BUILDING A REAL-TIME DATABASE REPLICATION AND CHANGE DATA CAPTURE SOLUTION WITH MSKCONNECTOR, MSK CLUSTER AND DEBEZIUM MYSQL CONNECTOR

ABSTRACT

This solution explores real-time database replication and Change Data Capture processing. MSKConnect is The central piece of the solution, it provides a framework for an MSK cluster to connect with external systems such as databases, Amazon S3, Search indexes and file systems. MSKConnect is a plugin-based framework, it uses plugins as the base mechanism for enabling connectivity and data movement between an MSK Cluster and other external systems. For this solution, we are using Debezium MySQL source connector (Aurora DB being the source of our real-time data) plugin to perform both replication and Change Data Capture based on tasks spurn by MSK Workers. The IAC delivery tool for this solution is Terraform.

Table of Contents

1.0 Introduction

This guide is a Terraform implementation inspired by the AWS workshop https://catalog.us-east-1.prod.workshops.aws/workshops/c2b72b6f-666b-4596-b8bc-bafa5dcca741/en-US/mskconnect/overview
This solution explores real-time database replication and Change Data Capture processing. The central piece of this solution is MSKConnect, which provides a framework for MSK cluster to connect with external systems such as databases, Amazon S3, Search indexes and file systems. MSKConnect is a plugin-based framework, it uses plugins as the base mechanism for enabling connectivity and data movement between an MSK Cluster and other external systems. For this solution, we are using Debezium MySQL source connector (Aurora DB being the source of our real-time data) plugin to perform both replication and Change Data Capture based on tasks spurn by MSK Workers. A Worker Is a JVM process that runs the connector logic. Details of the steps involved in implementation this solution is provided in the subsequent sections below.

We hope you will find this solution useful for your next real-time database replication and CDC processing project.

2.0 Clone Repository and Initialize Modules and Apply

To begin, please clone the repository to your workspace. Execute the command shown below:

cd /home/dev01 (this workspace may be different in your own case)
git clone https://github.com/sitadconsulting/aws_infra.git
cd aws_infra/projects/proj3 (review the files in this directory and update the "dev.auto.tfvars" file to reflect values from your own environment – ***Please ensure you at the least assign value for kafka_client_instance_public_key, remote_access_to_kafka_client_instance_port_22 remote_access_to_kafka_client_instance_port_8081***)

```
[dev01@houssitdvm02 proj3]$ terraform init
Initializing the backend...
Initializing modules...
- acmpca_ca in ../../modules/acmpca_certificate_authority
- acmpca_ca_certificate in ../../modules/acmpca_certificate_authority_certificate
- acmpca_certificate in ../../modules/acmpca_certificate
- debezium_mysql_connector in ../../modules/mskconnect_connector
- debezium_mysql_custom_plugin in ../../modules/mskconnect_custom_plugin
- debezium_mysql_worker_configuraion in ../../modules/mskconnect_worker_configuration
- ec2_iam_role in ../../modules/iam_role
- ec2_iam_role_policy_attachment in ../../modules/iam_role_policy_attachment
- eip in ../../modules/eip
- iam_instance_profile in ../../modules/iam_instance_profile
- internet_gateway in ../../modules/internet_gateway
- internet_gateway_route in ../../modules/route
- kafka_authorization_iam_policy in ../../modules/iam_policy
- kafka_client_instance in ../../modules/instance
- kafka_connect_iam_policy in ../../modules/iam_policy
- kafka_connect_iam_role in ../../modules/iam_role
- kafka_connect_iam_role_policy_attachment in ../../modules/iam_role_policy_attachment
- kafka_connect_target_bucket in ../../modules/s3_bucket
- key_pair in ../../modules/key_pair
- launch_template in ../../modules/launch_template
- msk_broker_log_group in ../../modules/cloudwatch_log_group
- msk_cluster in ../../modules/msk_cluster
- msk_configuraion in ../../modules/msk_configuration
- msk_kms_key in ../../modules/kms_key
- mskconnect_log_group in ../../modules/cloudwatch_log_group
- nat_gateway in ../../modules/nat_gateway
- nat_gateway_route in ../../modules/route
- private_route_table in ../../modules/route_table
- private_route_table_association in ../../modules/route_table_association
- private_subnet_1a in ../../modules/subnet
- private_subnet_1b in ../../modules/subnet
- private_subnet_1c in ../../modules/subnet
- public_route_table in ../../modules/route_table
- public_route_table_association in ../../modules/route_table_association
- public_subnet_1a in ../../modules/subnet
- public_subnet_1b in ../../modules/subnet
- public_subnet_1c in ../../modules/subnet
- s3_object in ../../modules/s3_object
- s3_object_file_upload in ../../modules/s3_object
- sales_db_parameter_group in ../../modules/db_parameter_group
- sales_db_rds_cluster in ../../modules/rds_cluster
- sales_db_rds_cluster_instance in ../../modules/rds_cluster_instance
- sales_db_rds_cluster_parameter_group in ../../modules/rds_cluster_parameter_group
- sales_db_subnet_group in ../../modules/db_subnet_group
- security_group in ../../modules/security_group
- time_sleep in ../../modules/time_sleep
- vpc in ../../modules/vpc
- vpc_dhcp_options in ../../modules/vpc_dhcp_options
- vpc_dhcp_options_assoc in ../../modules/vpc_dhcp_options_assoc
- vpc_security_group_rule in ../../modules/vpc_security_group_rule

Initializing provider plugins...
- Finding hashicorp/local versions matching "~> 2.3.0"...
- Finding latest version of hashicorp/template...
- Finding latest version of hashicorp/time...
- Finding hashicorp/aws versions matching "5.23.1"...
- Installing hashicorp/local v2.3.0...
- Installed hashicorp/local v2.3.0 (signed by HashiCorp)
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (signed by HashiCorp)
- Installing hashicorp/time v0.9.1...
- Installed hashicorp/time v0.9.1 (signed by HashiCorp)
- Installing hashicorp/aws v5.23.1...
- Installed hashicorp/aws v5.23.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
[dev01@houssitdvm02 proj3]$ terraform apply
```

Once terraform apply execution completes, terraform outputs some values, we will use the value of the output shown below:

sales_db_rds_cluster_instance_endpoint – Aurora DB endpoint used to connect to the DB instance via Mysql coomand from Kafka Client Instance.

3.0 Verify MSKConnector has performed an initial creation of an empty salesdb database Kafka Topic and an internal schema history Topic

Get logged on the kafka client instance

```
[dev01@houssitdvm02 proj3]$ ssh -i "kafka_client_instance-key.pem" ec2-user@ec2-34-230-39-175.compute-1.amazonaws.com

Warning: Identity file kafka_client_instance-key.pem not accessible: No such file or directory.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  7 10:51:36 2023 from 81.153.115.90
[ec2-user@ip-10-0-1-176 ~]$
```

Switch to kafka user

```
[ec2-user@ip-10-0-1-176 ~]$ sudo su - kafka
[kafka@ip-10-0-1-176 ~]$
```

List of kafka Topics showing internal schema history topic and the empty salesdb topic

```
[kafka@ip-10-0-1-176 ~]$ bin/kafka-topics.sh --list --bootstrap-server $MSK_BOOTSTRAP_IAM_ADDRESS --command-config /etc/kafka/security/iam/client.properties
__amazon_msk_canary
__amazon_msk_connect_configs_debezium-mysql-source-connector_e6ef2014-7c9e-441b-bbf6-c752e70f4112-4
__amazon_msk_connect_status_debezium-mysql-source-connector_e6ef2014-7c9e-441b-bbf6-c752e70f4112-4
__consumer_offsets
internal.dbhistory.salesdb
offsets_my_debezium_source_connector
salesdb
[kafka@ip-10-0-1-176 ~]$
```

4.0 Load Data into SalesDB database

Get logged on the kafka client instance on another terminal

```
[dev01@houssitdvm02 proj3]$ ssh -i "kafka_client_instance-key.pem" ec2-user@ec2-34-230-39-175.compute-1.amazonaws.com

Warning: Identity file kafka_client_instance-key.pem not accessible: No such file or directory.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  7 10:51:36 2023 from 81.153.115.90
[ec2-user@ip-10-0-1-176 ~]$
```

Switch to dba user and logon to the Aurora database instance

```
[ec2-user@ip-10-0-1-176 ~]$ sudo su – dbauser01

[dbauser01@ip-10-0-1-176 ~]$ mysql -u master -h <output of sales_db_rds_cluster_instance_endpoint >  -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 410
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Populate salesdb database

```
mysql> source /home/dba/dbauser01/scripts/salesdb.sql
Query OK, 0 rows affected (0.00 sec)
:
Database changed
:
Query OK, 0 rows affected (0.01 sec)
:
Query OK, 2000 rows affected (0.07 sec)
Records: 2000  Duplicates: 0  Warnings: 0
:
```

```
Query OK, 4000 rows affected (0.11 sec)
Records: 4000  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
:
Query OK, 1000 rows affected (0.03 sec)
Records: 1000  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
:
Query OK, 15 rows affected (0.01 sec)
Records: 15  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
:
Query OK, 23001 rows affected (0.53 sec)
Records: 23001  Duplicates: 0  Warnings: 0
Query OK, 5999 rows affected (0.13 sec)
Records: 5999  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.01 sec)
:
Query OK, 23003 rows affected (0.59 sec)
Records: 23003  Duplicates: 0  Warnings: 0
Query OK, 5997 rows affected (0.11 sec)
Records: 5997  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
:
Query OK, 13746 rows affected (0.56 sec)
Records: 13746  Duplicates: 0  Warnings: 0
Query OK, 13603 rows affected (0.37 sec)
Records: 13603  Duplicates: 0  Warnings: 0
Query OK, 13603 rows affected (0.39 sec)
Records: 13603  Duplicates: 0  Warnings: 0
Query OK, 13480 rows affected (0.50 sec)
Records: 13480  Duplicates: 0  Warnings: 0
Query OK, 13238 rows affected (0.33 sec)
Records: 13238  Duplicates: 0  Warnings: 0
Query OK, 13239 rows affected (0.47 sec)
Records: 13239  Duplicates: 0  Warnings: 0
Query OK, 13188 rows affected (0.35 sec)
Records: 13188  Duplicates: 0  Warnings: 0
Query OK, 3903 rows affected (0.13 sec)
Records: 3903  Duplicates: 0  Warnings: 0
:
Records: 0  Duplicates: 0  Warnings: 0
Query OK, 98000 rows affected (3.94 sec)
Rows matched: 98000  Changed: 98000  Warnings: 0
:
Query OK, 98000 rows affected (3.14 sec)
Records: 98000  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
:
Query OK, 100 rows affected (0.01 sec)
Records: 100  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)

mysql>
```

5.0 Verify MSKConnector creation of Kafka topic for each salesdb database table

Still logged on the first terminal as Kafka user, list kafka topics showing salesdb tables

```
[kafka@ip-10-0-1-176 ~]$ bin/kafka-topics.sh --list --bootstrap-server
$MSK_BOOTSTRAP_IAM_ADDRESS  --command-config /etc/kafka/security/iam/client.properties
__amazon_msk_canary
__amazon_msk_connect_configs_debezium-mysql-source-connector_e6ef2014-7c9e-441b-bbf6-
c752e70f4112-4
__amazon_msk_connect_status_debezium-mysql-source-connector_e6ef2014-7c9e-441b-bbf6-
c752e70f4112-4
__consumer_offsets
internal.dbhistory.salesdb
offsets_my_debezium_source_connector
salesdb
salesdb.salesdb.CUSTOMER
salesdb.salesdb.CUSTOMER_SITE
salesdb.salesdb.PRODUCT
salesdb.salesdb.PRODUCT_CATEGORY
salesdb.salesdb.SALES_ORDER
salesdb.salesdb.SALES_ORDER_ALL
salesdb.salesdb.SALES_ORDER_DETAIL
salesdb.salesdb.SALES_ORDER_DETAIL_DS
salesdb.salesdb.SUPPLIER
[kafka@ip-10-0-1-176 ~]$
```

6.0 Verify MSKConnector replication of salesdb database to salesdb Kafka topic

Start a Kafka consumer for the salesdb topic, using a new terminal console

```
[kafka@ip-10-0-1-176 ~]$ bin/kafka-console-consumer.sh --bootstrap-server $MSK_BOOTSTRAP_IAM_ADDRESS --consumer.config /etc/kafka/security/iam/client.properties --from-beginning --topic salesdb | jq --color-output
.
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699352599964,
    "snapshot": "last",
    "db": "",
    "sequence": null,
    "table": null,
    "server_id": 0,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 153,
    "row": 0,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699352600965,
  "databaseName": "",
  "schemaName": null,
  "ddl": "SET character_set_server=utf8mb4, collation_server=utf8mb4_0900_ai_ci",
  "tableChanges": []
}
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": null,
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 226,
    "row": 0,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699354624446,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "CREATE DATABASE IF NOT EXISTS salesdb",
  "tableChanges": []
}
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 419,
    "row": 0,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699354624578,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "CREATE USER 'debezium'@'%' IDENTIFIED WITH 'mysql_native_password' AS
'*A06791A44B817A1A84D38FABE76938F0183EE3F0'",
  "tableChanges": []
}
.
.
.
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 184859,
    "row": 165,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699354625450,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "/*!40000 ALTER TABLE `CUSTOMER_SITE` DISABLE KEYS */",
  "tableChanges": []
}
.
.
.
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354636000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53807579,
    "row": 99,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699354660341,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "/*!50001 DROP VIEW IF EXISTS `SALES_ORDER_V`*/",
  "tableChanges": []
}
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354636000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "SALES_ORDER_V",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53807788,
    "row": 99,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699354660768,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "CREATE ALGORITHM=UNDEFINED DEFINER=`master`@`%` SQL SECURITY DEFINER VIEW `SALES_ORDER_V` AS select `SALES_ORDER`.`ORDER_ID` AS `ORDER_ID`,`SALES_ORDER`.`SITE_ID` AS `SITE_ID`,`SALES_ORDER`.`ORDER_DATE` AS `ORDER_DATE`,`SALES_ORDER`.`SHIP_MODE` AS `SHIP_MODE` from `SALES_ORDER` where (`SALES_ORDER`.`ORDER_DATE` < str_to_date('2015-02-01 00:00:00.0','%Y-%m-%d %H:%i:%s '))",
  "tableChanges": []
}
```

7.0 Verify MSKConnector replication of salesdb Customer table to Kafka Customer topic

Start a Kafka consumer for the salesdb Customer topic

On another terminal start a kafka consumer to verify salesdb Customer topic has been populated

```
[kafka@ip-10-0-1-176 ~]$ bin/kafka-console-consumer.sh --bootstrap-server $MSK_BOOTSTRAP_IAM_ADDRESS --consumer.config /etc/kafka/security/iam/client.properties --from-beginning --topic salesdb.salesdb.CUSTOMER | jq --color-output .
{
  "before": null,
  "after": {
    "CUST_ID": 4.0,
    "NAME": "Customer Name 4",
    "MKTSEGMENT": "Market Segment 5"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 94415,
    "row": 3,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699354625057,
  "transaction": null
}
.
.
.

“source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 176339,
    "row": 162,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699354625364,
  "transaction": null
}
{
  "before": null,
  "after": {
    "CUST_ID": 2000.0,
    "NAME": "Customer Name 2000",
    "MKTSEGMENT": "Market Segment 7"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699354624000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 176339,
    "row": 165,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699354625364,
  "transaction": null
}
```

8.0 Verify MSKConnector replication of salesdb schema change

Add a new table to salesdb table

On the terminal where you have your connection to the Aurora database instance, execute the script below

```
mysql> USE salesdb;
Database changed
mysql> CREATE TABLE SALES_ORDER_VI (
   ->      order_id VARCHAR(255),
   ->      customer_id VARCHAR(255),
   ->      item_description VARCHAR(255),
   ->      price DECIMAL(6,2),
   ->      order_date DATETIME DEFAULT CURRENT_TIMESTAMP
   -> );
Query OK, 0 rows affected (0.07 sec)
```

Verify the schema change has been replicated on the same terminal where you started the salesdb Kafka consumer

```
{
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699355300000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "SALES_ORDER_VI",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53808324,
    "row": 99,
    "thread": null,
    "query": null
  },
  "ts_ms": 1699355300675,
  "databaseName": "salesdb",
  "schemaName": null,
  "ddl": "CREATE TABLE SALES_ORDER_VI (\n    order_id VARCHAR(255),\n    customer_id VARCHAR(255),\n    item_description VARCHAR(255),\n    price DECIMAL(6,2),\n    order_date DATETIME DEFAULT CURRENT_TIMESTAMP\n)",
  "tableChanges": [
    {
      "type": "CREATE",
      "id": "\"salesdb\".\"SALES_ORDER_VI\"",
      "table": {
        "defaultCharsetName": "utf8mb4",
        "primaryKeyColumnNames": [],
        "columns": [
{
  "name": "order_id",
  "jdbcType": 12,
  "nativeType": null,
  "typeName": "VARCHAR",
  "typeExpression": "VARCHAR",
  "charsetName": "utf8mb4",
  "length": 255,
  "scale": null,
  "position": 1,
  "optional": true,
  "autoIncremented": false,
  "generated": false,
  "comment": null,
  "defaultValueExpression": null,
  "enumValues": null
},
{
  "name": "customer_id",
  "jdbcType": 12,
  "nativeType": null,
  "typeName": "VARCHAR",
  "typeExpression": "VARCHAR",
  "charsetName": "utf8mb4",
  "length": 255,
  "scale": null,
  "position": 2,
  "optional": true,
  "autoIncremented": false,
  "generated": false,
  "comment": null,
  "defaultValueExpression": null,
  "enumValues": null
},
{
  "name": "item_description",
  "jdbcType": 12,
  "nativeType": null,
  "typeName": "VARCHAR",
  "typeExpression": "VARCHAR",
  "charsetName": "utf8mb4",
  "length": 255,
  "scale": null,
  "position": 3,
  "optional": true,
  "autoIncremented": false,
  "generated": false,
  "comment": null,
  "defaultValueExpression": null,
  "enumValues": null
},
{
  "name": "price",
  "jdbcType": 3,
  "nativeType": null,
  "typeName": "DECIMAL",
  "typeExpression": "DECIMAL",
            "charsetName": null,
            "length": 6,
            "scale": 2,
            "position": 4,
            "optional": true,
            "autoIncremented": false,
            "generated": false,
            "comment": null,
            "defaultValueExpression": null,
            "enumValues": null
          },
          {
            "name": "order_date",
            "jdbcType": 93,
            "nativeType": null,
            "typeName": "DATETIME",
            "typeExpression": "DATETIME",
            "charsetName": null,
            "length": null,
            "scale": null,
            "position": 5,
            "optional": true,
            "autoIncremented": false,
            "generated": false,
            "comment": null,
            "defaultValueExpression": "1970-01-01 00:00:00",
            "enumValues": null
          }
        ],
        "comment": null
      }
    }
  ]
}
```

9.0 Verify MSKConnector replication of salesdb update change

Insert some records into the salesdb customer table

On the terminal where you are maintaining your SQL connection to the Aurora database instance, execute below database inserts

```
mysql> INSERT INTO CUSTOMER VALUES ('2001', 'Customer Name 2001', 'Market Segment 7');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER VALUES ('2002', 'Customer Name 2002', 'Market Segment 1');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER VALUES ('2003', 'Customer Name 2003', 'Market Segment 3');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER VALUES ('2004', 'Customer Name 2004', 'Market Segment 5');
Query OK, 1 row affected (0.01 sec)

mysql>
```

Verify the update change has been replicated on the same terminal where you started the salesdb Kafka consumer

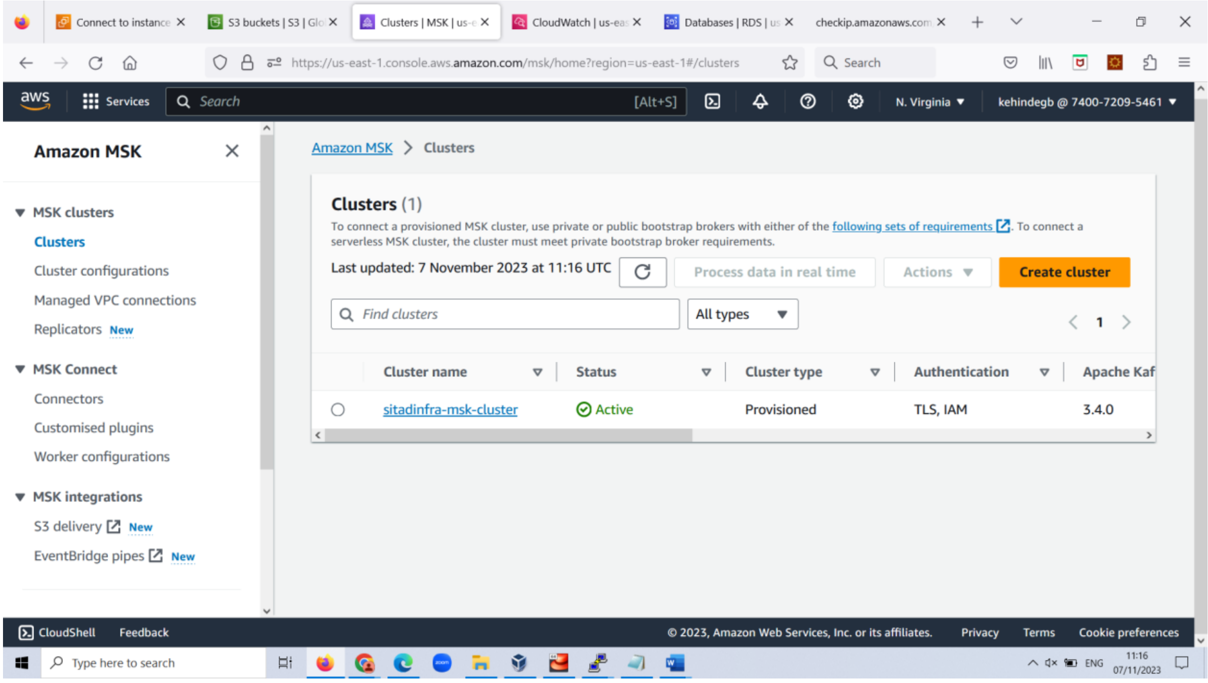
```
{
  "before": null,
  "after": {
    "CUST_ID": 2003.0,
    "NAME": "Customer Name 2003",
    "MKTSEGMENT": "Market Segment 3"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699355421000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53809762,
    "row": 0,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699355421043,
  "transaction": null
}
```

```
{
  "before": null,
  "after": {
    "CUST_ID": 2001.0,
    "NAME": "Customer Name 2001",
    "MKTSEGMENT": "Market Segment 7"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699355421000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53808936,
    "row": 0,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699355421031,
  "transaction": null
}
{
  "before": null,
  "after": {
    "CUST_ID": 2002.0,
    "NAME": "Customer Name 2002",
    "MKTSEGMENT": "Market Segment 1"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699355421000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53809349,
    "row": 0,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699355421036,
  "transaction": null
}
```

```
{
  "before": null,
  "after": {
    "CUST_ID": 2004.0,
    "NAME": "Customer Name 2004",
    "MKTSEGMENT": "Market Segment 5"
  },
  "source": {
    "version": "2.3.4.Final",
    "connector": "mysql",
    "name": "salesdb",
    "ts_ms": 1699355422000,
    "snapshot": "false",
    "db": "salesdb",
    "sequence": null,
    "table": "CUSTOMER",
    "server_id": 1454232132,
    "gtid": null,
    "file": "mysql-bin-changelog.000003",
    "pos": 53810175,
    "row": 0,
    "thread": 410,
    "query": null
  },
  "op": "c",
  "ts_ms": 1699355422404,
  "transaction": null
}
```

Appendix

For completeness we provide some screen shots of the solution below:



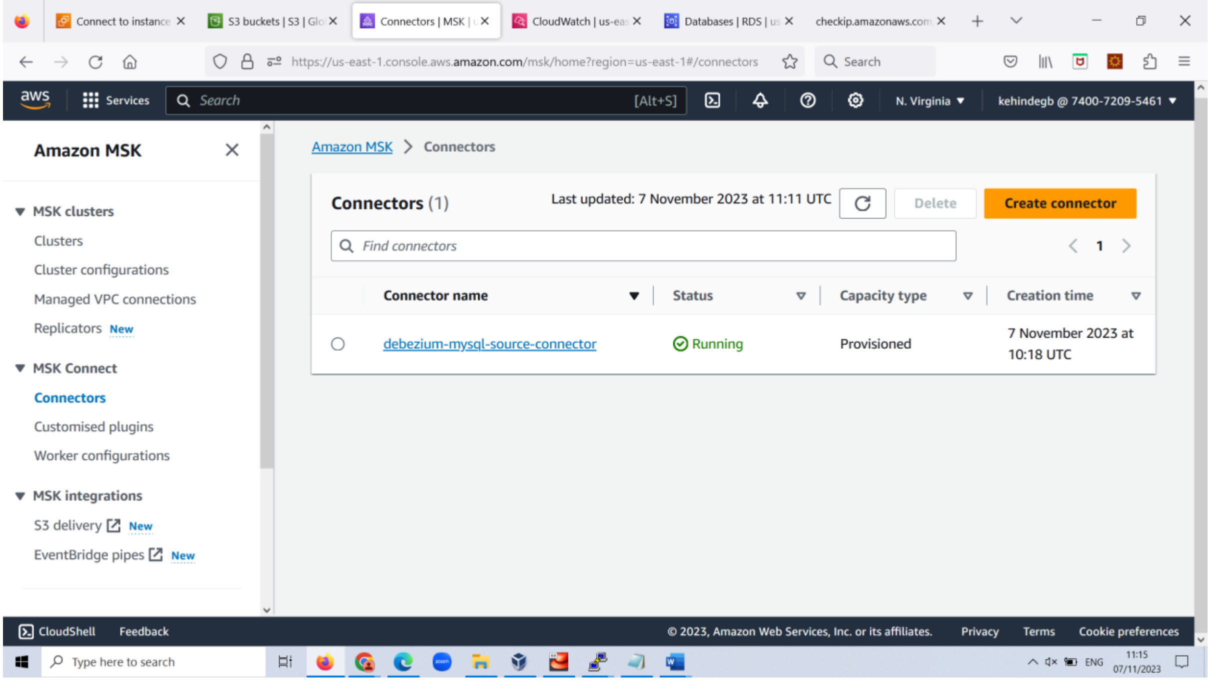
Clusters (1)

To connect a provisioned MSK cluster, use private or public bootstrap brokers with either of the following sets of requirements. To connect a serverless MSK cluster, the cluster must meet private bootstrap broker requirements.

Last updated: 7 November 2023 at 11:16 UTC

	Cluster name	Status	Cluster type	Authentication	Apache Kaf
○	sitadinfra-msk-cluster	Active	Provisioned	TLS, IAM	3.4.0

Fig 1: MSK Cluster with TLS and IAM Authentication

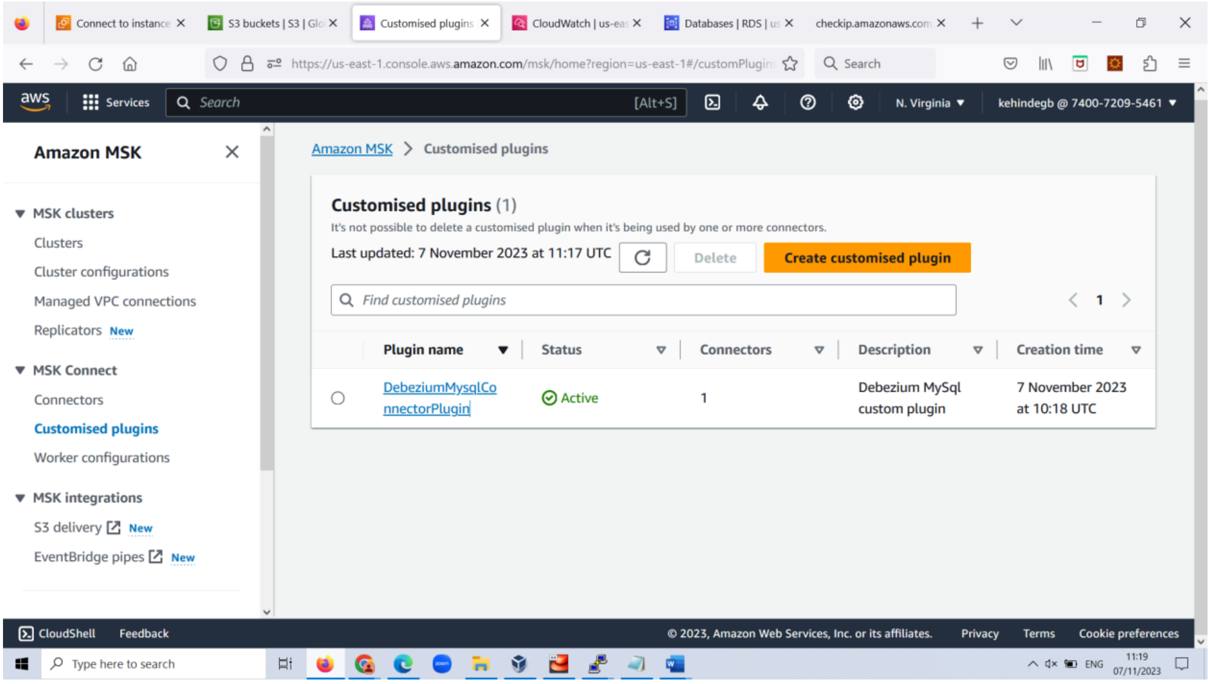


Connectors (1)

Last updated: 7 November 2023 at 11:11 UTC

	Connector name	Status	Capacity type	Creation time
○	debezium-mysql-source-connector	Running	Provisioned	7 November 2023 at 10:18 UTC

Fig 2: Depicts MSKConnector in running state



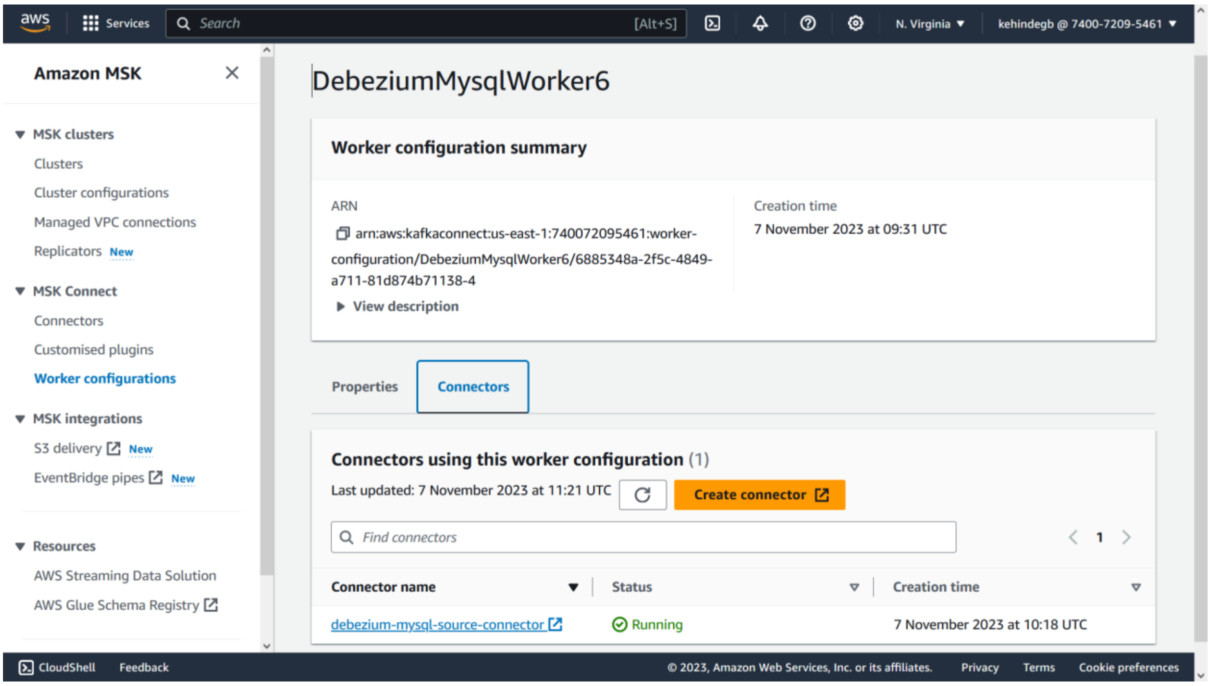
Customised plugins (1)

It's not possible to delete a customised plugin when it's being used by one or more connectors.

Last updated: 7 November 2023 at 11:17 UTC

	Plugin name	Status	Connectors	Description	Creation time
○	DebeziumMysqlConnectorPlugin	Active	1	Debezium MySql custom plugin	7 November 2023 at 10:18 UTC

Fig 3: Shows Debezium MySql plugin in a active state.



DebeziumMysqlWorker6

Worker configuration summary

ARN
arn:aws:kafkaconnect:us-east-1:740072095461:worker-configuration/DebeziumMysqlWorker6/6885348a-2f5c-4849-a711-81d874b71138-4

▶ View description

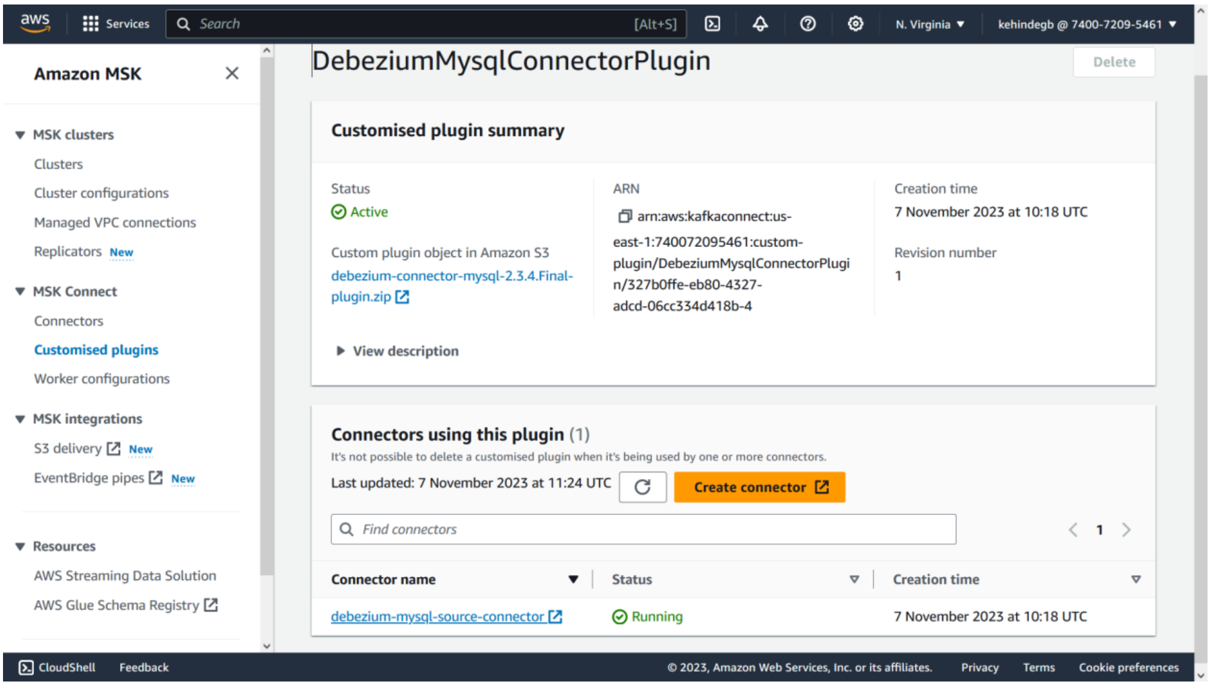
Creation time
7 November 2023 at 09:31 UTC

Connectors using this worker configuration (1)

Last updated: 7 November 2023 at 11:21 UTC

Connector name	Status	Creation time
debezium-mysql-source-connector	Running	7 November 2023 at 10:18 UTC

Fig 4: Shows Worker and Connector association



DebeziumMysqlConnectorPlugin

Customised plugin summary

Status
✓ Active

Custom plugin object in Amazon S3
debezium-connector-mysql-2.3.4.Final-plugin.zip

ARN
arn:aws:kafkaconnect:us-east-1:740072095461:custom-plugin/DebeziumMysqlConnectorPlugin/327b0ffe-eb80-4327-adcd-06cc334d418b-4

Creation time
7 November 2023 at 10:18 UTC

Revision number
1

Connectors using this plugin (1)

It's not possible to delete a customised plugin when it's being used by one or more connectors.

Last updated: 7 November 2023 at 11:24 UTC

Connector name	Status	Creation time
debezium-mysql-source-connector	✓ Running	7 November 2023 at 10:18 UTC

Fig 5: Shows plugin and connector association