



**ORACLE  
OPEN  
WORLD**

**HOL6286**

## **Analyzing GoldenGate Streams with Data Integration Platform Cloud**

Alex Kotopoulos  
Director of Product Management

Thomas Vengal  
Director of Product Management

**ORACLE®**

## Hands on Lab HOL6286 – Analyzing GoldenGate Streams with Data Integration Platform Cloud

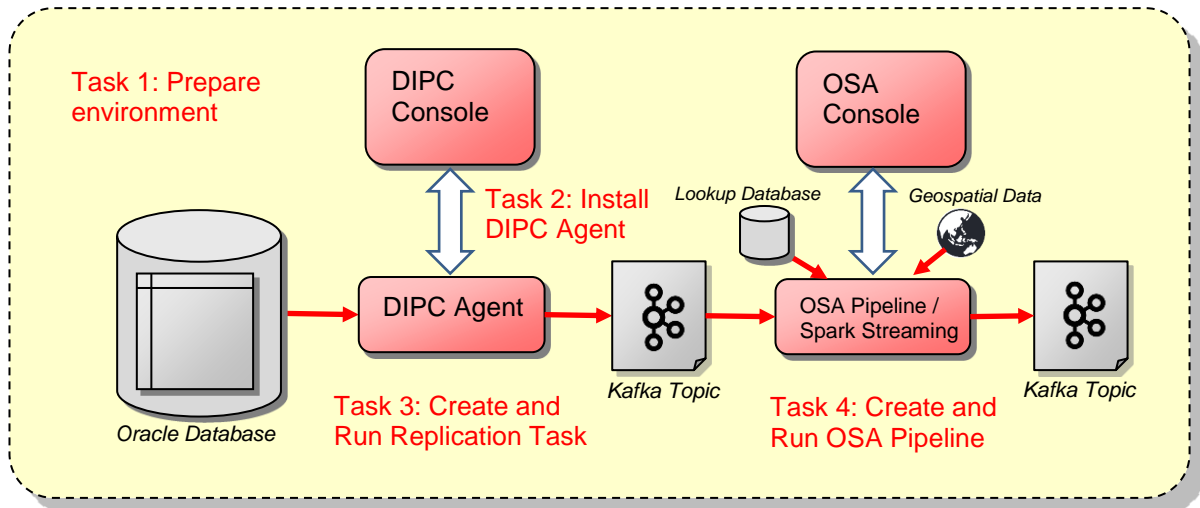
**Abstract:** This lab will highlight to Data Engineers, Software Developers, DBAs, Data Analysts and Architects some simple and cloud based technology to quickly analyze real-time data streams using Oracle Data Integration Platform Cloud and Oracle Stream Analytics. In this lab, participants will walk through the steps that are needed to replicate change data capture from databases and streaming data into Kafka and then analysing the data in real-time for complex events.

The following lessons will walk us through various steps that are needed to create a replication task on Data Integration Platform Cloud, configuration to Kafka and Oracle Stream Analytics to replicate data from an on-premises database.

HANDS ON LAB HOL6286 – ANALYZING GOLDENGATE STREAMS WITH DATA INTEGRATION PLATFORM CLOUD .....	2
ARCHITECTURE OVERVIEW .....	3
OVERVIEW .....	3
<i>Time to Complete</i> .....	3
<i>Prerequisites</i> .....	3
TASK 1: PREPARATION STEPS.....	4
TASK 2: DIPC AGENT DOWNLOAD AND INSTALL.....	5
TASK 3: CREATING CONNECTIONS AND REPLICATION TASK .....	10
TASK 4: ORACLE STREAM ANALYTICS FROM KAFKA .....	20
SUMMARY .....	29

## Architecture Overview

This Hands-on lab is based on replicating transactional data from an Oracle Database to events in Kafka and performing stream analytics on these events. The database contains user location data, and the stream analytics pipeline is creating marketing offers to users based on their geographic location close to retail stores.



We are distributing the work into 4 tasks:

1. Prepare the VM, start processes and initialize database
2. Download a DIPC Agent and configure to on-premises DB
3. Create connections and replicate task on DIPC with Oracle source and Kafka target
4. Analyze data using Oracle Stream Analytics from Kafka

## Overview

### Time to Complete

Perform all tasks – 60 Minutes

### Prerequisites

Before you begin this tutorial, you should

- Have a general understanding of RDBMS concepts.
- Have a general understanding of streaming and Kafka concepts.
- Good to have some basics of Complex Event Processing concepts.

## **Task 1: Preparation Steps**

*In these steps you will setup the environment for this exercise*

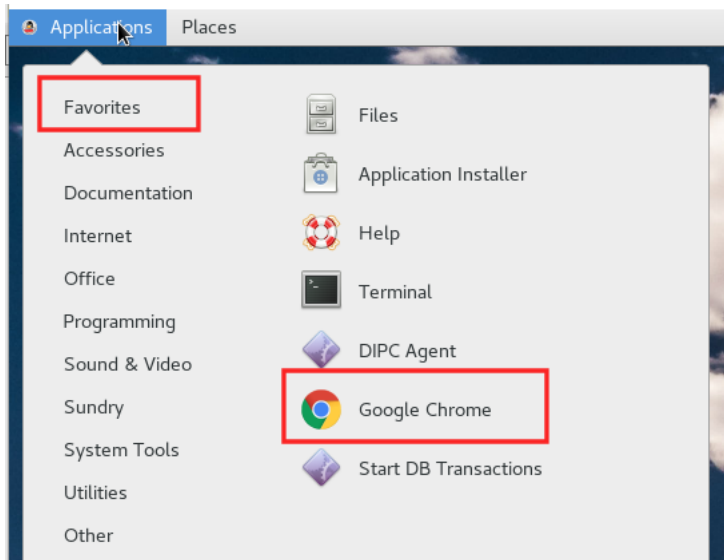
This hands-on lab assumes that an Oracle Ravello cloud instance has been set up for you, you have been given the IP address for this instance and used a VNC client to open a desktop to the Ravello instance.

The demo virtual machine simulates an Oracle Data Integration Platform Cloud (DIPC) cloud environment on a single virtual machine.

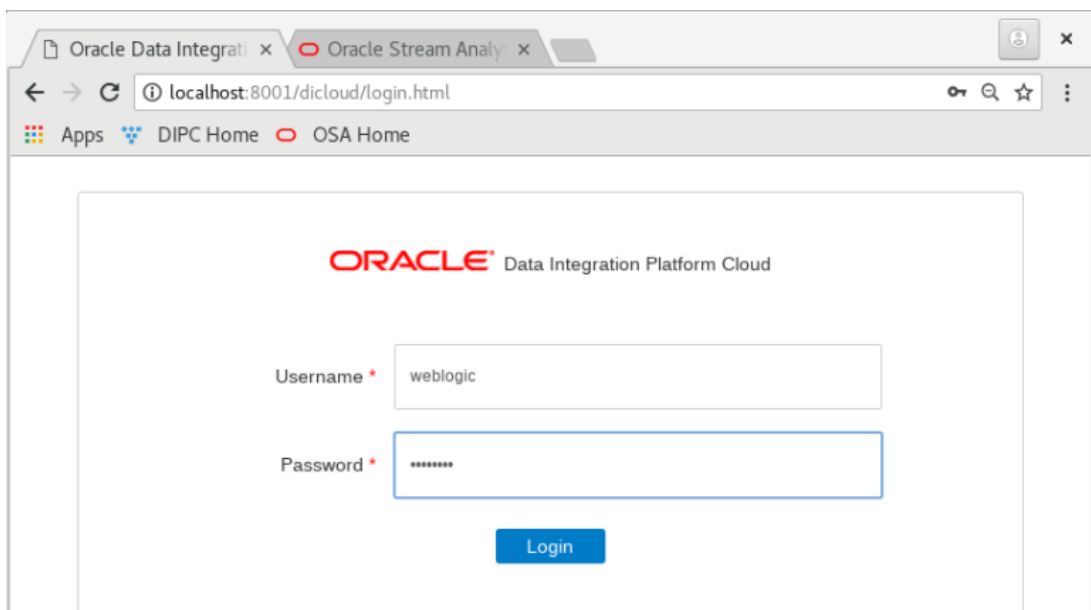
## Task 2: DIPC agent review and setup

This task will take you through the Data Integration Platform Cloud (DIPC) system login and understand how to download and install a DIPC agent to an on-premises system

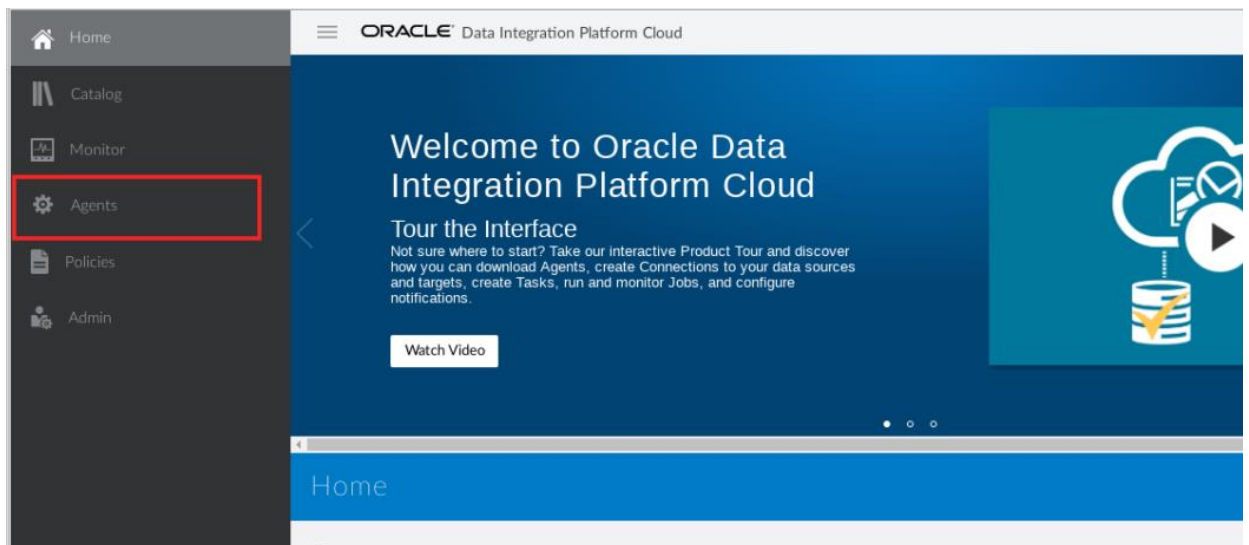
1. Open Google Chrome on the Linux desktop. You can open it from the Applications > Favorites menu



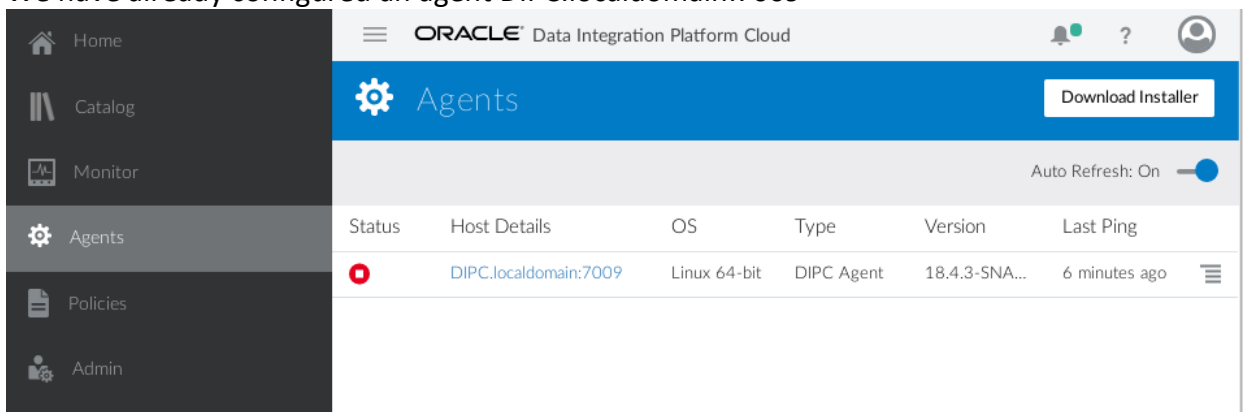
2. The browser should open with two tabs for DIPC (<http://localhost:8001/dicloud>) and OSA(<http://localhost:9080/osa>). Open the first tab and login as  
Username: weblogic  
Password: welcome1



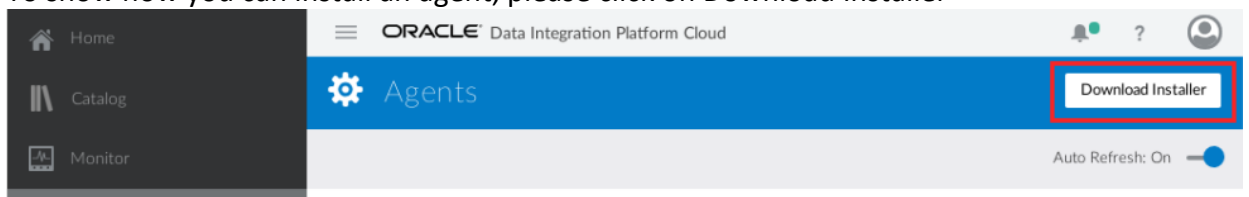
3. Click on the Agents link on the left Menu



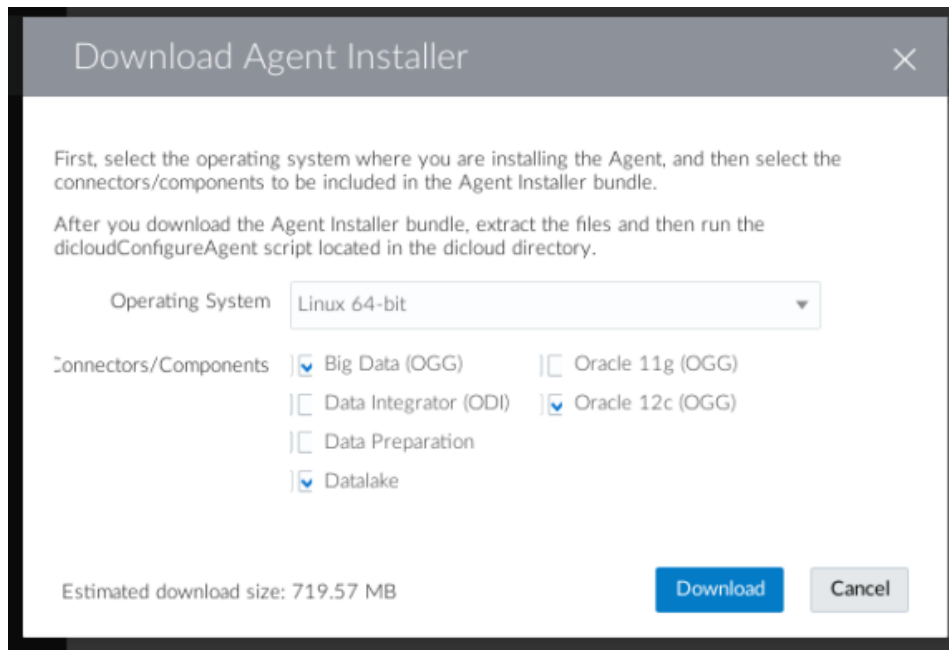
4. We have already configured an agent DIPC.localdomain:7009



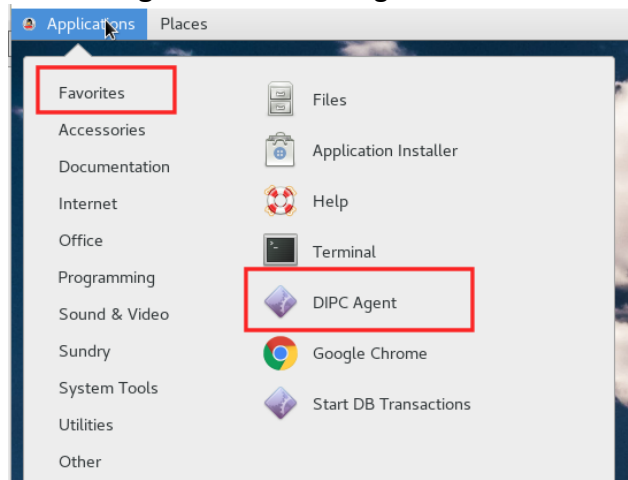
5. To show how you can install an agent, please click on Download Installer



- You can now select the components of the Agent to download. Please **Cancel** out of this screen since we already have an Agent installed and running.



- Now we will start the Agent. Please open the Applications > Favorites menu and click on **DIPC Agent** to start the agent on the local machine.

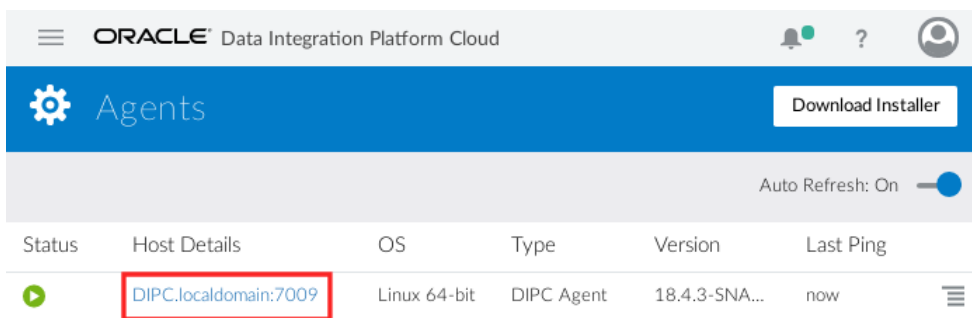


- A dialog opens and shows you the progress of the Agent start. It will take **about 90 seconds** to fully start the Agent.

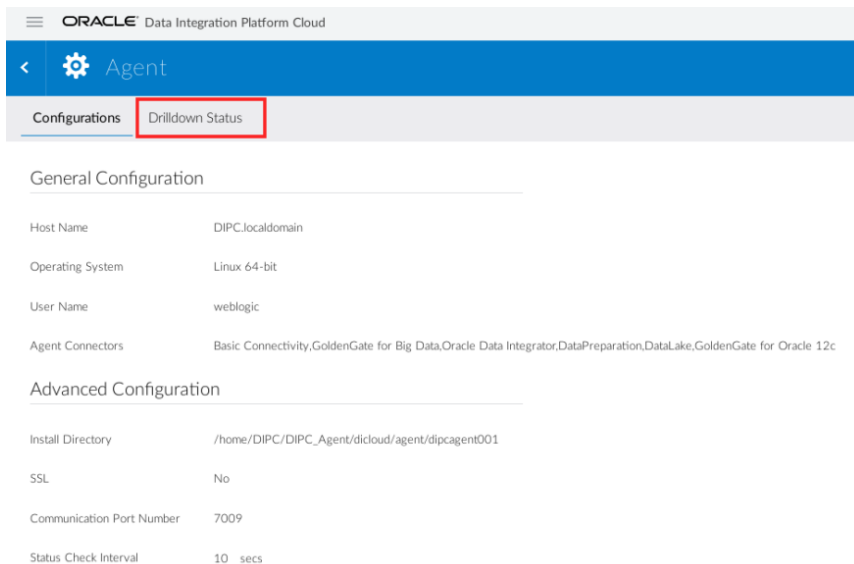
## HOL6286 – Analyzing GoldenGate Streams with Data Integration Platform Cloud

```
DIPC Agent
File Edit View Search Terminal Help
2018-10-16T21:39:59.930776Z 0 [Note] Plugin 'FEDERATED' is disabled.
2018-10-16T21:39:59.936935Z 0 [Note] InnoDB: Loading buffer pool(s) from /home/D
IPC/DIPC_Agent/dicloud/mysql_home/data/ib_buffer_pool
2018-10-16T21:39:59.943031Z 0 [Note] Found ca.pem, server-cert.pem and server-ke
y.pem in data directory. Trying to enable SSL support using them.
2018-10-16T21:39:59.943049Z 0 [Note] Skipping generation of SSL certificates as
certificate files are present in data directory.
2018-10-16T21:39:59.945422Z 0 [Warning] CA certificate ca.pem is self signed.
2018-10-16T21:39:59.945494Z 0 [Note] Skipping generation of RSA key pair as key
files are present in data directory.
2018-10-16T21:39:59.945812Z 0 [Note] Server hostname (bind-address): '*'; port:
3307
2018-10-16T21:39:59.945943Z 0 [Note] IPv6 is available.
2018-10-16T21:39:59.945958Z 0 [Note] - '::' resolves to '::';
2018-10-16T21:39:59.945992Z 0 [Note] Server socket created on IP: '::'.
2018-10-16T21:39:59.953039Z 0 [Note] InnoDB: Buffer pool(s) load completed at 18
1016 21:39:59
2018-10-16T21:39:59.975597Z 0 [Note] Event Scheduler: Loaded 0 events
2018-10-16T21:39:59.976550Z 0 [Note] /home/DIPC/DIPC_Agent/dicloud/mysql_home/bi
n/mysql: ready for connections.
Version: '5.7.21-enterprise-commercial-advanced' socket: '/home/DIPC/DIPC_Agent
/dicloud/mysql_home/socket' port: 3307 MySQL Enterprise Server - Advanced Edit
ion (Commercial)
```

9. Go back to the browser. Click on the agent name:



10. You can now see a screen with Agent Configuration details. Click on Drilldown Status link:






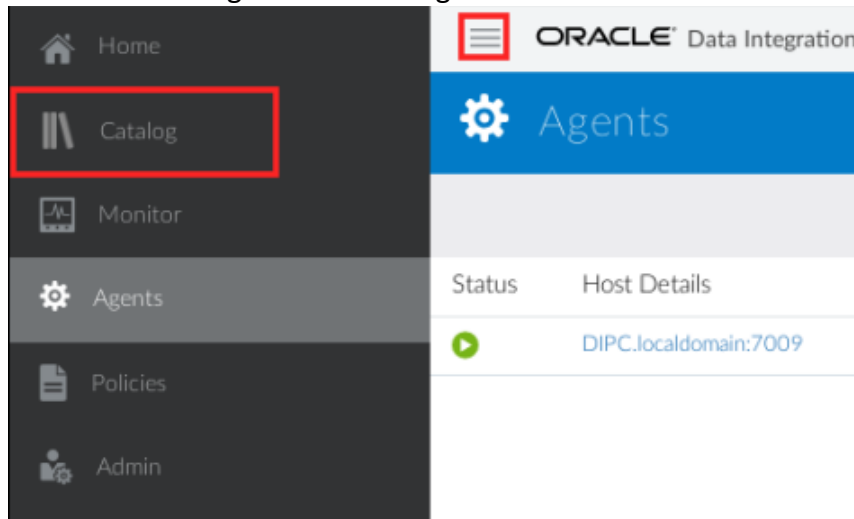
11. You should be able to see the drill down status in the Agent See whether all the components are running well and in green color.  
As the Agent is starting, some of the components might briefly show a failed state as they are still starting. This should change within a few seconds.

ORACLE Data Integration Platform Cloud		
Agent		
Configurations		Drilldown Status
<div> DIPC.localdomain:7009 DIPC Agent 18.4.3-SNAPSHOT Last Ping: Oct 11, 2018 11:41:42 PM </div>		
Status	Component	Status Message
▶	Agent Connectivity Plugin	
▶	GoldenGate BigData Manager GoldenGate Home : /home/DIPC/DIPC_Agent/dicloud/gghomebigdata	
▶	GoldenGate BigData Performance Metrics Server for BigData	
▶	GoldenGate BigData Agent Plugin	
▶	Agent GoldenGate BigData Monitoring Task	
▶	GoldenGate BigData Process Status Monitor	
▶	GoldenGate BigData Process Metrics Collector	
▶	GoldenGate BigData Process Table-level Metrics Collector	
▶	Oracle Data Integrator Plugin	
▶	Data Preparation Plugin	
▶	Datalake Builder Plugin	
▶	GoldenGate Manager GoldenGate Home : /home/DIPC/DIPC_Agent/dicloud/gghome	
▶	GoldenGate Performance Metrics Server	
▶	GoldenGate Agent Plugin	
▶	Agent GoldenGate Monitoring Task	
▶	GoldenGate Process Status Monitor	
▶	GoldenGate Process Metrics Collector	
▶	GoldenGate Process Table-level Metrics Collector	
▶	Agent Commands Listener	

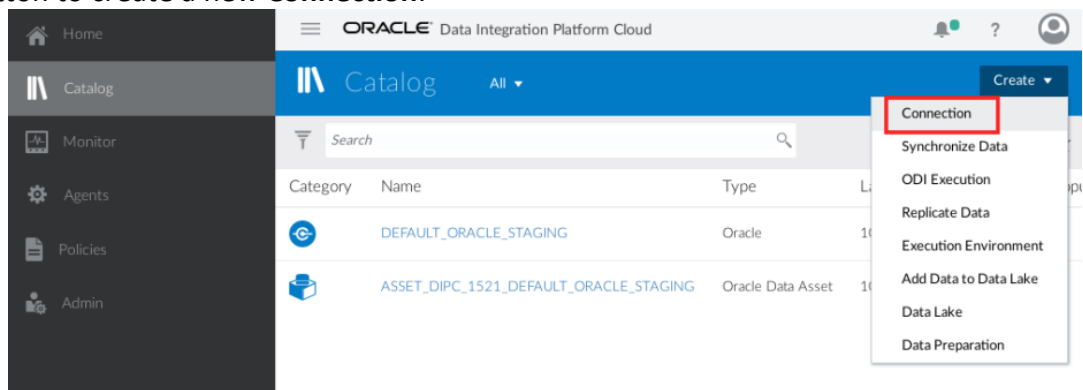
## Task 3: Creating Connections and Replication Task

In this task we will create a source connection from Oracle database for capturing all the change data capture and a target connection to Apache Kafka using the Kafka Connect interface. We will also create a Replicate task to replicate the data from Oracle DB to Kafka.

1. Navigate to the Catalog page through the left-hand menu. The menu can be opened and closed through the “Hamburger” button: 



2. You can now see the catalog with all objects for this DIPC instance. Press the **Create** button to create a new **Connection**.



3. Create a new Oracle CDB Connection by providing the following values:

- Name : **ORACLE\_CDB**
- Agent: **DIPC.localdomain:7009**
- Type: **Oracle Database**
- Subtype: **Oracle CDB**
- Hostname: **DIPC**
- Port : **1521**
- Username : **C##GGSRC**
- Password: **welcome1**
- Service name: **orcle.us.oracle.com** (Yes, it's spelled orcle ;-)

Click Test connection and Save

The screenshot shows a 'Connection' dialog box with a green banner at the top stating 'Connection test succeeded'. The dialog is divided into two main sections: 'General Information' and 'Connection Settings'.

**General Information:**

- Name \*: ORACLE\_CDB
- Identifier \*: ORACLE\_CDB
- Description: (empty)
- Agent \*: DIPC.localdomain:7009
- Type \*: Oracle Database
- Subtype \*: Oracle CDB

**Connection Settings:**

- Hostname \*: DIPC
- Port \*: 1521
- Username \*: C##GGSRC
- Password \*: (masked with dots)
- Service Name \*: orcle.us.oracle.com

At the bottom of the dialog, there are three buttons: 'Test Connection', 'Cancel', and 'Save'.

4. Create another new Connection to Oracle database by providing the following values:

- Name : **ORCL\_SRC**
- Agent: **DIPC.localdomain:7009**
- Type: **Oracle Database**
- Subtype: **Oracle**
- Hostname: **DIPC**
- Port: **1521**
- Username: **osadata**
- Password: **welcome1**
- Service Name: **orclpdb.us.oracle.com**
- Schema name: **OSADATA**
- CDB Connection: **ORACLE\_CDB**

Click **Test Connection** and **Save**

Connection

✓ Connection test succeeded

General Information

Name \* ORCL\_SRC

Identifier \* ORCL\_SRC

Description

Agent \* DIPC.localdomain:7009

Type \* Oracle Database

Subtype \* Oracle

Connection Settings

Hostname \* DIPC

Port \* 1521

Username \* osadata

Password \* .....

Service \* ☒ Service Name \* orclpdb.us.oracle.com

☐ SID

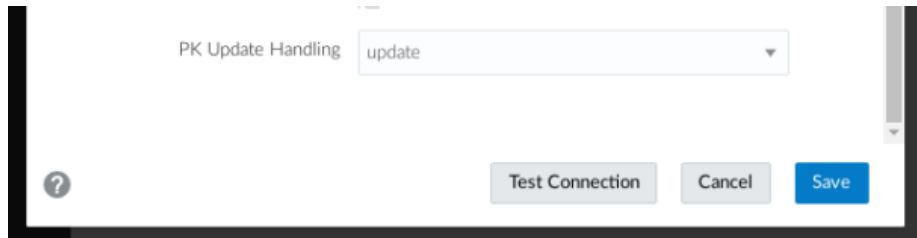
Schema Name OSADATA

CDB Connection ORACLE\_CDB

Test Connection Cancel Save

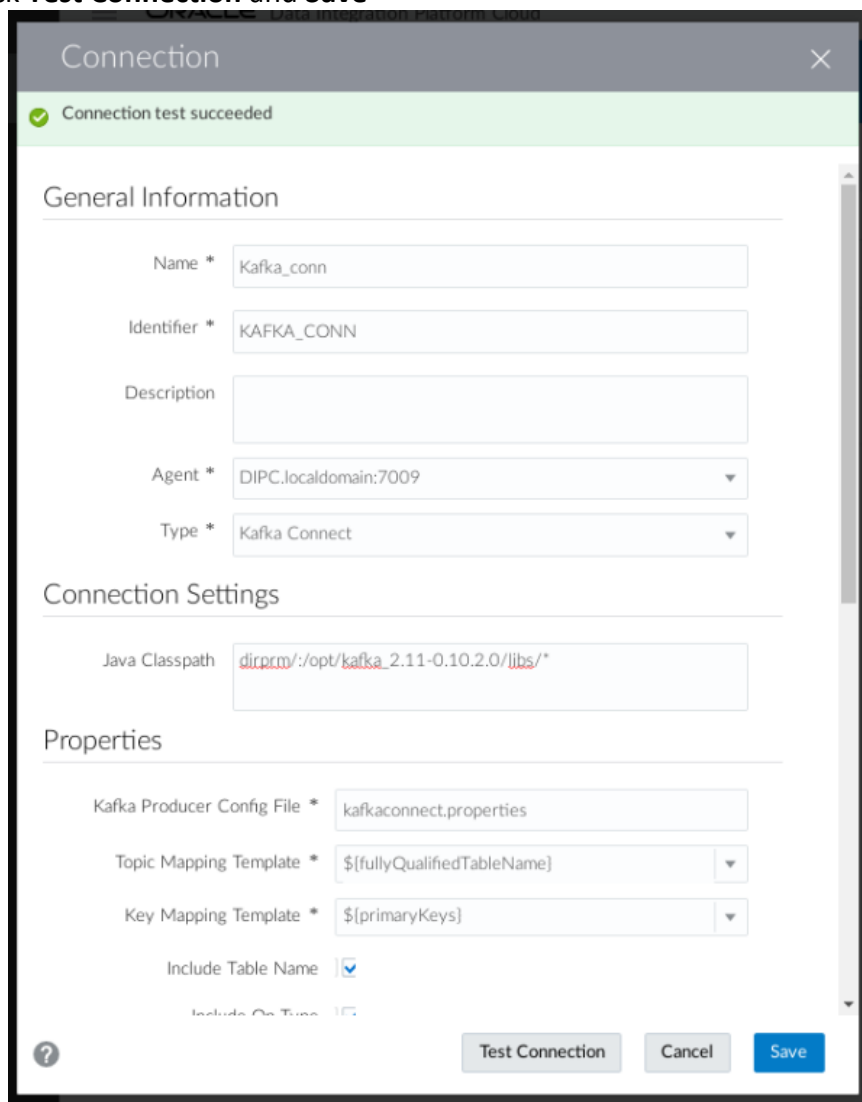
5. Create a new Connection to Kafka with the following values

- Name : **Kafka\_conn**
- Agent: **DIPC.localdomain:7009**
- Type: **Kafka Connect**
- Java Class Path: **dirprm/./opt/kafka\_2.11-0.10.2.0/libs/\***
- KafkaProducerConfigFile: **kafkaconnect.properties**
- pkUpdateHandling : **update** (At the lower end of the dialog)



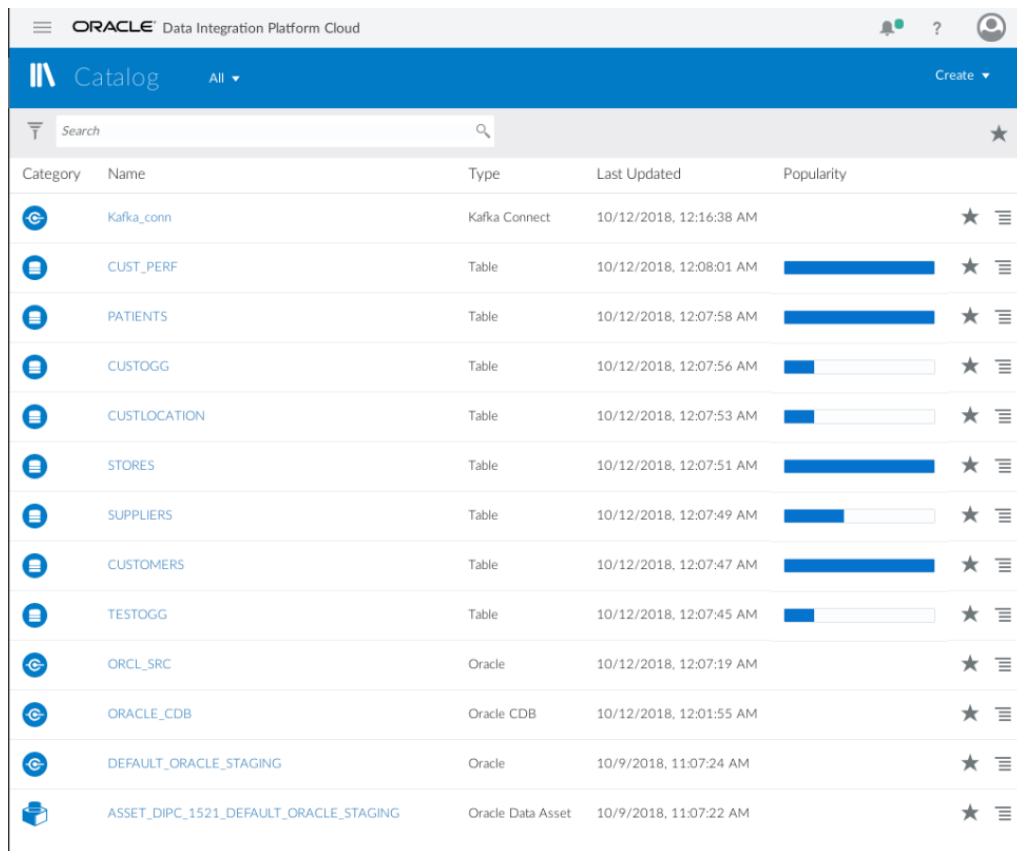
A screenshot of a dropdown menu for 'PK Update Handling'. The dropdown is open, showing the selected value 'update'. Below the dropdown are three buttons: 'Test Connection', 'Cancel', and 'Save'.

Click **Test Connection** and **Save**



A screenshot of the 'Connection' dialog box. At the top, a green banner indicates 'Connection test succeeded'. The dialog is divided into three sections: 'General Information', 'Connection Settings', and 'Properties'. The 'General Information' section contains fields for Name (Kafka\_conn), Identifier (KAFKA\_CONN), Description, Agent (DIPC.localdomain:7009), and Type (Kafka Connect). The 'Connection Settings' section contains a field for Java Classpath (dirprm/./opt/kafka\_2.11-0.10.2.0/libs/\*). The 'Properties' section contains fields for Kafka Producer Config File (kafkaconnect.properties), Topic Mapping Template (\$[fullyQualifiedTableName]), Key Mapping Template (\$[primaryKeys]), and a checked checkbox for 'Include Table Name'. At the bottom are buttons for 'Test Connection', 'Cancel', and 'Save'.

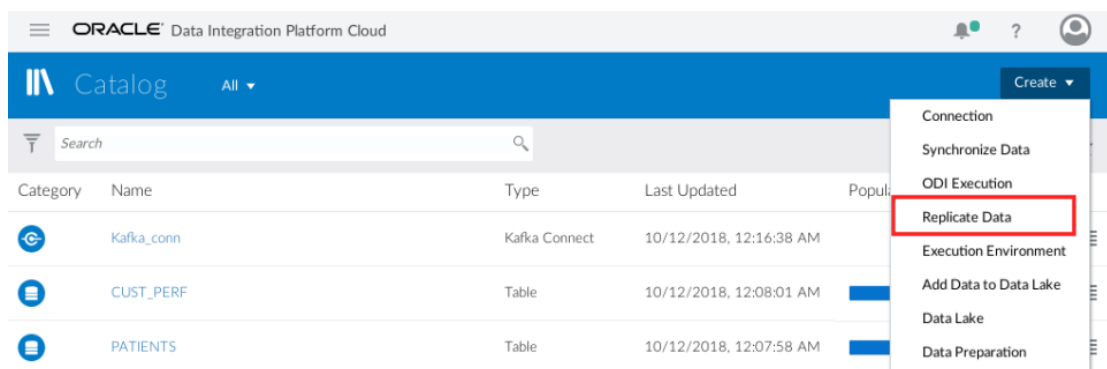
6. When looking at the catalog you will notice that tables from the configured Oracle connection have been automatically detected and are now listed as Data Entities in the catalog.



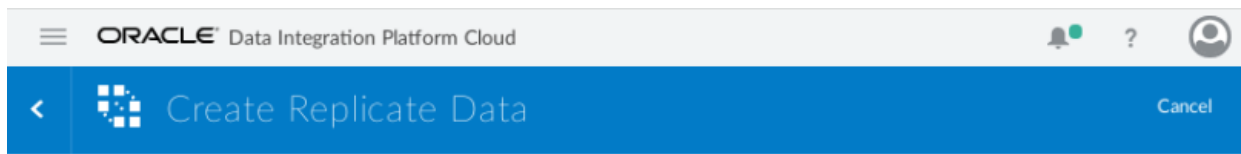
The screenshot shows the Oracle Data Integration Platform Cloud Catalog page. The header includes the Oracle logo, the text 'Data Integration Platform Cloud', and a 'Create' button. Below the header is a search bar and a table of data entities. The table has columns for Category, Name, Type, Last Updated, and Popularity. The entities listed include Kafka connections, Oracle tables, and Oracle data assets.

Category	Name	Type	Last Updated	Popularity
Kafka	Kafka_conn	Kafka Connect	10/12/2018, 12:16:38 AM	★
Table	CUST_PERF	Table	10/12/2018, 12:08:01 AM	★
Table	PATIENTS	Table	10/12/2018, 12:07:58 AM	★
Table	CUSTOGG	Table	10/12/2018, 12:07:56 AM	★
Table	CUSTLOCATION	Table	10/12/2018, 12:07:53 AM	★
Table	STORES	Table	10/12/2018, 12:07:51 AM	★
Table	SUPPLIERS	Table	10/12/2018, 12:07:49 AM	★
Table	CUSTOMERS	Table	10/12/2018, 12:07:47 AM	★
Table	TESTOGG	Table	10/12/2018, 12:07:45 AM	★
Oracle	ORCL_SRC	Oracle	10/12/2018, 12:07:19 AM	★
Oracle CDB	ORACLE_CDB	Oracle CDB	10/12/2018, 12:01:55 AM	★
Oracle	DEFAULT_ORACLE_STAGING	Oracle	10/9/2018, 11:07:24 AM	★
Oracle Data Asset	ASSET_DIPI_1521_DEFAULT_ORACLE_STAGING	Oracle Data Asset	10/9/2018, 11:07:22 AM	★

7. Create a Replicate Data task from the Create menu in the Catalog page



8. In the **Create Replicate Data** screen name the task **Replicate to Kafka**

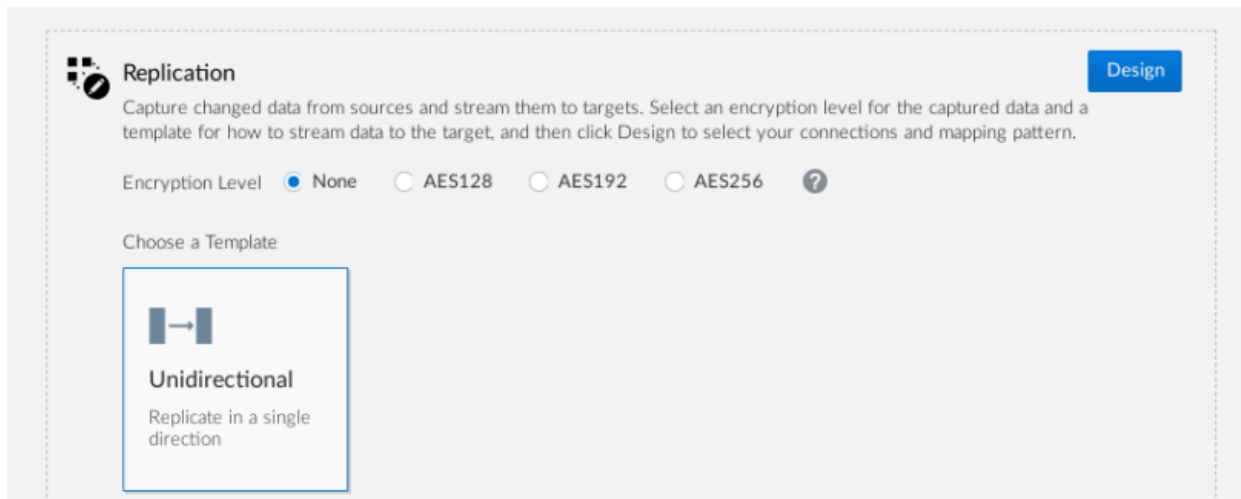


### General Information

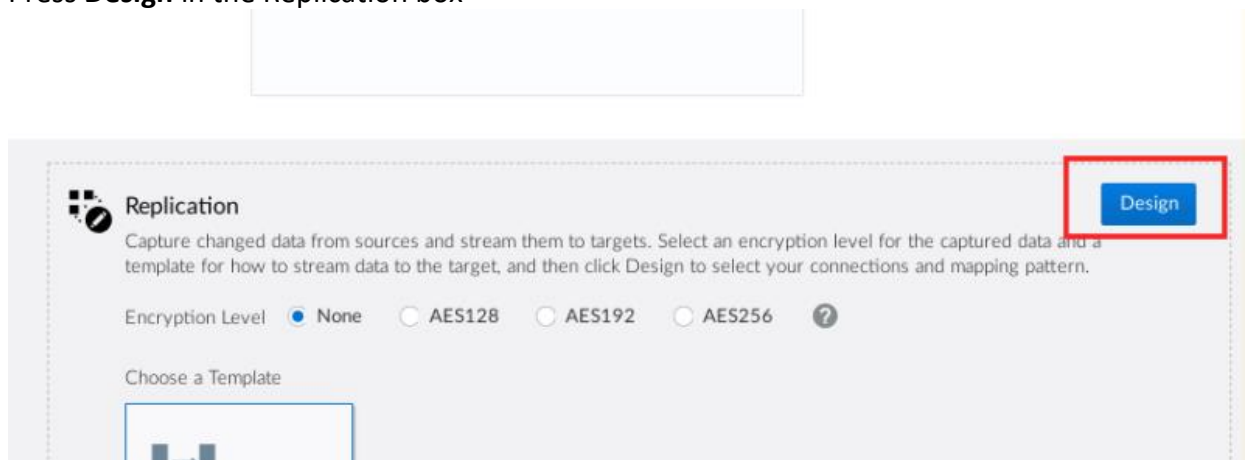
Name \*  Enter a name with maximum 255 characters

Identifier \*

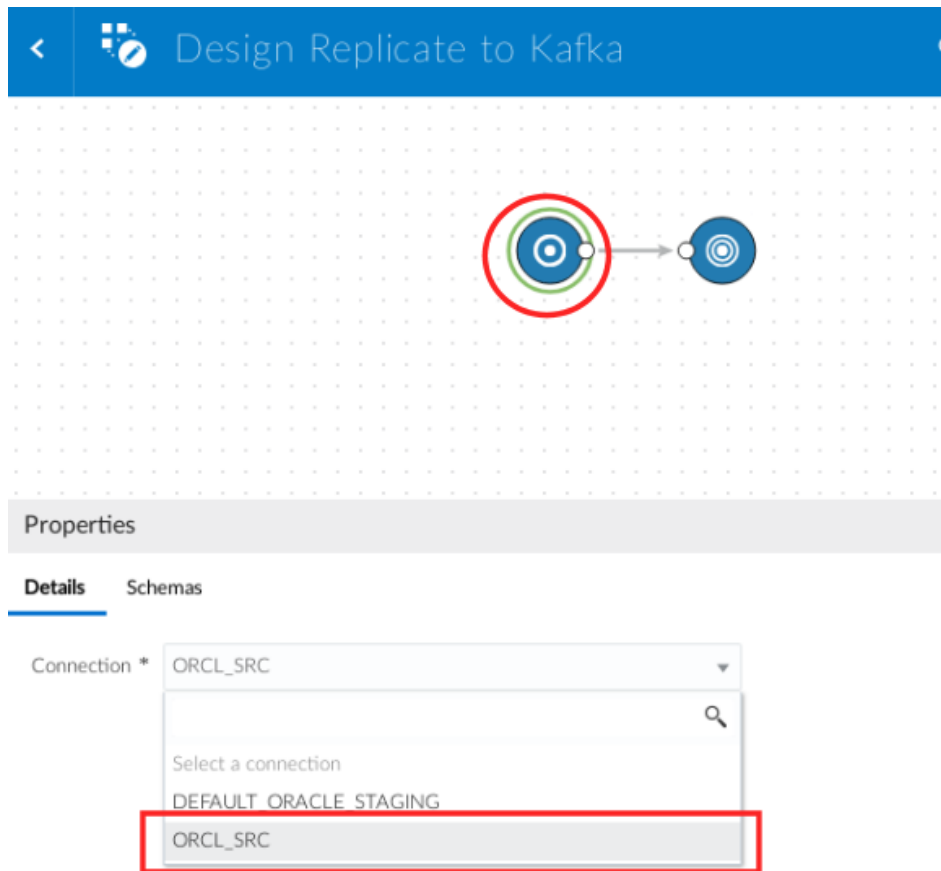
Description



9. Press **Design** in the Replication box



10. In the diagram screen select the source component and choose the **ORCL\_SRC** connection.



11. Review the Schemas tab of the source. The OSADATA schema is pre-selected on the right, we will keep this setting.

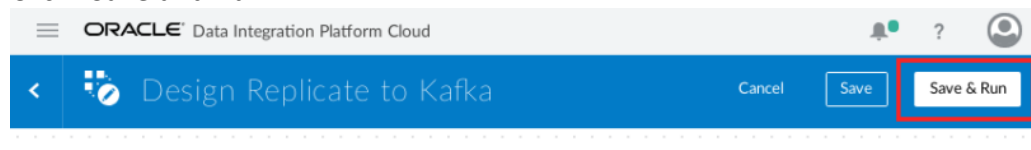
Properties			
Details		Schemas	
OSADATA		Rule <a href="#">Reset</a>	
#	Schema (63)	#	Rule Applied
1	OSADATA	1	Include OSADATA.*
2	ANONYMOUS		
3	APPQOSSYS		
4	AUDSYS		
5	C##GGSRC		
6	CTXSYS		
7	DBSFUSER		
8	DBSNMP		
9	DEV_DICLOUD_DATA_DQ		
10	DEV_DICLOUD_JMS		



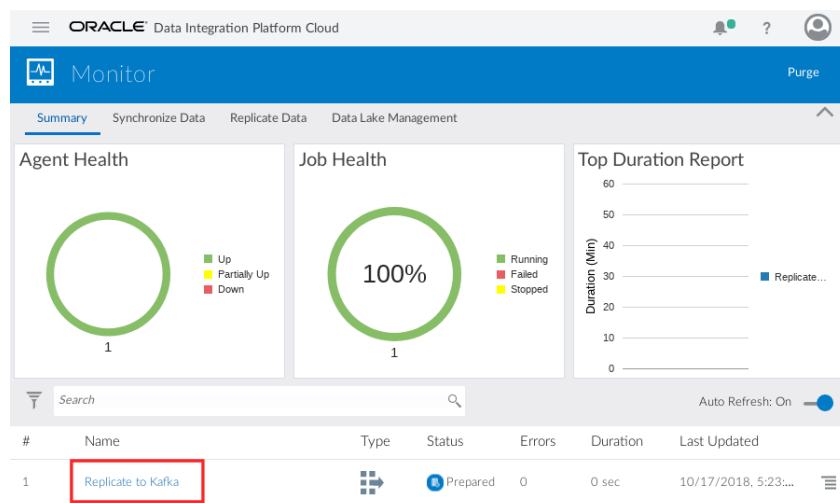
12. Select the target component in the diagram and select Kafka\_conn



13. Click “Save and Run”

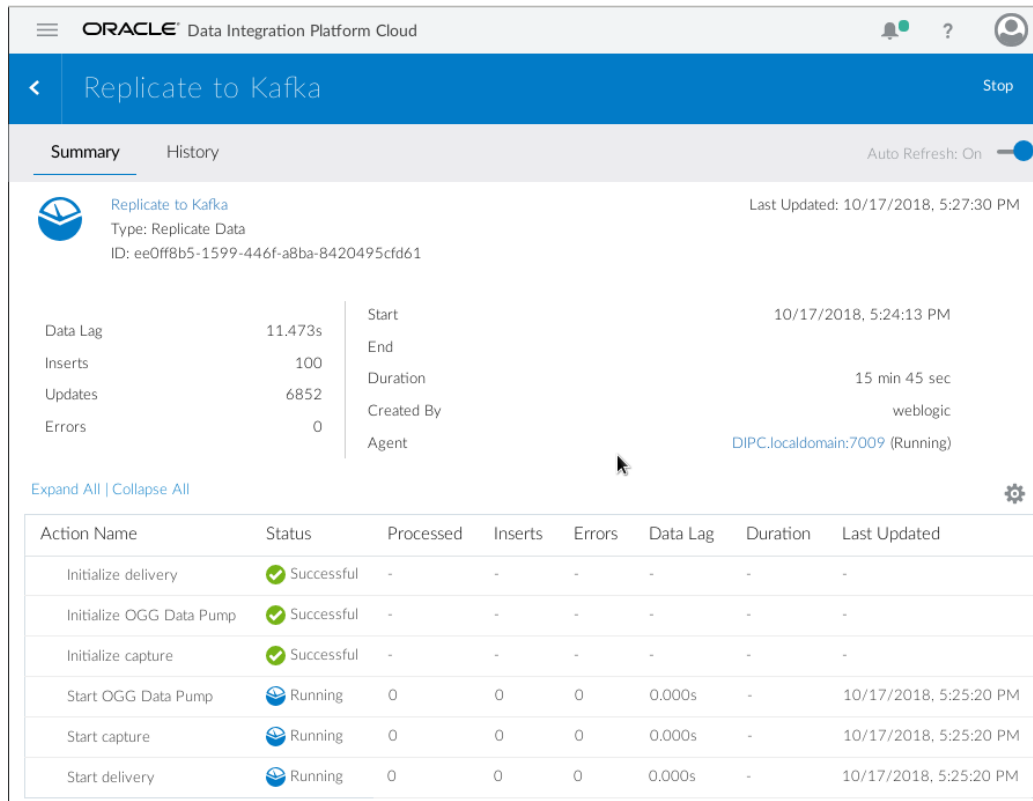


14. This automatically opens the Monitor page to watch the progress of the job.



Click on Replicate to Kafka in the task list to open details for this task.

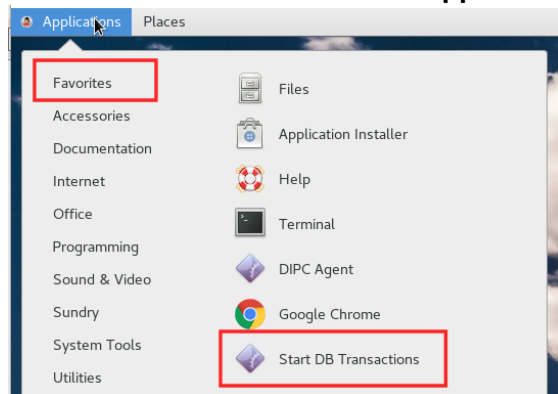
15. Wait for all Actions within the Replicate to Kafka task to show as running. This will take about 4 minutes to complete.



The screenshot shows the Oracle Data Integration Platform Cloud interface. The main heading is 'Replicate to Kafka' with a 'Stop' button. Below this, there are tabs for 'Summary' and 'History'. The 'Summary' tab is active, showing a 'Replicate to Kafka' task with a 'Type: Replicate Data' and 'ID: ee0ff8b5-1599-446f-a8ba-8420495cfd61'. The task is 'Running' and was 'Created By: weblogic' on '10/17/2018, 5:27:30 PM'. The 'Agent' is 'DIPC.localdomain:7009 (Running)'. The 'Data Lag' is '11.473s'. The 'Inserts' are '100', 'Updates' are '6852', and 'Errors' are '0'. The 'Duration' is '15 min 45 sec'. The 'Start' time is '10/17/2018, 5:24:13 PM'. Below this, there is a table with columns: Action Name, Status, Processed, Inserts, Errors, Data Lag, Duration, and Last Updated. The table contains the following data:

Action Name	Status	Processed	Inserts	Errors	Data Lag	Duration	Last Updated
Initialize delivery	Successful	-	-	-	-	-	-
Initialize OGG Data Pump	Successful	-	-	-	-	-	-
Initialize capture	Successful	-	-	-	-	-	-
Start OGG Data Pump	Running	0	0	0	0.000s	-	10/17/2018, 5:25:20 PM
Start capture	Running	0	0	0	0.000s	-	10/17/2018, 5:25:20 PM
Start delivery	Running	0	0	0	0.000s	-	10/17/2018, 5:25:20 PM

16. For simulating new transactions to the source Oracle database, you need to run **Start DB Transactions** from the **Applications > Favorites** menu:



17. A terminal window is opening showing the DB transactions. The transactions simulate location information of customers from mobile apps, this will be used later for the stream analytics demo. Please keep the window open for the remainder of the lab.

```

Terminal
File Edit View Search Terminal Help
Starting DB Transactions ...
drop table osadata.custlocation
create table osadata.custlocation (custid VARCHAR(200) PRIMARY KEY,prodid VARCHAR(200),lat NUMBER(20,10),lng NUMBER(20,10),tmstamp VARCHAR(200))
insert into osadata.custlocation values ('1','100',37.32612024000054,-121.94990541593428,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('2','100',37.32557720176031,-121.9492941130155,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('3','100',37.32470691313216,-121.94842512260713,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('4','400',37.327479486488976,-121.94556773261145,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('5','450',37.32070553347025,-121.94709894273106,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('6','500',37.32927192324816,-121.9446378679472,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('7','300',37.325017423321334,-121.94419756737982,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('8','450',37.31964917649496,-121.95023614807512,'2018/08/07 17:50:10')
insert into osadata.custlocation values ('9','300',37.32078309897001,-121.93836047453568,'2018/08/07 17:50:10')

```

18. The new database transactions will be to be captured by the Replication Task and replicated to Kafka. To observe the status, look at the the **Replicate To Kafka** task in DIPC web console. You will be able to see the progress of processed transactions in the **Start OGG Data Pump, Start capture** and **Start delivery** actions:

ORACLE Data Integration Platform Cloud

Replicate to Kafka Stop

Summary History Auto Refresh: On

Replicate to Kafka Last Updated: 10/17/2018, 5:27:30 PM

Type: Replicate Data  
ID: ee0ff8b5-1599-446f-a8ba-8420495cfd61

Data Lag	11.473s	Start	10/17/2018, 5:24:13 PM
Inserts	100	End	
Updates	6852	Duration	15 min 45 sec
Errors	0	Created By	weblogic
		Agent	DIPC.localdomain:7009 (Running)

[Expand All](#) | [Collapse All](#)

Action Name	Status	Processed	Inserts	Errors	Data Lag	Duration	Last Updated
Initialize delivery	Successful	-	-	-	-	-	-
Initialize OGG Data Pump	Successful	-	-	-	-	-	-
Initialize capture	Successful	-	-	-	-	-	-
Start OGG Data Pump	Running	6988	100	0	3.130s	-	10/17/2018, 5:39:51 PM
Start capture	Running	7008	100	0	1.483s	-	10/17/2018, 5:39:51 PM
Start delivery	Running	6952	100	0	6.860s	-	10/17/2018, 5:39:50 PM

## Task 4: Oracle Stream Analytics from Kafka

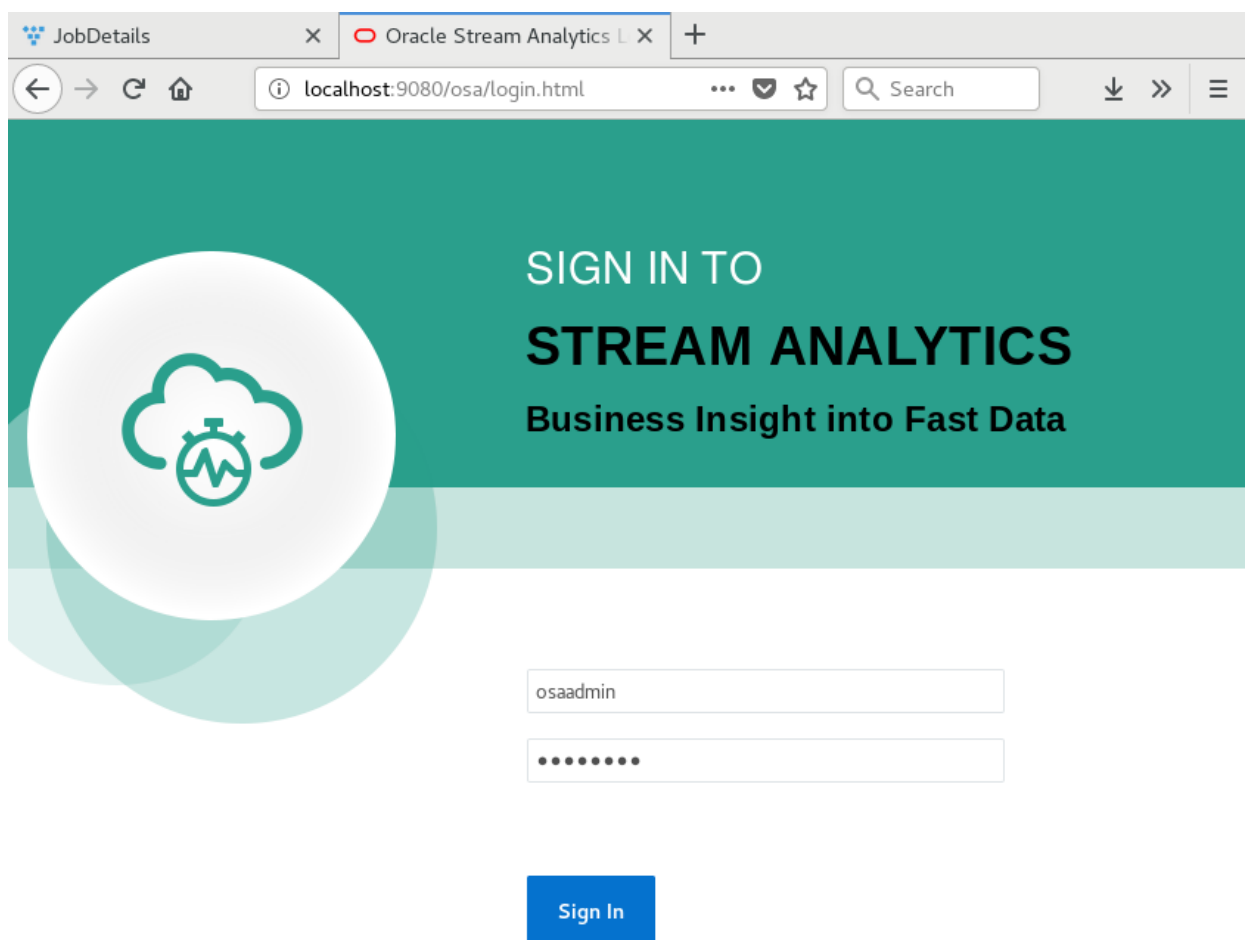
This task will take you through the design of a stream analytics pipeline for analysing the customer location stream provided by the Replication Task created earlier.

1. Open the Stream Analytics Console in the browser

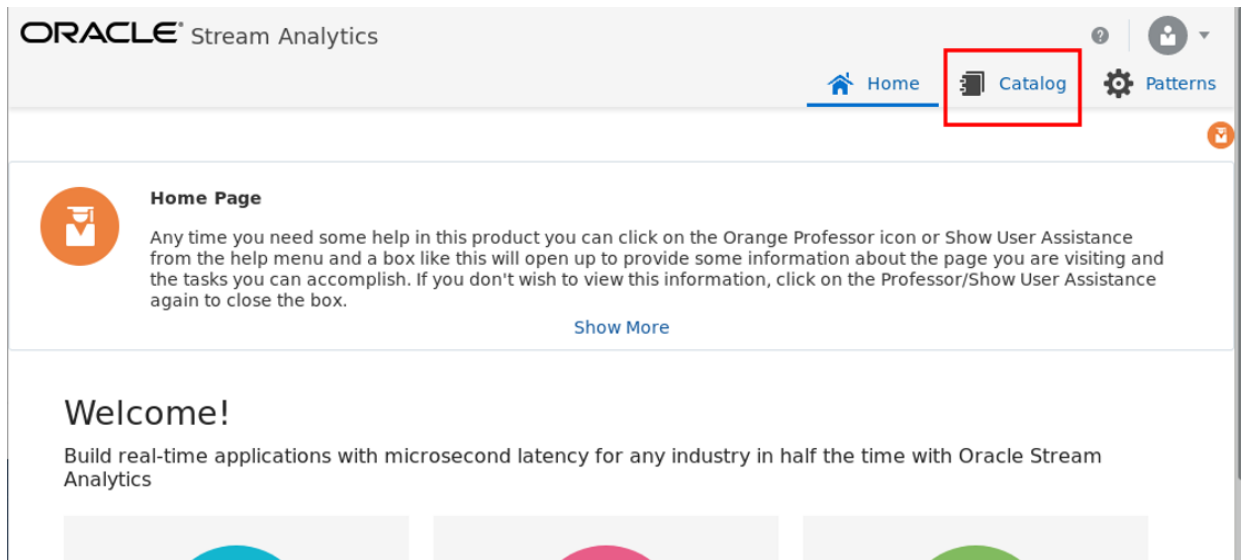
URL: <http://localhost:9080/osa>

User Name: osaadmin

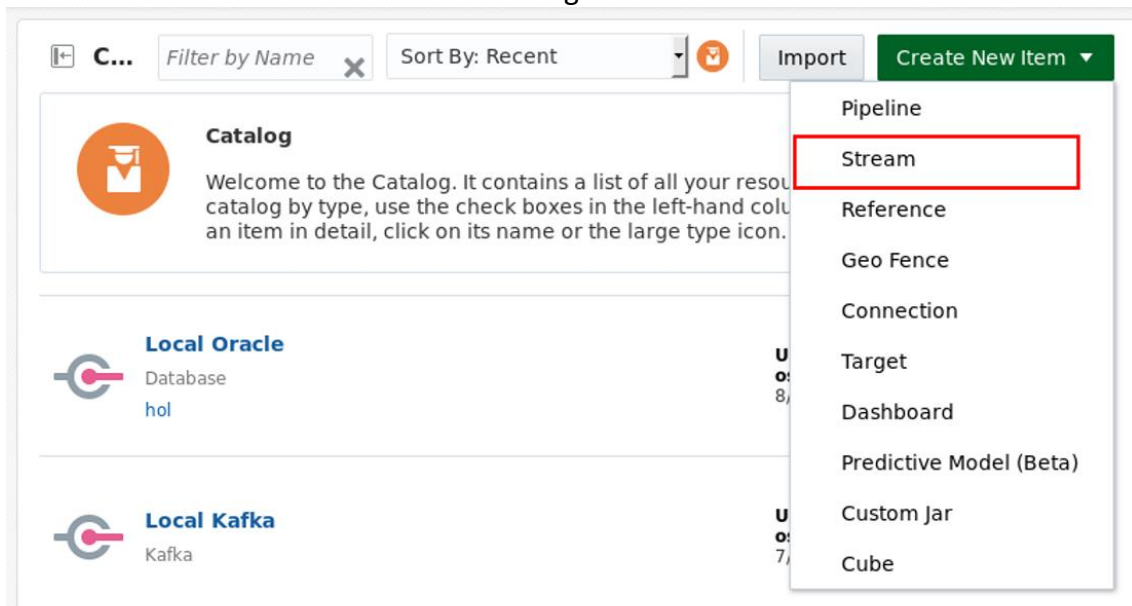
Password: welcome1



1. Select **Catalog** in the upper right menu



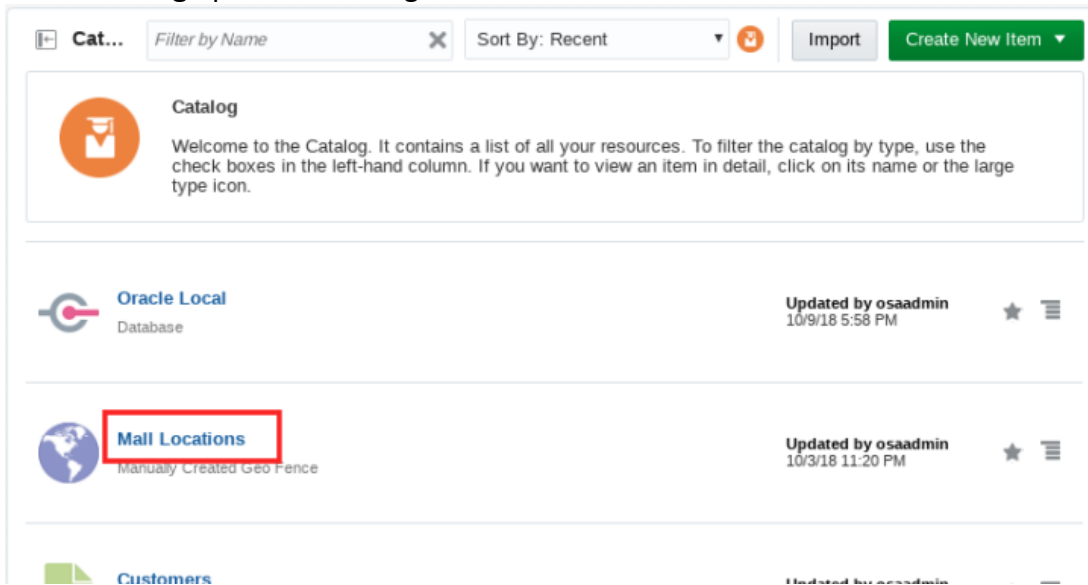
## 2. Select **Create New Item > Stream** in Catalog



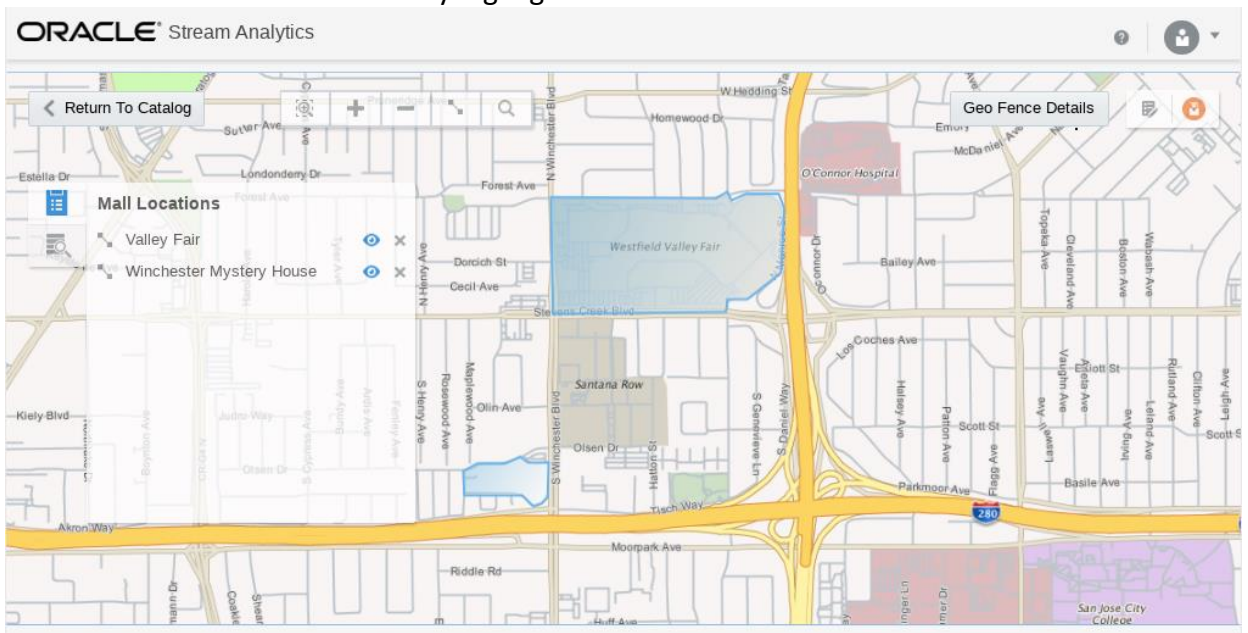
## 3. Enter the following data in the **New Stream** dialog, pressing **Next** on every page and **Save** at the end.


Type Properties	Source Details	Data Format	Shape
Name: <b>CustLocations</b> Stream Type: <b>Kafka</b>	Connection: <b>Local Kafka</b> Topic Name: <b>ORCLPDB1.OSADATA.CUSTLOCATION</b>	Leave all defaults	Leave all defaults, wait for <b>Infer Successful</b> to be shown

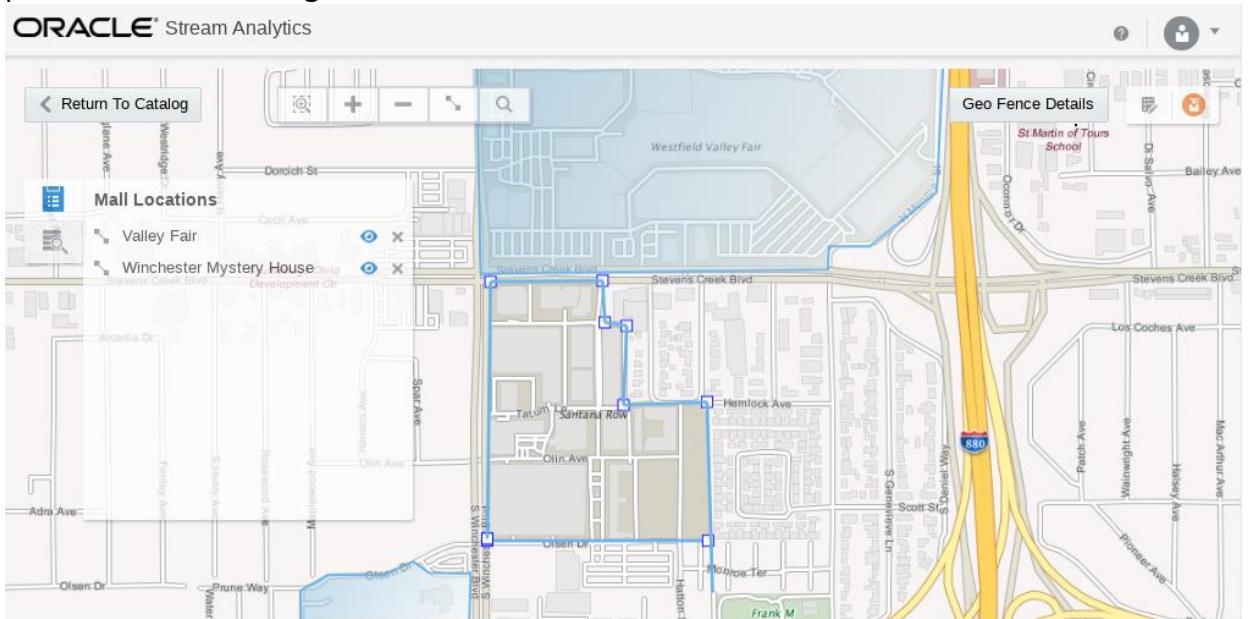
4. In the **Catalog** open the existing Geo Fence named **Mall Locations**



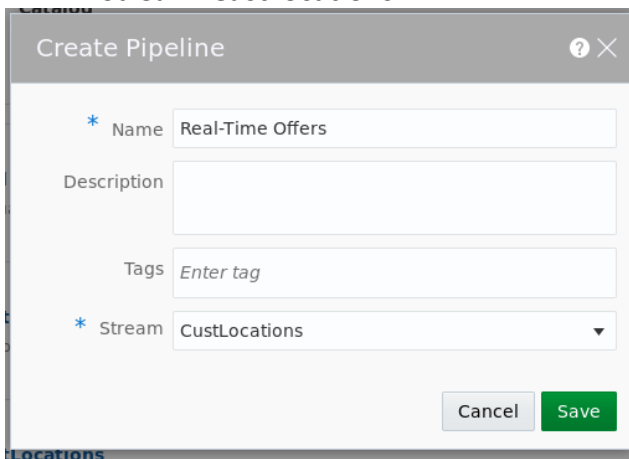
5. The Geo Fence editor opens with a partial Geo Fence for local malls in San Jose. You can see two areas that are already highlighted in blue.



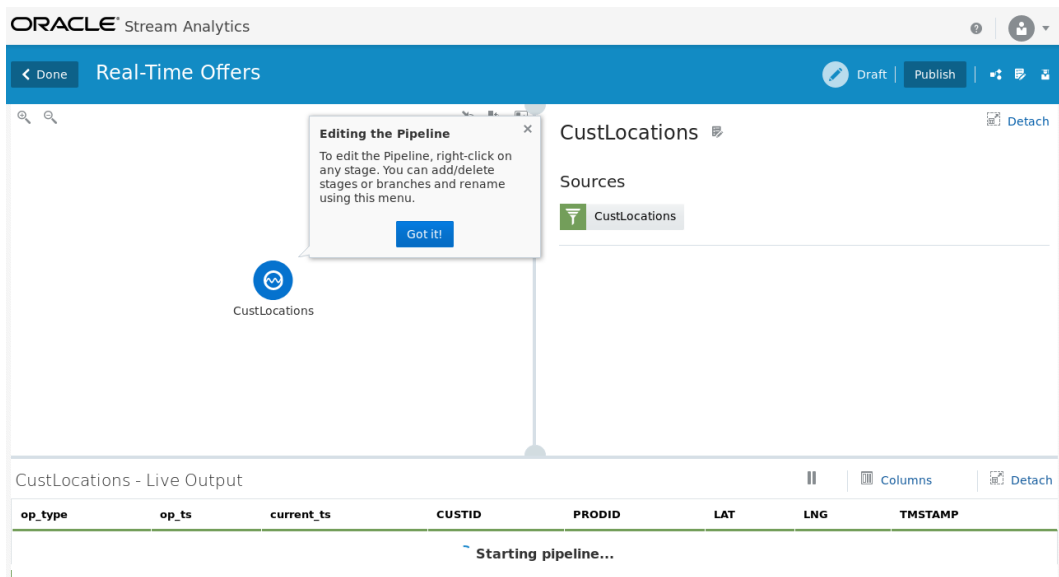
6. Press the Fence Icon  and drag an outline of another mall, until the fence is closed. You can use the + and – buttons to zoom in/out. It doesn't have to be exact, and you can repeat this with multiple areas on the screen. This step can also be skipped. Then press **Return To Catalog**.



7. Select **Create New Item > Pipeline**
8. Enter the following data:
  - Name: **Real Time Offers**
  - Stream: **CustLocations**



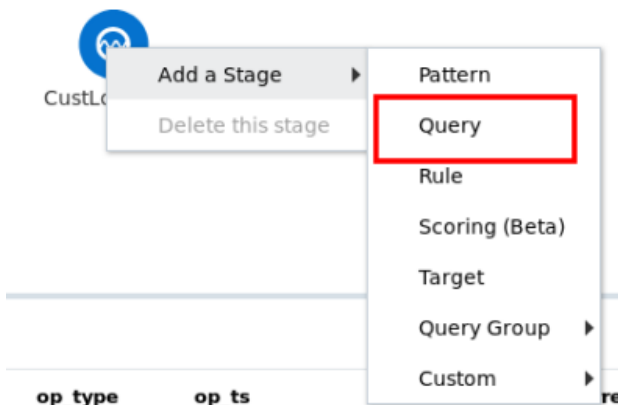
9. The editor appears and the pipeline is started in the background. Starting the pipeline might take **up to 60 sec.**



10. After the pipeline is started, live output of the stream should be visible on the bottom screen:

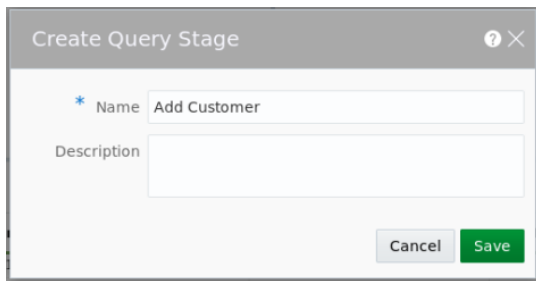
op_type	op_ts	current_ts	CUSTID	PRODID	LAT	LNG	TMESTAMP
U	8/14/18 10:54:21 PM	8/14/18 10:59:22 PM	79	100	37.3224446702	-121.9530087506	2018/08/07 17:50:10
U	8/14/18 10:54:21 PM	8/14/18 10:59:22 PM	78	200	37.3218715301	-121.947087715	2018/08/07 17:50:10
U	8/14/18 10:54:21 PM	8/14/18 10:59:21 PM	77	100	37.3198094817	-121.9427449716	2018/08/07 17:50:10
U	8/14/18 10:54:21 PM	8/14/18 10:59:21 PM	76	500	37.3265699809	-121.9542114311	2018/08/07 17:50:10
U	8/14/18 10:54:21 PM	8/14/18 10:59:21 PM	75	500	37.3269512031	-121.9504204014	2018/08/07 17:50:10
U	8/14/18 10:54:20 PM	8/14/18 10:59:21 PM	74	300	37.3250900504	-121.9471617449	2018/08/07 17:50:10
U	8/14/18 10:54:20 PM	8/14/18 10:59:21 PM	73	200	37.32182984	-121.944486759	2018/08/07 17:50:10
U	8/14/18 10:54:20 PM	8/14/18 10:59:21 PM	72	300	37.3220298981	-121.9438845523	2018/08/07 17:50:10

11. Right-click on **CustLocations** to create a Query Stage in the Pipeline



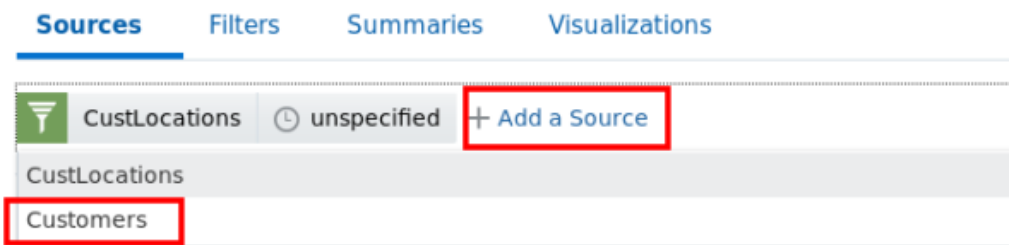
12. Fill in the Create Query Stage dialog and use the name “**Add Customer**” and press **Save**.





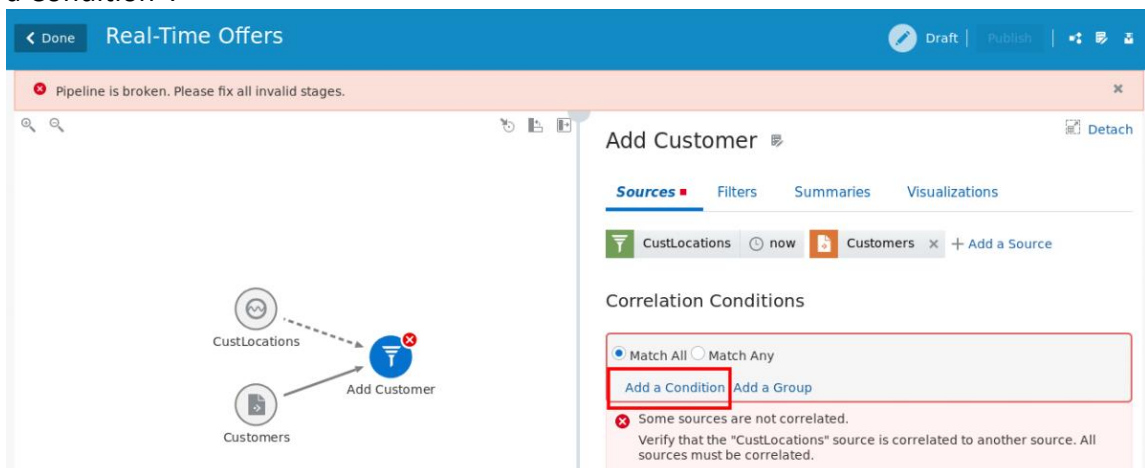
A dialog box titled "Create Query Stage" with a close button (X) in the top right corner. It contains two input fields: "Name" with the text "Add Customer" and "Description" which is empty. At the bottom, there are two buttons: "Cancel" and "Save".

13. In the Add Customer Properties pane press “Add a Source” and select Customers  
**Add Customer**



The "Add Customer" pane with tabs for Sources, Filters, Summaries, and Visualizations. The Sources tab is active, showing a funnel icon, "CustLocations", a clock icon, "unspecified", and a red-bordered "+ Add a Source" button. Below this, a list shows "CustLocations" and "Customers", with "Customers" highlighted by a red box.

14. A validation error is shown because no correlation condition has been set. Press “Add a Condition”.



The "Real-Time Offers" pipeline editor interface. At the top, a blue bar contains "< Done", "Real-Time Offers", "Draft", "Publish", and icons for settings, share, and help. Below this, a red error banner states: "Pipeline is broken. Please fix all invalid stages." The main workspace shows a diagram with "CustLocations" and "Customers" sources pointing to an "Add Customer" stage, which has a red error icon. On the right, the "Add Customer" pane is open, showing the Sources tab with "CustLocations" (funnel icon), "now" (clock icon), "Customers" (document icon), and "+ Add a Source". Below the sources, the "Correlation Conditions" section has "Match All" selected, and a red-bordered "Add a Condition" button. A red error message at the bottom of the pane reads: "Some sources are not correlated. Verify that the 'CustLocations' source is correlated to another source. All sources must be correlated."

15. Select the following fields and operators:



The "Correlation Conditions" configuration area. It starts with "Match All" selected. Below, there are four components in a row: an orange box with "CUSTID\_1" and a dropdown arrow, a white box with "equals (case sensitive)" and a dropdown arrow, a green box with "CUSTID" and a dropdown arrow, and a green box with "CustLocations" and a gear icon. To the right of these is a red "X" icon. At the bottom, there are links for "Add a Condition" and "Add a Group".

After selecting the last field, click on an empty space next to the condition to apply the changes.

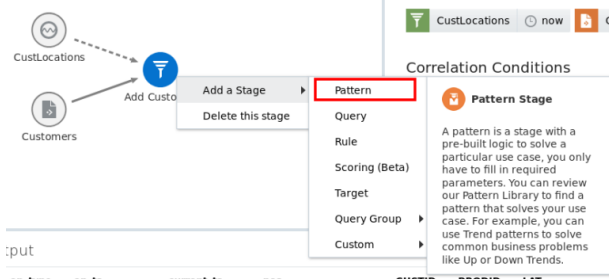
16. The Live Output now shows streaming data enriched by a join with a database table:

Add Customer - Live Output

op_type	op_ts	current_ts	CUSTID	PRODID	LAT	LNG	TMSTAMP	CUSTID_1	NAME	STREET	CITY	FCUSTID
U	8/14/18 11:03:12 PM	8/14/18 11:04:57 PM	64	450	37.3175299201	-121.940589061	2018/08/07 17:50:20	64	Sansa Danza	4547 Market Dr	Clyde	na
U	8/14/18 11:03:12 PM	8/14/18 11:04:57 PM	63	200	37.3302118131	-121.9449346808	2018/08/07 17:50:20	63	Ellie Stone	4175 Leigh Str	Tampa	na
U	8/14/18 11:03:12 PM	8/14/18 11:04:57 PM	62	450	37.326788551	-121.9445806664	2018/08/07 17:50:20	62	Jill Bara	2503 Washington Str	Blythewood	na
U	8/14/18 11:03:12 PM	8/14/18 11:04:57 PM	61	200	37.3230458207	-121.9381193051	2018/08/07 17:50:20	61	Arya Solo	3752 Ash S	Delta Junction	na

■ CustLocations ■ Customers

17. We now want to limit customers who are entering the Mall Location. Right-click on Add Customer to add a Pattern stage.



18. In the Select a Pattern to Create page select Spatial > Geo Fence

Select a Pattern to Create

**Geo Fence**

Use this pattern to analyze streams containing geolocation data. Determine how events relate to pre-defined geo-fences in your maps.

**Category**

- Trend
- Spatial**
- State
- Enrichment
- Missing Event
- Filter
- Outlier
- Shape Detector
- Finance
- Inclusion
- Statistical

**Pattern**

- Geo Fence**
- Spatial: Point to Polygon
- Proximity: Stream with Geofence
- Reverse Geo Code: Near By
- Geo Filter
- Proximity: Single Stream
- Interaction: Two Stream
- Spatial: Speed
- Interaction: Single Stream
- Proximity: Two Stream
- Reverse Geo Code: Near By Place
- Geo Code
- Direction

19. Enter the following data:

Name: **In Mall**

Create Pattern Stage

\* Name: In Mall

Description:

Cancel Save

20. Fill in the following values:

In Mall 

 Detach



Geo Fence

**Parameters** Visualizations

\* Geo Fence



\* Latitude



\* Longitude

\* Object Key   

\* Tracking Events ☐ Near ☒ Enter ☐ Exit ☐ Stay

\* Coordinate System

Distance Buffer   

Stay Duration   

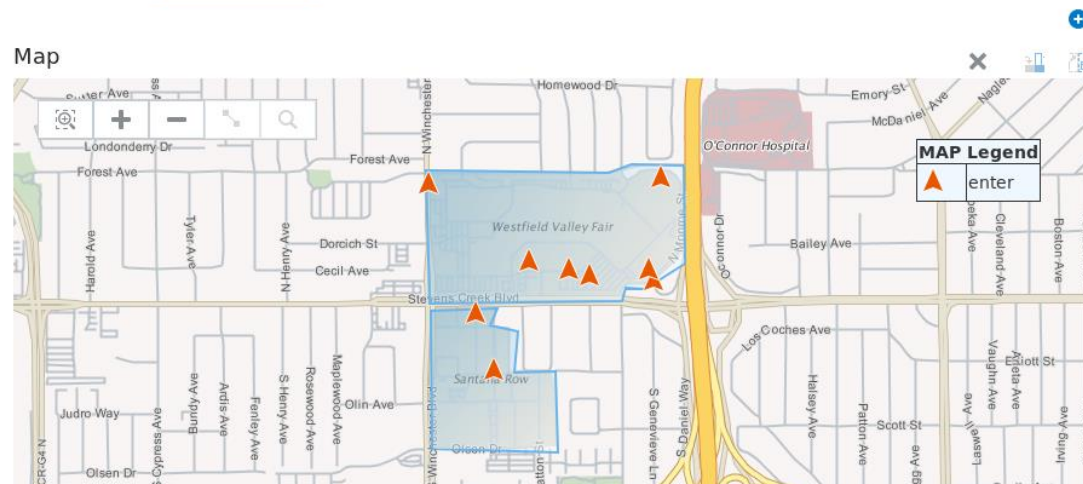
21. Select the Visualizations tab. After waiting for **about 30 sec** until all changes are applied, you will see customers who have entered the mall area.

In Mall 

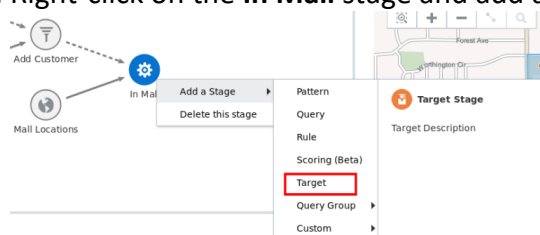
 Detach

Geo Fence

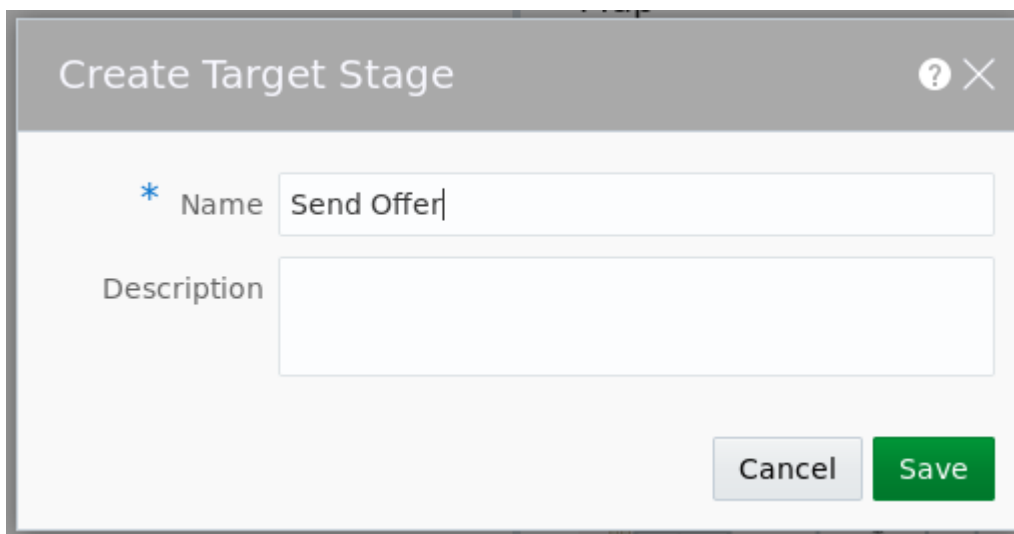
Parameters **Visualizations**



22. Right-click on the **In Mall** stage and add a Target Stage



23. Name the new Target Stage as **Send Offer** and press **Save**.



The 'Create Target Stage' dialog box has a title bar with a question mark and a close button. It contains two input fields: 'Name' with the value 'Send Offer' and 'Description' which is empty. At the bottom right, there are 'Cancel' and 'Save' buttons.

24. In the **Send Offer > Target Mapping** screen press **Create**.

Send Offer

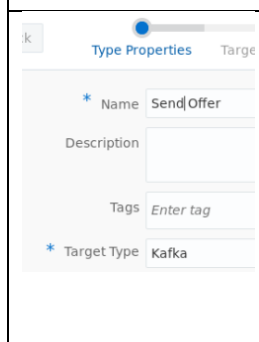
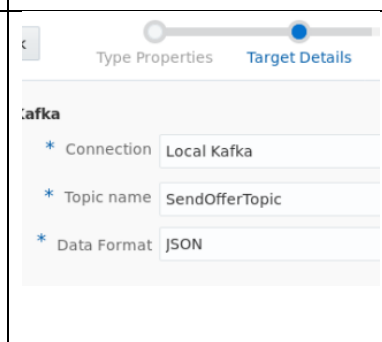
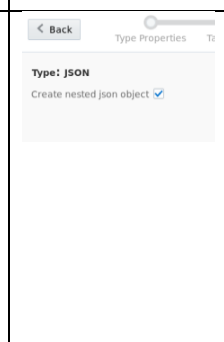
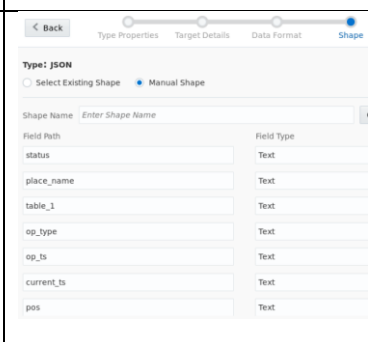
Target Mapping

Target No Target

Create

Edit

25. Enter the following data in the **Create Target** dialog, pressing **Next** on every page and **Save** at the end.

Type Properties	Target Details	Data Format	Shape
Name: <b>Send Offer</b> Target Type: <b>Kafka</b>	Connection: <b>Local Kafka</b> Topic Name: <b>SendOfferTopic</b>	Leave all defaults	Leave all defaults
			

26. Press **Publish** in the header bar to publish the pipeline

The screenshot shows the Oracle Stream Analytics interface for a pipeline named 'Send Offer'. The pipeline diagram on the left shows data flowing from 'CustLocations' and 'Customers' through an 'Add Customer' operation, then through 'Mail Locations' and an 'In Mail' operation, finally reaching the 'Send Offer' operation. The 'Send Offer' operation is configured with a 'Target Mapping' table that maps input properties to output stream properties.

Target Property	Output Stream Property
status	status
place_name	place_name
op_type	op_type
op_ts	op_ts
current_ts	current_ts
CUSTID	CUSTID

Below the configuration, the 'Send Offer - Live Output' table displays real-time data:

status	place_name	op_type	op_ts	current_ts	CUSTID	PROID	LAT	LNG	TMSTAMP	CUSTID_1	NAME	STREET	CITY	FCUSTID	STATE	AVGDISCOUNT	WEBSALES	STORESALES
ENTER	New Geo-fence	U	8/14/18 11:13:10 PM	8/14/18 11:2 9:52 PM	37	300	37.3260957631	-121.9501711132	2018/08/07 17:50:11	37	Jim Zeus	1430 Clinton Dr	Kapowsin	null	WA	7.0	1070.0	3579
ENTER	New Geo-fence	U	8/14/18 11:13:09 PM	8/14/18 11:2 9:01 PM	27	300	37.3223684039	-121.9484444088	2018/08/07 17:50:11	27	Don Wong	1672 Darwin Rd	Knightdale	null	NC	10.0	203.0	380
ENTER	New Geo-fence	U	8/14/18 11:13:08 PM	8/14/18 11:2 9:01 PM	25	300	37.3199806323	-121.9494702215	2018/08/07 17:50:11	25	Jill Danza	1099 Lincoln Rd	Christiansburg	null	VA	12.0	2751.0	1177
ENTER	New Geo-fence	U	8/14/18 11:13:08 PM	8/14/18 11:2 9:01 PM	24	400	37.3253920251	-121.9416228854	2018/08/07 17:50:11	24	Jill Stone	3204 Lincoln Str	Cameron	null	NY	34.0	2566.0	4450

27. In the Publish dialog keep all defaults and press **Publish**

The screenshot shows the 'Publish' dialog box with the 'Pipeline Settings' tab selected. The settings are as follows:

- Batch Duration: 1000 (Milliseconds)
- Executor Count: 1
- Cores per Executor: 2
- Executor Memory: 1800 (Megabytes)
- Cores per Driver: 1
- Driver Memory: 1500 (Megabytes)
- High Availability: ☐ Enabled

Buttons at the bottom include 'Cancel' and 'Publish'.

## Summary

You have now successfully completed the Hands on Lab, and have successfully performed an end-to-end real-time data ingest to Kafka from Oracle database using Data Integration Platform Cloud. You have also completed how to analyze the data in real time using Oracle Stream Analytics with simple steps.