Linking an Amazon Oracle RDS source database to the Delphix Dynamic Data Platform

Applicable Delphix Versions

<table>
<thead>
<tr>
<th>Major Release</th>
<th>All Sub Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2</td>
<td>5.2.2.0, 5.2.2.1, 5.2.3.0</td>
</tr>
<tr>
<td>5.1</td>
<td>5.1.0.0, 5.1.1.0, 5.1.2.0, 5.1.3.0, 5.1.4.0, 5.1.5.0, 5.1.5.1, 5.1.6.0, 5.1.7.0, 5.1.8.0, 5.1.8.1, 5.1.8.2</td>
</tr>
</tbody>
</table>

Linking Oracle RDS source databases with Delphix

Background

It is possible to link Amazon AWS Oracle RDS instances to Delphix as a source of data for virtual databases provisioned from the Delphix Dynamic Data Platform. The process for performing this however is not as straight forward as linking a non-AWS RDS database to Delphix. The reason for this is Delphix requires rman to perform the ingestion of a regular dSource and Amazon do not provide shell access or the ability to spawn RMAN processes at the operating system level with Amazon RDS instances. The only access provided by Amazon to AWS Oracle RDS instances is through SQLNet. As such using Oracle RDS databases as a source for provisioning Delphix VDB’s requires the use of Amazon Data Migration Services (DMS). A DMS instance is used to replicate the data out of the RDS instance to a non-RDS based Oracle database.

The Process Overview

Linking an Amazon Oracle RDS source database to a Delphix Engine can effectively be broken down into the following basic steps.

- Create the Amazon Oracle RDS source database using the AWS website. This RDS based database will be the source of data and source that Amazon Data Migrations Services will be linked to.
- Create a temporary Oracle dSource and add this Oracle database to Delphix as a dSource.
- From this temporary dSource create an Oracle VDB. This VDB will become the target database for Amazon Data Migration Services and the source of all data that subsequent VDB’s will be provisioned from.
- Establish an Amazon Data Migration Services instance
- Create Source and Target Endpoint for the Amazon DMS instance to use. The Source endpoint will be the AWS Oracle RDS instance and the target endpoint will be the staging VDB established through Delphix.
- Create an Amazon DMS task to perform the replication of data between the source and target endpoint.
The Environment used in this example

- An AWS Oracle RDS instance (rds1) running 12.1 Enterprise Edition of Oracle
- An AWS Data Migration Services instance running a 2.4.1 AWS DMS instance
- An on-premise Delphix Engine running 5.2.2.0 of the Delphix Dynamic Data Platform
- A temporary Oracle dSource (rdstemp) running the 12.1 Enterprise Edition of Oracle created on an on-premise linux host
- A Delphix Virtual Database (vrdstg1) created from the temporary dSource running from an on-premise linux host

An overview of the core components of the configuration

The Complete Process from Start to Finish

Create the Amazon Oracle RDS Instance

Log in to AWS

Logging into AWS you are presented with the AWS service list. For the purpose of this exercise we are only interested in Relational Database Services and Migration, Database Migration Services.
Select Relation Database Service

Database

Relational Database Service
DynamoDB
ElastiCache
Amazon Redshift

From Create Instance choose Launch a DB Instance

Create instance

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Restore from S3  Launch a DB Instance

Note: your DB instances will launch in the US West (Oregon) region
In this case as its only a test environment the **Dev/Test Use** case is selected.

Configure the AWS RDS Oracle Instance setting the licence model and version of Oracle you want to run.

The class of instance in this case is the smallest available give its a test environment.

Set the AWS unique name for this database and the master username and password. This user could be likened to the
SYS user in Oracle although it is not exactly the same.

Configure the RDS instances networking environment (Virtual Private Cloud).

As the instance is to be accessed from my on premise environment it was opened up for public access.

Set the database name and the listener port incoming SQLNet connections will use to attach to this Oracle RDS database.
The summary screen will appear indicating that the AWS is in the process of creating the Oracle RDS instance rds1.
The Oracle RDS Instance will become available once it has been started and backed up for the first time.

Select the instance to expose the configuration details including the connection endpoint, security groups and availability zone information.

Test a connection to the new Oracle RDS instance using sqlplus to ensure it can be connected to remotely.

Connecting to the RDS Instance requires the construction of a SQLNet connect string using the endpoint information from the RDS Instance settings. The following example details the format required to perform the remote SQLNet connection to the AWS Oracle RDS instance.

```
[oracle@oel7si2 ~]$ sqlplus 'delphixrds/
password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(NOCACHE)(SOURCE=rds1.cpomexylatjw.us-west-2.rds.amazonaws.com)\(PORT=1521\))(CONNECT_DATA=(SID=rds1)))'
SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 7 20:24:29 2018
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL>
```

**AWS Oracle RDS Source Database User Creation**

Amazon Data Migration Services require a user be created and granted specific privileges in this database so that change data capture can be performed in the source database.

Creating a user for AWS Data Migration Services to use to connect to the source Oracle RDS instance is performed using the following process:

- Log into the Oracle RDS source database using the Oracle RDS administrator account, in this case "delphixrds".
Create the new Oracle RDS source database user, "AWSADMIN"

```
SQL> create user AWSADMIN identified by delphix123;
SQL> grant connect,resource to AWSADMIN;
SQL> grant unlimited tablespace to AWSADMIN;
```

Grant the following privileges to the new Oracle RDS user "AWSADMIN". Note the SQL prompts have been omitted so cut and paste can be used to create a script containing these grants.

These privileges have been extracted from the AWS documentation in regards to what is required. The following link contains the instructions from which this privilege list was gathered.

AWS Data Migration Services User Requirements for a Source Oracle RDS instance

```sql
grant select any table to AWSADMIN;
grant select any view to AWSADMIN;
grant select any transaction to AWSADMIN;
exec rdsadmin.rdsadmin_util.grant_sys_object('V$_ARCHIVED_LOG','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_LOG','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_LOGFILE','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_DATABASE','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_THREAD','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_PARAMETER','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_NLS_PARAMETERS','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_TRANSACTION','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_TIMEZONE_NAMES','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_INDICATIONS','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_TABLEPARTITIONS','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('OBJ$','AWSADMIN','SELECT');
grant select on all catalog to AWSADMIN;
grant select on all_cons_columns to AWSADMIN;
grant select on all_cons_columns to AWSADMIN;
grant select on all_objects to AWSADMIN;
grant select on all_indexes to AWSADMIN;
grant select on all_tables to AWSADMIN;
grant select on all_users to AWSADMIN;
grant select on all_constraints to AWSADMIN;
grant select on all_cons_columns to AWSADMIN;
grant select on all_tab_cols to AWSADMIN;
grant select on all_ind_columns to AWSADMIN;
grant select on all_log_groups to AWSADMIN;
exect rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('OBJ$','AWSADMIN','SELECT');
grant select on dba_tablespaces to AWSADMIN;
grant select on all_table_partitions to AWSADMIN;
exect rdsadmin.rdsadmin_util.grant_sys_object('ALL_ENCRYPTED_COLUMNS','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_LOGMNR_LOGS','AWSADMIN','SELECT');
exect rdsadmin.rdsadmin_util.grant_sys_object('V$_LOGMNR_CONTENTS','AWSADMIN','SELECT');
```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOGMNR','AWSADMIN','EXECUTE');
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');

Create the temporary Oracle dSource that will be linked to the Delphix Engine.

Using the Oracle database configuration assistant create a new unused database.

This new oracle database will become be a temporary dSource from which the VDB RDS staging database will be created.

The temporary dSource will need a delphix database user be created before it can be linked to the Delphix Engine.

This environment and user must be configured in the same manner as any other dSource environment. The Delphix database user must be created using the createDelphixDBUser.sh script downloaded from the Delphix documentation.


Link the temporary Oracle dSource to the Delphix Engine.

Following the process for linking from the Delphix documentation link the temporary dSource to the Delphix Engine.

Detach the temporary Oracle dSource.

A single snapshot of the temporary dSource will have been created during the linking process. Once this database has been ingested by the engine and this snapshot has been generated the temporary dSource can be detached from the Delphix Engine.

Shutdown the temporary Oracle dSource.

As the temporary dSource is not really required from this point on it can be shutdown.

```
[oracle@oel7sl1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 7 11:54:53 2018
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

Create the staging VDB.

Create a VDB called vplbstg from the snapshot of the temporary dSource created in the previous steps.

This VDB will become the AWS DMS target instance and the source for any VDB's created from what is ultimately the Amazon Oracle RDS Source.

Using the process for provisioning VDB's create a new VDB.
Configure the staging VDB for AWS Data Migration Services use.

Configure the staging VDB according to the DMS target requirements set out in the AWS documentation:


```
SQL> create user rdsadmin identified by tiger account unlock;
User created.

SQL> grant connect,resource to rdsadmin;
Grant succeeded.

SQL> grant unlimited tablespace to rdsadmin;
Grant succeeded.

SQL> grant select any transaction to rdsadmin;
Grant succeeded.

```

Configure the AWS Data Migration Services replication instance.

This is performed through the AWS Management Console.

Select Replication Instances and Create replication instance and using the wizard establish a new replication instance which will be used to replicate data from the AWS RDS instance to the on-premise staging VDB.
Name the AWS RDS replication instance and set the class and virtual network security group required.

Set out a maintenance schedule should minor upgrades be required.
Configure the AWS Data Migration Services source and target replication endpoints.

Select the RDS instance from the Management Console and create a source (in this case the AWS Oracle RDS instance) and a target (in this case the on-premise staging VDB) for replicating data.

![AWS Data Migration Services](image1)

An example of the AWS RDS instance and source end point details:

![AWS RDS Instance](image2)

An Example of the on-premise staging VDB target end point details. Appropriate network paths and firewall allowances between the on-premise and AWS environments must exist and allow connectivity between the site. Configuring this correctly is up to the sites network administrators, OS administrators and DBA’s.
Testing the endpoints.

The AWS Management Console provides functionality to test the connectivity of each endpoint. Until these tests succeed for both the source and target endpoints AWS DMS replication cannot be performed successfully.

Configure the AWS Data Migration Services replication task.

From the AWS Management console for DMS establish a replication task.

Link the replication instance, the source and target endpoints and establish the replication task type.
Establish the tasks selection rules including the source schema and schema objects it is to pull data from and transfer to the target endpoint.

Start the replication task and perform the replication of the data from the AWS Oracle RDS source instance and the on-premise staging VDB provisioned from the Delphix Engine.

Once the replication task has completed checking the target staging VDB and ensure that the data specified in the replication task has arrived at the VDB.

The staging VDB can now be used as the source of data for other VDB's that may be required by developers, etsters etc on-premise to perform tasks using data from the AWS Oracle RDS instances.
External Links

AWS DMS

https://aws.amazon.com/dms/

Source Endpoints supported

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Introduction.Sources.html

Target Endpoints supported


How DMS Works

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Introduction.html