## Tutorial– Building a stateful decision service using ODM and Cloudant

#### Introduction

This tutorial section walks you through the implementation of a stateful decision service using ODM on Cloud and Cloudant. It uses the customer discount example described in the article. The tutorial can be run using IBM trial accounts.

#### Solution Design

ODM on Cloud provides the stateless decision service and Cloudant persists the purchase history. Node-JS serves as the ‘glue’ tying both of these components together. Node-JS also provides a lightweight web GUI. This design is shown below:



The sequence of events is as follows:

1. A customer selects a product
2. The purchase history of the customer is retrieved from Cloudant
3. The Decision Service is invoked, passing in the purchase history. If the rule fires, a discount is applied to the product
4. The discount is displayed to the customer

#### The ODM Decision Service

The decision service contains two business rules. The first simply adds the current purchase to the purchase history:

***then***

add the purchase of **'the customer'** to the purchase history of **'the customer'**;

The second rule checks whether the total number of purchases in the last sixty days is more than two. If so, a discount is applied:

***definitions***

**set** **purchasesInLastSixtyDays** **to** the purchase history of **'the customer'**

**where** the purchase timestamp of **each purchase** is after 60 days before now

***if***

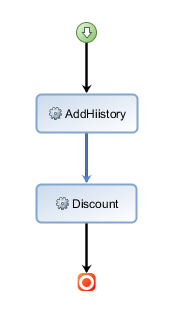
**the** **number** **of** **purchases** **in** **purchasesInLastSixtyDays** is more than 2

***then***

set the discount of **'the customer'** to 10;

set the discount reason of **'the customer'** to **"10% discount for 3 purchases in last 60 days"**;

The rule flow is as follows:



The decision service uses a *customer* json object as the input/output parameter.

Using the Stateful Decision Service Façade Design Pattern described in the article, the approach is to pass state into the decision service via the json input payload.

#### Persisting State using Cloudant

**Cloudant** is an IBM software product, which is primarily delivered as a cloud-based service. Cloudant is a non-relational, distributed database. Cloudant is ideal for persisting ODM state for the following reasons:

* The JSON model generated by ODM can be used as the database schema
* Cloudant is a document store. There are no complex queries or relationships between tables to worry about.
* Cloudant is low cost, robust, scalable and easy to use

#### What you need to start the tutorial

You will now be shown how to build and run the customer discount service using the following free trials from IBM:

* ODM on Cloud. See [ODM on Cloud](https://www-03.ibm.com/software/products/en/ibm-operational-decision-manager-on-cloud) Trial
* Cloudant. See [Cloudant Trial](https://www.ibm.com/cloud/cloudant)
* ODM Cloud. See [IBM Cloud](https://console.bluemix.net/registration) Trial

You must download the tutorial materials provided on GitHub here: [**Get the code**](https://github.com/ncrowther/OdmHadoopAdaptor)

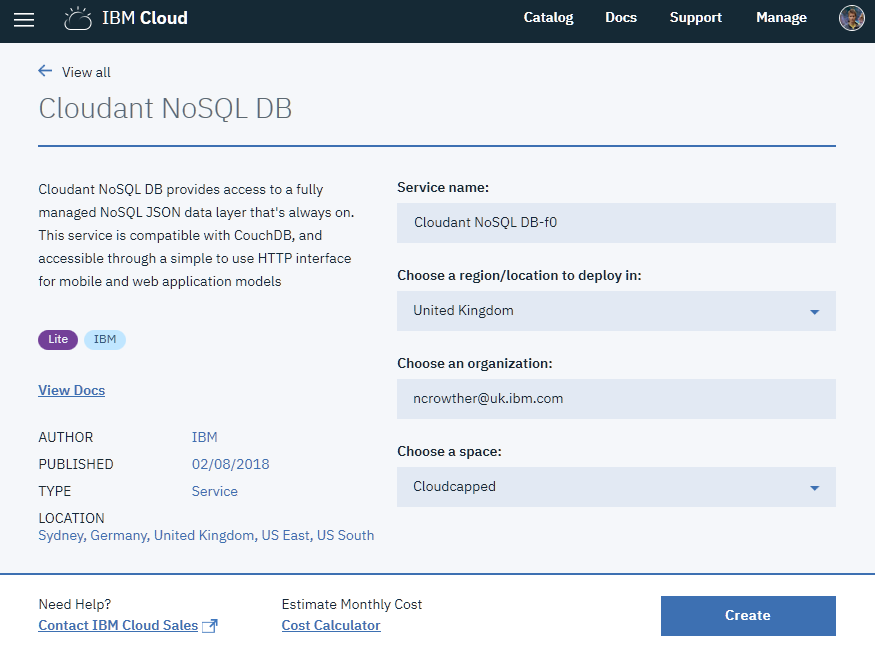
\*\* GIT HUB pic here

Click the Clone or Download button and save the code as a .zip.

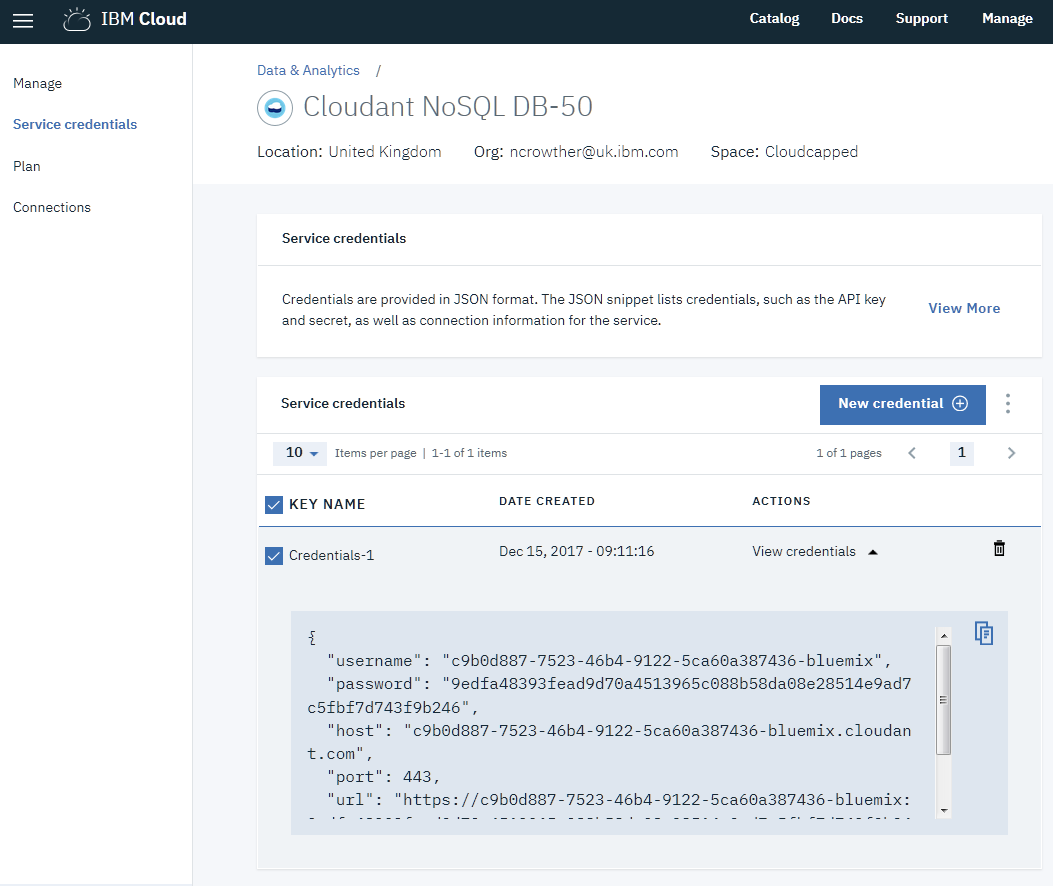
Open the zip and extract the following artifacts from the *quickstart* directory within the zip:

### Step 1. Create Cloudant Service

1. Log in to your IBM Cloud account and create the Cloudant service as shown in the following screen shot from the IBM Cloud dashboard.

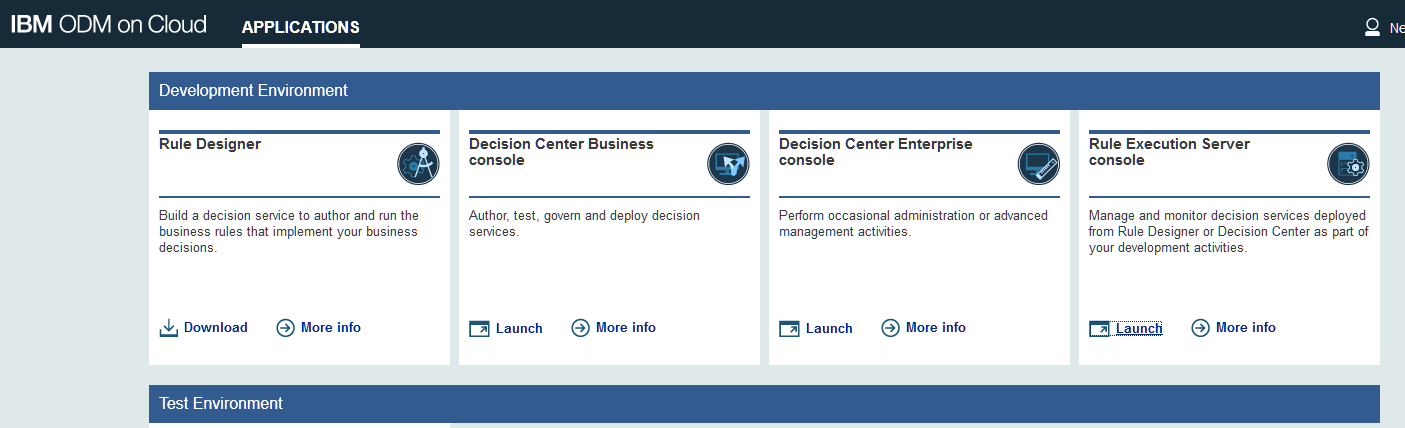


2. From the Cloudant service, click on the *Service Credentials* tab and press the New Credential button. Take note of the user name, password, host, port and URL. See following example:



#### Deploy the Decision Service

On you ODM on Cloud trial, go to the following screen:

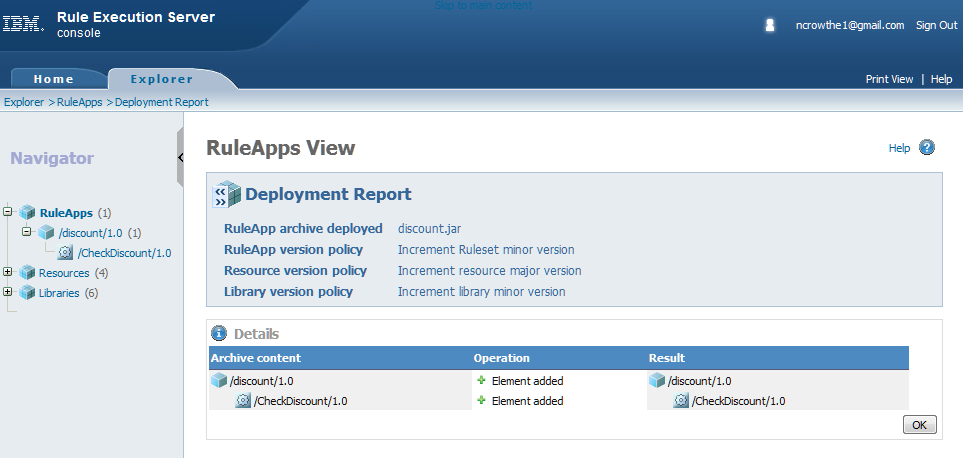


Launch the Rule Execution Server Console. Go to the **Explorer** tab and select **Deploy RuleApp Archive**, as shown in the following example.

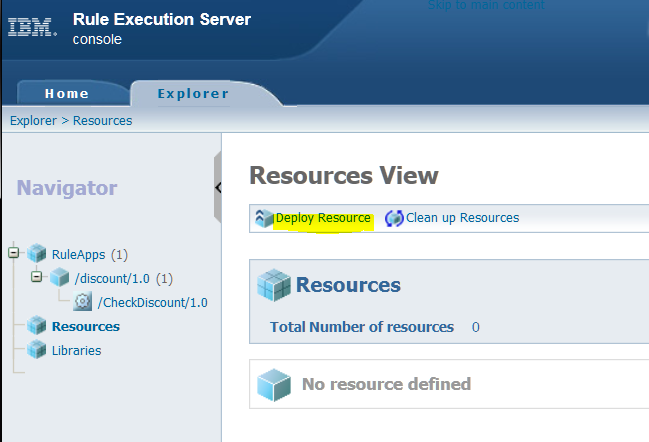


In the **Deploy RuleApp Archive** window, click the **Browse** button and select the checkDiscount.jar that you downloaded from the quickstart folder of the GIT repo. Leave the default versioning policy and click **Deploy**.

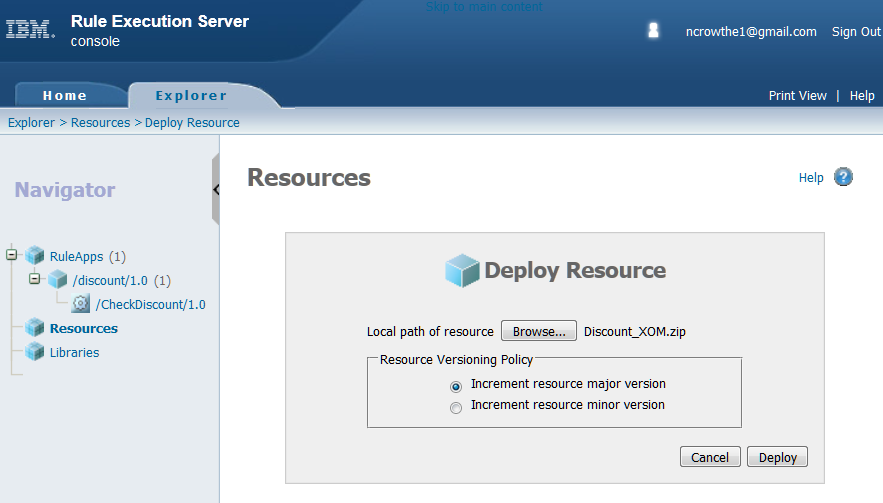
You should see the successful deployment of the discount rule app as shown below:



Now deploy the XOM. In the Explorer tab, navigate to the Resources folder and click ‘Deploy Resource’. See below:



Now browse to the *Discount\_XOM.zip* downloaded from the quickstart folder of the GIT repo. Press the Deploy button:

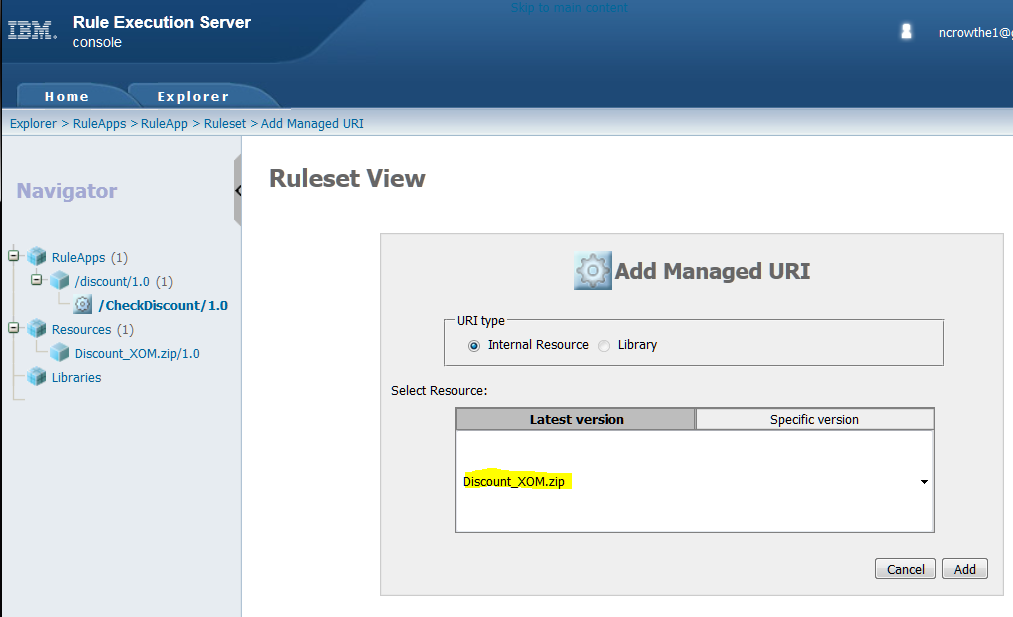


To make development changes to this decision service, you will need to download Rule Designer. This is beyond the scope of this tutorial, but the source code for the rules is also provided in section z

Go to the CheckDiscount service and select Add Managed URI:

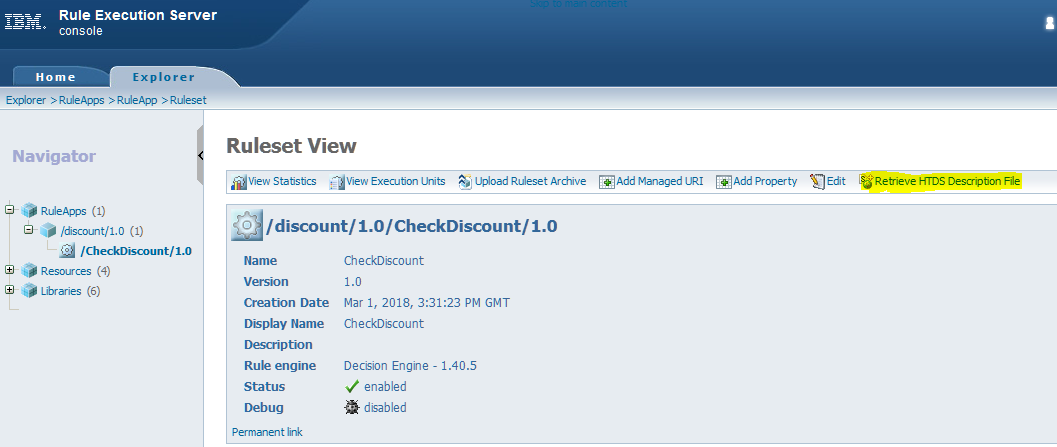


Add Discount\_XOM.zip:

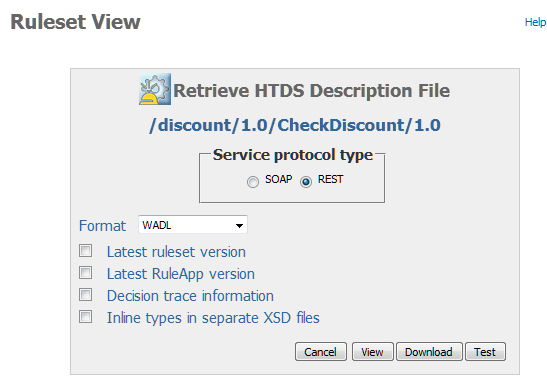


#### Test the decision service

Go back to the RES Explorer tab and select *Retrieve HTDS Description File*, highlighted below:



Select REST as the Service Protocol Type:



Press the *Test* button. Select JSON as the Execution Request type. Paste the following JSON payload in to the request:

{  
 "\_\_DecisionID\_\_": "0",  
 "customer": {  
 "customerId": "123456",  
 "purchase": {  
 "sku": "3",  
 "purchaseTimestamp": "2018-03-08T01:49:45.000+0000"  
 },  
 "discount": 0,  
 "discountReason": "",  
 "purchaseHistory": [  
 {  
 "sku": "1",  
 "purchaseTimestamp": "2018-03-01T01:49:45.000+0000"  
 },  
 {  
 "sku": "2",  
 "purchaseTimestamp": "2018-02-28T01:49:45.000+0000"  
 }  
 ]  
 }  
}

You should see the following result:

{

"\_\_DecisionID\_\_": "0",

"customer": {

"customerId": "123456",

"purchase": {

"sku": "1",

"purchaseTimestamp": "2018-03-08T01:49:45.000+0000"

},

"discount": 10,

"discountReason": "10% discount for 3 purchases in last 60 days",

"purchaseHistory": [

{

"sku": "1",

"purchaseTimestamp": "2018-03-01T01:49:45.000+0000"

},

{

"sku": "2",

"purchaseTimestamp": "2018-02-28T01:49:45.000+0000"

},

{

"sku": "3",

"purchaseTimestamp": "2018-02-08T01:49:45.000+0000"

}

]

}

}

This response proves the discount has been correctly applied.

#### Stitching it ODM and Cloudant together with Node-JS

Open the

#### Running the application

#### Where to go now

Now that you have successfully run your first ODM Cloudant application you can start to look at developing your own stateful decision services. We suggest changing the sample code first to get a feel for what it is doing. Then feel free to create your own rule projects and see where it takes you!

Please contact the authors if you need any assistance.