Baffle Guide for AWS Demo for Lower Environment (Static Masking)

Baffle is the easiest way to protect data.

Static Masking

Static masking is a process of removing sensitive data from a database that can't be recovered. In some cases, this might be accomplished with partial or full masking. However, in this example, encryption will be used and the encrypted data will be considered "destroyed" in the lower-environment because there is no access to the encryption keys.

The purpose of this demonstration is to simulate simultaneously copying and de-identifying production data to a lower environment in real-time. Refer to figure 1.

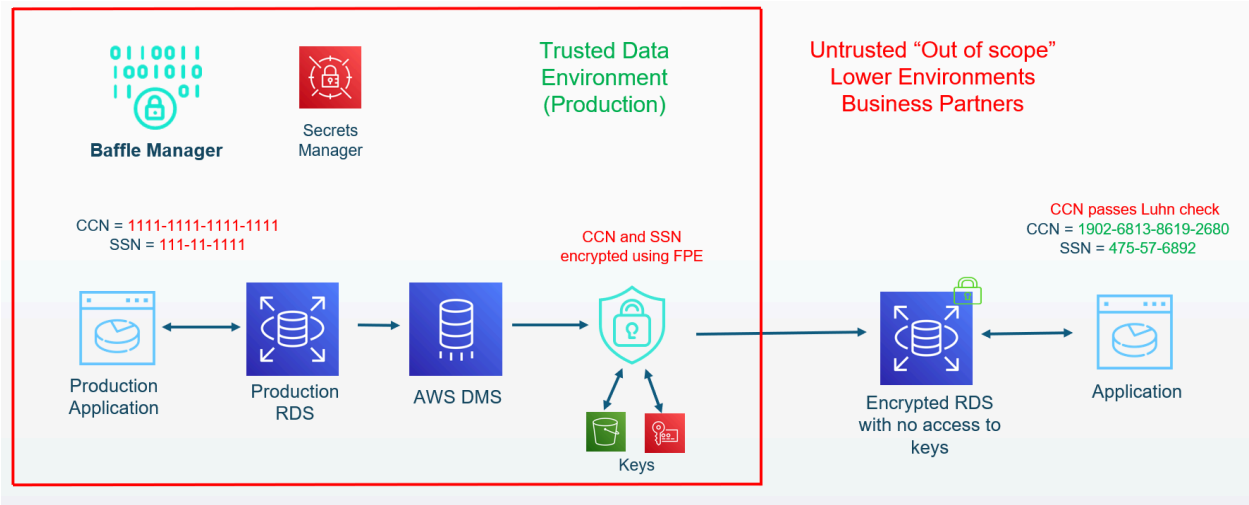

Figure 1. Setup for Static Masking

It is assumed that there is a trusted production environment with an AWS RDS database and application. The production database is operating with data in the clear (though it should be noted that the Baffle product could be placed between the application and database to encrypt data even in the production database). Development and testing engineers need data in their environments that closely resembles the quantity and type of data in production. However, the sensitive data can't be used in these lower environments without significantly increasing security and privacy risks. Baffle is used to implement format preserving encryption (FPE) to de-identify the data as AWS DMS copies from the production database to the database in the lower environment. The advantage of FPE over traditional encryption is

that the ciphertext is the same datatype and length of the original clear text, thus application and human testing is not affected. The de-identified data in the lower environment is updated in real time if DMS is set for continuous backup.

Note that this demonstration also simulates sharing data with business partners and/or migration to cloud environments where some (or all) of the data is sensitive and must be de-identified.

Prerequisites:

1. A browser that can connect to your AWS environment.
2. AWS Console access to an AWS environment with the ability to deploy Cloudformation, AWS RDS, AWS DMS, and an EC2.
    a. If using an Admin, then skip the "AWS IAM setup" and continue to "Setup the AWS Infrastructure".
    b. If you prefer to create a user specifically for this demo, Baffle has created a CloudFormation template to set policy and a user group. Start with "AWS IAM Setup"

AWS IAM Setup

1. Use CloudFormation (CF) to create the user group and policies.
    a. Download the "create_group_role_template.yaml" file from this location: baffle-public/market-place/cloudformation-template/lower-environment/create_group_policy at master · baffle/baffle-public (github.com)
    b. As an AWS Admin, log in to the AWS console and go to the CloudFormation service
    c. Click **Stacks**->**Create Stack**->**With new resources(standard)**
    d. Select **Choose and existing template** then **Upload a template file** then **Choose file** and navigate to the file. Select the file and click **Open.** Click **Next**
    e. In *Stack Name*, enter a name for the stack. This will also become the name of the resulting user group, with "-group" appended. ie <iam-stack-name>-group. Click **Next**
    f. Leave all defaults. Click **Next**
    g. Under *Capabilities*, check the box acknowledging that IAM resources will be created. Click **Submit**
2. Create a user and attach to the user group.
    a. As an AWS Admin, log in to the AWS console and go to the IAM service
    b. Go to the IAM services and click **User groups**. Ensure the new user group is there.
    c. Click **Users**->**Create User**. Enter a User Name. Check the box to provide user access to the AWS management console.
    d. Select the method you prefer for setting the password for console access and click **Next**.
    e. In *Permissions Option*s, select **Add user to group** and then below that, check the box that corresponds to the group created above. Click **Next**.
    f. Click **Create user**
    g. Note the password or download the csv file. Click **Return to users list**

Setup the AWS infrastructure

A CloudFormation (CF) template will be used to create AWS infrastructure that simulates Figure 1. This includes an RDS Postgres instance with two logical databases in it. One called "dms_source_db" and one called "dms_target_db" which represent production and development databases, respectively. A DMS instance will be created that will copy from the production database to the development database. Baffle software consists of Baffle Manager and Baffle Shield. Baffle Shield is the reverse proxy that does the encryption as DMS moves the data. Baffle Manager will program Baffle Shield. Secrets Manager will store the database credentials. An S3 bucket will store the Data Encryption Key (DEK) and AWS KMS will store the key encryption key (KEK) which is a customer managed key (CMK) in AWS terms. Finally, both applications will be represented by a pgAdmin connection.

1. Download the "create_baffle_workflows_template.yaml" from this location: baffle-public/market-place/cloudformation-template/create_baffle_workflows_template.yaml at master · baffle/baffle-public (github.com)
2. Either as an AWS admin or the user created in the "AWS IAM Setup" section above, log into the AWS console and go to the CloudFormation service.
3. Click **Stacks**->**Create Stack**->**With new resources(standard)**
4. Select **Choose an existing template** then **Upload a template file** then **Choose file** and navigate to the file. Select the file and click **Open.** Click **Next**
5. In *Stack Name*, enter a name for the stack, we will refer to it as <baffle-stack-name>
6. In *Parameters*
    a. *Workflow* - Select STATIC_MASK from the dropdown for this demo. (This same CF template can create several demos).
    b. *UserEmail* - create a username (email suggested) for Baffle Manager and pgAdmin
    c. *UserPassword*- create a password for Baffle Manager and pgAdmin
    d. *DBPassword* - create a password for the RDS database that will be created
    e. *MyIP* - enter the IPv4 address of your location which can be found at https://checkip.amazonaws.com This is to add you to the security group so you can access the AWS infrastructure that this CF template will create.
    f. Take note of all the credentials as they will be needed later. Click **Next**
7. Leave the defaults. Click **Next**
8. Under *Capabilities*, check the box acknowledging that IAM resources might be created (though none will be). Click **Submit**

    Note! It can take up to 20 minutes for this CF template to run. This demo can't continue until the stack status = CREATE_COMPLETE

Launch Baffle Manager

1. Login into the AWS Console and navigate to the CloudFormation service. Click **Stacks**->**(baffle-stack-name)** and a window slides in from the right.

2. Click the **Outputs** tab, Find and right click the link next to the **BaffleManagerURL** and select “open in new tab”. A new browser tab for Baffle Manager will appear.
3. In the new tab, click through the privacy warnings because a self-signed certificate was used. In production, proper certificates and trust would have to be established.
4. Log in with the Baffle Manager credentials you provided above.

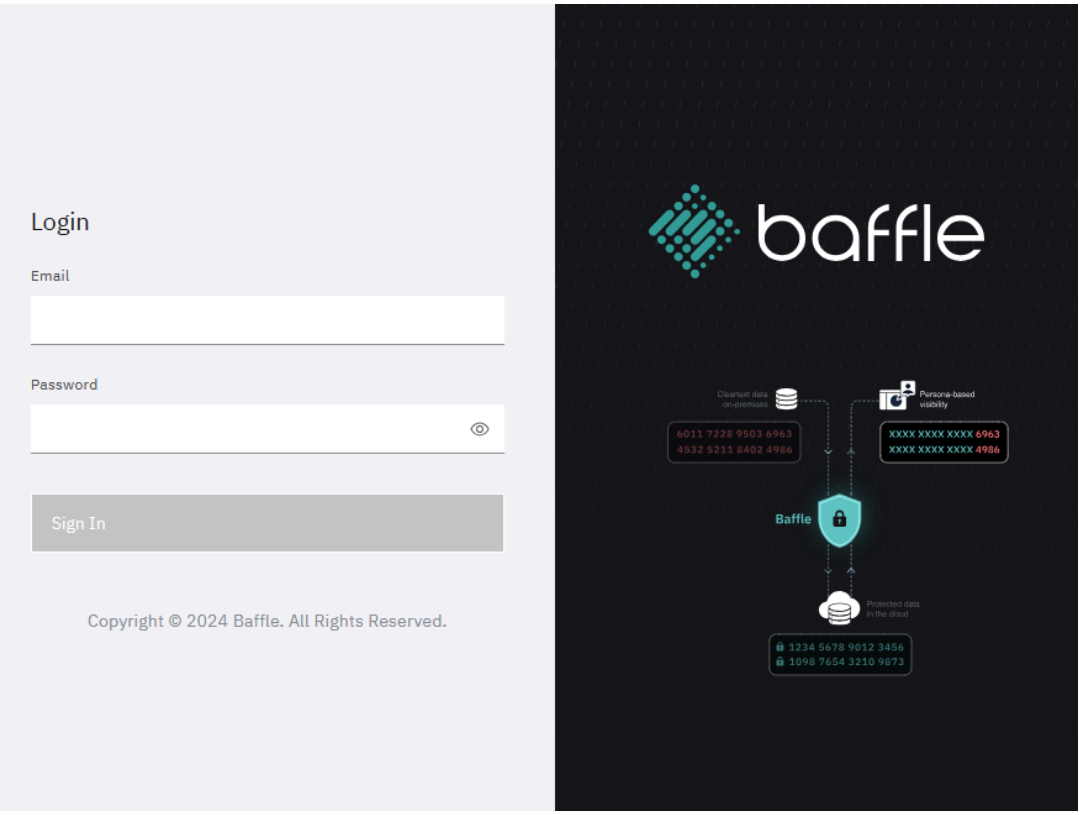

Figure 2. Baffle Manager Login Page

Once logged in, note the left navigation pane (Figure 3). This is where all processes start. The first section is Database Proxies and is the focus of this demonstration. The next section is for managing the Data Proxy, which is a separate Baffle product. Just note that several sub-headers have the same name in both the Database and Data proxy sections, so make sure to use the Database section.

For this demonstration, Baffle Manger has already been programmed through our API. However, we will step through the artifacts here.

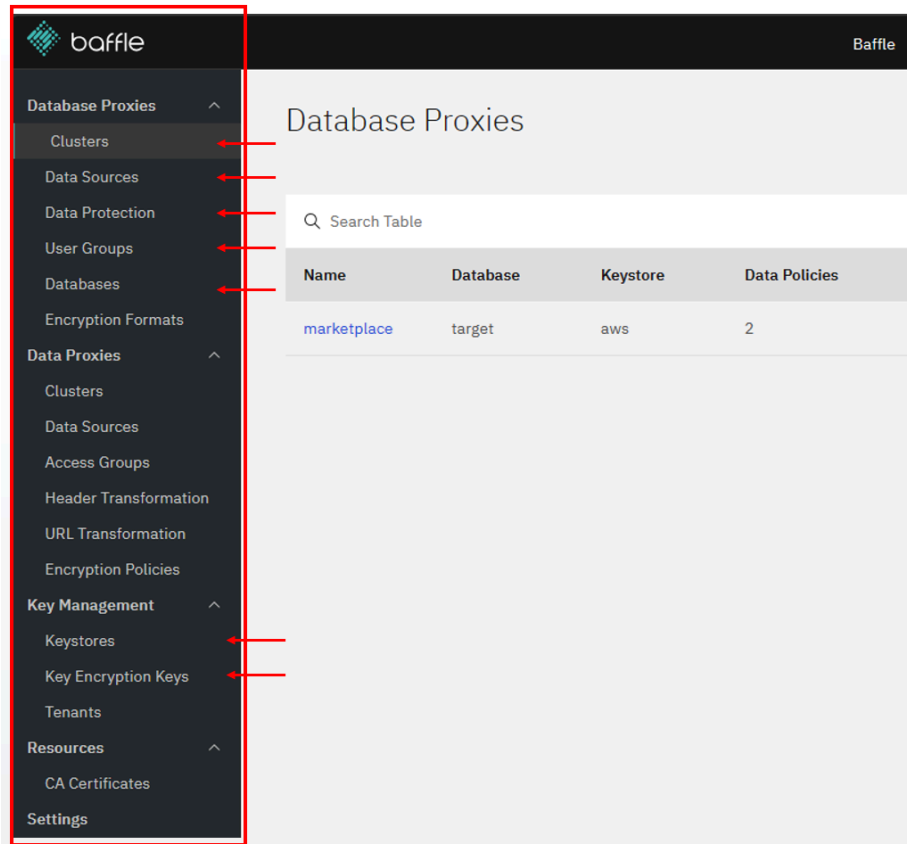

Figure 3. Baffle Manager Left Navigation

Check to ensure CloudFormation created the following:

- Key Management -> Key Encryption Keys -> alias/<stack_name>-baffle-shield-key. The KEK and related DEK are specified here.
- Key Management -> Keystore -> aws-kms. This specifies AWS KMS and S3 location for storing the key encryption key (KEK) and data encryption key (DEK) respectively.
- Database Proxies -> Databases -> PostgresSQL This is the target database that will contain the de-identified data
- Database Proxies -> Data Protection -> ccn-fpe-cc and ssn-fpe-decimal. This is the policy for encrypting ssn and ccn using format-preserving encryption. The ccn encryption has an additional step for ensuring the ciphertext passes the Luhn check.
- Database Proxies -> Data Sources-> ccn_dms_ds and ssn_dms_ds. This is the location of the social security numbers (ssn) and credit card numbers (ccn) to be protected. This includes the database, schema, table, column, and datatype.
- Database Proxy -> Clusters -> proxy-_static_mask. This is where Baffle Shield is configured and programmed.

Turn DMS on

Trigger DMS to start copying any data in the production database (dms_source_db) to the lower environment database (dms_target_db) on a continual basis

1. Login into the AWS Console and navigate to the CloudFormation service. Click **Stacks**->**(baffle-stack-name)** and a window slides in from the right.
2. Click the **Outputs** tab, Find and right click the link next to the **DMSMigrationTaskURL** and select "open in new tab". A new browser tab for DMS will appear.
3. In the top right of the new tab, click **Actions**->**Restart/Resume**

Load Data Into Production

Put sensitive ccn and ssn data into the production database, dms_source_db.

a. Login into the AWS Console and navigate to the CloudFormation service. Click **Stacks**->**(baffle-stack-name)** and a window slides in from the right.

b. Click the **Outputs** tab, Find and right click the link next to the *PGAdminURL* and select **open in new tab**. A new browser tab for pgAdmin will appear.

c. In the new tab, log into pgAdmin using the credentials you created (same as Baffle Manager credentials). The pgAdmin dashboard should appear.

d. On the left navigation pane called *Object Explorer*, will be a database server called "dms_static_mask", click on the chevron by it to show the different connections.

e. Click the chevron next to **direct@baffle** and a window will pop-up. Enter the RDS database password you created earlier and click **OK**.

f. Click the chevron next to **Databases**. Right click **dms_source_db** and then **Query Tool.** A new SQL command window will open on the right.

g. Copy sample data into the customers table using these SQL commands:

```
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('c19213ad-dff9-49a5-8ef0-fbe1964d6f66',
'Charley', '1111-1111-1111-1111', '111-11-1111', 1);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('1c598c48-2cda-4e70-99b1-a8adcc99a0a9',
'Cacilie', '2222-2222-2222-2222', '2222-22-2222', 2);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('bd58c5f8-5ca8-4a98-8a81-cf1575a7cf06',
'Jaye', '3333-3333-3333-3333', '333-33-3333', 3);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('bf9634a8-6d0b-4a4b-9658-5ca3d7a8679c',
'Katharyn', '4444-4444-4444-4444', '444-44-4444', 4);
```

```
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('df59fe01-debe-4bd8-9b34-25d0469fab76',
'Val', '5555-5555-5555-5555', '555-55-5555', 5);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('7790c775-f4e1-40ca-94d9-3bc85f12ca09',
'Jefferson', '6666-6666-6666-6666', '666-66-6666', 6);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('dc23cb1a-62f1-4e2d-917f-92f24fd6578a',
'Merrielle', '7777-7777-7777-7777', '777-77-7777', 7);
insert into customers (uuid, first_name, ccn, ssn,
entity_id) values ('d1c12f3b-9895-478f-9f28-6773415e3373',
'Gilberte', '8888-8888-8888-8888', '888-88-8888', 8);
```

Click the “play” button ▶ to run the query.

h. Verify the data is in the table with this command

```
select * from customers;
```

Highlight that command and click the “play” button ▶ to run the query.

See the data in the data output pane below. Note these clear ccn and ssn values are highly contrived to be easy to identify.

Observe De-Identified Data

Once DMS senses new data in production, it copies that data over to the lower environment, dsm_target_db through Baffle Shield. At this time, Baffle Shield will de-identify ccn and ssn with format preserving encryption.

1. Back in the *Object Explorer*, Under dms_static-mask->direct@baffle->Databases, right click **dms_target_db** and then **Query Tool.** A new SQL command pane opens.
2. Use the following command to select all from the table “customers”.

```
SELECT * FROM customers;
```

Click the “play” button ▶ to run the query.

Observe that the ssn and ccn of each user is encrypted. Because the encryption is format preserving, it may not not normally be obvious, but since the clear numbers were all easy to see (ie 111-11-1111), it is now easy to see the encrypted values are very different.

(Optional) Show Continuous Updates

This is just to show that any future production changes are captured by DMS and updated but de-identified in the lower environment. These ongoing updates are an advantage over many other approaches to static masking that have to be done periodically and can be disruptive.

1. Go back to the Query tool for the production (sales) database. Along the top see the tab that says “dms_source_db/baffle@direct@baffle*”

2. Delete a the entry for Charley and enter a new row for Horace by entering this command:

   ```
   delete from customers where first_name = 'Charley';
   insert into customers (uuid, first_name, ccn, ssn, entity_id)
   values ('8ac6425d-9895-478f-9f28-b37841ac39ee', 'Horace',
   '9999-9999-9999-9999', '999-99-9999', 9);
   ```

   Highlight only that command and click the “play” button  to run the query.

3. Go back to the Query tool for the lower environment (sales_dev) database. Along the top see the tab that says “dms_target_db/baffle@direct@baffle*”

4. Select all from the table “customers”.

   ```
   SELECT * FROM customers;
   ```

   Click the “play” button  to run the query.

   Observe that Charley was deleted and Horace added with a de-identified ccn and ssn.

Delete the CloudFormation Entries to Release the Resources

1. Go back to the DMS page and at the top right, click **Actions**->**Stop** to stop the DMS job. In the warning pop-up window, also click **stop**
2. Navigate to the CloudFormation service. Click **Stacks**->**(baffle-stack-name)** and a window slides in from the right.
3. At the top of the new window, click **Delete**. A warning window will pop-up. Click **Delete**
4. If you used the CloudFormation template to create a new user group and policy, then delete that CF stack as well. That will delete the user group and policy, but not the user.