Integration of AWS Redshift with CipherTrust REST Data Protection (CRDP)

AWS Redshift [Overview]

This document describes how to configure and integrate CipherTrust Manager with AWS Redshift.

Amazon Redshift is a data warehouse product which forms part of the larger cloud-computing platform Amazon Web Services. It is built on top of technology from the massive parallel processing (MPP) data warehouse company ParAccel to handle large scale data sets and database migrations. Redshift differs from Amazon's other hosted database offering, Amazon RDS, in its ability to handle analytic workloads on big data data sets stored by a column-oriented DBMS principle. Redshift allows up to 16 petabytes of data on a cluster compared to Amazon RDS Aurora's maximum size of 128 tebibytes. Thales provides a couple of different methods to protect sensitive data in AWS Redshift.

Bring Your Own Encryption (BYOE)

- **Data Ingest** – with Thales Batch Data Transformation (BDT)
- **Data Access** – external functions for column level encrypt and decryption using Ciphertrust REST Data Protection (CRDP) Protect/Reveal API's.

Bring/Hold Your Own Key (BYOK) (HYOK)

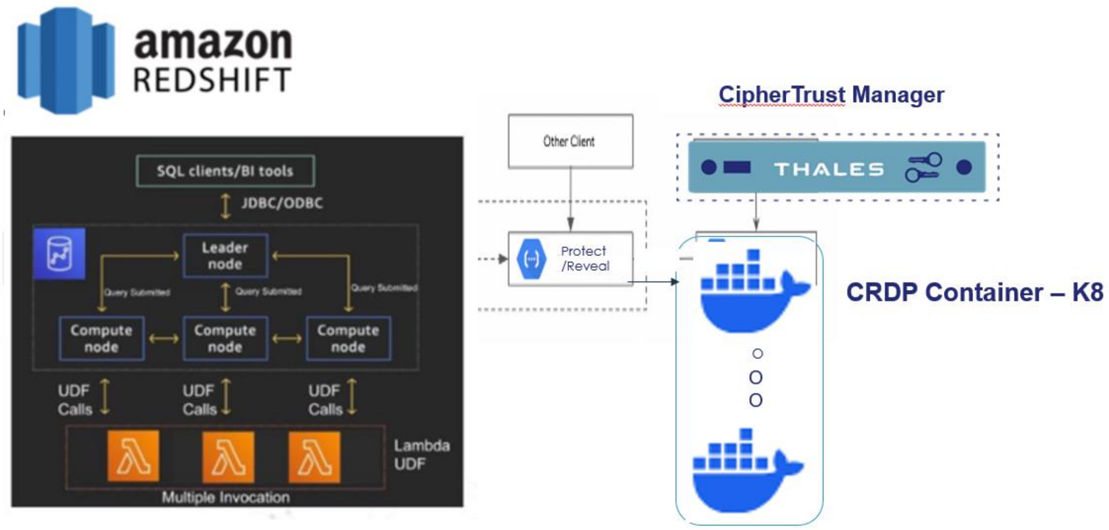
- **AWS Redshift Customer Managed Keys-** with Thales CM CCKM BYOK and HYOK.

The above methods are NOT mutually exclusive. All methods can be used to build a strong defense in depth strategy to protect sensitive data in the cloud. The focus of this integration will be on Data Access protecting sensitive data in AWS Redshift columns by using CRDP to create User Defined Functions (UDF) for encryption and decryption of sensitive data.

Architecture

The examples provided in this document use a capability AWS Redshift called "External Function". A Redshift external function lets you incorporate functionality with software outside of Redshift by providing a direct integration with AWS Lambda Functions. A Redshift external function allows you to implement your function in other languages other than SQL. Listed below is a diagram of how this integration works.

AWS Redshift integration with Thales CRDP (AWS Lambda Function)



Supported Product Versions

- **CipherTrust Manager** CipherTrust Manager 2.14 and higher
- **CRDP** 1.0 and higher
- **AWS Redshift**

This integration is validated AWS Lambda Functions and Java 11 along with CM 2.14

Prerequisites

Steps performed for this integration were provided by this AWS link:
https://docs.aws.amazon.com/redshift/latest/dg/udf-creating-a-lambda-sql-udf.html

https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_EXTERNAL_FUNCTION.html

- Ensure that CRDP for is installed and configured. Refer to https://thalesdocs.com/ctp/con/crdp/latest/admin/index.html
- Ensure that the CipherTrust Manager is installed and configured. Refer to the **CipherTrust Manager documentation** for details.
- AWS Lambda functions communicates with the CRDP Container using REST API's

Steps for Integration

- **[Installing and Configuring Thales CRDP for Java]**
- **[Download code from Thales github and compile]**
- **[Publish jar/zip file to AWS Lambda Function (endpoint)]**
- **[Create AWS Redshift External Function]**
- **[Integration with Thales CipherTrust Manager]**
- **[Environment Variables]**

Installing and Configuring CRDP Container

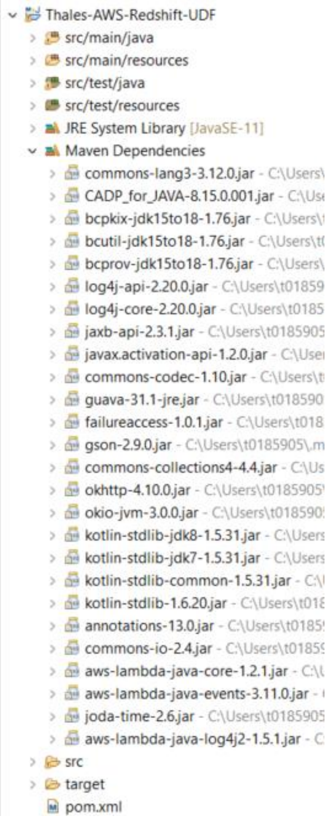
To install and configure **CDRP,** refer to Quick Start.

Eclipse development tool was used for these examples. Here is the version used for testing along with the Maven plugin for Eclipse.

eclipse.buildId=4.15.0.I20200305-0155

m2e - Maven Integration for Eclipse

Here is a screenshot in eclipse of the pom file used for these examples:

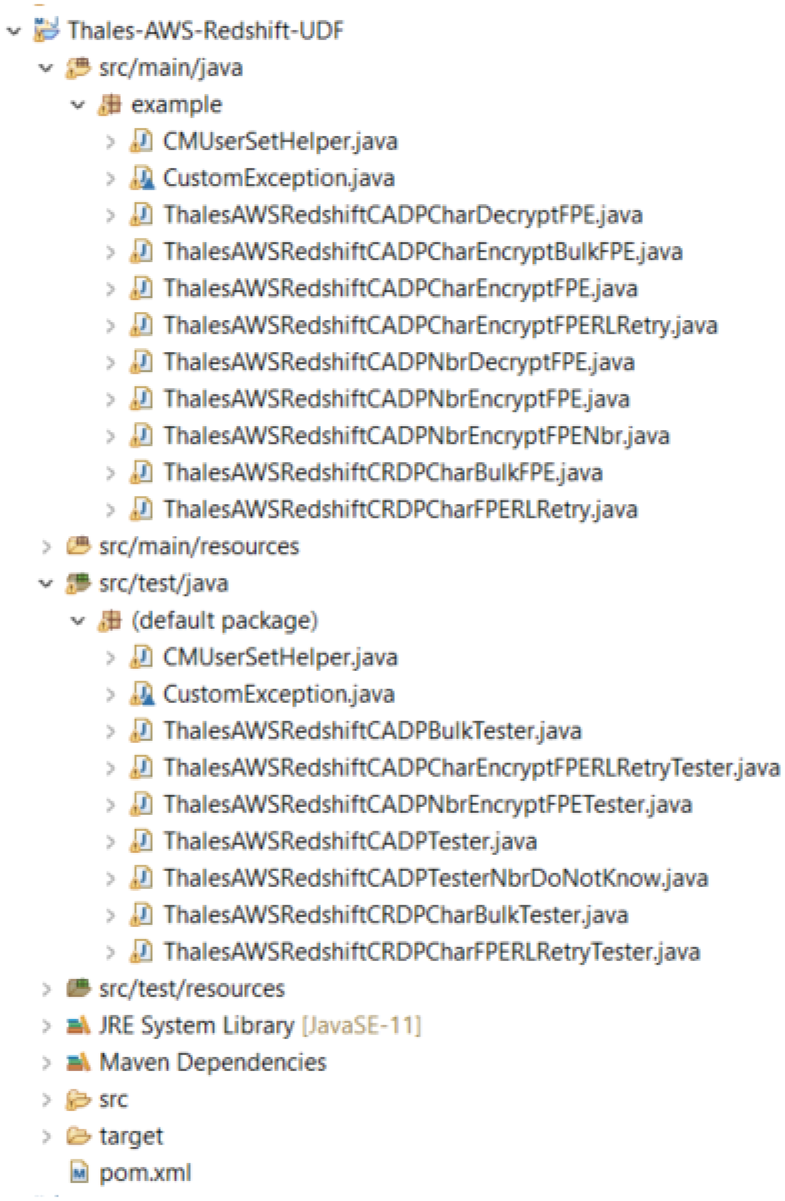


```
Thales-AWS-Redshift-UDF
  src/main/java
  src/main/resources
  src/test/java
  src/test/resources
  JRE System Library [JavaSE-11]
  Maven Dependencies
    commons-lang3-3.12.0.jar - C:\Users\
    CADP_for_JAVA-8.15.0.001.jar - C:\Us
    bcpkix-jdk15to18-1.76.jar - C:\Users\
    bcutil-jdk15to18-1.76.jar - C:\Users\t
    bcprov-jdk15to18-1.76.jar - C:\Users\
    log4j-api-2.20.0.jar - C:\Users\t01859
    log4j-core-2.20.0.jar - C:\Users\t0185
    jaxb-api-2.3.1.jar - C:\Users\t0185905
    javax.activation-api-1.2.0.jar - C:\Use
    commons-codec-1.10.jar - C:\Users\t
    guava-31.1-jre.jar - C:\Users\t018590
    failureaccess-1.0.1.jar - C:\Users\t018
    gson-2.9.0.jar - C:\Users\t0185905\.m
    commons-collections4-4.4.jar - C:\Us
    okhttp-4.10.0.jar - C:\Users\t0185905
    okio-jvm-3.0.0.jar - C:\Users\t018590
    kotlin-stdlib-jdk8-1.5.31.jar - C:\Users
    kotlin-stdlib-jdk7-1.5.31.jar - C:\Users
    kotlin-stdlib-common-1.5.31.jar - C:\
    kotlin-stdlib-1.6.20.jar - C:\Users\t01
    annotations-13.0.jar - C:\Users\t0185
    commons-io-2.4.jar - C:\Users\t0185
    aws-lambda-java-core-1.2.1.jar - C:\
    aws-lambda-java-events-3.11.0.jar -
    joda-time-2.6.jar - C:\Users\t0185905
    aws-lambda-java-log4j2-1.5.1.jar - C
  src
  target
  pom.xml
```

Download code from github and compile.

git clone https://github.com/ThalesGroup/CipherTrust_Application_Protection.git

The database directory has all the code for AWS Redshift. You should see the classes in your project.



```
v Thales-AWS-Redshift-UDF
  v src/main/java
    v example
      > CMUserSetHelper.java
      > CustomException.java
      > ThalesAWSRedshiftCADPCharDecryptFPE.java
      > ThalesAWSRedshiftCADPCharEncryptBulkFPE.java
      > ThalesAWSRedshiftCADPCharEncryptFPE.java
      > ThalesAWSRedshiftCADPCharEncryptFPERLRetry.java
      > ThalesAWSRedshiftCADPNbrDecryptFPE.java
      > ThalesAWSRedshiftCADPNbrEncryptFPE.java
      > ThalesAWSRedshiftCADPNbrEncryptFPENbr.java
      > ThalesAWSRedshiftCRDPCharBulkFPE.java
      > ThalesAWSRedshiftCRDPCharFPERLRetry.java
  > src/main/resources
  v src/test/java
    v (default package)
      > CMUserSetHelper.java
      > CustomException.java
      > ThalesAWSRedshiftCADPBulkTester.java
      > ThalesAWSRedshiftCADPCharEncryptFPERLRetryTester.java
      > ThalesAWSRedshiftCADPNbrEncryptFPETester.java
      > ThalesAWSRedshiftCADPTester.java
      > ThalesAWSRedshiftCADPTesterNbrDoNotKnow.java
      > ThalesAWSRedshiftCRDPCharBulkTester.java
      > ThalesAWSRedshiftCRDPCharFPERLRetryTester.java
  > src/test/resources
  > JRE System Library [JavaSE-11]
  > Maven Dependencies
  > src
  > target
    pom.xml
```

CRDP supports a bulk API which allows for CRDP to batch requests before calling protect or reveal. A separate class file for each datatype is available for testing with this API. There is a limit of 10,000 items to protect but it also depends on the size of the items as well so review the CRDP documentation to make sure those thresholds are not exceeded. Typically, the point of diminishing returns is around 200.

Assuming you have your CM and your CRDP container already configured the `ThalesAWSRedshiftCRDPCharFPERLRetryTester` can be used to test basic connection to the CRDP container to make sure your CM environment is configured correctly. You will need to modify the keyname to make sure it exists in CM and there are environment variables that must be set as well. If you plan on using UserSets the user must also be in the Application Data Protection Client Group. See section on Environment Variables for more details.

Generate the jar file to upload to the CSP.

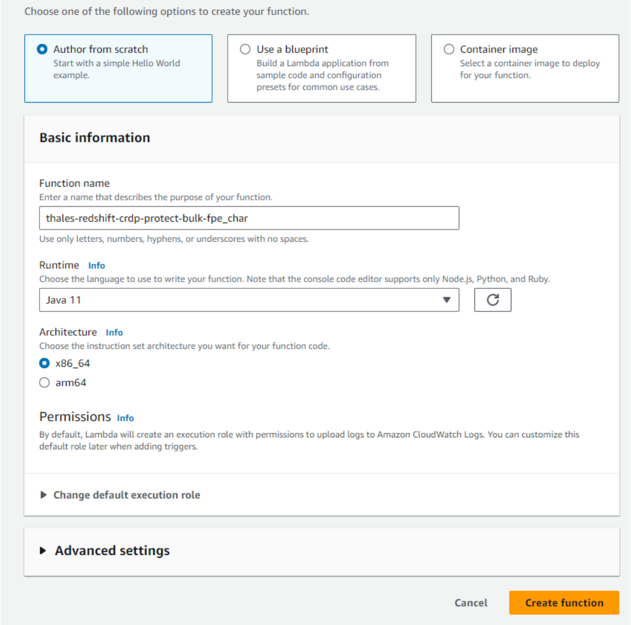
To compile and generate the target jar file to be uploaded to AWS Lambda select the project and choose “Run As” “maven install” to generate the target.

```
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing C:\Users\t0185905\workspace\Thales-AWS-Redshift-UDF\target\Thales-
AWS-Redshift-UDF-0.0.1-SNAPSHOT.jar with C:\Users\t0185905\workspace\Thales-AWS-
Redshift-UDF\target\Thales-AWS-Redshift-UDF-0.0.1-SNAPSHOT-shaded.jar
[INFO]
[INFO] --- maven-install-plugin:3.0.1:install (default-install) @ Thales-AWS-
Redshift-UDF ---
[INFO] Installing C:\Users\t0185905\workspace\Thales-AWS-Redshift-UDF\pom.xml to
C:\Users\t0185905\.m2\repository\ThalesCPL\Thales-AWS-Redshift-UDF\0.0.1-
SNAPSHOT\Thales-AWS-Redshift-UDF-0.0.1-SNAPSHOT.pom
[INFO] Installing C:\Users\t0185905\workspace\Thales-AWS-Redshift-UDF\target\Thales-
AWS-Redshift-UDF-0.0.1-SNAPSHOT.jar to
C:\Users\t0185905\.m2\repository\ThalesCPL\Thales-AWS-Redshift-UDF\0.0.1-
SNAPSHOT\Thales-AWS-Redshift-UDF-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.056 s
[INFO] Finished at: 2024-07-10T12:26:14-04:00
[INFO] ------------------------------------------------------------------------
```

Publish jar/zip file to AWS Lambda Function. (endpoint)

Once you have generated the jar file to upload you can then create the CSP function.

AWS Lambda Function



Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
thales-redshift-crdp-protect-bulk-fpe_char
Use only letters, numbers, hyphens, or underscores with no spaces.

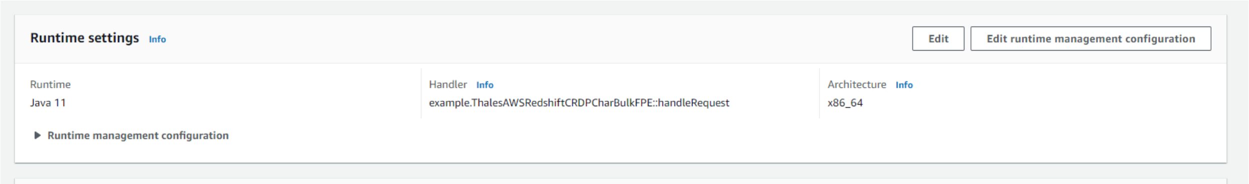
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Java 11

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Click Create function.

Set environment variables appropriate values (See section on Environment Variables for details) and then select Next.   Upload the jar or zip file on the next screen. Be sure to change the entry point to reflect your class name. ***The example below is*** `ThalesAWSRedshiftCRDPCharBulkFPE`***This should match your code path.***



Runtime settings Info

Edit
Edit runtime management configuration

Runtime
Java 11

Handler Info
example.ThalesAWSRedshiftCRDPCharBulkFPE::handleRequest

Architecture Info
x86_64

▶ Runtime management configuration

Be sure to have a role that allows for AWSLambdaBasicExecutionRole.   512MB of RAM is plenty. You can monitor your tests and possibly lower it.

Once you have created the functions above and if you have already configured and setup CM with the key and all the environment variables you can test the function with the test tab. You will need to provide the appropriate json to test. Here is an example:

```
{
  "request_id": "23FF1F97-F28A-44AA-AB67-266ED976BF40",
  "cluster": "arn:aws:redshift:xxxx",
  "user": "adminuser",
  "database": "db1",
  "external_function": "public.foo",
  "query_id": 5678234,
  "num_records": 3,
  "arguments": [
    [
      "The. king. and I me"
    ],
    [
      "The mroe data "
    ],
    [
      "test dada"
    ]
  ]
}
```

Create AWS Redshift External Function.

Here are some links that provide details.

https://docs.aws.amazon.com/redshift/latest/dg/udf-creating-a-lambda-sql-udf.html

https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_EXTERNAL_FUNCTION.html

https://aws-dojo.com/excercises/excercise31/

As noted above the steps are:

1. Create the AWS Lambda Function. (should already be done from above)
2. Create external function object in AWS Redshift

Examples:

Redshift Function Definitions.

The functions are created using the following format:

```
CREATE OR REPLACE EXTERNAL FUNCTION
public.thales_crdp_protect_bulk_fpe_char(sensitivedata varchar)
 RETURNS varchar
 STABLE
lambda 'thales-redshift-crdp-protect-bulk-fpe-char'
iam_role 'arn:aws:iam::yoruproject:role/redshiftpoweruseraccess'

CREATE OR REPLACE EXTERNAL FUNCTION
public.thales_crdp_protect_fpe_char(sensitivedata varchar)
 RETURNS varchar
 STABLE
```

```
lambda 'thales-redshift-crdp-protect-fpe-char'
iam_role 'arn:aws:iam:: yoruproject:role/redshiftpoweruseraccess'
```

Integration with CipherTrust Manager.

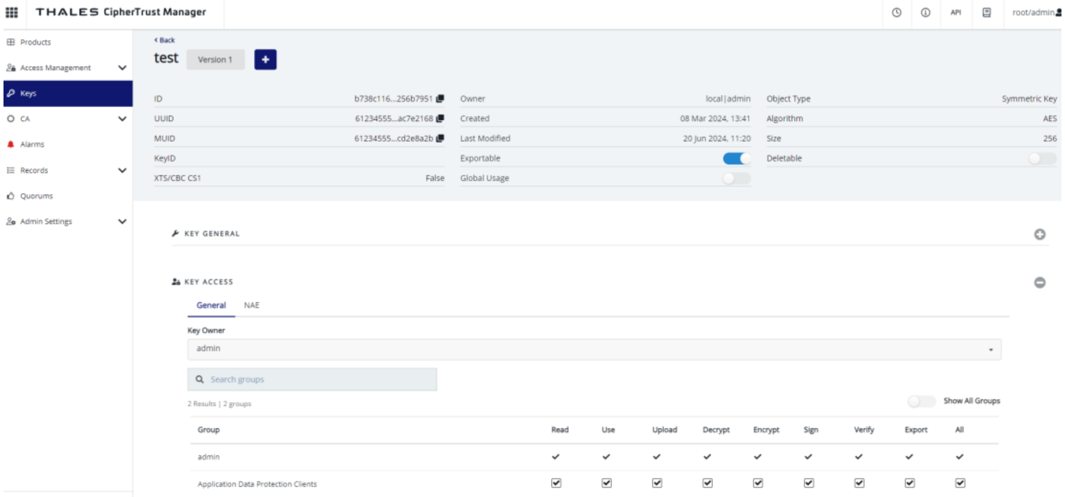
Here is a link to a demo that shows all the steps required to setup CRDP in CM.

thales.navattic.com/thalesprotectreveal

Here is a list of the steps.

1. Create an Encryption Key
2. Create a CRDP Application
3. Create a User Set
4. Create an Access Policy
5. Create a Protection Policy
6. Perform Integration

When creating a key to be used by the protection policy make sure that it is in the Application Data Protection Clients Group.



test

ID	b738c116...256b7951	Owner	local|admin	Object Type	Symmetric Key
UUID	61234555...ac7e2168	Created	08 Mar 2024, 13:41	Algorithm	AES
MUID	61234555...cd2e8a2b	Last Modified	20 Jun 2024, 11:20	Size	256
KeyID		Exportable		Deletable	
XTS/CBC CS1	False	Global Usage			

Key Owner: admin

2 Results | 2 groups

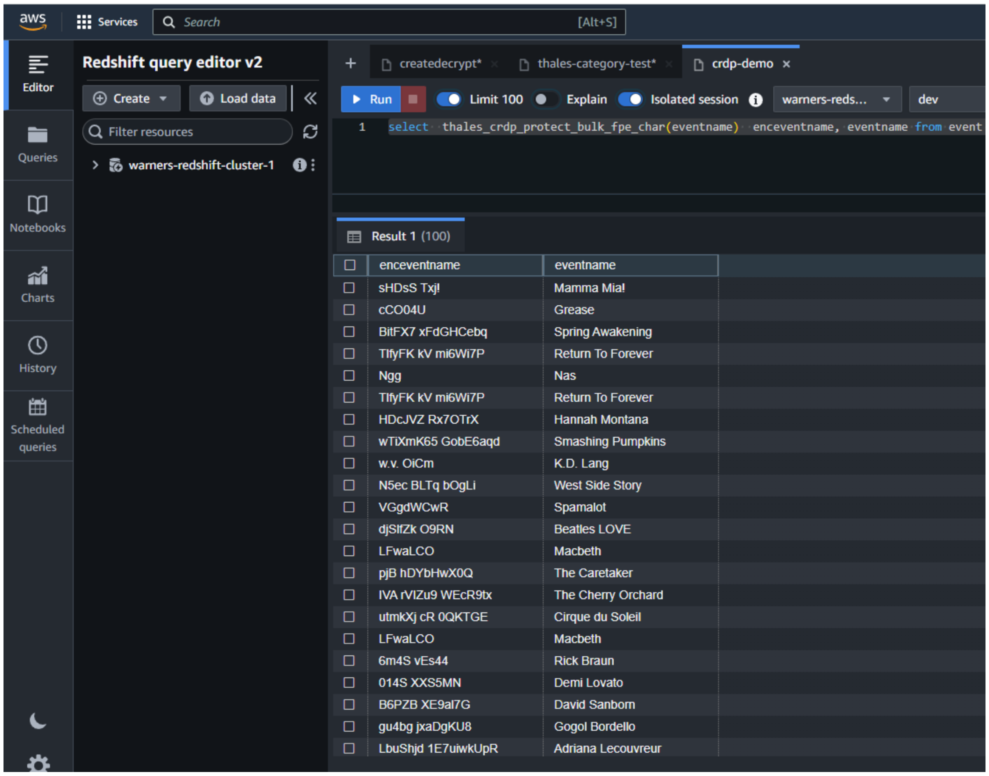
Group	Read	Use	Upload	Decrypt	Encrypt	Sign	Verify	Export	All
admin	✓	✓	✓	✓	✓	✓	✓	✓	✓
Application Data Protection Clients	☑	☑	☑	☑	☑	☑	☑	☑	☑

If userset lookups are desired, then when creating a user it should have the ability to export the key and also be in the Application Data Protection Clients Group. This user will also be needed to be provided as an environment variable for the function along with the password The examples provided have the key as a hardcoded value, but this can be easily altered to be provided as an environment variable, obtained from a secrets manager or in the userDefinedContext of the create function statement.

As noted above there is a test class that can be used to test connectivity with CM without having to publish the Function.

When all the above steps are performed you should see your UDF's in AWS Redshift UI. Here is a sample query using one of the UDF's.

Sample Results:



```
select  thales_crdp_protect_bulk_fpe_char(eventname)  enceventname, eventname from event
```

enceventname	eventname
sHDsS Txj!	Mamma Mia!
cCO04U	Grease
BitFX7 xFdGHCebq	Spring Awakening
TIfyFK kV mi6Wi7P	Return To Forever
Ngg	Nas
TIfyFK kV mi6Wi7P	Return To Forever
HDcJVZ Rx7OTrX	Hannah Montana
wTiXmK65 GobE6aqd	Smashing Pumpkins
w.v. OiCm	K.D. Lang
N5ec BLTq bOgLi	West Side Story
VGgdWCwR	Spamalot
djSlfZk O9RN	Beatles LOVE
LFwaLCO	Macbeth
pjB hDYbHwX0Q	The Caretaker
IVA rVIZu9 WEcR9tx	The Cherry Orchard
utmkXj cR 0QKTGE	Cirque du Soleil
LFwaLCO	Macbeth
6m4S vEs44	Rick Braun
014S XXS5MN	Demi Lovato
B6PZB XE9al7G	David Sanbom
gu4bg jxaDgKU8	Gogol Bordello
LbuShjd 1E7uiwkUpR	Adriana Lecouvreur

Environment Variables

Listed below are the environment variables currently required for the Cloud Function.

```
String crdpip = System.getenv("CRDPIP");  //CRDP Container IP
String userName = System.getenv("CMUSER");
// If you plan on using UserSets the cmuser must also be in the Application Data
Protection Client Group.
String password = System.getenv("CMPWD");
// returnciphertextforuserwithnokeyaccess = is na environment variable to express
// how data should be returned when the user above does not have access to the key
//and if doing a lookup in the userset and the user does not exist. If
//returnciphertextforuserwithnokeyaccess = null
// then an error will be returned to the query, else the results set will provide
ciphertext.
// validvalues are yes,no
// yes will return cipher text
// no will return error.
String returnciphertextforuserwithnokeyaccess =
System.getenv("returnciphertextforuserwithnokeyaccess");
// usersetlookup = should a userset lookup be done on the user from Big Query? 1
// = yes,no
String usersetlookup = System.getenv("usersetlookup");
```

```
// usersetID = should be the usersetid in CM to query.
String usersetID = System.getenv("usersetidincm");
// usersetlookupip = this is the IP address to query the userset. Currently it
// is the userset in CM but could be a memcache or other in memory db.
String userSetLookupIP = System.getenv("usersetlookupip");
String keymetadatalocation = System.getenv("keymetadatalocation");
//Valid values are internal, external to indicate where the metadata resides.
String external_version_from_ext_source = System.getenv("keymetadata");
//This is only to be used for demo purposes since it will need to obtained from a
database or some other external source. This should be the 7 byte header needed to
reveal data that is using a protection profile that contains an ext
// How many records in a chunk. Testing has indicated point of diminishing
// returns at 100 or 200, but
// may vary depending on size of data.
int batchsize = Integer.parseInt(System.getenv("BATCHSIZE"));
```

Application Data Protection UserSets

Application Data Protection UserSets are currently used for DPG to control how the data will be revealed to users. These UserSets can also be independent of any Access Policy. Most cloud databases have some way to capture who is running the query and this information can be passed to CM to be verified in a UserSet to ensure the person running the query has been granted proper access. In github there is a sample class file called CMUserSetHelper that can be used to load a userset with values from an external identity provider such as LDAP. The name of this method is `addAUserToUserSetFromFile`. Once users have been loaded into this userset the usersetid must be captured and used as an environment variable to the Function. The function has a number of environment variables that must be provided for the function to work. Please review the section on Environment Variables for more details.

Options for keyname

Currently the sample code uses a hard coded key name. Other options to be considered are:

1.)keyname as an environment variable.
2.)keyname as a AWS secret.
3.)keyname passed in an attribute to the function. Example with a database view.
create view employee as select first_name, last_name,
thales_CRDP_aws_decrypt_char('testfaas', email) as email from emp_basic

Options for handling null values.

Since it is not possible to encrypt a column that contains null values or any column that has 1 byte it is necessary to skip those to avoid getting an error when running the query. There are a couple of ways to handle this use case.

Option 1. Modify the queries.

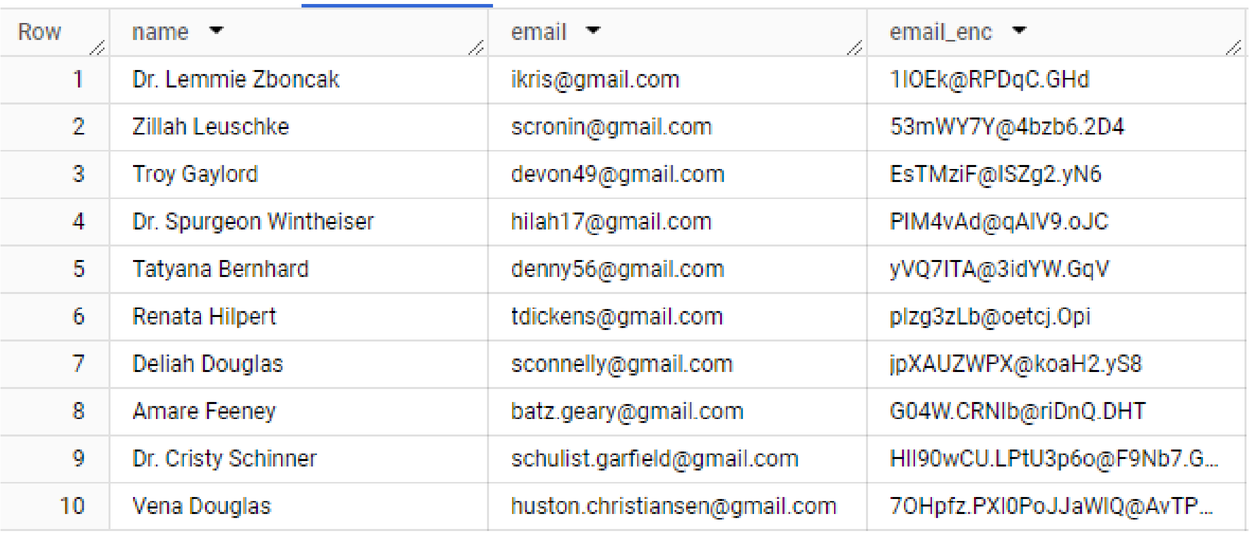
Many times, simply adding a where clause to exclude values that have nulls or less than 2 bytes can avoid query errors. For example: select * from FROM mw_demo_dataset_US.plaintext50CRDPemailprotected_nulltest where email is not null and length(email) > 1; For those scenarios where that is not suffice some other examples are listed below.   Here is an example of a select statement that can be modified to handle null values: Here is an example of a select statement that can be modified to handle null values:

```
SELECT
  name,
  CASE
    WHEN email IS NULL THEN 'null'
    WHEN length(email) < 2 then email
    WHEN email = 'null' then email
    ELSE `cpl-sales-l-app-us-01.mw_demo_dataset_US.thales_CRDP_encrypt_char`(email)
  END AS email
FROM
  mw_demo_dataset_US.plaintext50CRDPemailprotected_nulltest;
```

The above use case was for situations where the column contained both null and the word 'null'.

Here is an example of a select statement that can be modified to handle null values in the where clause.

```
    SELECT name, email, email_enc
FROM
  mw_demo_dataset_US.plaintext50CRDPemailprotected_nulltest
  where CASE
    WHEN email_enc IS NULL THEN 'null'
    WHEN length(email_enc) < 2 then email_enc
    WHEN email_enc = 'null' then email_enc
    ELSE  `cpl-sales-l-app-us-
01.mw_demo_dataset_US.thales_CRDP_decrypt_char`(email_enc)
  END  like "%gmail%"
```



Row	name	email	email_enc
1	Dr. Lemmie Zboncak	ikris@gmail.com	1lOEk@RPDqC.GHd
2	Zillah Leuschke	scronin@gmail.com	53mWY7Y@4bzb6.2D4
3	Troy Gaylord	devon49@gmail.com	EsTMziF@ISZg2.yN6
4	Dr. Spurgeon Wintheiser	hilah17@gmail.com	PIM4vAd@qAIV9.oJC
5	Tatyana Bernhard	denny56@gmail.com	yVQ7ITA@3idYW.GqV
6	Renata Hilpert	tdickens@gmail.com	plzg3zLb@oetcj.Opi
7	Deliah Douglas	sconnelly@gmail.com	jpXAUZWPX@koaH2.yS8
8	Amare Feeney	batz.geary@gmail.com	G04W.CRNIb@riDnQ.DHT
9	Dr. Cristy Schinner	schulist.garfield@gmail.com	HIl90wCU.LPtU3p6o@F9Nb7.G...
10	Vena Douglas	huston.christiansen@gmail.com	7OHpfz.PXI0PoJJaWIQ@AvTP...

Option 2. Modify the Cloud Function to skip encrypting these values.

Include an if statement checking for null values the word ‘null’ or any value less than 2.

```
if (redshiftrow != null && redshiftrow.size() > 0) {
        JsonElement element = redshiftrow.get(0);
        if (element != null && !element.isJsonNull()) {
                sensitive = element.getAsString();
                if (sensitive.isEmpty() || sensitive.length() < 2) {
                        if (sensitive.isEmpty()) {
                                JsonElement elementforNull = new JsonPrimitive("null");
                                sensitive = elementforNull.getAsJsonPrimitive().toString();
                        } else {
                                sensitive = element.getAsJsonPrimitive().toString();
                        }
                        encdata = sensitive;
                } else {
                        if (sensitive.equalsIgnoreCase("null")) {
                                sensitive = element.getAsJsonPrimitive().toString();
                                encdata = sensitive;
                        } else {
                                sensitive = element.getAsJsonPrimitive().toString();
                                byte[] outbuf = encryptCipher.doFinal(sensitive.getBytes());
                        }
                }
        } else {
                JsonElement elementforNull = new JsonPrimitive("null");
                sensitive = elementforNull.getAsJsonPrimitive().toString();
                encdata = sensitive;
        }
} else {
        JsonElement elementforNull = new JsonPrimitive("null");
        sensitive = elementforNull.getAsJsonPrimitive().toString();
        encdata = sensitive;
}
```

Note: When using this logic it is important to know that for any column that is null will return ‘null’. This should be fine for use cases where the query is not updating the column. For use cases where the source system is expecting null vs the word ‘null’ additional testing should be conducted on the systems that rely on this data type as being null vs the word ‘null’.

Sample CRDP docker commands.

As noted in the link above there are number of ways to deploy the CRDP container. For demo purposes here are the steps that were performed for a standalone use case.

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl
software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable"
sudo apt update
apt-cache policy docker-ce
sudo apt install docker-ce
sudo systemctl status docker
sudo usermod -aG docker crdpuser

su – crdpuser
docker ps
docker pull thalesciphertrust/ciphertrust-restful-data-
protection:latest
exit
//Back on as root
```

```
docker run -e KEY_MANAGER_HOST=192.168.159.134 -e
REGISTRATION_TOKEN=vw886rxzMolgtL1Gz950Ta9kVbDDvo0qgawGHthOc7iKn
m8b1VP6R2ps3qifVDTN -p 8090:8090 -e SERVER_MODE=no-tls
thalesciphertrust/ciphertrust-restful-data-protection
```

```
{"level":"info","time":"Thu, 20 Jun 2024 14:56:26 +0000","msg":"going to initialize shield"}

{"level":"info","time":"Thu, 20 Jun 2024 14:56:26 +0000","msg":"registerClient: Going to
register the client"}

{"level":"info","time":"Thu, 20 Jun 2024 14:56:26 +0000","msg":"registerClient: Register the
client successfully"}

{"level":"info","time":"Thu, 20 Jun 2024 14:56:32
+0000","msg":"","app_name":"crdpapp1","audit":true,"client_id":"be458360-e656-4faf-b7ee-
0d210626b214","endpoint":"/v1/protect","jwt_username":"","key_name":"test","key_version":
"1","method":"POST","protection_policy_name":"plain-alpha-
internal","protection_policy_version":"4","source_ip":"192.168.159.1","status":"Success"}

{"level":"info","time":"Thu, 20 Jun 2024 14:56:45
+0000","msg":"","app_name":"crdpapp1","audit":true,"client_id":"be458360-e656-4faf-b7ee-
0d210626b214","endpoint":"/v1/protect","jwt_username":"","key_name":"test","key_version":
```

```
"1","method":"POST","protection_policy_name":"plain-alpha-
internal","protection_policy_version":"4","source_ip":"192.168.159.1","status":"Success"}

{"level":"info","time":"Thu, 20 Jun 2024 14:56:51
+0000","msg":"","app_name":"crdpapp1","audit":true,"client_id":"be458360-e656-4faf-b7ee-
0d210626b214","endpoint":"/v1/protect","jwt_username":"","key_name":"test","key_version":
"1","method":"POST","protection_policy_name":"plain-alpha-
internal","protection_policy_version":"4","source_ip":"192.168.159.1","status":"Success"}

{"level":"error","time":"Thu, 20 Jun 2024 14:56:57 +0000","msg":"error while encrypting the
data, err: Input buffer is too short (len=1), it has to be at least 2 characters long."}

{"level":"error","time":"Thu, 20 Jun 2024 14:56:57 +0000","msg":"Input buffer is too short
(len=1), it has to be at least 2 characters
long.","app_name":"crdpapp1","audit":true,"client_id":"be458360-e656-4faf-b7ee-
0d210626b214","endpoint":"/v1/protect","jwt_username":"","key_name":"test","key_version":
"1","method":"POST","protection_policy_name":"plain-alpha-
internal","protection_policy_version":"4","source_ip":"192.168.159.1","status":"Error"}
```