

DATABASE MIGRATION SERVICE

Migrating Oracle Database from On-Premises/EC2 to RDS



Table of Contents

Introduction	3
Environment Details	3
Approach	3
Starting DMS service	6
Monitoring DMS service	13
Procedures, Functions, Packages and other object migration	14
Post migration activity and checks	15
Conclusion	16

Introduction

Managing Oracle Databases in On-Premises/EC2 instances require more resources and cost. Hence moving it to RDS instance will ease our job, optimizing the IT budget and also providing us with various features like Multi-AZ, Scalability, Automatic Backups...etc.

In our case we are migrating 120 GB Oracle Database from EC2 instance to RDS using the DMS service which migrates the data very faster and effortlessly .The task can also be monitored and optimized through the DMS service. The same approach is applicable while migrating the Oracle Database from On-Premises to RDS.

Environment Details

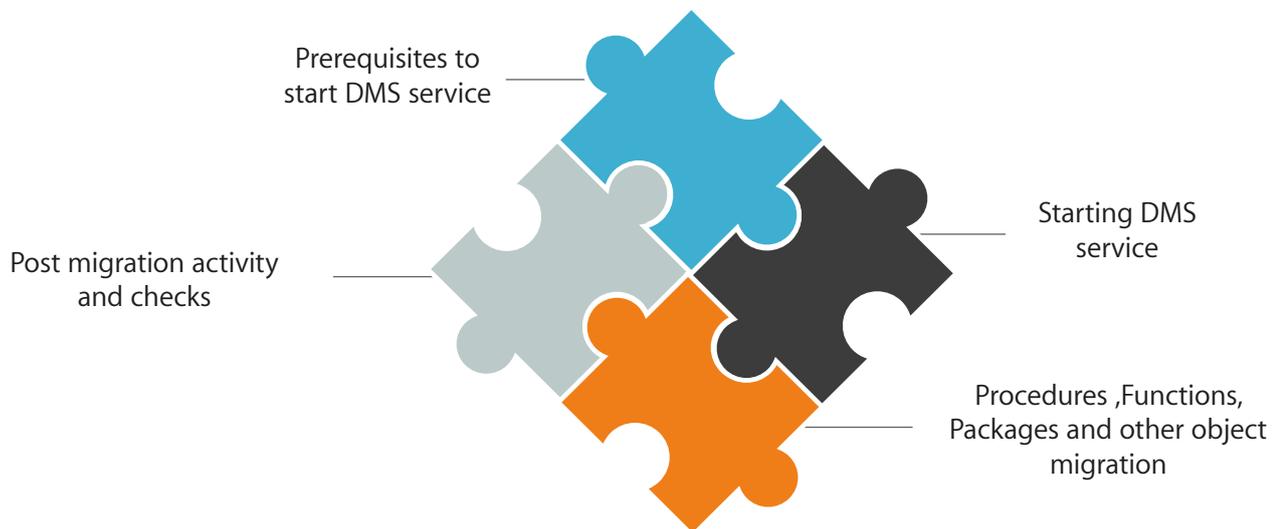
Below are the current EC2 and opted RDS environment details involved in this migration.

EC2 Instance Type	M4.large
Database Size	120 GB
Database Version	11.2.0.4
DB Instance Name	TEST
Schemas to migrate	DOCUMENTS, PROD, HOW,OWS, BACKUP

RDS Instance Type	T2.medium
Allocated Size	200 GB
Database Version	11.2.0.4
RDS Instance Name	PRODTEST

Approach

The below approach is followed to migrate the Database:



Prerequisites to start DMS service:

- Source database with Instance and Listener services need to be up and running.
- Target RDS database should be up and running.
- Target RDS database should have sufficient storage for data migration.
- Source database should be in "archive log" mode.
- Create the necessary tablespaces on the target database.
- Create schemas on the target database which needs to be migrated.

Below screen shots confirms the prerequisites are met in our migration.

a. Source database

The source database instance "TEST" and the "LISTENER" are up and running fine.

```

[oracle@ ~]$ ps -ef|grep pmon
oracle 16416      1  0 03:32 ?        00:00:00 ora_pmon_ TEST
oracle 17183 17157  0 05:21 pts/4    00:00:00 grep pmon
[oracle@ ~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.4.0 - Production on 12-SEP-2016 05:21:20

Copyright (c) 1991, 2013, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=) (PORT=1521))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.4.0 - Production
Start Date                12-SEP-2016 05:19:17
Uptime                    0 days 0 hr. 2 min. 3 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /oradata/app/oracle/product/11.2.0/network/admin/listener.ora
Listener Log File         /oradata/app/oracle/diag/tnslsnr/.../listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=...ec2.internal)(PORT=1521)))
Services Summary...
Service "TEST" has 1 instance(s).
  Instance "TEST", status READY, has 1 handler(s) for this service...
Service "SIKDB" has 1 instance(s).
  Instance "TEST", status READY, has 1 handler(s) for this service...
The command completed successfully
  
```

b. Target database

The target RDS instance is up and ready to be used

```

Oracle EE  prodtest  available  0.42%  0 Connections  None  db.t2.medium
  
```

c. Storage space in target database.

The target database is equipped with enough storage space.

Monitoring							
	CURRENT VALUE	THRESHOLD	LAST HOUR		CURRENT VALUE	LAST HOUR	
CPU	0.92%			Read IOPS	8.32/sec		
Memory	3,000 MB			Write IOPS	2.71/sec		
Storage	198,000 MB			Swap Usage	0 MB		

d. Source database in archive log mode.

The archive log mode is enabled in the source database.

```

SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
TEST         READ WRITE

SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /oradata/app/oracle/product/11.2.0/dbs/arch
Oldest online log sequence     917
Next log sequence to archive   919
Current log sequence           919
SQL>
  
```

e. Tablespace creation in the target database.

Tablespaces "DOCUMENTS","PROD","HOW","OWS" and "BACKUP" are created.

```
SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
PRODTEST     READ WRITE

SQL> CREATE TABLESPACE DOCUMENTS;

Tablespace created.

SQL> CREATE TABLESPACE PROD;

Tablespace created.

SQL> CREATE TABLESPACE HOW;

Tablespace created.

SQL> CREATE TABLESPACE OWS;

Tablespace created.

SQL> CREATE TABLESPACE BACKUP;

Tablespace created.
```

f. Schema creation and grant privileges in the target database.

Schemas "DOCUMENTS","PROD","HOW","OWS" and "BACKUP" are created and the required privileges are granted as per the application/user needs.

Schemas are created

```
SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
PRODTEST     READ WRITE

SQL> CREATE USER DOCUMENTS IDENTIFIED BY DOCUMENTS
  2 DEFAULT TABLESPACE DOCUMENTS
  3 TEMPORARY TABLESPACE TEMP
  4 QUOTA UNLIMITED ON DOCUMENTS;

User created.

SQL> CREATE USER PROD IDENTIFIED BY PROD
  2 DEFAULT TABLESPACE PROD
  3 TEMPORARY TABLESPACE TEMP
  4 QUOTA UNLIMITED ON PROD;

User created.

SQL> CREATE USER HOW IDENTIFIED BY HOW
  2 DEFAULT TABLESPACE HOW
  3 TEMPORARY TABLESPACE TEMP
  4 QUOTA UNLIMITED ON HOW;

User created.

SQL> CREATE USER OWS IDENTIFIED BY OWS
  2 DEFAULT TABLESPACE OWS
  3 TEMPORARY TABLESPACE TEMP
  4 QUOTA UNLIMITED ON OWS;

User created.

SQL> CREATE USER BACKUP IDENTIFIED BY BACKUP
  2 DEFAULT TABLESPACE BACKUP
  3 TEMPORARY TABLESPACE TEMP
  4 QUOTA UNLIMITED ON BACKUP;

User created.
```

Privileges are granted

```
SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
PRODTEST     READ WRITE

SQL> GRANT CONNECT TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT RESOURCE TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE INDEXTYPE TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE TYPE TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE SESSION TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT ALTER SESSION TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE TABLE TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE VIEW TO DOCUMENTS ;

Grant succeeded.

SQL> GRANT CREATE SYNONYM TO DOCUMENTS ;

Grant succeeded.
```

Note: Same privileges been granted to all the schemas.

Starting DMS service.

Step 1:

Login to the AWS console and Click on "DMS" option under Database Services.



Step 2:

Click on “Create Migration” button

The screenshot shows the AWS DMS console dashboard. On the left is a navigation menu with options like 'Dashboard', 'Get started', 'Tasks', 'Endpoints', 'Certificates', 'Replication instances', and 'Subnet groups'. The main content area features a 'What's new' section with updates on AWS DMS expansion and support for SAP ASE and SCT. Below this is an 'Active tasks' section showing 'No active tasks' and a 'Get started' section with a 'Create migration' button highlighted in red. To the right, there are sections for 'Additional info' (with links like 'Getting started', 'Overview and features'), 'Related services' (listing 'Amazon RDS' and 'Data Pipeline'), and 'AWS Schema Conversion Tool' with a 'Download' link.

Step 3:

Since this is an Oracle to Oracle database migration, downloading the schema conversion tool is optional. However it is applicable if the migration is heterogeneous. Click on “Next” button.

The screenshot displays the 'Setting up your first replication task' wizard, specifically the 'Step 1: Welcome' screen. The title is 'Welcome to AWS Database Migration Service'. Below the title, there is explanatory text about source, target, and replication instances, with a link to 'Download AWS Schema Conversion Tool' highlighted in red. A diagram illustrates the data flow: 'Source Database' (represented by a cylinder icon) points to 'Task on replication instance' (represented by a gear icon), which then points to 'Target Database' (represented by a cloud icon). Below the diagram, it states: 'Data is replicated from a source database to a target database by a task running on a replication instance'. At the bottom right, the 'Next' button is highlighted in red, along with 'Cancel' and 'Previous' buttons.

Step 4:

Under the “Create replication Instance” window, provide a meaningful replication instance name, description of the migration, Instance class and VPC where the RDS instance must reside. Select Multi-AZ to YES for high availability and redundant replication streaming. Check the publicly accessible box to access the replication instance through internet.

The screenshot shows the 'Step 2: Replication instance' screen of the wizard. The title is 'Create replication instance'. Below the title, there is a brief description of a replication instance. The form contains the following fields: 'Name' (text input with 'default'), 'Description' (text input with 'oracle db migration from ec2 needed to rds profile'), 'Instance class' (dropdown menu with 'dms.c3.medium'), 'VPC' (dropdown menu with 'rds-ec2-2018 - VPC - prod/uklan-p'), 'Multi-AZ' (dropdown menu with 'No'), and 'Publicly accessible' (checkbox checked). Each field has a help icon to its right.

Step 5:

Under advanced section enter allocated storage for DMS instance, Replication subnet group based on the VPC selected in the previous step, VPC security group(s) to allow inbound/outbound traffic, KMS master key as default if there is no existing key for database volume encryption. Click on "Next" button.

Advanced

Allocated storage (GB): 50

Replication Subnet Group: default-vpc-e63b3583

Availability zone: us-west-2a

VPC Security Groups: default, launch-wizard-10, launch-wizard-13, launch-wizard-14

KMS master key: (Default) aws/dms

Description: Default master key that protects my DMS replication instance volumes when no other key is defined

Account: 841764865526

Key ARN: iam:aws:kms:us-west-2:841764865526:key/371ab7c2-990e-4306-9331-e2916c8038e

Cancel Previous **Next**

Step 6:

Under "connect source and target database endpoints" window, enter source and target database details to create endpoints. Provide a meaningful endpoint identifier name, Source and Target engines (Database engines), Server name (Database server IP /end point), Port (Database listening port), SSL mode (To encrypt the connection) which is none for oracle. Username in source identifier should have data dictionary views access privileges and the Username in target identifier should be the schema name which needs to be migrated.

Connect source and target database endpoints

Replication instance created successfully

Your database endpoint can be on-premise, in EC2, RDS or in the cloud. Define the connection details below. It is recommended that you test your endpoint connections here to avoid errors later.

Source database connection details

Endpoint identifier: TEST

Source engine: oracle

Server name: [REDACTED]

Port: 1521

SSL mode: none

User name: SYSTEM

Password: [REDACTED]

SID: TEST

Advanced

Run test

Target database connection details

Endpoint identifier: PROCTEST

Target engine: oracle

Server name: [REDACTED]

Port: 1521

SSL mode: none

User name: DOCUMENTS

Password: [REDACTED]

SID: PROCTEST

Advanced

Run test

In our case the username in the source identifier is "SYSTEM" which by default has all the privileges and the target username is "DOCUMENTS" which needs to be migrated.

Step 7:

Click “Run test” to check the connectivity between replication instance and source, target databases. You should receive a message stating “Connection tested successfully” and click on “Next” button.

The screenshot displays two side-by-side configuration panels for source and target database connections. The source panel is titled 'Source database connection details' and the target panel is 'Target database connection details'. Both panels include fields for Endpoint identifier, Source/Target engine, Server name, Port, SSL mode, User name, Password, and SID. Below these fields, there are 'Advanced' sections. In both advanced sections, a 'Run test' button is highlighted with a red box, and a green checkmark with the text 'Connection tested successfully' is visible below it. At the bottom right of the entire configuration area, the 'Next' button is also highlighted with a red box.

Step 8:

Under the “Create task” window provide a meaningful task name and task description. Select Migration type as “Migrate existing data” and select the check box “Start task on create” to start the activity automatically post task creation.

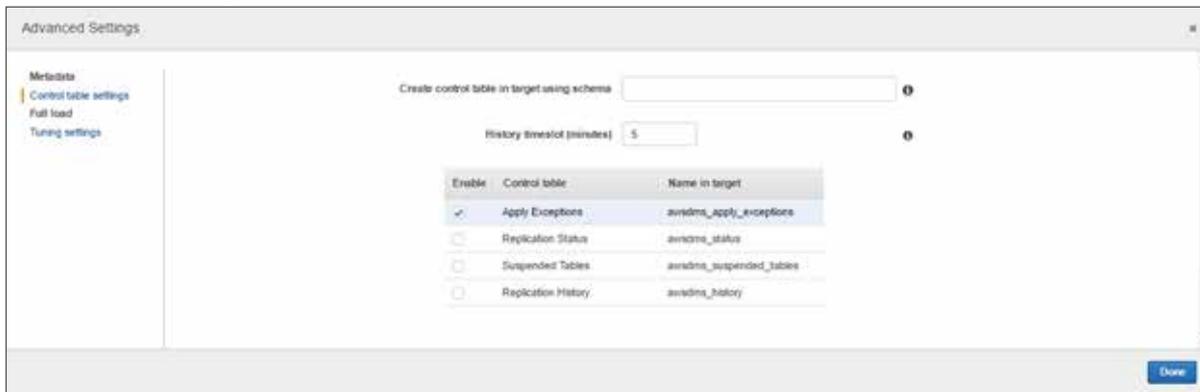
The screenshot shows the 'Setting up your first replication task' window. On the left, there is a navigation pane with steps: Step 1: Welcome, Step 2: Replication instance, Step 3: Database endpoints, and Step 4: Task. The main area is titled 'Create task' and contains the following fields: Task name (DocumentMigration), Task description (Migrating Document schema to RDS), Source endpoint (test), Target endpoint (prodtst), Replication instance (dmsinst), Migration type (Migrate existing data), and a checked 'Start task on create' checkbox. A note at the top of the main area states: 'A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it can be created automatically.'

Step 9:

Under Task settings select the “Target table preparation mode” as “Do nothing” since it is a fresh migration. Select Full LOB mode in “Include LOB columns in replication” if the size of the LOB is unknown and to migrate complete LOBs regardless of size. The option “Limited LOB mode” can be selected if the LOB chunk size is known. The option “Don’t include LOB columns” can be selected to exclude LOB columns. Check “Enable logging” box to capture the errors and warnings in the logs. Select advanced settings to configure “Control Tables” and “Tuning” settings to optimize the data migration.

The screenshot shows the 'Task Settings' window. It includes the following settings: Target table preparation mode (Do nothing), Include LOB columns in replication (Limited LOB mode), Max LOB size (kb) (32), and Enable logging (checked). A blue box at the bottom contains the text: 'CloudWatch Logs usage will be charged at standard rates. See here for more details.' The 'Advanced Settings' button at the bottom is highlighted with a red box.

Under advanced settings



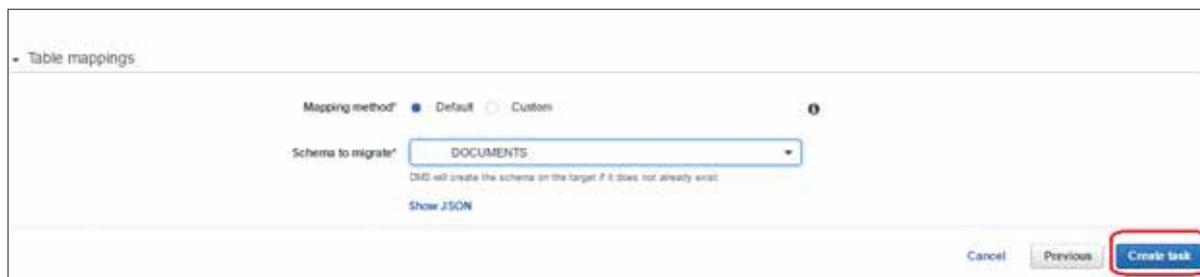
Under advanced settings



Step 10:

Under Table mappings select the “Mapping method” as default to migrate the table with the same name to target database. To remap with a different table name click on “Custom” and edit the JSON script for relevant mapping. Select the migration schema name from “Schema to migrate” drop down list. Click on “Create task” button.

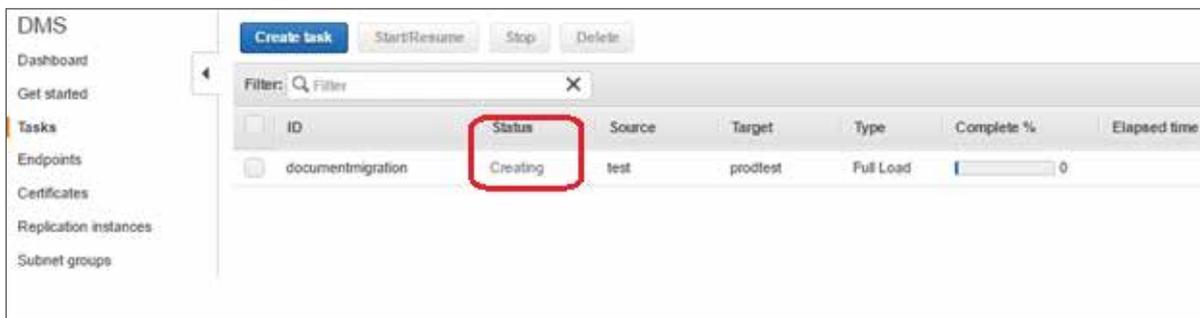
Table mappings option



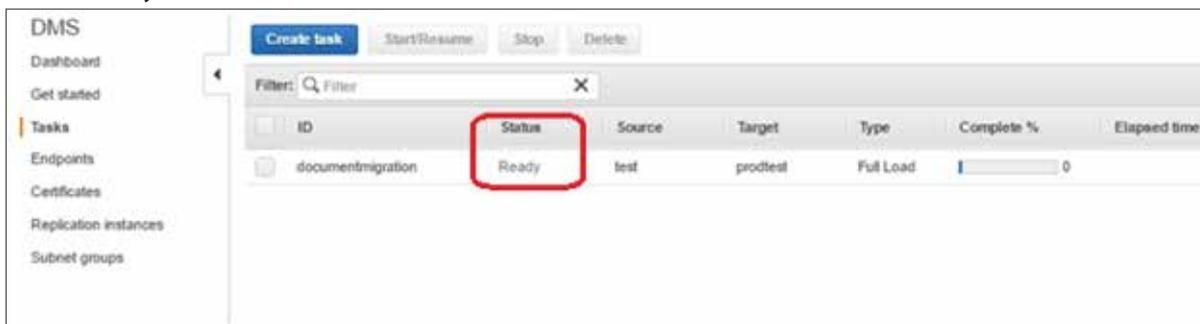
Step 11:

The Task has been created successfully and the status is changed from “Creating” to “Ready”.

Task is in creating status



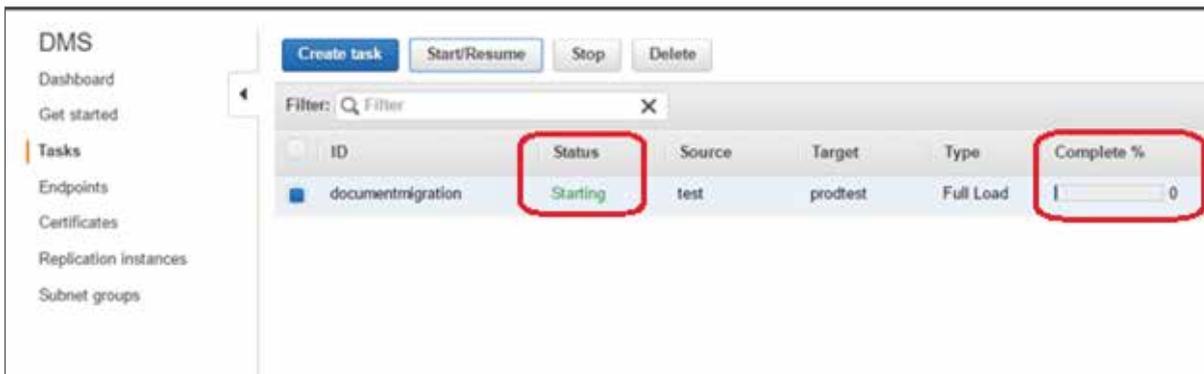
Task is in ready status



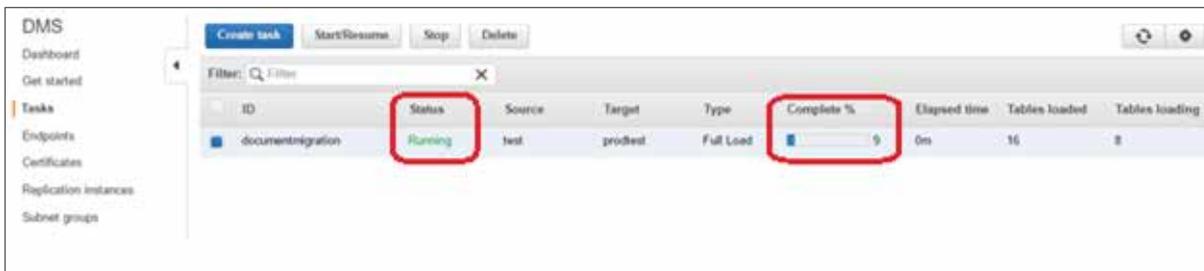
Step 12:

If the task is not started automatically, select the appropriate task and click on “Start/Resume” button to initiate the migration activity. Once the activity is initiated the status will change from “Ready” to “Starting” and then eventually to “Running” state. The progress bar shows the percentage of the task completion.

Task is in starting status

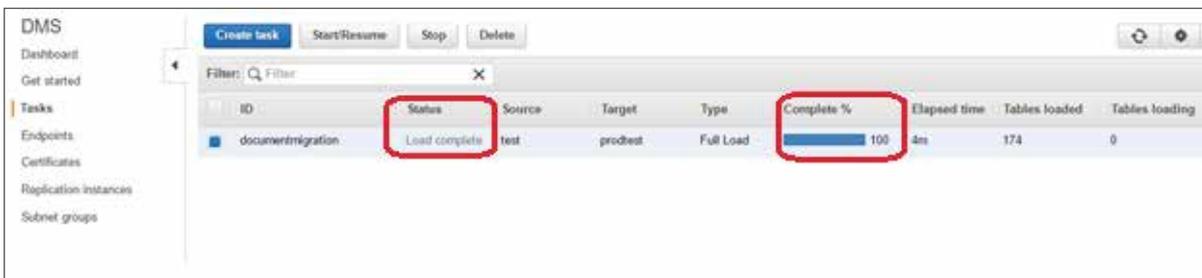


Task is in running status



Task is in Load complete status

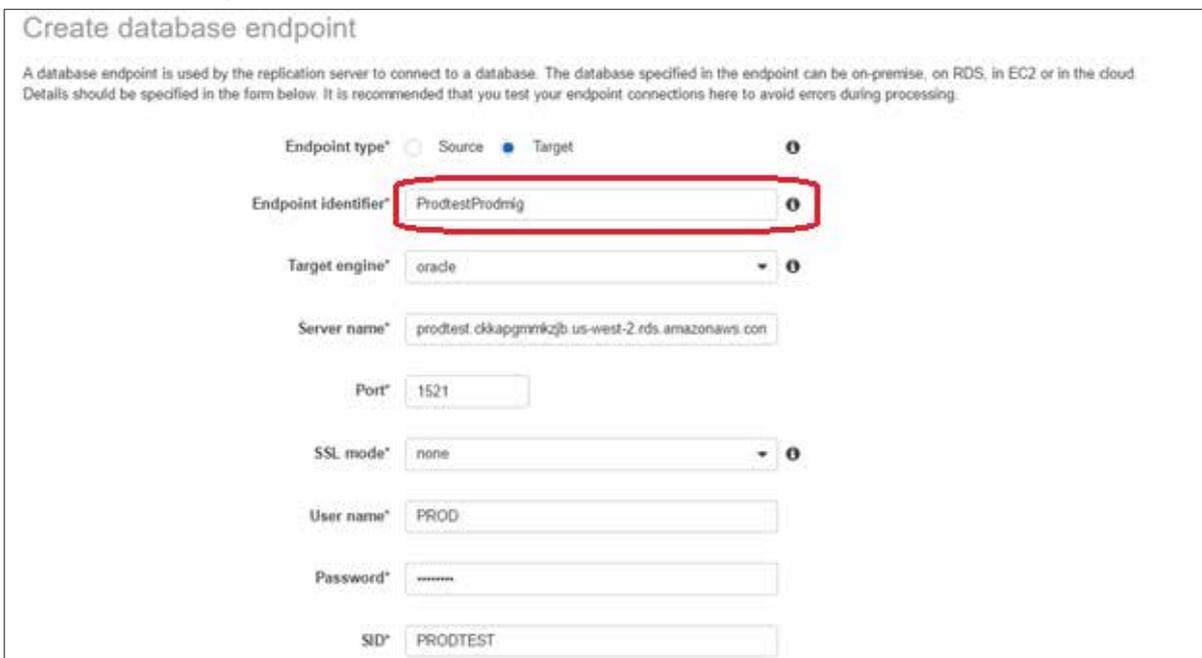
The Progress bar is changed to 100% and the status shows as “Load Complete”. This indicates that one schema migration has completed from source to target Database successfully.



Step 13:

Create a new endpoint, new task and repeat the same steps to migrate all schemas to the target database.

New “ProdtestProdmig” endpoint is created.



Advanced

Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC* vpc-e63b3583 - VPC_prabhakaran.p

Replication instance* dmtest - vpc-e63b3583

Refresh schemas after successful connection test

Run test

✔ Connection tested successfully

Cancel **Save**

Prodtestprodmig endpoint is in active status

DMS

Dashboard

Get started

Tasks

Endpoints

Certificates

Replication instances

Subnet groups

Create endpoint Modify Test connection Refresh schemas Delete

Filter: Filter

Identifier	Type	Status	Engine	Server name	Port
prodtest	target	active	oracle	prodtest.ckkapgmmkzjb.us-w	1521
prodtestprodmig	target	active	oracle	prodtest.ckkapgmmkzjb.us-w	1521
test	source	active	oracle	52.5.143.197	1521

New Task "prodmigration" is created.

Create task

A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it can be created automatically.

Task name* prodmigration

Replication instance* dmtest - vpc-e63b3583

Source endpoint* test

Target endpoint* prodtest

Migration type* Migrate existing data

Start task on create

Task Settings

Target table preparation mode* Do nothing Drop tables on target Truncate

Include LOB columns in replication* Don't include LOB columns Full LOB mode Limited LOB mode

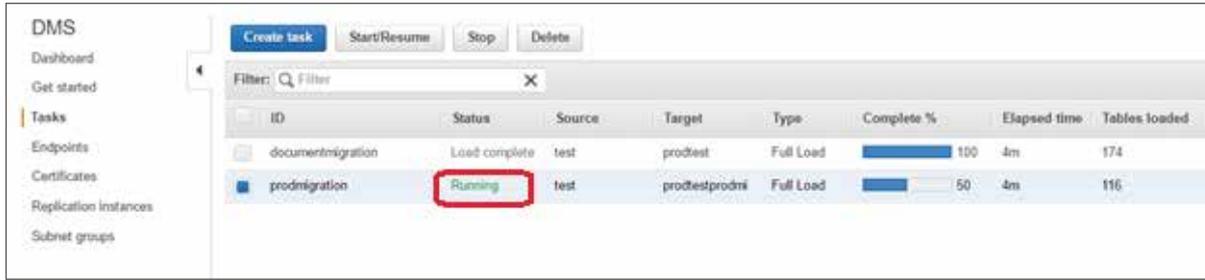
Max LOB size (kb)* 32

Enable logging

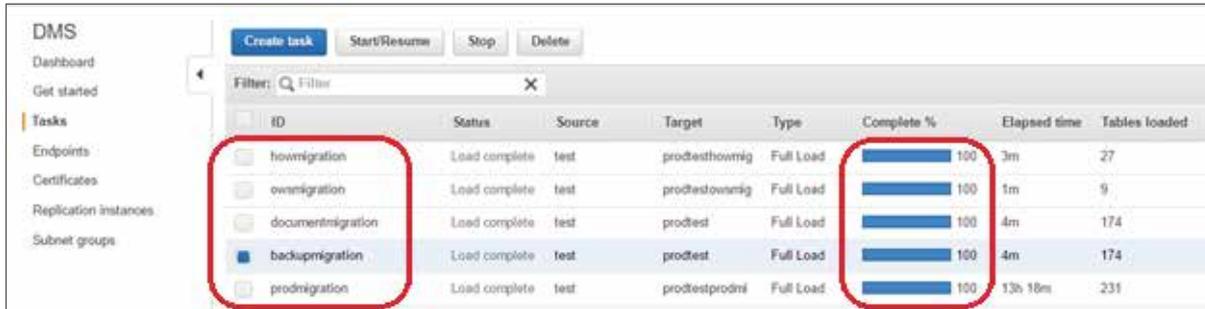
CloudWatch Logs usage will be charged at standard rates. See here for more details.

Advanced Settings

Prodmigration task is in "Running" state



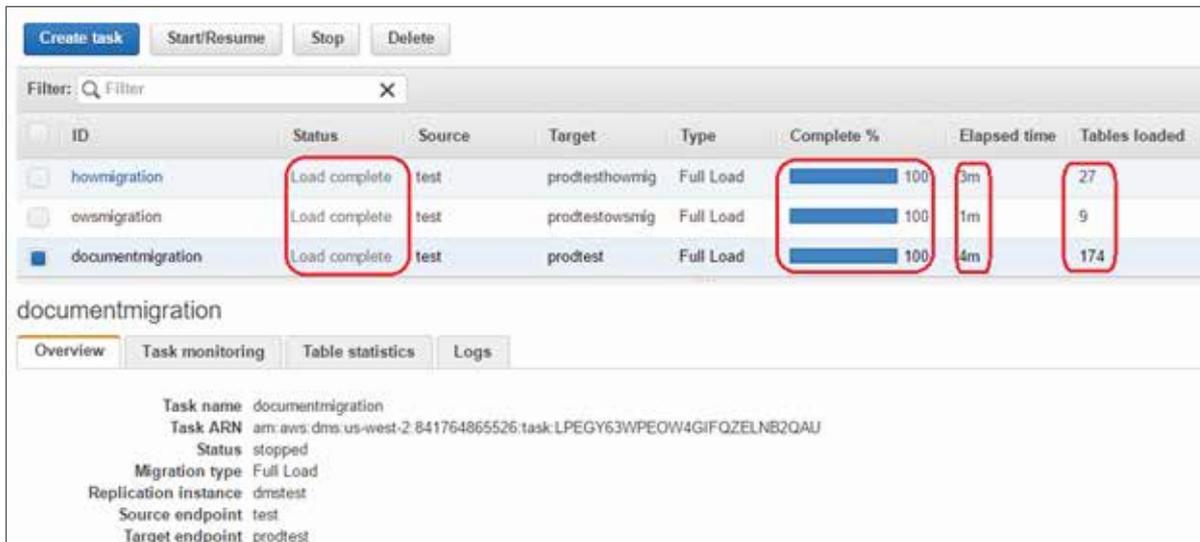
The same way backupmigration, owsmigration, howmigration tasks are created and successfully migrated all the schemas from source to target database.



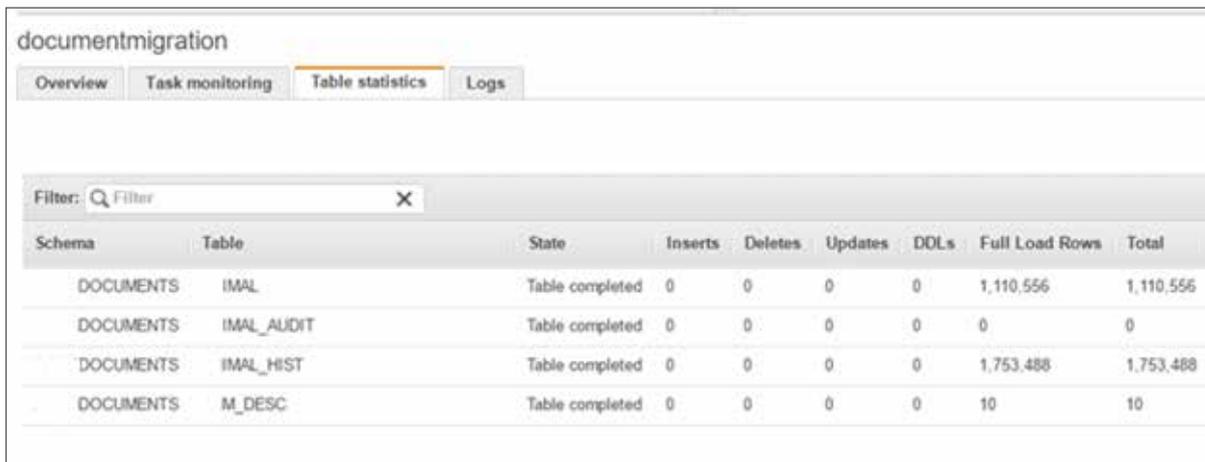
Monitoring DMS service Task

The progress of the task and resources used can be monitored through DMS console.

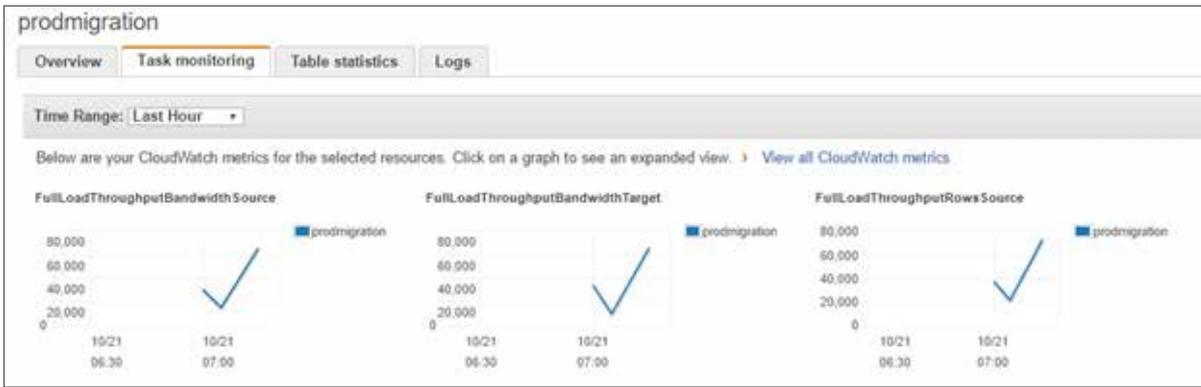
a. Task status, complete %, Elapsed time and table loaded can be monitored.



b. Table Statistics tab shows the table migration details as shown below.



c. Task monitoring tab shows the detailed cloud watch metrics.



d. Migration logs shows the details of the migration, error and warnings with time stamp under the logs tab.

The screenshot shows the 'Logs' tab in the AWS console for the 'prodmigration' task. It displays a list of migration logs with columns for 'Time (UTC +00:00)' and 'Message'. The logs include timestamps and messages such as 'Unload finished for table DOCUMENTS_EMPDOE_EVAL_HOLD (id = 173): 40 rows sent. (streamcomponent.c:2567)', 'Subtask #7 ended (replicationtask_util.c:925)', and 'Load finished for table DOCUMENTS_EMPDOE_EVAL_HOLD (id = 171): 447 rows received. 0 rows skipped. Volume trans'.

Time (UTC +00:00)	Message
2016-10-21 06:08:06	2016-10-21T06:08:06 [SOURCE_UNLOAD]: Unload finished for table 'DOCUMENTS_EMPDOE_EVAL_HOLD' (id = 173): 40 rows sent. (streamcomponent.c:2567)
2016-10-21 06:08:06	2016-10-21T06:08:06 [TASK_MANAGER]: Table 'DOCUMENTS_ATT00_CODE' (id = 169) Loading finished by subtask 7. 16121 records transferred. (replicationtask_util.c:925)
2016-10-21 06:08:06	2016-10-21T06:08:06 [TASK_MANAGER]: Subtask #7 ended (replicationtask_util.c:925)
2016-10-21 06:08:08	2016-10-21T06:08:08 [TARGET_LOAD]: Load finished for table 'DOCUMENTS_EMPDOE_EVAL_HOLD' (id = 171): 447 rows received. 0 rows skipped. Volume trans

Procedures, Functions, Packages and other object migration:

All schemas tables are migrated successfully through the DMS service. The below method is followed to migrate the other object types like procedures, functions, packages, package body's, views and indexes.

Step 1:

Exporting objects from source database except tables which is already been migrated.

```

C:\oracle\11g\bin> sqlplus /nolog
SQL> connect sys/password@prod as sysdba
SQL> expdp sys/password@prod directory=D:\dms\expdp_dir exclude=table dumpfile=fulldbexcludetab.dmp logfile=fulldbexcludetab.log
Export Release 11.2.0.4.0 - Production on Fri Oct 7 02:43:22 2016
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Exported by Oracle Database 11g Release 11.2.0.4.0 - 64bit Production
Starting 'SYS_EXPORT_FULL_01'. ***** AS SYSDBA *****
Starting in progress using BLOCKS method...
REAL destination using BLOCKS method: 8 B
Processing object type DATABASE_EXPORT/TABLESPACE
Processing object type DATABASE_EXPORT/PROFILE
Processing object type DATABASE_EXPORT/SYS_USER/USER
Processing object type DATABASE_EXPORT/SCHEMA/USER
Processing object type DATABASE_EXPORT/ROLE
Processing object type DATABASE_EXPORT/GRANT/SYSTEM_GRANT/PROC_SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/DEFINER_ROLE
Processing object type DATABASE_EXPORT/SCHEMA/TABLESPACE_QUOTA
Processing object type DATABASE_EXPORT/RESOURCE_COST
  
```

Step 2:

Creating database link from source to target database.

```

SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
TEST         READ WRITE

SQL> create database link to_prodtestrds connect to myadmin identified by myadmin using 'prodtest';
Database link created.
  
```

Step 3:

Testing the connection from source to target Database.

```

SQL> select sysdate from dual@to_prodtestrds;

SYSDATE
-----
17-OCT-16

SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
TEST         READ WRITE
  
```

Step 4:

Dump file is transferred from source to target database through DBMS_FILE_TRANSFER.

```
SQL> select name,open_mode from v$database;

NAME          OPEN_MODE
-----
TEST         READ WRITE

SQL>
SQL> BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE (
    source_directory_object      => 'DATAPUMP_DIR',
    source_file_name             => 'fulldbexcludetab.dmp',
    2   3   4   5   destination_directory_object => 'DATA_PUMP_DIR',
    destination_file_name       => 'fulldbexcludetab.dmp',
    destination_database        => 'to_prodttestrds'
  );
END; 6   7   8   9
10 /

PL/SQL procedure successfully completed.
```

Step 5:

Importing the Dump file from the source to the target database.

```
ls -l /u01/app/oracle/oradata/prodttest/DIRECTORY_DATA_PUMP_DIR/FULL/P dumpfile=fulldbexcludetab.dmp logfile=fulldbexcludetab.log
```

Post migration activity and checks

Performed invalid object compilation and object comparison between source and target databases.

Below command is used to compile the invalid objects in the target RDS database.

```
exec SYS.UTL_RECOMP.RECOMP_SERIAL('PROD');
```

Source database object count:

```
SQL> select object_type,status,count(*) from dba_objects where owner in( 'PROD') group by object_type,status;

OBJECT_TYPE      STATUS      COUNT(*)
-----
FUNCTION         VALID        138
PROCEDURE        INVALID      1
DATABASE LINK    VALID        3
VIEW             INVALID     16
SEQUENCE         VALID       127
TYPE            VALID        1
MATERIALIZED VIEW VALID        1
PACKAGE          INVALID      1
TABLE           VALID       231
INDEX           VALID       161
PROCEDURE       VALID        10
PACKAGE BODY    INVALID      1
LOB            VALID        2
VIEW           VALID       105

14 rows selected.
```

Target database object count:

```
SQL> exec SYS.UTL_RECOMP.RECOMP_SERIAL('PROD');

PL/SQL procedure successfully completed.

SQL> select object_type,status,count(*) from dba_objects where owner= 'PROD' group by object_type,status;

OBJECT_TYPE      STATUS      COUNT(*)
-----
FUNCTION         VALID        138
PROCEDURE        INVALID      1
DATABASE LINK    VALID        3
VIEW             INVALID     16
SEQUENCE         VALID       127
TYPE            VALID        1
MATERIALIZED VIEW VALID        1
PACKAGE          INVALID      1
TABLE           VALID       231
INDEX           VALID       161
PROCEDURE       VALID        10
PACKAGE BODY    INVALID      1
LOB            VALID        2
VIEW           VALID       105

14 rows selected.
```

The same way the objects can be compared for all the schemas.If there are any missing objects it can be created manually.

Note:

Below are the required source database user privileges to migrate the schemas through DMS. We have used SYSTEM user to migrate all schema's hence granting the below privileges are not required.

SELECT ANY TRANSACTION	CREATE ANY VIEW
SELECT on V\$NLS_PARAMETERS	DROP ANY VIEW
SELECT on V\$TIMEZONE_NAMES	CREATE ANY PROCEDURE
SELECT on ALL_INDEXES	ALTER ANY PROCEDURE
SELECT on ALL_OBJECTS	DROP ANY PROCEDURE
SELECT on DBA_OBJECTS	CREATE ANY SEQUENCE
SELECT on ALL_TABLES	ALTER ANY SEQUENCE
SELECT on ALL_USERS	DROP ANY SEQUENCE
SELECT on ALL_CATALOG	SELECT on DBA_USERS
SELECT on ALL_CONSTRAINTS	SELECT on DBA_TAB_PRIVS
SELECT on ALL_CONS_COLUMNS	SELECT on DBA_OBJECTS
SELECT on ALL_TAB_COLS	SELECT on DBA_SYNONYMS
SELECT on ALL_IND_COLUMNS	SELECT on DBA_SEQUENCES
DROP ANY TABLE	SELECT on DBA_TYPES
SELECT ANY TABLE	SELECT on DBA_INDEXES
INSERT ANY TABLE	SELECT on DBA_TABLES
UPDATE ANY TABLE	SELECT on DBA_TRIGGERS

Conclusion:

Database migration has been successfully completed from EC2 oracle database to RDS oracle database using DMS service.

About Author:

Prabhakaran Ponnusamy a Cloud Architect at 8k Miles, an Oracle expertise along with multiple database technologies hands-on. Passionate in learning, experimenting and exploring new technologies and making business much more than usual.

San Ramon, CA (HQ)

12647 Alcosta Boulevard,
Suite 450, San Ramon, CA 94583, USA
www.8kmiles.com
sales@8kmiles.com
Direct : 925-270-4800
Toll-Free : 855-856-4537

Chicago, IL

1827 Walden Office Square
Suite #460
Schaumburg, IL 60173
Phone : 708-289-5111

Dallas, TX

17740 Preston Road
Suite #200
Dallas, TX -75252
Phone : 214-272-2404

Chennai, India

Srinivasa Towers
New No.5, Old No. 11,
Cenotaph Road,
Alwarpet, Chennai – 600 018
Phone : +91-44-6602-8000
Fax : +91-44-4300-9049

Ontario, Canada

4 Robert Speck Parkway,
Suite 1500,
Mississauga, Ontario L4Z 1S1
Phone : 416-366-7762

Sharjah, UAE

Q1-05-109/C
SAIF Zone
PO Box 121213
Sharjah-UAE