get the code for this lab by entering the following command:

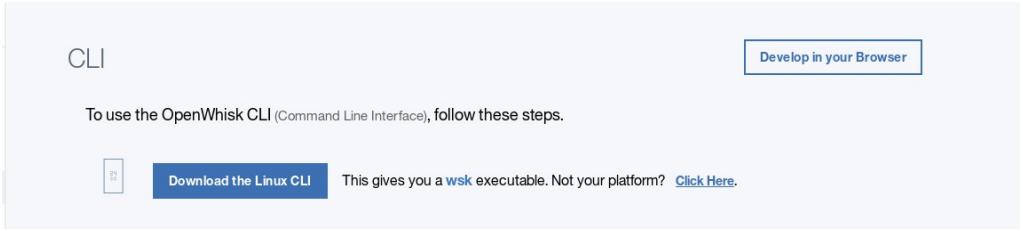
```
git clone -b openwhisk https://github.com/bluemix-enablement/BMX2450OpenWhisk.git
```

This creates a directory named BMX2450OpenWhisk.

3. Open a new browser window and download the OpenWhisk CLI for your platform from this link https://console.ng.bluemix.net/openwhisk/cli. When the Bluemix Console opens, verify that you are in the org and space that correspond to where your Cloudant service was created.



4. Click **Download the Linux CLI**. If the button shows a different platform, click where it says `Click Here`.



5. The file name is **OpenWhisk_CLI-linux.tgz**. Click **Save File** and then **OK**.
6. In the terminal Xterm window, change to the download directory and enter the following command to extract the command line interface:

```
tar -xvf OpenWhisk_CLI-linux.tgz
```

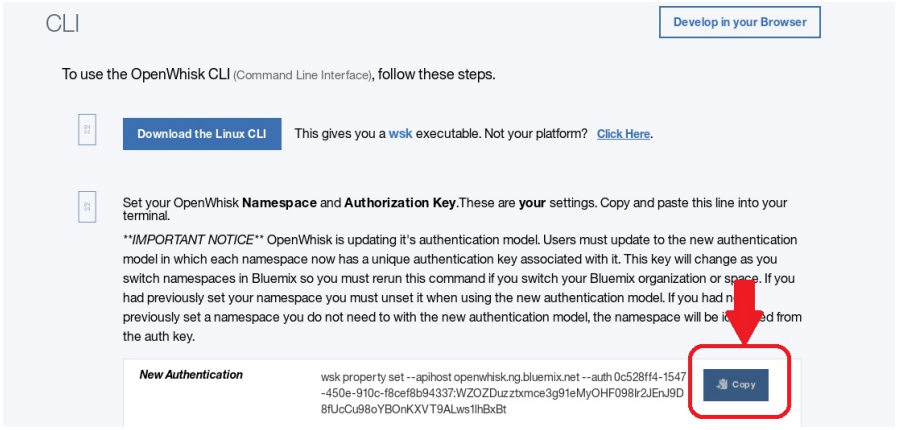
7. You must add the *wsk* path to the PATH variable. First, you must find the full path to where you just downloaded the file. In the terminal, type
   `pwd wsk`
8. Note the response, which is something like *home/usr/Downloads*.
   NOTE: normally, you would move this file to a more appropriate directory. For this exercise, you can just leave it in the *Downloads* directory.
9. In the terminal, type the following:
   `export PATH=$PATH:/[path to your file]`
10. Verify that the cli works by entering the command:

```
wsk
```

Usage information for the wsk command is dislayed.

```
bmxuser@bluecompute:~$ cd Downloads/
bmxuser@bluecompute:~/Downloads$ ls
OpenWhisk_CLI-linux.tgz
bmxuser@bluecompute:~/Downloads$ tar -xvf OpenWhisk_CLI-linux.tgz
wsk
bmxuser@bluecompute:~/Downloads$ pwd
/home/bmxuser/Downloads
bmxuser@bluecompute:~/Downloads$ export PATH=$PATH:/home/bmxuser/Downloads
bmxuser@bluecompute:~/Downloads$ wsk


Usage:
  wsk [command]

Available Commands:
  action          work with actions
  activation      work with activations
  package         work with packages
  rule            work with rules
  trigger         work with triggers
  sdk             work with the sdk
  property        work with whisk properties
  namespace       work with namespaces
  list            list entities in the current namespace
  api-experimental work with APIs

Flags:
      --apihost HOST         whisk API HOST
      --apiversion VERSION   whisk API VERSION
  -u, --auth KEY             authorization KEY
  -d, --debug                debug level output
  -h, --help                 help for wsk
  -i, --insecure             bypass certificate checking
  -v, --verbose              verbose output

Use "wsk [command] --help" for more information about a command.
bmxuser@bluecompute:~/Downloads$
```

11. Return to the browser tab with the OpenWhisk CLI information and copy the command to configure the OpenWhisk CLI. Note that this is customized for your Bluemix id and the org and space that you are logged into, which is why you verified this information in step one.



12. Paste the link into your terminal window and run it.

```
bmxuser@bluecompute:~/Downloads$ wsk property set --apihost openwhisk.ng.bluemix.net --auth 0c528ff4-1547-
450e-910c-f8cef8b94337:WZOZDuzztxmce3g91eMyOHF098Ir2JEnJ9D8fUcCu98oYBOnKXVT9ALws1lhBxBt
ok: whisk auth set to 0c528ff4-1547-450e-910c-f8cef8b94337:WZOZDuzztxmce3g91eMyOHF098Ir2JEnJ9D8fUcCu98oYBO
nKXVT9ALws1lhBxBt
ok: whisk API host set to openwhisk.ng.bluemix.net
```

This sets your OpenWhisk namespace and authorization key.

13. Run the following to automatically create OpenWhisk packages with the Cloudant credentials in your space:

```
wsk package refresh
```

This should result in a package containing your Cloudant database credentials.

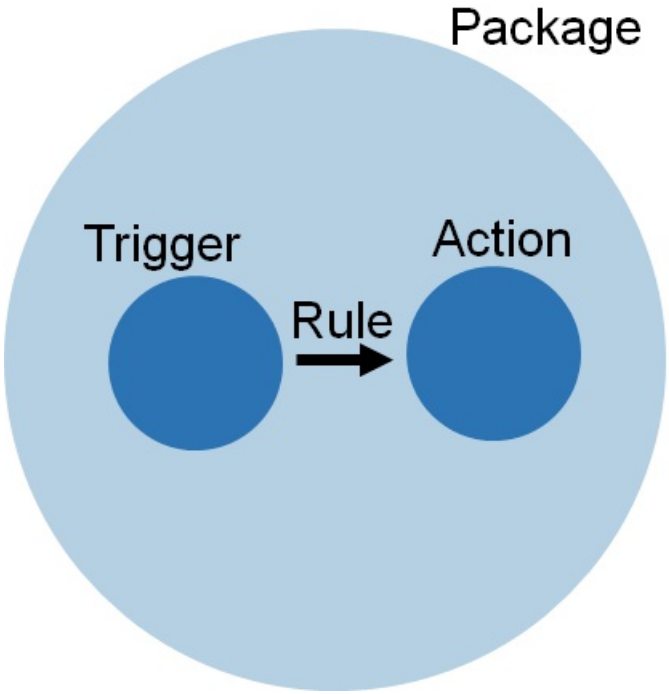
```
bmxuser@bluecompute:~/Downloads$ wsk package refresh
_ refreshed successfully
created bindings:
updated bindings:
Bluemix_cloudant-socialreview-pfg_Credentials-1
deleted bindings:
```

Enter the following to list your OpenWhisk packages.

```
wsk package list
```

```
bmxuser@bluecompute:~/Downloads$ wsk package list
packages
/pgeiger@us.ibm.com_cloudnative-dev/Bluemix_cloudant-socialreview-pfg_Credentials-1 private
```

Deploy the OpenWhisk package and actions



Some OpenWhisk terminology, as a reminder:

Event Sources

Event sources, such as devices, queues, databases, and webhooks, emit classes of events in the form of triggers.

Triggers

Triggers are the class of events (including device readings, published messages, and data changes) that are emitted by event sources.

Actions

Actions are functions that encapsulate code – written in any supported language by implementing a single method signature – to be executed in response to a trigger.

Rules

Rules represent the declarative association between a trigger and an action, defining which action(s) should be executed in response to an event.

Packages

Packages encapsulate external services in a reusable manner and assemble them into triggers and actions.

For these lab exercises, a set of actions is being provided. In this case, the actions are written in JavaScript, but can also be Swift, or run in a Docker container. You will first create a socialreview package and then add the actions to that package. The properties that are added to the package are available to the actions in the package, so the access information for the Cloudant database and the Watson Tone Analyzer only need to be defined at the package level.

1. Change directories to the BMX2450OpenWhisk directory that contains the code that you downloaded earlier. If you are currently in the Downloads directory, the command

will be:

```
cd ../BMX2450OpenWhisk/
```

2. Use the OpenWhisk CLI to create a `socialreview` package. Pass the `url` property from the Cloudant service instance created, and the `username`, `password`, and `url` properties from the Watson Tone Analyzer service instance.

```
 wsk package create socialreview --param cloudant_url <cloudant url> --param
watson_url <watson tone analyzer url> --param watson_username <watson tone analyzer
username> --param watson_password <watson tone analyzer password --param
cloudant_reviews_db socialreviewdb
```

It looks similar to the following

```
bmxuser@bluecompute:~/BMX2450OpenWhisk/openwhisk/actions$ wsk package create socialreview --param cloudant
_url https://d20ca3f7-373b-46cc-b955-ec15327ae373-bluemix:99e7cd9bf0f0c654d37e99228b6bcf831b2de5b46cfe37f4
bf1b4418c8573ed8@d20ca3f7-373b-46cc-b955-ec15327ae373-bluemix.cloudant.com --param watson_url https://gate
way.watsonplatform.net/tone-analyzer/api --param watson_username 99daa030-c736-400b-be61-c6617b9bf81f --pa
ram watson_password SBpaBKFbsO5F --param cloudant_reviews_db socialreviewdb
```

3.. You are creating four OpenWhisk actions:

- **initCloudant**: connects the the Cloundant instance that is running in Bluemix using the url that you defined when you created the package. The database credentials are retrieved from Bluemix and the socialreviewdb is created if it hasn't been created yet. The name fo the database when also defined when you created the package, in the cloudant_reviews_db parameter.
- **saveReview**: This action saves new reviews to the database.
- **getReviews**: Retrieves reviews from the database.
- **analyzeTone**: Invokes Tone Analyzer to analyze the tone of the review, and stores the analysis in the cloudant database with the review entry. It uses the Tone analyzer url, username, and password that you defined when you created the package in the previous step.

3. Upload all of the actions under the created package. The code for the actions is in the openwhisk/actions directory. All of the actions in the package inherit the properties we created in the package (`cloudant_url`, `watson_url`, `watson_username`, `watson_password`, `cloudant_reviews_db`) :

```
# wsk action create socialreview/initCloudant openwhisk/actions/initCloudant.js
# wsk action create socialreview/saveReview openwhisk/actions/saveReview.js
# wsk action create socialreview/getReviews openwhisk/actions/getReviews.js
# wsk action create socialreview/analyzeTone openwhisk/actions/analyzeTone.js
```

The four actions are created in the socialreview package:

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk action create socialreview/initCloudant openwhisk/actions/init
Cloudant.js
ok: created action socialreview/initCloudant
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk action create socialreview/saveReview openwhisk/actions/saveRe
view.js
ok: created action socialreview/saveReview
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk action create socialreview/getReviews openwhisk/actions/getRev
iews.js
ok: created action socialreview/getReviews
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk action create socialreview/analyzeTone openwhisk/actions/analy
zeTone.js
ok: created action socialreview/analyzeTone
```

3. View the actions and packages. From the command line, enter

```
#wsk package list
```

You now have the socialreview package in addition to the cloudant package.

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk package list
packages
/pgeiger@us.ibm.com_cloudnative-dev/socialreview                     private
/pgeiger@us.ibm.com_cloudnative-dev/Bluemix_cloudant-socialreview-pfg_Credentials-1 private
```

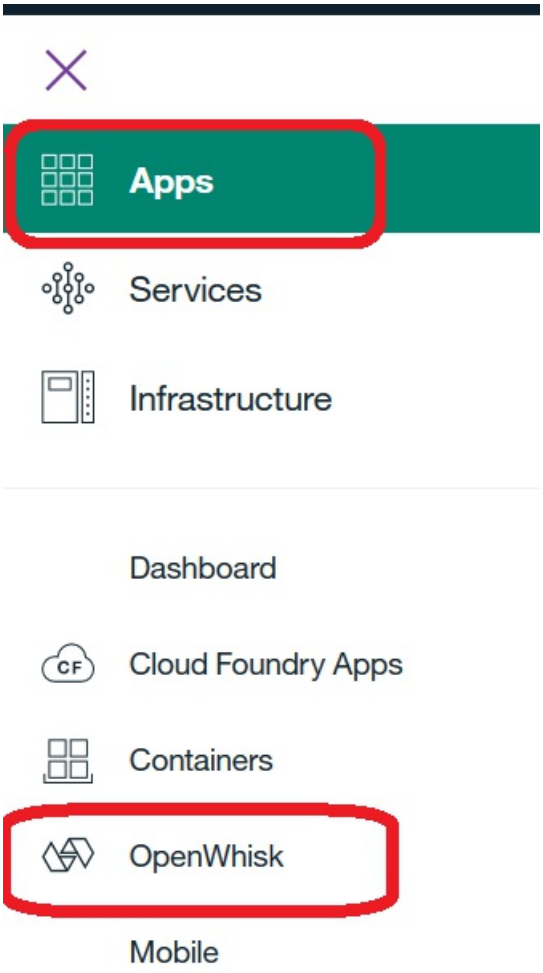
To see the actions, enter:

```
#wsk action list
```

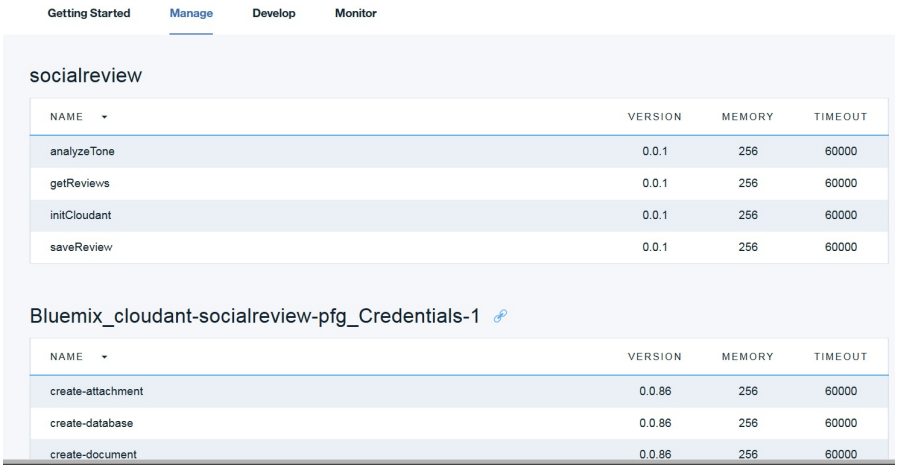
You see the four actions that you just added to the socialreview package,

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk action list
actions
/pgeiger@us.ibm.com_cloudnative-dev/socialreview/analyzeTone          private nodejs:6
/pgeiger@us.ibm.com_cloudnative-dev/socialreview/getReviews           private nodejs:6
/pgeiger@us.ibm.com_cloudnative-dev/socialreview/saveReview           private nodejs:6
/pgeiger@us.ibm.com_cloudnative-dev/socialreview/initCloudant         private nodejs:6
```

4. View the information in the OpenWhisk Dashboard. The OpenWhisk Dashboard in Bluemix can be used to Develop, Monitor, and Manage your OpenWhisk packages, actions, and triggers as well. In your Bluemix interface, open the navigation menu and click **Apps** then **OpenWhisk** to open the Dashboard.



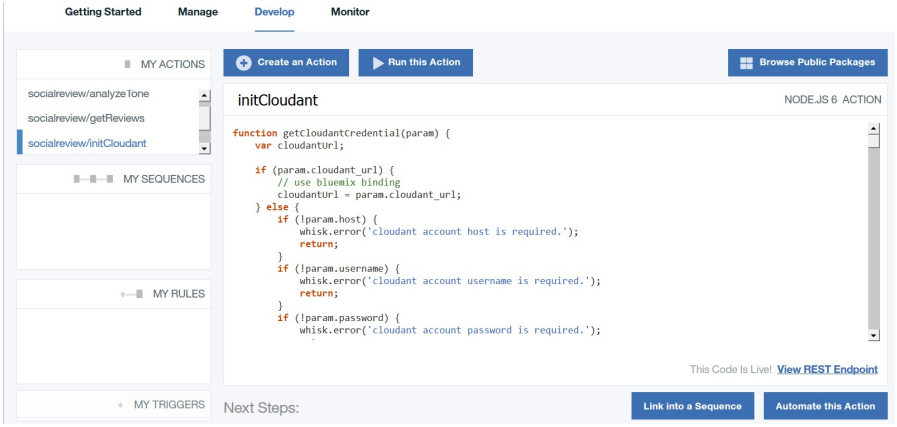
The Dashboard opens to the Getting Started Window. Click **Manage** to see your packages and actions. You see the socialreview package with the four actions that you added, as well as the package for your Cloudant service. Scroll through the list to see the actions that are available as part of that package.



5. Actions can be executed directly from the Dashboard or from the command line, as well as invoked automatically by a trigger. We'll take a look at each of these options. First, execute the initCloudant OpenWhisk action to create the Cloudant databases and

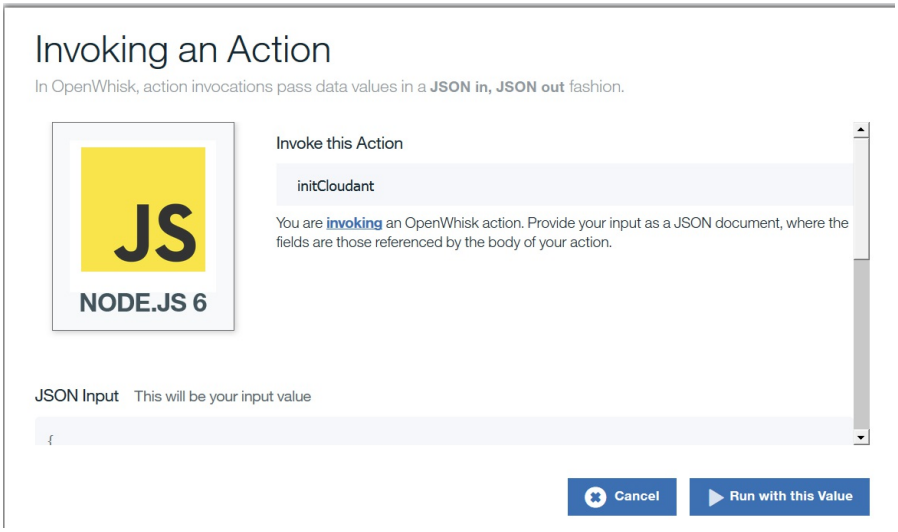
indexes required by the Social Review microservice. Since you are already looking at the Dashboard, let's invoke this action here.

- In the OpenWhisk Dashboard, click **Develop**. The actions that you created are listed under My Actions. Any Sequences, Rules, and Triggers would be displayed here as well.
- Click **initCloudant** to select that action. The Node.js source for the action is displayed.

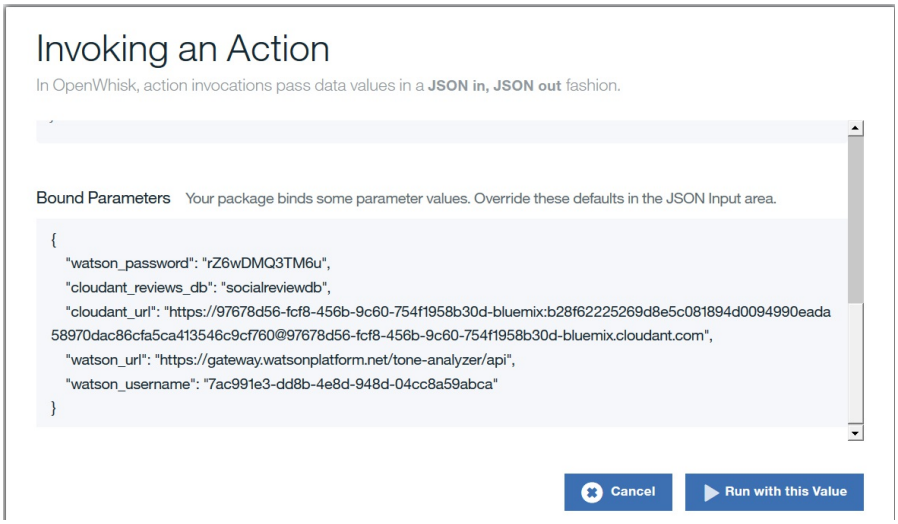


Notice that you have a number of options. You can Create a new action, run this action, link the action into a sequence, or automate the action. You can also make changes to the action source code from here as needed.

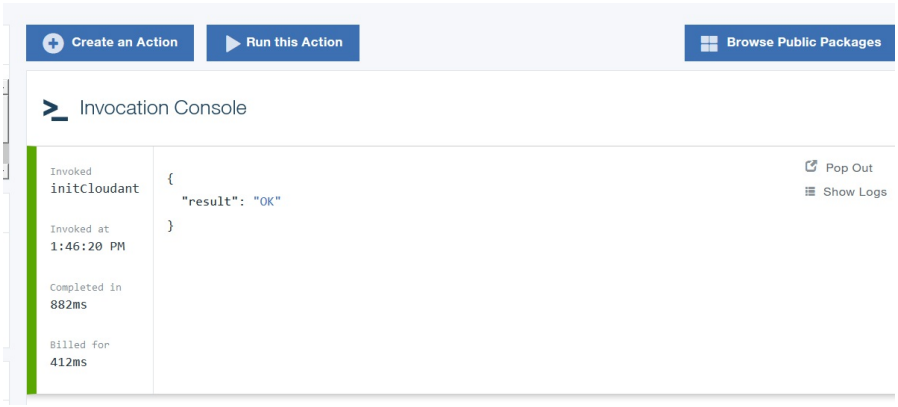
- Click Run this Action. The Invoking an action window is displayed. You can specify JSON input from this window that will be passed as input to the action.



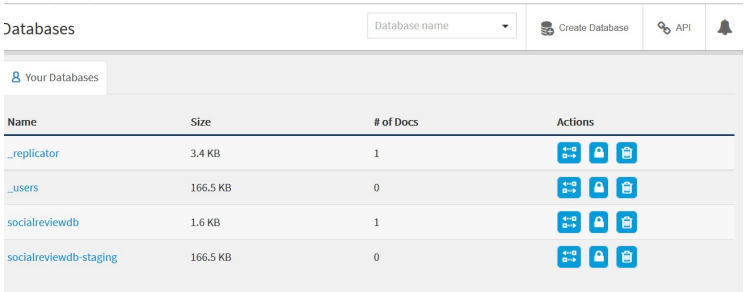
- Scroll down to see the parameters that are bound to the action. These are the parameters that you defined to the socialreview package. You can use the JSON input field to override any of these parameters, as needed.



- The only input that this action requires is the Cloudant URL, so click **Run with this value**. If you configured your Cloudant URL properly, you should see a successful result.



6. Verify that the database was initialized successfully.
    - From the Bluemix navigation menu, click **Services** and **Dashboard**.
    - Locate your Cloudant DB instance and click the name to open the Cloudant Manage window.
    - Click **Launch** to launch the Cloudant Dashboard.
    - Click Databases. You see that a socialreviewdb and a socialreview-staging database have been created.



7. Create the OpenWhisk REST API gateway for the OpenWhisk actions. The OpenWhisk API gateway is a new, experimental feature, which enables you to easily expose your OpenWhisk actions as RESTful endpoints. You can assign actions to specific

endpoints, and even have verbs (get, put, post delete) from the same endpoint assigned to different actions. Here you expose the getReviews and saveReview actions as REST APIs with the following commands:

```
# wsk api-experimental create /api /reviews/list get socialreview/getReviews
# wsk api-experimental create /api /reviews/comment post socialreview/saveReview
```

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk api-experimental create /api /reviews/list get socialreview/getReviews
ok: created API /api/reviews/list GET for action /_/socialreview/getReviews
https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/list
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk api-experimental create /api /reviews/comment post socialreview/saveReview
ok: created API /api/reviews/comment POST for action /_/socialreview/saveReview
https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/comment
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk api-experimental list
ok: APIs
Action                                  Verb  API Name  URL
/pgeiger@us.ibm.com_cloudnative-dev/soci   get      /api  https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/list
/pgeiger@us.ibm.com_cloudnative-dev/soci   post     /api  https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/comme
nt
```

8. Create an OpenWhisk trigger called `reviewTrigger` on the staging database `socialreviewdb-staging`. This uses the Whisk built-in trigger from the generated Cloudant package. Replace `<org>` and `<space>` with the Bluemix org and space that you are using for this lab and replace `<your Cloudant db name>` with the name of your Cloudant database. You can retrieve this information by running the **wsk package list** command. Reviews are initially added to this staging database, and then moved to the socialreview database after being analyzed.

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk package list
packages
/pgeiger@us.ibm.com_cloudnative-dev/Bluemix_cloudant-socialreview-pfg_Credentials-1 private
/pgeiger@us.ibm.com_cloudnative-dev/socialreview                                    private
```

Org Space Cloudant DB Name

```
# wsk trigger create reviewTrigger --feed /<org>_<space>/<your Cloudant DB
name>/changes --param dbname socialreviewdb-staging
```

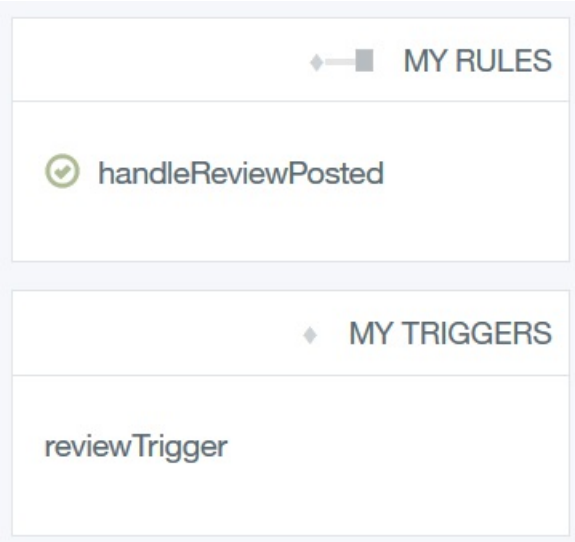
The trigger is successfully created.

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk trigger create reviewTrigger --feed /pgeiger@us.ibm.com_cloudnative-dev/Bluemix_cloudant-socialreview-p
fg_Credentials-1/changes --param dbname socialreviewdb-staging
ok: invoked /pgeiger@us.ibm.com_cloudnative-dev/Bluemix_cloudant-socialreview-pfg_Credentials-1/changes with id 88ee29724b844763ba59e6465a638be1
{
    "namespace": "pgeiger@us.ibm.com_cloudnative-dev",
    "name": "changes",
    "version": "0.0.102",
    "subject": "pgeiger@us.ibm.com",
    "activationId": "88ee29724b844763ba59e6465a638be1",
    "start": 1487271891770,
    "end": 1487271892336,
    "duration": 566,
    "response": {
        "status": "success",
        "statusCode": 0,
        "success": true,
        "result": {}
    },
    "logs": [],
    "annotations": [
        {
            "key": "limits",
            "value": {
                "logs": 10,
                "memory": 256,
                "timeout": 90000
            }
        },
        {
            "key": "path",
            "value": "whisk.system/cloudant/changes"
        }
    ],
    "publish": false
}
ok: created trigger reviewTrigger
```

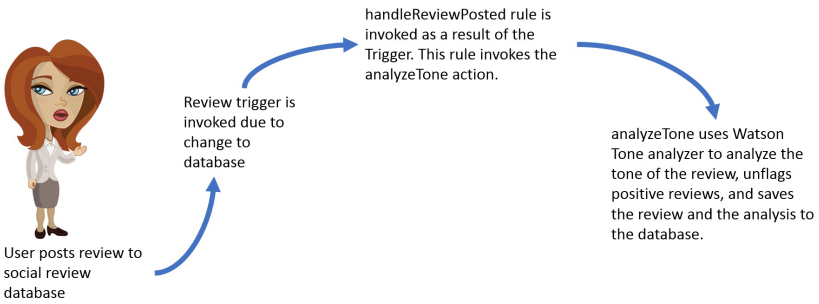
6. Create a rule that fires the `analyzeTone` action when `reviewTrigger` is triggered. This analyzes the text of posted reviews and uses the output to decide whether to unflag the review so it is returned by the API. Once the text is analyzed, it will be inserted into the socialreviewdb database.

```
# wsk rule create handleReviewPosted reviewTrigger socialreview/analyzeTone
```

- Return to the OpenWhisk Dashboard and click **Develop**. Your new rule and new trigger are now listed.



The flow you just created looks like this:



Verify the Social Review Microservice

1. Check the created OpenWhisk endpoints, for example:

```
# wsk api-experimental list
```

You see your GET and POST APIs.

```
bmxuser@bluecompute:~/BMX2450OpenWhisk$ wsk api-experimental list
ok: APIs
Action                                  Verb  API Name  URL
/pgeiger@us.ibm.com_cloudnative-dev/soci   get      /api  https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/list
/pgeiger@us.ibm.com_cloudnative-dev/soci   post     /api  https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api/reviews/comment
```

2. Create a positive review using the API, replacing the API endpoint with your API endpoint from the listing in the previous step. You can change the reviewer name, comment, review date, and email as desired.

```
# curl -X POST -H "Content-Type: application/json"    -d '{ "comment": "I love this product!", "rating":
5, "reviewer_name": "Pam Geiger", "review_date": "01/19/2016",
"reviewer_email": "pgeiger@us.ibm.com"}' <your api endpoint>/reviews/comment?itemId=13402
```

The command completes successfully.

3. Verify the Results.

- Open the OpenWhisk Dashboard in Bluemix.
- Click **Monitor** You see activity information for all of the OpenWhisk activities. In the activity log, you see that the `saveReview` action is called, which saves the review to the `socialreviewdb` database, initially flagging it.
- the `reviewTrigger` is fired,
- which triggers the `handleReviewPosted` rule,
- which executes the `analyzeTone` action. the review text, "I love this product!", is analyzed and determined to be positive, and the comment is unflagged and updated into the `socialreviewdb` database with the JSON document returned by the Watson Tone Analyzer attached.

4. Call the GET API to get the reviews for the item:

```
# curl -X GET -H "Accept: application/json" <your url endpoint>/api/reviews/list?
itemId=13402
```

It has been stored in the socialreview database and is not flagged.

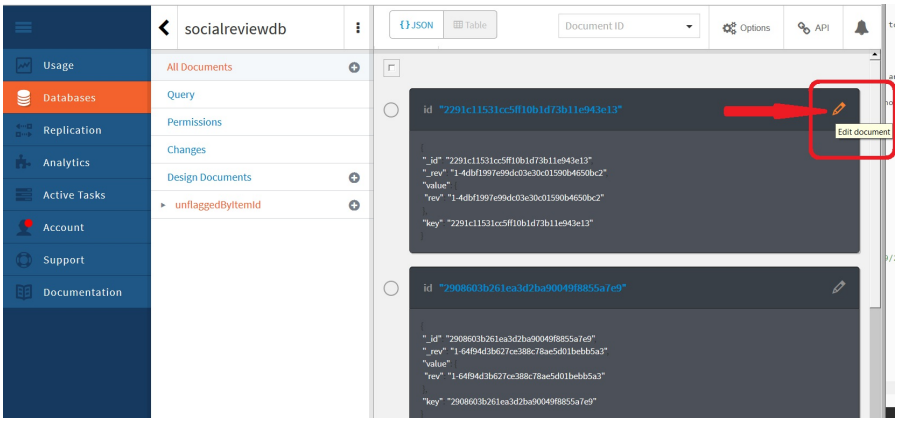
5. Now submit a negative review:

```
# curl -X POST -H "Content-Type: application/json" -d '{ "comment": "I hate this
product!", "rating": 1, "reviewer_name": "Jack Sprat", "review_date": "01/19/2016",
"reviewer_email": "jsprat@gmail.com"}' <your api endpoint>/api/reviews/comment?
itemId=13402
```

```
}bmxuser@bluecompute:~/BMX24500penWhisk$ curl -X POST -H "Content-Type: application/json"  -d '{ "comment": "I hate this product!", "rating": 1,"reviewer_n
me": "Jack Sprat", "review_date": "01/19/2016","reviewer_email": "jsprat@gmail.com"}'  https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.ne
t/api/reviews/comment?itemId=13402
{
  "result": "OK",
  "message": {
    "ok": true,
    "id": "1ad0139853f8f64732b7fe69823c3b1e",
    "rev": "1-47943a4e0f6c5fd41cbc0b7abf7093e6"
  }
bmxuser@bluecompute:~/BMX24500penWhisk$  curl -XGET -H "Accept: application/json" https://acf54f3b-6648-45ec-88c3-3423a0b6997c-gws.api-gw.mybluemix.net/api
/reviews/list?itemId=13402
{
  "docs": [{
    "reviewer_email": "pgeiger@us.ibm.com",
    "review_date": "01/19/2016",
    "flagged": false,
    "rating": 5,
    "comment": "I love this product!",
    "itemId": 13402,
    "reviewer_name": "Pam Geiger"
  }, {
    "reviewer_email": "jsprat@gmail.com",
    "review_date": "01/19/2016",
    "rating": 1,
    "comment": "I hate this product!",
    "itemId": 13402,
    "reviewer_name": "Jack Sprat"
  }]
```

6. Observe in the OpenWhisk monitor that the same sequence is fired.
7. Open the Cloudant Dashboard and click **Databases**

- Click socialreviewdb to open the database.
- Click **All Documents**
- Click **Edit Document** for the first document in the database.



- You see the review with the tone analysis information. Note that for the negative review the scores are high for anger, disgust and feat.

- Click **Cancel** and edit the postive review. Notice that this is reviews is not flagged since it is positive, and the ratings for anger, disgust, and fear are low,while the rating for joy is high.

This completes the lab exercises.