PCS Partner Playbook (Jan 20)

Private Lift is a measurement solution that uses encrypted data and is powered by secure multi-party computation (MPC) with select partners. Data is encrypted so that each participating partner's data is kept private from the other. Upon completion of the MPC, each participating partner is only able to view the aggregated output statistics of the computation. Previously, this type of reporting required at least one party to learn which specific people converted after seeing an ad. Meta has the information about who saw an ad and the advertiser has information on who converted. MPC and encryption make it possible for both parties to learn insights about how the ads worked, without the need for either party to see the other's data sets.

You'll need the below work to be done by someone with permissions and familiarity with the following components:
1. Domain name service (for setting DNS A record for Conversions API Gateway subdomain)
2. Basic knowledge and permissions to access AWS services like IAM, S3 - Creating and Reading , VPC - creation, Peering, Route Tables (all these creations will happen through scripts).
3. Making API calls (for using Private Lift Graph API)
4. Familiarity with running shell commands
5. Debugging and log reading
6. (Only for manual file uploading for clients without CAPI-G or requiring app events) SQL and hashing (for conversion data preparation)

Nice-to-have skills (don't require prior knowledge as Meta will provide step-by-step instructions)
1. AWS Kinesis (for data ingestion and running tasks)
2. Docker, ECR, & ECS (for running ECS tasks)

Timeline:
Week 1: Kickoff call (1 hour)
Week 2: Conversions API Gateway Setup & Lift Study Launch (1 day)
Week 3: Private Computation Environment Setup, Testing, & Data Prep (3 days)
Week 4: Real Data Run (2 hours)

Please Note: Partners should not be using AWS EC2-Classic. They should be open to use EC2 VPC for deploying this solution.

Please complete the following steps in order to complete your Private Lift setup:

Private Computation Infrastructure Setup

1) Install Conversions API Gateway (Estimated Time: 2 hours)

   To run the commands to install the Private Lift infrastructure (specified in step 2 below), install conversions API gateway, by referring to the following guide: https://developers.facebook.com/docs/marketing-api/conversions-api/guides/gateway/setup

Note: if you DO NOT PLAN to send CAPI signals to Meta via Conversions API Gateway or if you have CAPI already setup through another source, please complete below steps:

- Open Conversions API Gateway Shell: `https://<capig.instance.url>/hub/shell`
- Run following command(Please replace the <PIXEL_ID> with your pixel id): `config write ConversionsApi /CONFIG_PER_PIXEL/<PIXEL_ID>/PUBLISH_TO_API false`
- Verify results by running following command: `config read ConversionsApi --pretty`

2) Setup Private Lift Environment (Estimated Time: 3 hours)

To setup the Private lift infrastructure using your AWS account, please complete the following steps:

- Go to Private Lift shell: `https://<capig.instance.url>/pl/shell/shell.html` (Replace the <capig.instance.url> with your CAPIG's subdomain.
- Run Deploy command (if unclear, run `deploy --help` to confirm the right usage):

  ```
  deploy [-b] <AWS region> <Your AWS account number> <Publisher AWS account number> <Publisher VPC ID> <Your aws_access_key_id> <Your aws_secret_access_key> <Tag>
  ```

  Please substitute the above placeholders with following values:
  - **-b**: with this optional argument, deployment service will add the semi automated data pipeline (for app data ingestion).
  - **AWS region:** the AWS region you would like to deploy the AWS infrastructure to. We should have alignment on this beforehand so that we all in the same region.
  - **Your AWS account number:** The account number of the AWS account you're using.
  - **Publisher AWS account number:** We will provide this to you beforehand.
  - **Publisher VPC ID:** We will provide this to you beforehand.
  - **Your aws_access_key_id and aws_secret_access_key:** These are your AWS credentials so that this command could deploy AWS infrastructure to your AWS account.
  - **Tag:** this is a string that will be appended to the name or tag of AWS resources to be created. It will be easier for you to identify which AWS resources are created
  - **Specifying S3 buckets**
    - **S3 config bucket prefix:** The S3 bucket that will be used to store

config files during the setup. It could be an existing bucket in your AWS account, or a totally new one (it will be created automatically during the setup). The full s3 bucket name will be (prefix + tag)
    - **S3 data bucket prefix:** The S3 Bucket that will be used to store data ingestion from Cloudbridge. Will be created during the setup. The full s3 bucket name will be (prefix + tag)
    - `Example: deploy <AWS region> <Your AWS account number> <Publisher AWS account number> <Publisher VPC ID> -d <S3 config bucket prefix> -s <S3 data output prefix> <Your aws_access_key_id> <Your aws_secret_access_key> <Tag>`
- **CLI help**
    - `deploy --help`
    - `undeploy --help`
- **`CLI status (draft)`**
    - `status`
        - `Streaming after CAPIG ver. 1.0.2`

After the deploy command is successful, please go to your AWS console and check the deployed AWS resources. Several key resources to check:

a. One ECS cluster with name “onedocker-cluster-<Tag>”
b. One VPC with name “onedocker-vpc-<Tag>”
c. VPC > subnets: subnets with name “onedocker-subnet-<Tag>”
d. Kinesis > Delivery streams with name “cb-data-ingestion-stream-<Tag>”
e. Other resources: with the VPC ID, you can find the corresponding Internet Gateway, Security Group, and Route Table.

Note: The “<Tag>” is the string you provide to the “deploy” command.

- The config.yml file will be used to run the Private Lift product. We need to make sure it is being uploaded to the s3 config bucket that you provided to the “deploy” command. Navigate to the s3 config bucket that passed to the “deploy” command and check if there is a file with name “config.yml”.
- Run the PCE validator to check that everything is set up correctly. This will help ensure all the resources for networking and computation are set up successfully. You will need 3 parameters to run the PCE Validator:
    a. key-id: AWS access key
    b. Key-data: AWS access secret
    c. pce-id: <Tag> used for the deployment in above steps
- Debugging: If you run into any issues during this step, see:
  Debugging the PL Deployment and Data Infra Pipeline

Please note

1. If the deploy failed halfway (passed input validation stage, but failed with terraform apply), run undeploy with the same set of inputs:

```
undeploy [-b] <AWS region> <Your AWS account number> <Publisher AWS account number> <Publisher VPC ID> <Your aws_access_key_id> <Your aws_secret_access_key> <Tag>
```

2. Ask advertisers to carefully store the set of input params they used, especially `<AWS region>` and `<Tag>`

3) Prepare for Data Ingestion (Estimated Time: 20 mins)

Data will be uploaded for Private Lift automatically through CAPIG. To enable the data ingestion to S3 using CAPIG, please complete below steps:

- Open Conversions API Gateway Shell: 4
- Run the following update commands (**Please follow the exact format and values!**):
    - config write Kinesis /PUBLISH_TO_KINESIS true
    - config write Kinesis /FIREHOSE_DELIVERY_STREAM_NAME "cb-data-ingestion-stream-<Tag>"
    - config write Kinesis /AWS_ACCESS_KEY "<Your aws_access_key_id>"
    - config write Kinesis /AWS_SECRET_KEY "<Your aws_secret_access_key>"

(The AWS credentials must have the policy "AmazonKinesisFirehoseFullAccess". To attach the policy to a user, go to IAM > Policies > select "AmazonKinesisFirehoseFullAccess" > Policy usage > click "Attach" > select user and click "Attach policy".)

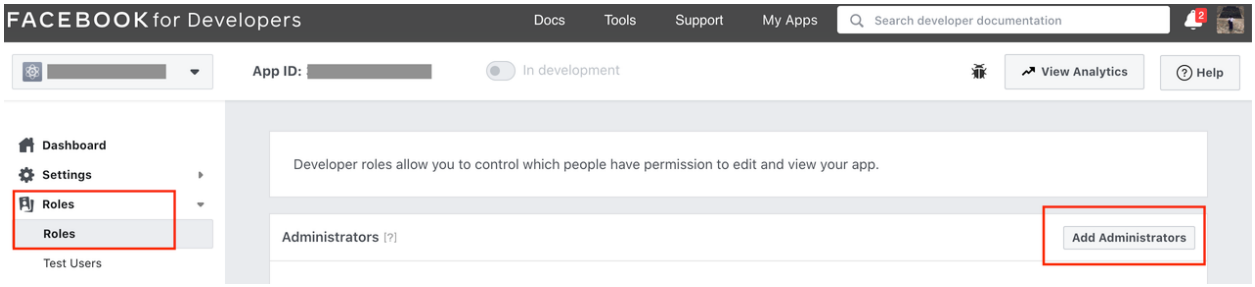
    - config write Kinesis /AWS_REGION "<AWS region>"
- Verify changes by running following command: config read Kinesis --pretty

App Permissions and Lift Test Setup

1) Set up a MPC-based lift test (Estimated Time: 1 hour)

There are two ways to setup MPC-based lift test:

- Your Client Solutions Manager will set the test up for you once you provide them with a timespan, holdout percentage, list of objective, and campaign IDs **make sure they include the business ID** in the UI when they set the study up or we won't have the correct permissions to pull the data
- You can also follow below steps for creating the lift study via the Lift API yourself:
    a. Go to Graph API Explorer.
    b. On the right panel, there is one dropdown selector under "Facebook App"
        - If there is the app you are using for Lift studies, select it, go to step "f".
        - If the app does not appear in the dropdown, go to step "c".
    c. [to be performed by a person with admin role] Go to the Facebook Developer Apps. On the left panel, click "Roles" → "Roles", and click "Add Administrators", and let the admin add you



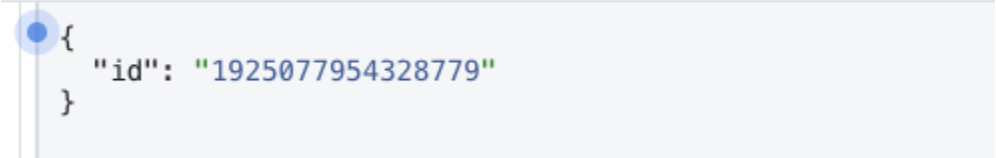
d. You can go back to Graph API Explorer, you should see a blue banner on the top stating you have a pending request, follow that to accept it, and you'll be added as the admin of that app.
e. Repeat steps a and b.
f. Switch call type to "POST", select the version to be the latest, input your user id, or business id, followed by "/ad_studies"
Note: to get business id, go to business settings > business info > Find info in the top section

g. On the left column, click "+ Add parameter", and the key, value, examples are illustrated in the table below make sure to include your business ID in the setup or the MPC game will not run

Key	Value	Example
name	<Study_Name>	PL Study XXXX
type	<Study_Type>	LIFT
start_time	Timestamp_of_study_start_time	1619422398
cooldown_start_time	Timestamp_of_study_start_time	1619422398
end_time	Timestamp_of_study_end_time	1619422398

observation_end_time	Timestamp_of_study_end_time	1619422398
cells	[{name:"cell 1",description:"description of my cell",treatment_percentage:90,control_percentage:10,adaccounts:[<ACCOUNT_ID1>,<ACCOUNT_ID2>]}]	[{"name": "cell 1","description": "description of my cell","treatment_percentage": 90,"control_percentage": 10,"adaccounts":["<ACCOUNT_ID>"]}]
objectives	[{name:"new objective",is_primary:true,type:"MAI",applications:[{id:<APP_ID>}]}]	[{name:"new objective",is_primary:true,type:"MPC_CONVERSION"}]

h. Click Submit, you should see result like below:



Note: Currently if you create a study using API, you are not allowed to cancel or modify it. Please contact your Meta sales representative if you need to make any changes or cancel a study after creating it.

2) Test Graph API Access (Estimated Time: 20 mins)

- Go to Graph API Explorer.
- Switch call type to “GET”, select the version to be the latest, replace the <Study_Id> with the study_id you got in the response when you created your study and click on Submit. You should not get permission related errors. In case of permission error, please reach out to your Meta POC.

Private Lift (AdHoc Run)

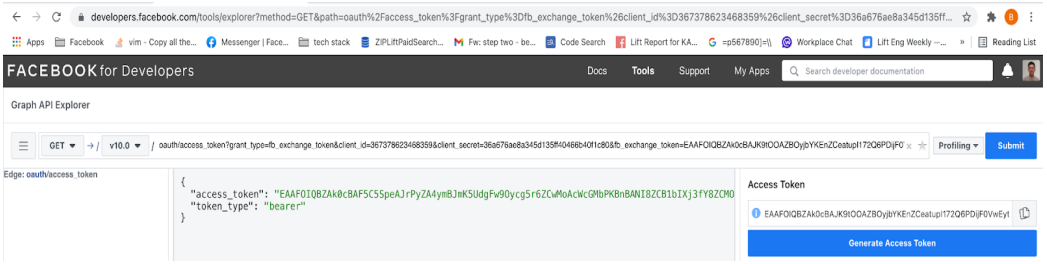
1) Run PL (Estimated Time: 5 hours)

To Run PL you will need to download the docker image shared on ECR which encapsulates all of the latest python code for the coordinator and libraries required to run the code. To begin with this process please go through the following steps:

Pre-test **[One time setup]**

1. Install Docker
2. Install AWS CLI
3. Run `aws configure` in your terminal
4. Download the config.yml file from the S3 bucket that you created while setting up the private lift infrastructure in Step 3.
   aws s3 cp s3://config-bucket-from-step-3/config.yml .
5. Fill in Auth Details in config.yml:
   a. To clear up TODO in AWS credentials (with enough permissions/policies):
      i. Set all instances (2 - 3 in total) of `access_key_id` with your IAM account's access key id
      ii. Set all instances (2 - 3 in total) of `access_key_data` with your IAM account's secret access key
   b. There should be no more remaining fields (filled up in the PC infra setup stage) except for the graphapi option. Just in case it does, then check out the **Appendix -- Update ALL TODOs in config.yml**
6. Obtain long-lived GraphAPI access token:
   a. NOTE: This token will expire and need to be renewed every 60 days

b. Find your <app_id>: Go to https://developers.facebook.com/apps. Copy the app_id of the app of your choice.
c. Go to https://developers.facebook.com/apps/<app_id>/settings/basic/, click 'show app secret', copy the app secret.
d. Obtain long-lived access token:
    i. Go to https://developers.facebook.com/tools/explorer. Select the app of your choice.
    ii. Paste oauth/access_token?grant_type=fb_exchange_token&client_id={app-id}&client_secret={app-secret}&fb_exchange_token={your-access-token} into request textfield
    iii. (Please remove parenthesis {app_id} before replacing). App-id: <app_id>, app-secret: use app secret, your-access-token: copy&paste the 'access token' on the right side bar.
    iv. Click 'submit'



    v. Copy the long-lived token
    vi. Put long-lived token in config file:
    vii. open config.yml
    viii. Delete the 'TODO' under "graphapi:token:", and paste the long-lived token there

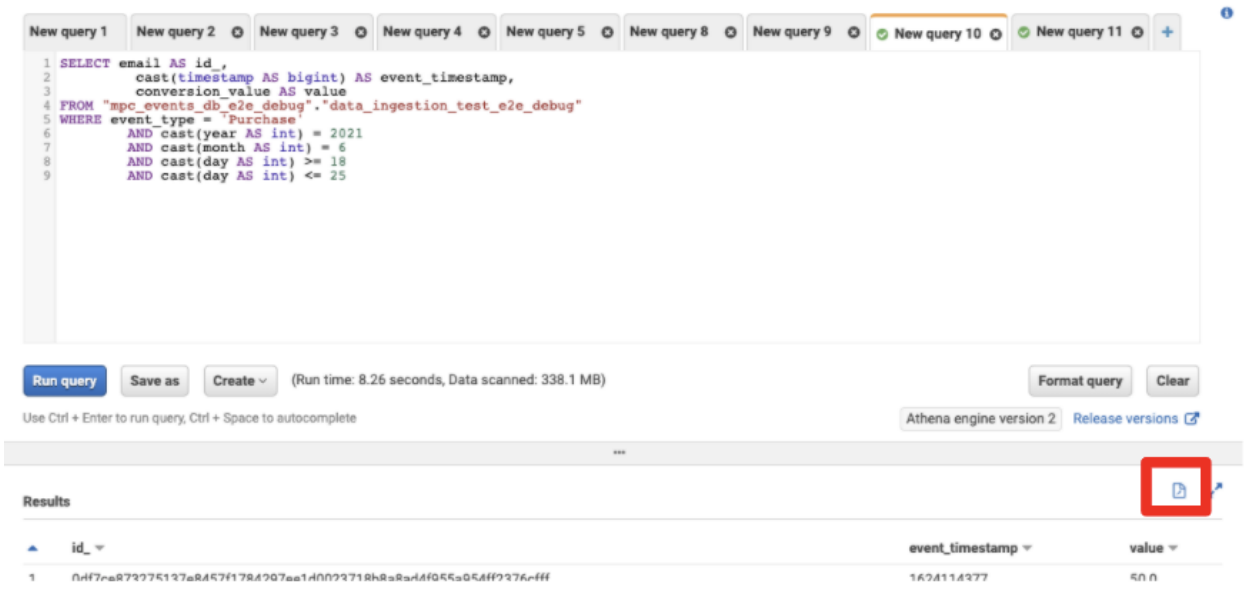
7. `curl -O https://raw.githubusercontent.com/facebookresearch/fbpcs/main/fbpcs/scripts/run_fbpcs.sh && chmod +x run_fbpcs.sh`
8. Your Solutions Engineering contact at Meta will grant you access to the ECR repository for the code, `539290649537.dkr.ecr.us-west-2.amazonaws.com/pl-coordinator-env`
    a. Your contact may also ask for a few details such as your AWS account ID and AWS canonical ID in order to grant you access to certain resources for use during the computation.

[Run every time you want to get results for a study]

1. If you haven't run PL in some time or are running it for the first time, make sure you have completed all the above steps and run the PCE validator to ensure your setup is still correct.
2. Once data is flowing to the data ingestion S3 bucket from Kinesis Stream - an Athena table is created - "mpc-events-db-<tag>"."<YOUR_TABLE_NAME>" . (Make sure you are in the correct AWS region, otherwise you will not see the tables!)
3. Once you see the above table in Athena, you are ready and good to run the query and generate csv for the PL run. Please follow below steps to generate the csv:

a. Look up the hash key for the study you're interested in by querying the Graph API endpoint for the study: GET /<ad_study_id>?fields=opp_data_information
    i. Under datasets_information, there will be a corresponding hash_key element for each study which can be pasted into the query below.
b. If first time using Athena, need to set **Query result location** in **Settings** (upper right corner)
c. Before running the query, please double check the correct 'key' is being used
    i. Make sure you are using PL key for running PL, and PA key for running PA.
    ii. Make sure the value of the key is correctly copied/pasted (e.g. no extra slash).
d. After you have everything setup, please run the below example query :
    i. Full list of possible filters available in Appendix: Athena Queries.
    ii.
```
SELECT
        TO_BASE64(
                HMAC_SHA256(
                  CAST(user_data.email AS VARBINARY),
                  FROM_BASE64('<YOUR_HASH_KEY>')
                  )
        ) AS id_,
        CAST(timestamp AS bigint) AS event_timestamp,
CAST(CAST(REPLACE(conversion_value, ',', '') AS DECIMAL(38,2))*100
AS BIGINT) AS value
FROM "mpc-events-db-<Tag>"."<YOUR_TABLE_NAME>"
WHERE event_type = '<event_type>'
AND (CAST(concat(year, '-' , month , '-', day) AS DATE)
BETWEEN CAST('<START_DATE_IN_YYYY-MM-DD>' AS DATE)
AND CAST('<END_DATE_IN_YYYY-MM-DD>' AS DATE))
AND user_data.email IS NOT NULL
```

e. Please replace the placeholders with the values marked in bold in the above query:
    i. "YOUR_HASH_KEY" - Hash key retrieved from step(a) above.
    ii. "mpc-events-db-<Tag>"."<YOUR_TABLE_NAME>" - replace <Tag> with your PCS deployment tag and <YOUR_TABLE_NAME> with your table name in athena DB.
    iii. "Event_type" - replace it with the event_types contained within the objective of study. For Ex - Purchase, AddToCart, etc. Please double check with your Meta POC.
    iv. START_DATE_IN_YYYY-MM-DD - start date of the study
    v. END_DATE_IN_YYYY - end date of the study
f. The download button is highlighted in red rectangle as in the screenshot below. Alternatively, go to the specified **Query result location** to find the resulting query.



g. Upload the CSV to any S3 bucket to which the private lift has access to. It can be any bucket till the time you have permission to access that bucket using the AWS access key and secret.

4. Once you've completed the above one-time setup and generated your CSV(s), you can run Private Lift end-to-end with the following command for each study:
   a. Create a new directory on your machine and move the updated config.yml file to this folder and open a new terminal window in this directory.
   b. Please run the following curl command on your terminal to get the updated copy of run_fbpcs.sh. You only have to do it once until you are told to do it again. We're actively working on making the update not manual.

      ```
      curl -O
      https://raw.githubusercontent.com/facebookresearch/fbpcs/main/fbpcs
      /scripts/run_fbpcs.sh && chmod +x run_fbpcs.sh
      ```

   c.
      ```
      ./run_fbpcs.sh run_study <study_id>
      --objective_ids=<objective_id_1>,<objective_id_2>,...
      --config=config.yml
      --input_paths=https://<s3_conversion_data_file_path_for_objective_1
      >,https://<s3_conversion_data_file_path_for_objective_2>,...
      --log_path=/fbpcs_instances/output.txt
      ```

      Example:

      ```
      ./run_fbpcs.sh run_study 446346650252929
      --objective_ids=277428597633624 --config=config.yml
      --input_paths=https://private-lift-devaccount.s3.us-west-1.amazonaw
      s.com/2cc-6e7a-4a9b-bes4b-b7ce511f067c.csv
      --log_path=/fbpcs_instances/output.txt
      ```

5. Wait up to 6 hours for the process to terminate. Please do not terminate the process before it finishes completely. If you encounter an issue during this step, reach out to your Meta POC.
6. Save the results:
    - Go to https://developers.facebook.com/tools/explorer?method=GET&path=<study_id>%2Finstances&version=v10.0, click 'submit'. Copy the response.
    - If there's an error, please send both the response and the output.txt file to your Meta contact for debugging. The output.txt file will have logging related to which stage is running and does **not** contain the output results of the MPC (metrics); so is safe to share.
    - The metrics should be present in the Lift UI within 24 hours.

Appendix

Update all the TODOs in `config.yml`

- Set all instances of the region to your aws region. Example: `us-west-2`
- Set cluster name. Example `onedocker-cluster-<postfix>`
- Set subnets using one of the subnets created while setting up infra in step 3 in PL Setup.
- Set all instances (2 - 3 in total) of `access_key_id` with your IAM account's access key id
- Set all instances (2 - 3 in total) of `access_key_data` with your IAM account's secret access key
- Set task definition. Example: `onedocker-task-<postfix>:1#onedocker-container-<postfix>`
  Note that: The number :1 could change if there was deploy/undeploy
- Ignore the graphapi option

Athena Queries

- If first time using Athena, need to set **Query result location** in **Settings** (upper right corner)
- Example query:
    - **If before CAPIG ver. 1.0.2** (or did not manually update crawler and data_transformation_lambda.py)

```
SELECT
        TO_BASE64(
```

```
                        HMAC_SHA256(
                            CAST(email AS VARBINARY),
                            FROM_BASE64('key')
                        )
            ) AS id_,
             CAST(timestamp AS bigint) AS event_timestamp,
          CAST(CAST(REPLACE(conversion_value, ',', '') AS
DECIMAL(38,2))*100 AS BIGINT) AS value
FROM
"mpc_events_db_e2e_debug"."data_ingestion_test_e2e_debug"
WHERE event_type = 'Purchase'
           AND (CAST(concat(year, '-' , month , '-', day) AS
DATE)
BETWEEN CAST('2021-06-18' AS DATE)
AND CAST('2021-06-25' AS DATE))
AND email IS NOT NULL
```

- *Sometimes we would like to filter out ios 14.5 + events, add the following condition to WHERE clause:*

```
AND user_data.device_os = 'iOS'
AND LENGTH(user_data.device_os_version) > 0
        AND ( COALESCE( try( CAST( (
    CASE
    WHEN try(SPLIT(user_data.device_os_version, '.')[1]) IS NULL THEN
    '0'
    ELSE SPLIT(user_data.device_os_version, '.')[1]
    END ) AS INTEGER ) ), 0 ) >= 15
        OR ( COALESCE( try( CAST( (
    CASE
    WHEN try(SPLIT(user_data.device_os_version, '.')[1]) IS NULL THEN
    '0'
    ELSE SPLIT(user_data.device_os_version, '.')[1]
    END ) AS INTEGER ) ), 0 ) = 14
        AND COALESCE( try( CAST( (
    CASE
    WHEN try(SPLIT(user_data.device_os_version, '.')[2]) IS NULL THEN
    '0'
    ELSE SPLIT(user_data.device_os_version, '.')[2]
    END ) AS INTEGER ) ), 0 ) >= 5 ) )
```

- If Athena complains for conversion_value column cannot be empty or non-numeric strings, replace that line with the following:

```
CAST (
  (CASE
    WHEN try(
      CAST(REPLACE(conversion_value, ',', '') AS DECIMAL(38, 2))
      ) IS NULL THEN 0 ELSE (
      CAST(REPLACE(conversion_value, ',', '') AS DECIMAL(38, 2))
    )
  END) * 100 AS BIGINT
) AS conversion_value
```