Configure DMS to Preserve Timestamp Data without Timezone

Background

Customer has a timestamp column in RDS (Aurora-MySQL), the value is in timestamp format (e.g. `2023-02-24 18:00:01`) instead of timestamptz format (e.g. `2023-02-24 18:00:01+8`). They have both RDS and Redshift timezone set to `Asia/Shanghai`.

They would like to use DMS to migrate the data to Redshift. They are expecting the timestamp value to remain the same after migration.

NOTE: The best practice is to use UTC timeznoe at all database layers, or use timestamptz value instead of timestamp in the data.

Findings

Following tests shows that, for an RDS (Aurora-MySQL) database to preserve its timestamp (without timezone) value whlie it is migrated to Redshift, they need to have following 2 settings in their DMS endpoints.

- In the source endpoint of DMS task, add extra connection attribute `serverTimezone=Asia/Shanghai`
- In the target endpoint of DMS task, add extra connection attributes `initstmt=SET TIMEZONE='Asia/Shanghai'`

Test

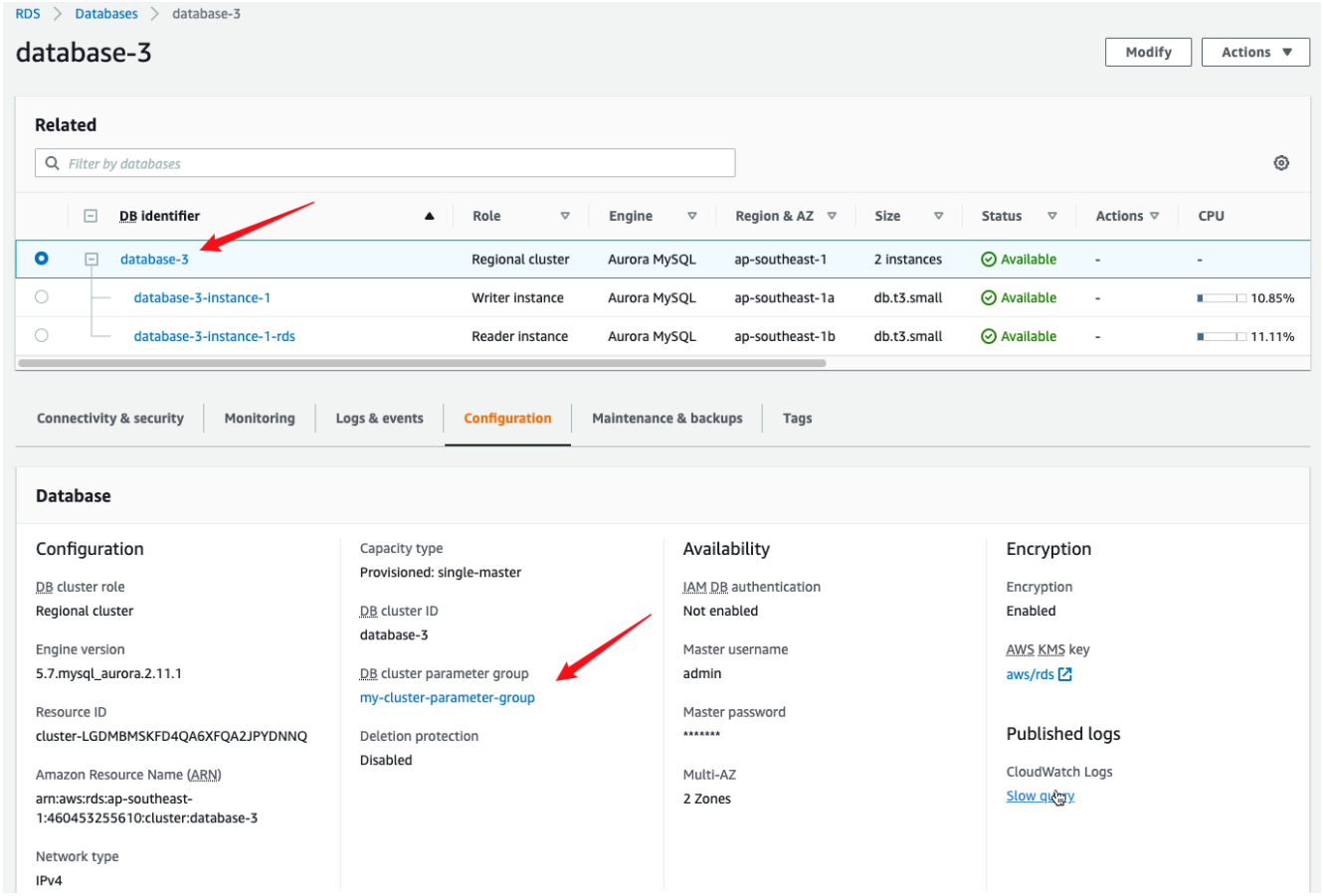
Prerequiste

Setup RDS(MySQL) Cluster, Redshift Cluster, and DMS instance with a task to prelicate data between the RDS and Redshift. By default, the new RDS and Redshift cluster has an initial timezone = UTC.

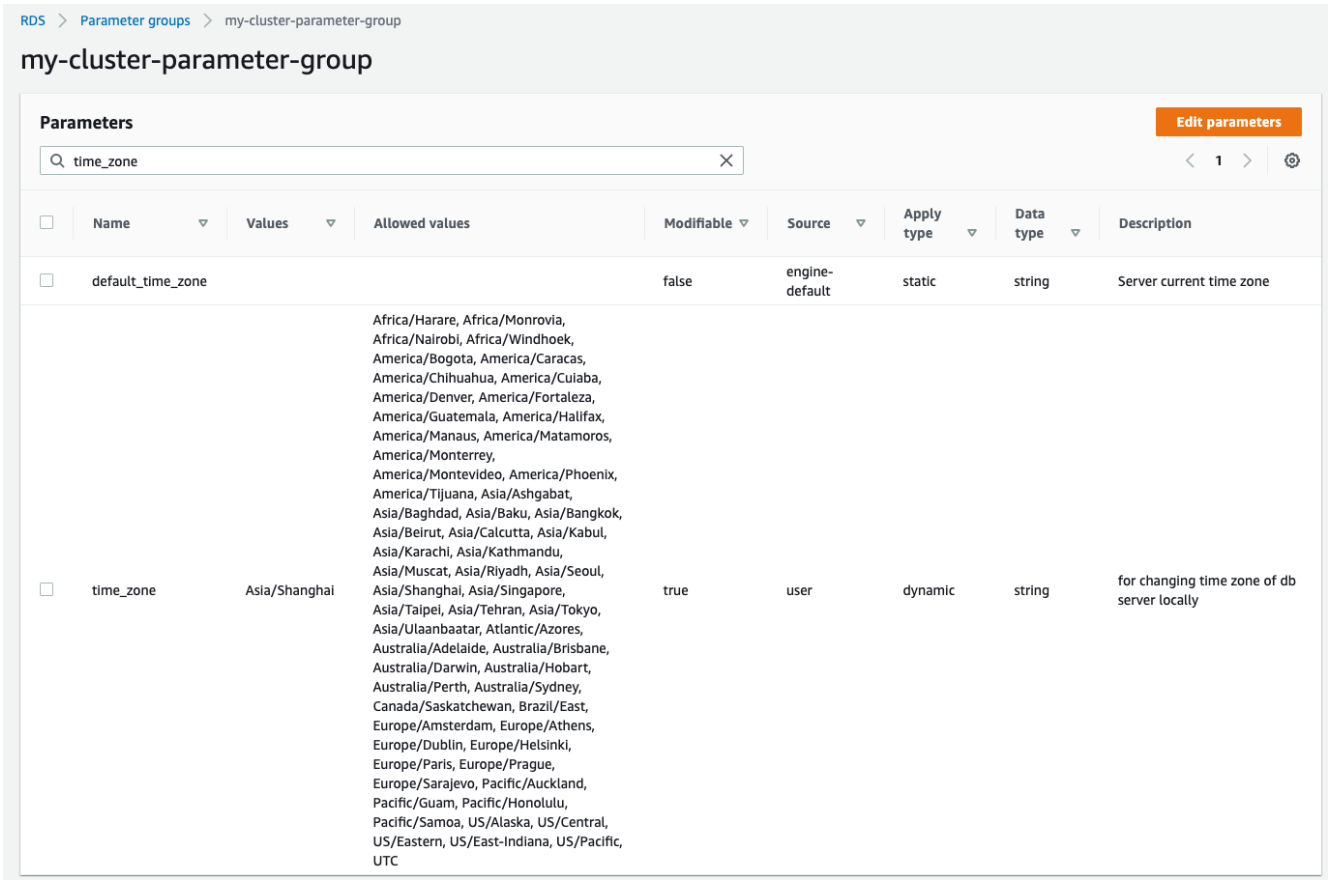
Set RDS Timezone

Set the timezone of RDS to `Asia/Shanghai`.

1. Update the cluster parameter group of RDS.



2. Update the parameter group by setting `time-zone` to `Asia/Shanghai`.



3. Must restart all DB instances in the cluster.
4. Examine the timezone setting in the MySQL terminal.

```
mysql> SELECT @@global.time_zone, @@session.time_zone;
+--------------------+---------------------+
| @@global.time_zone | @@session.time_zone |
+--------------------+---------------------+
| Asia/Shanghai      | Asia/Shanghai       |
+--------------------+---------------------+
1 row in set (0.00 sec)
```

Create Sample Database Table in RDS

When both RDS and Redshift are in deafult timezone of UTC, the migrated

1. Create a table `test` with a timestamp.

```
CREATE TABLE `test` (
  `id` bigint(20) NOT NULL,
  `first_name` text,
  `last_name` text,
  `email` text,
  `gender` text,
  `ip_address` text,
  `update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

2. Examine the table.

```
mysql> describe test;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| id          | bigint(20) | NO   | PRI | NULL    |       |
| first_name  | text       | YES  |     | NULL    |       |
| last_name   | text       | YES  |     | NULL    |       |
| email       | text       | YES  |     | NULL    |       |
| gender      | text       | YES  |     | NULL    |       |
| ip_address  | text       | YES  |     | NULL    |       |
| update_time | timestamp  | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

3. Insert some sample data in the table.

```
mysql> select * from test;
+----+------------+-----------+----------------+--------+------------+---------------------+
| id | first_name | last_name | email          | gender | ip_address | update_time         |
+----+------------+-----------+----------------+--------+------------+---------------------+
|  1 | alan       | alan      | alan@gmail.com | male   | 1.0.0.1    | 2023-02-24 18:00:01 |
|  2 | bob        | bob       | bob@gmail.com  | male   | 2.0.0.2    | 2023-02-24 15:00:01 |
+----+------------+-----------+----------------+--------+------------+---------------------+
2 rows in set (0.00 sec)
```

Test with Default Timezone for Redshift

1. Restart the DMS task to migrate data from RDS to Redshift.
2. Examine the data in Redshift. Their `update_time` field values are 8 hours earlier (-8) than the corresponding value in RDS.

id	first_name	last_name	email	gender	ip_address	update_time	operation
1	alan	alan	alan@gmail.com	male	1.0.0.1	2023-02-24 10:00:01	NULL
2	bob	bob	bob@gmail.com	male	2.0.0.2	2023-02-24 07:00:01	NULL

Test with Redshift Timezone = 'Asia/Shanghai'

1. Using Redshift Query Editor, check its current timezone setting. The default timezone value is UTC.

```
SHOW TIMEZONE;
```

2. For testing purpose, set the timezone value to `Asia/Singapore`, which is the same timezone +8 as `Asia/Shanghai`.

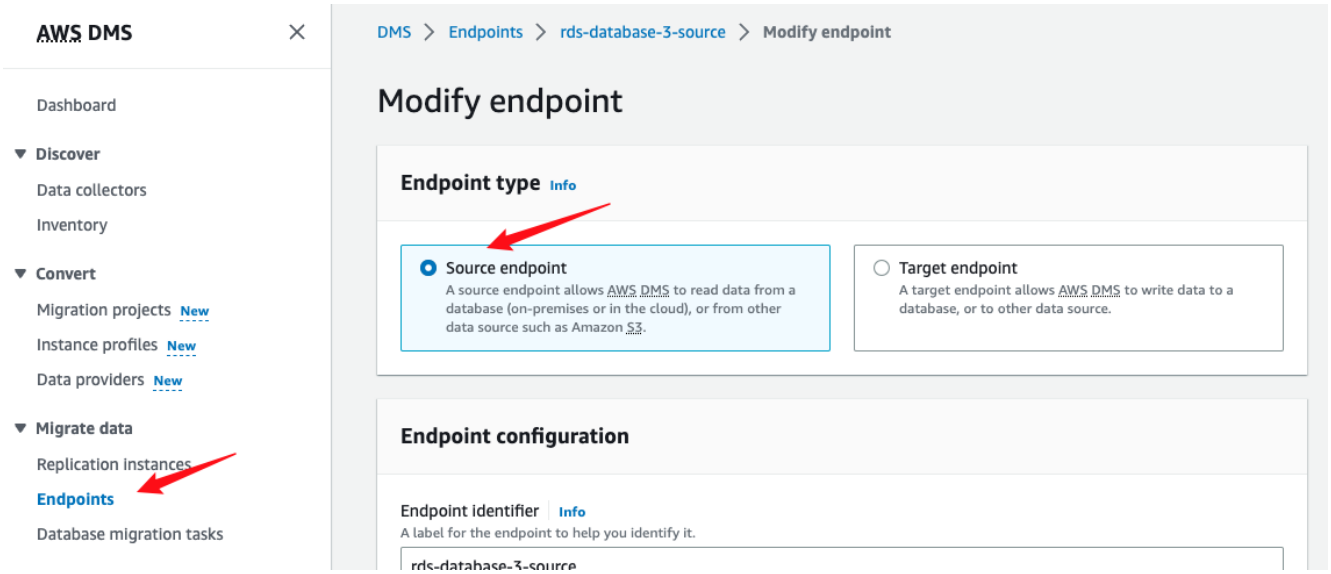
```
SET TIMEZONE='Asia/Shanghai';
SHOW TIMEZONE;
```

3. Restart the DMS task to migrate data from RDS to Redshift.
4. Examine the data in Redshift. Their `update_time` field values are 8 hours earlier (-8) than the corresponding value in RDS.

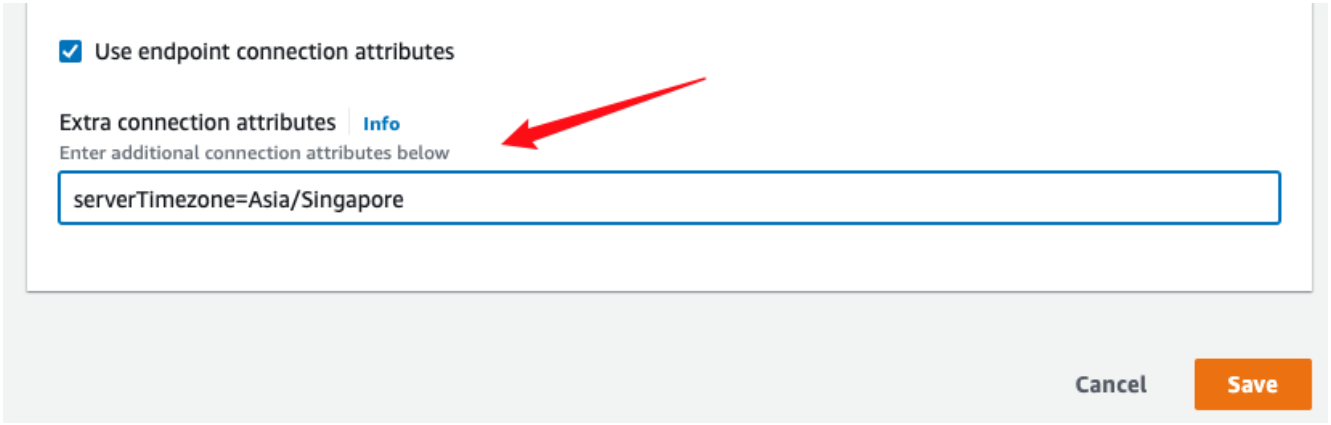
id	first_name	last_name	email	gender	ip_address	update_time	operation
1	alan	alan	alan@gmail.com	male	1.0.0.1	2023-02-24 10:00:01	NULL
2	bob	bob	bob@gmail.com	male	2.0.0.2	2023-02-24 07:00:01	NULL

Test by Modifying DMS Source Endpoint

1. Modify the DMS source endpoint, which is pointing to the RDS.



2. Add an extra connection attributes `serverTimezone=Asia/Singapore` to specify the server timezone.

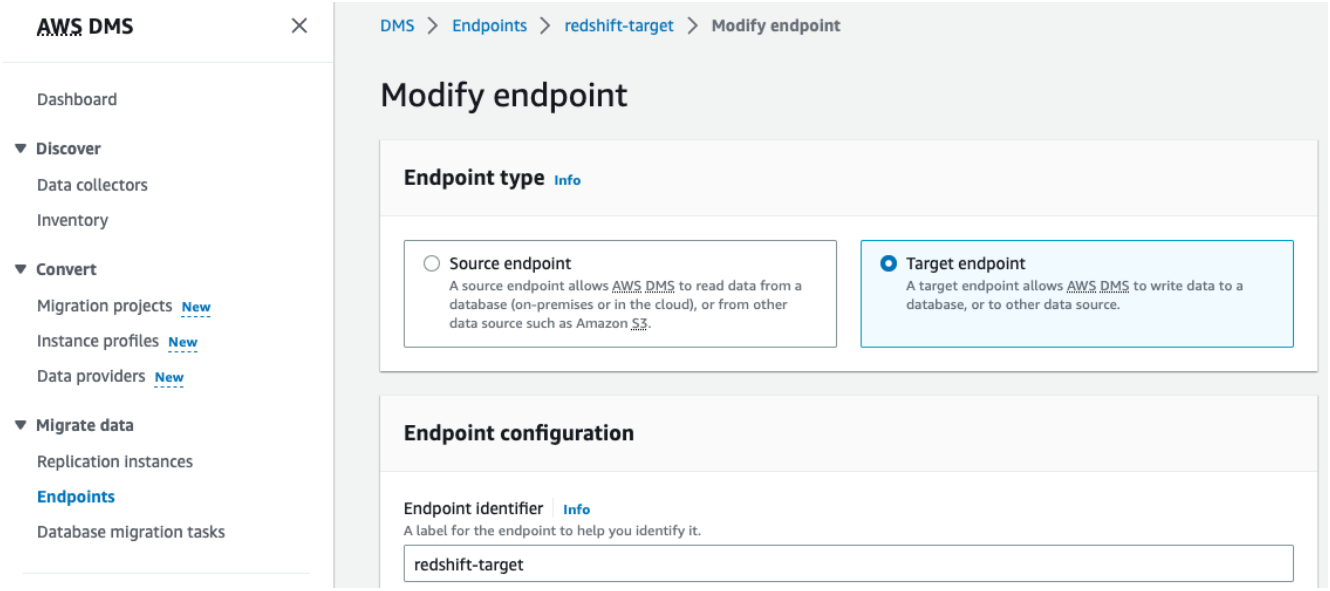


3. Examine the data in Redshift. Their `update_time` field values are 8 hours earlier (-8) than the corresponding value in RDS.

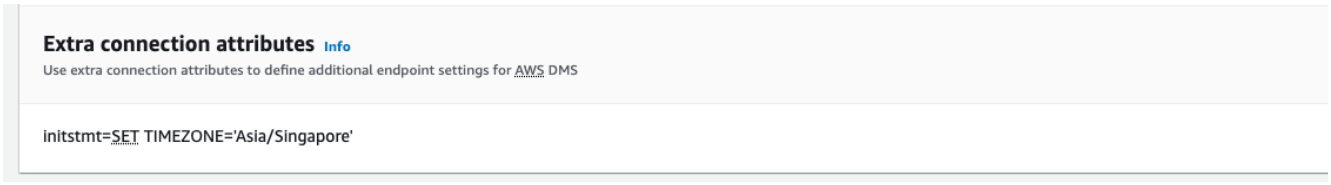
id	first_name	last_name	email	gender	ip_address	update_time	operation
1	alan	alan	alan@gmail.com	male	1.0.0.1	2023-02-24 10:00:01	NULL
2	bob	bob	bob@gmail.com	male	2.0.0.2	2023-02-24 07:00:01	NULL

Test by Modifying DMS Target Endpoint

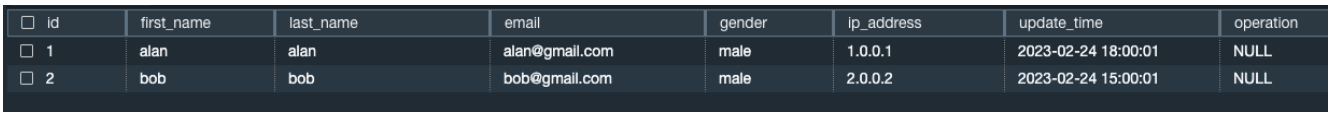
1. Stop the DMS task and modify the DMS target endpoint.



2. Update the extra connection attributes as `initstmt=SET TIMEZONE='Asia/Singapore'`, where `SET TIMEZONE='Asia/Singapore'` is the statement to set the timezone of Redshift.



3. Restart DMS task, which will truncate the target table and perform a full load.
4. Examine the data in Redshift. The `update_time` is migrated correctly.



id	first_name	last_name	email	gender	ip_address	update_time	operation
1	alan	alan	alan@gmail.com	male	1.0.0.1	2023-02-24 18:00:01	NULL
2	bob	bob	bob@gmail.com	male	2.0.0.2	2023-02-24 15:00:01	NULL