Private Computation Solutions Partner Playbook

Table of Contents

Introduction

This is the step-by-step guidebook for Meta platform advertisers to set up, install, and use privacy-enhanced ads measurement products (e.g., Private Lift and Private Attribution).

We recommend engineers to go through this resource as it requires certain knowledge or familiarity with network setup, cloud service, etc.

How to use the playbook

Follow the step-by-step instructions to setup, install, and use the product. Please contact your Meta representative for any questions/issues encountered.

Requirements

You’ll need the below work to be done by someone (engineers) 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 Computation Graph API)
4. Debugging and log reading
5. (**if not using UI**: familiarity with running shell commands)
6. (Only for clients who need/want to prepare your own conversion data): SQL and hashing
7. Please make sure you have reviewed the following AWS Prerequisites and Permission requirements.

a. Private Computation: Business pre-check questions
b. Private Computation: AWS pre-check questions
c. Private Computation: Guide to answering AWS pre-check questions

Private Computation Products overview

Both Private Lift and Private Attribution are measurement solutions that use encrypted data and are powered by secure multi-party computation (MPC) with select partners such that each participating partner's data is kept private from the other, and, 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, considering Meta has the information about who saw an ad and the advertiser has information on who converted. MPC makes it possible for both parties to only learn insights about ads performance, without the need for either party to see the other's data.

(Private) Lift is a powerful way to understand the incremental effect of your advertising on Meta's platform. This is a kind of experiment where we compare groups of people who did and did not have the opportunity to see your advertising to understand its causal impact on specific business objectives, such as brand recognition or conversion.

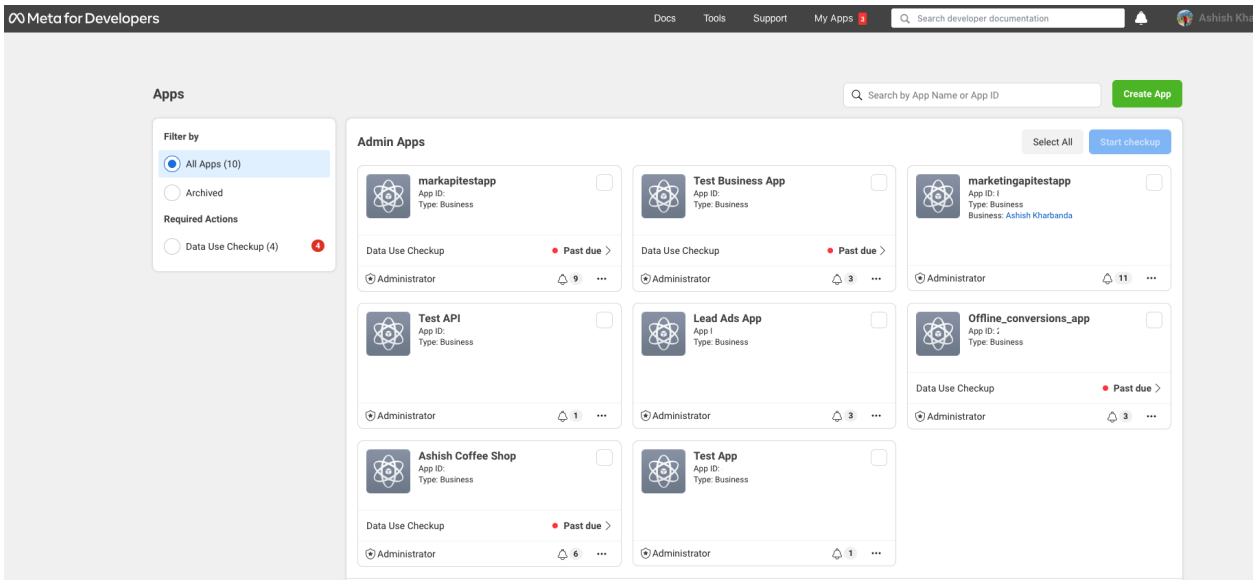
(Private) Attribution is a measurement product that determines the user actions that led to the desired outcome between the click of the ad and the conversion. We currently support 1-day click-through attribution.

Step 1: Conversions API Gateway Setup (1 Hour)

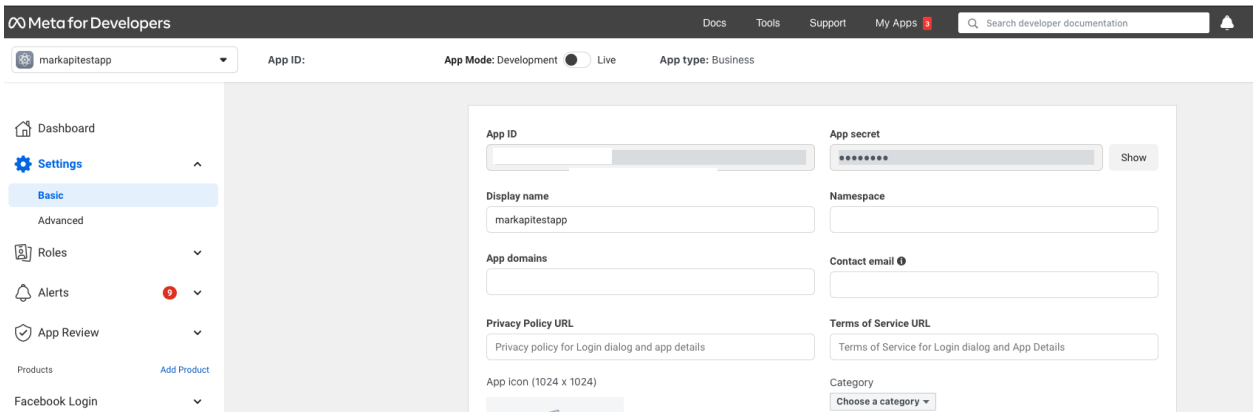
To run the commands to install the Private Lift infrastructure (specified in step 2 below), install Conversions API Gateway, by referring to the following guide. Please make sure to select at least an t2.xlarge EC2 instance type:
https://developers.facebook.com/docs/marketing-api/conversions-api/guides/gateway/setup

Step 2: Generating 60 days Access Token (10 Minutes)

- Go to developers.facebook.com/apps and select the app that you want to use.

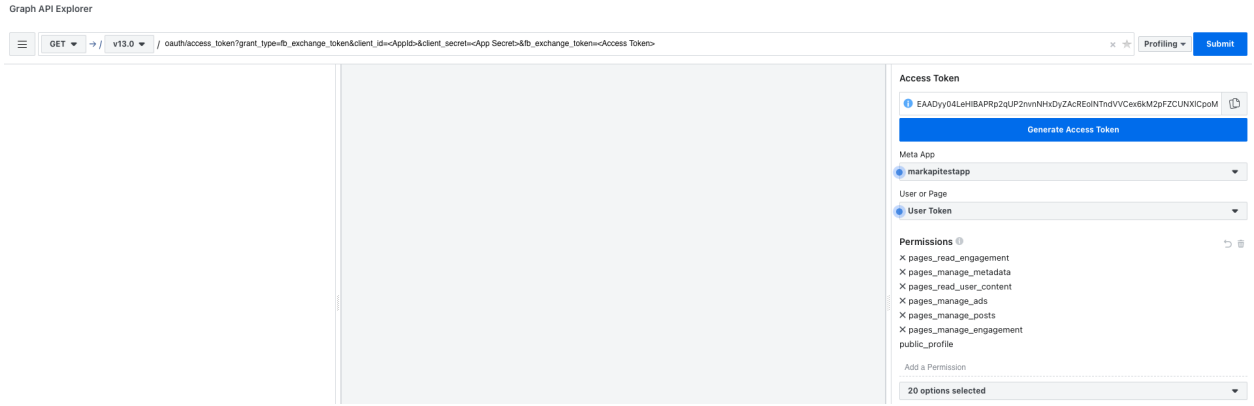


- Navigate to Settings → Basic. Click on “Show” near app secret and copy both App ID and App Secret.

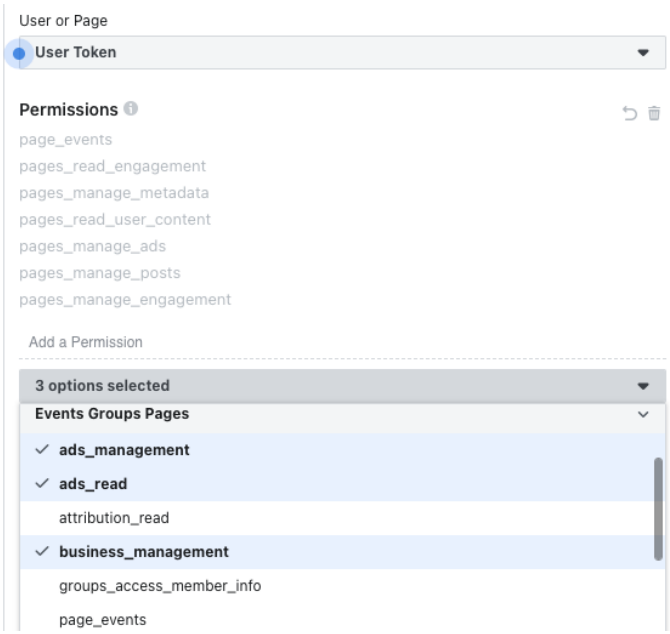


- Go to https://developers.facebook.com/tools/explorer
- In the `GET` request enter:
  `oauth/access_token?grant_type=fb_exchange_token&client_id=<AppId>&client_secret=<App Secret>&fb_exchange_token=<Access Token>`

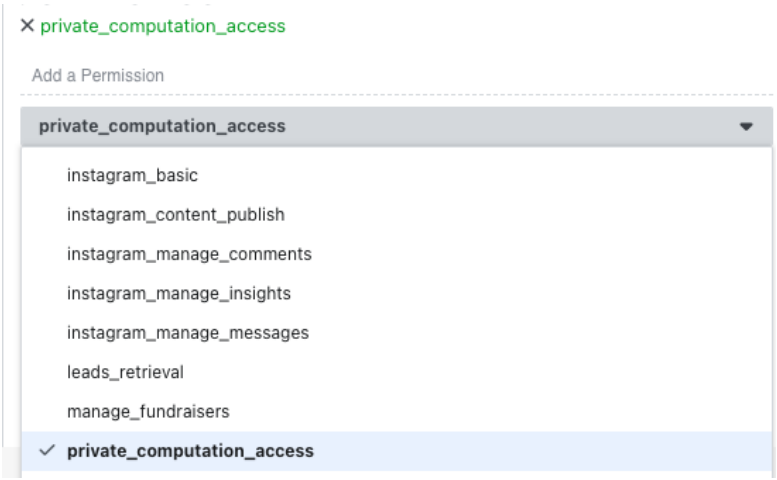
Replace <AppId> and <AppSecret> with the values copied in the previous step. Also replace the <Access Token> with the Access Token on the screen in the right corner (see below).



- Click on the User or Page dropdown and select User Token.
- Click on Add a Permission
    - → Events Groups Pages → Select **ads_management**, **ads_read** and **business_management**



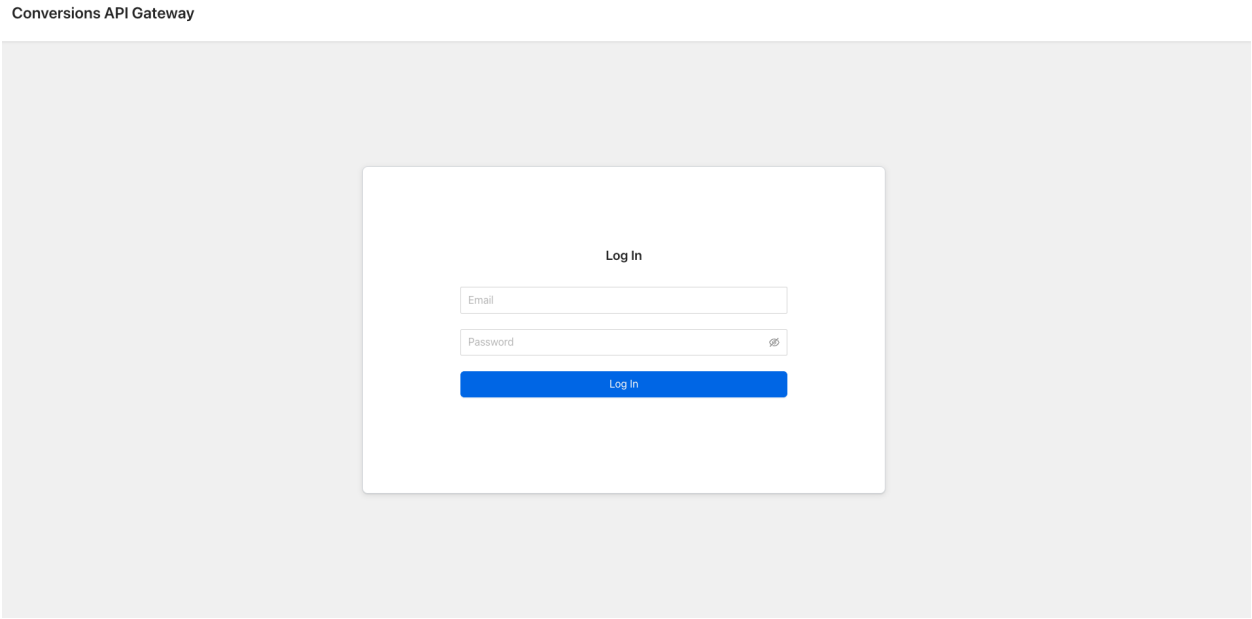
    - → Other→ Select **private_computation_access**.



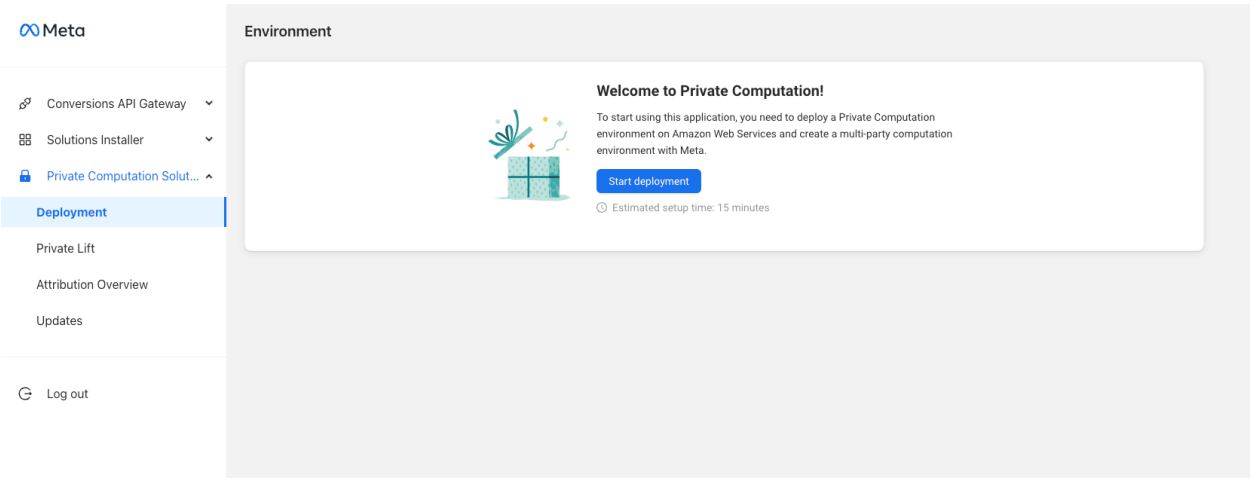
- Click on Submit Button and copy the access_token received in the response.
    - Please carefully store this token.

Step 3: PCS AWS Infrastructure Setup (30 Minutes)

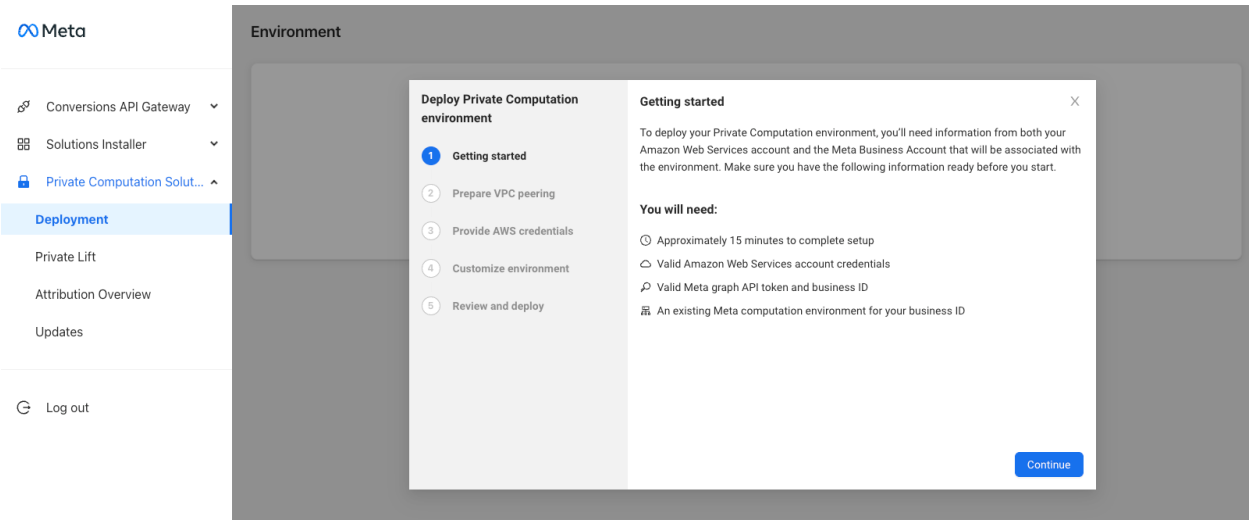
- Navigate to https://<capig.instance.url>/hub/ui. You should see the following window:



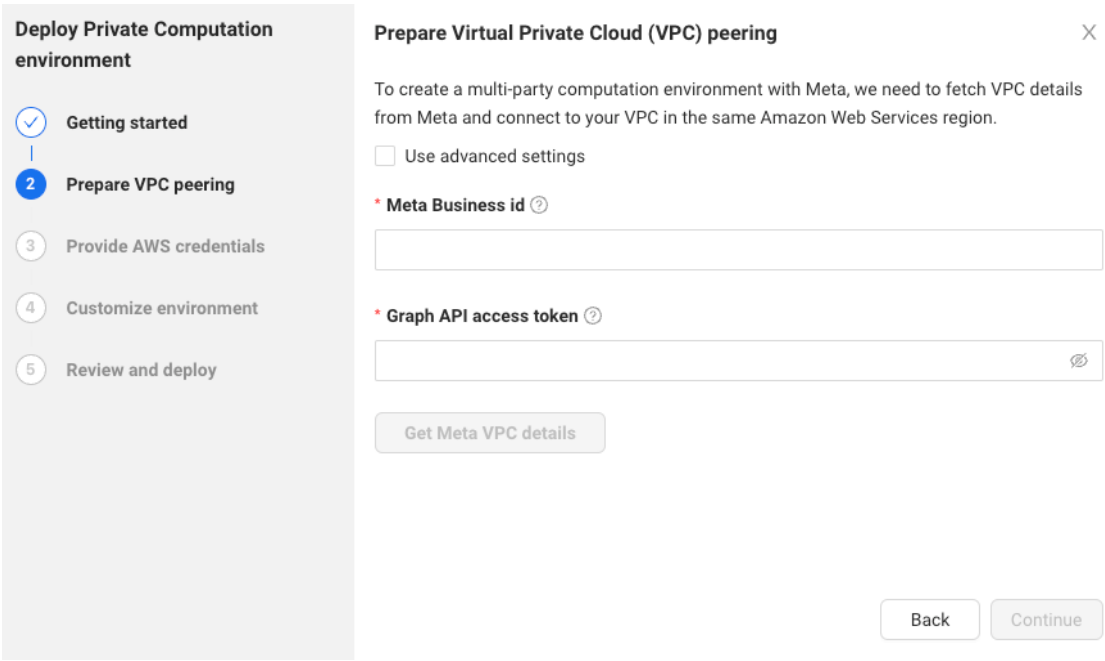
- Enter the credentials and login.
- Navigate to the Deployment Menu and click Start deployment.



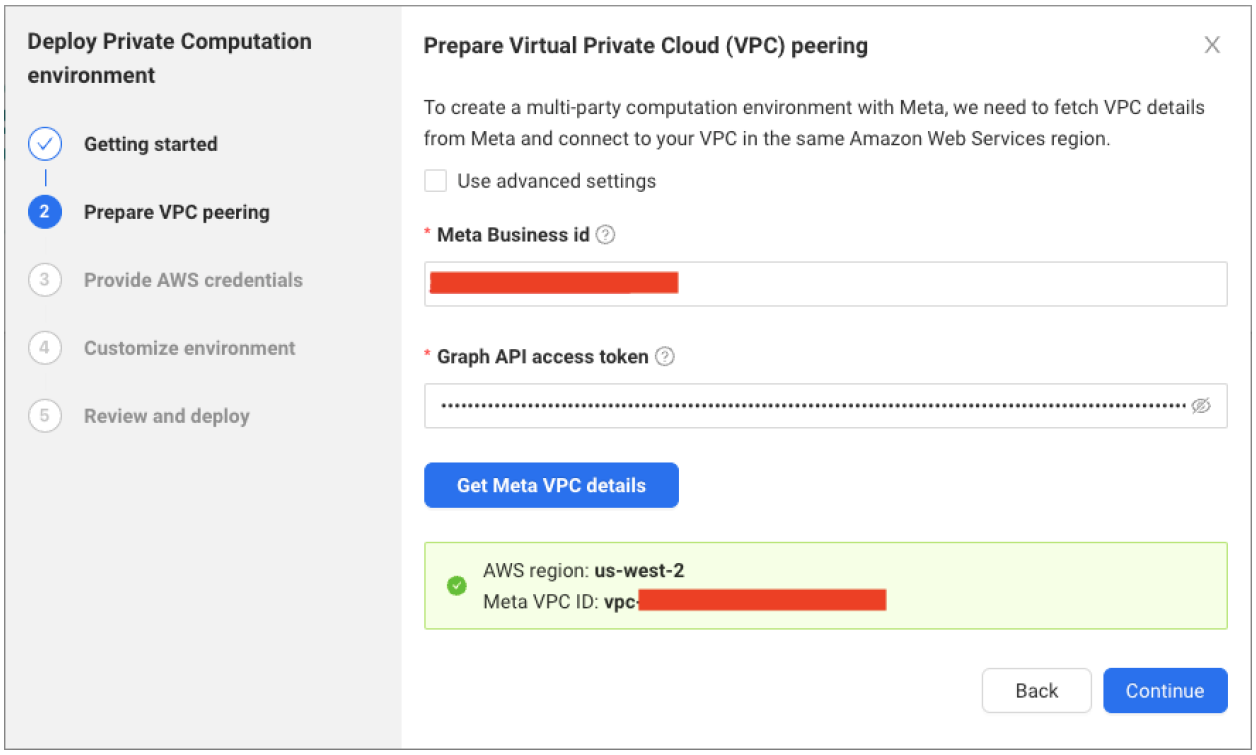
- A modal will pop up, and the screen will show on what you would require to have to complete the deployment



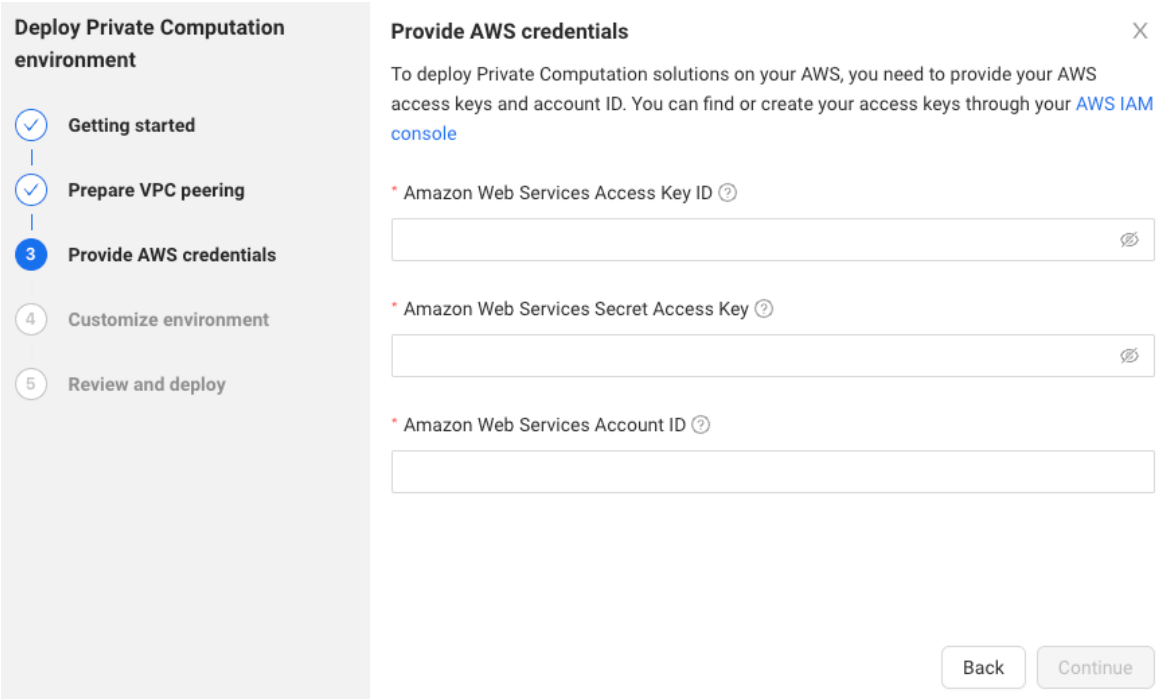
- Click continue, and go to the next step. You should see a screen that looks like below:



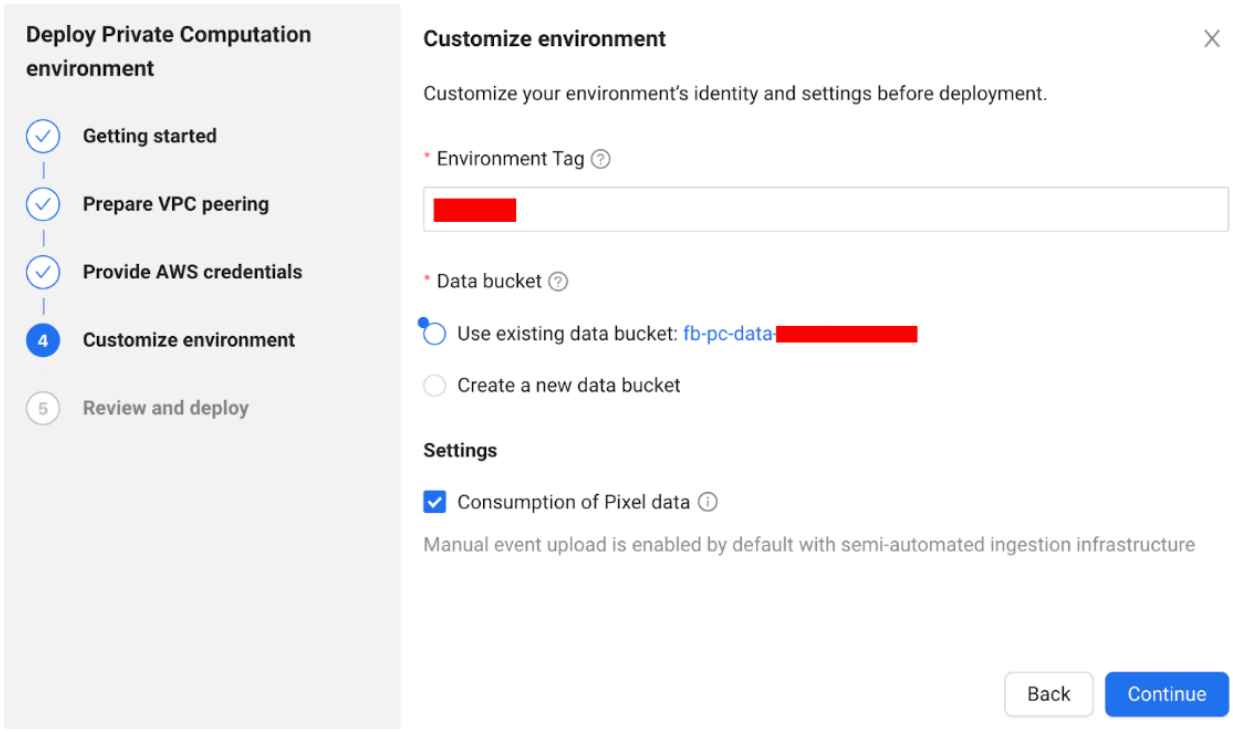
- Enter your business id, and the Graph API token generated in Step 2. Then click the button “Get Meta VPC details” the AWS region and peered Meta-side VPC ID should pop-up (as a reference).
- Note: Please Only click “use advanced settings” if advised by a META representative. The advanced settings option is described here.



- Press continue to go to the next screen.
- You should be in credential screen now as below:

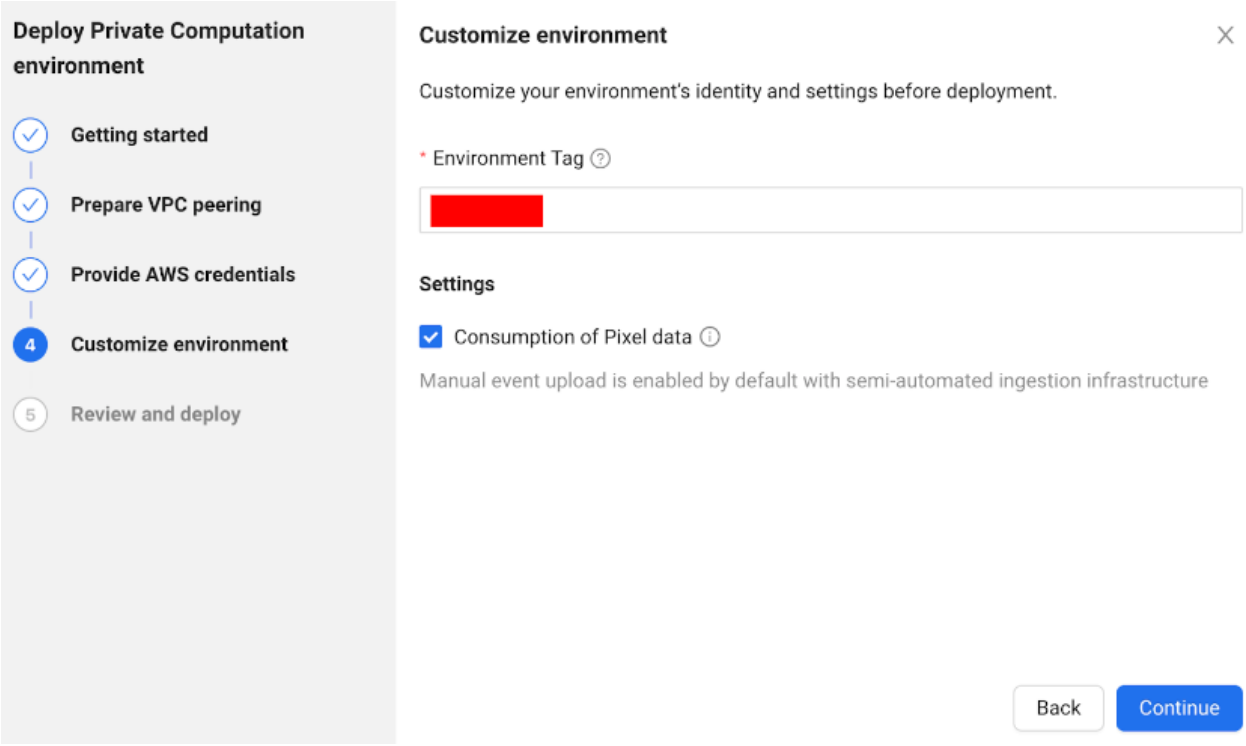


- Enter AWS Access Key ID and Secret Access Key, your AWS account ID and click on continue. These credentials should have admin access to create new components - S3 Buckets, Kinesis, VPC, Subnets, ECS Clusters.



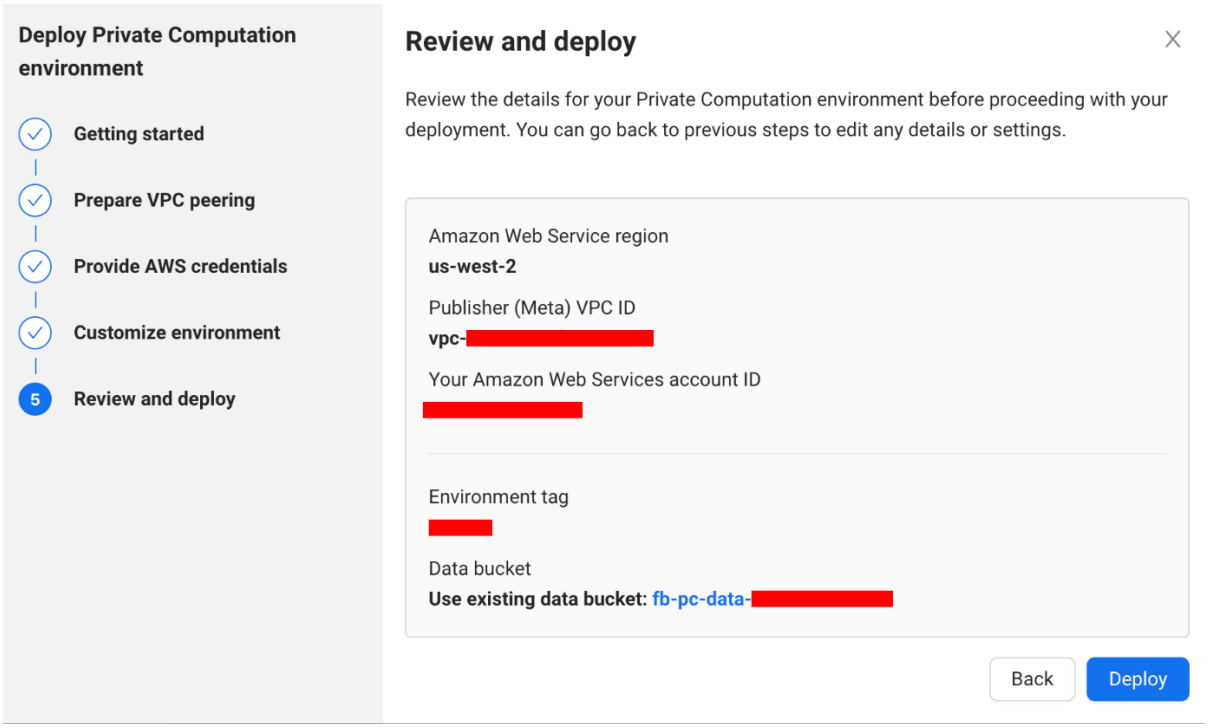
- In step 4, You can customize the environment. Please fill in the required fields and click on Next:
  Environment tag: a string that will be appended to the name or tag of AWS resources to be created. It will be easier for you to identify which AWS resources are created. For ease, we have pre-generated a tag for you (using “<month><day>” format), but you can change the tag based on your suitable name.
  Data bucket: this is the S3 bucket is where data for the computation is stored. If you are redeploying PCS, you have the option to reuse the existing data bucket or create a new bucket for this deployment.

  Note: If a suitable bucket could not be found, the screen will look as show below:

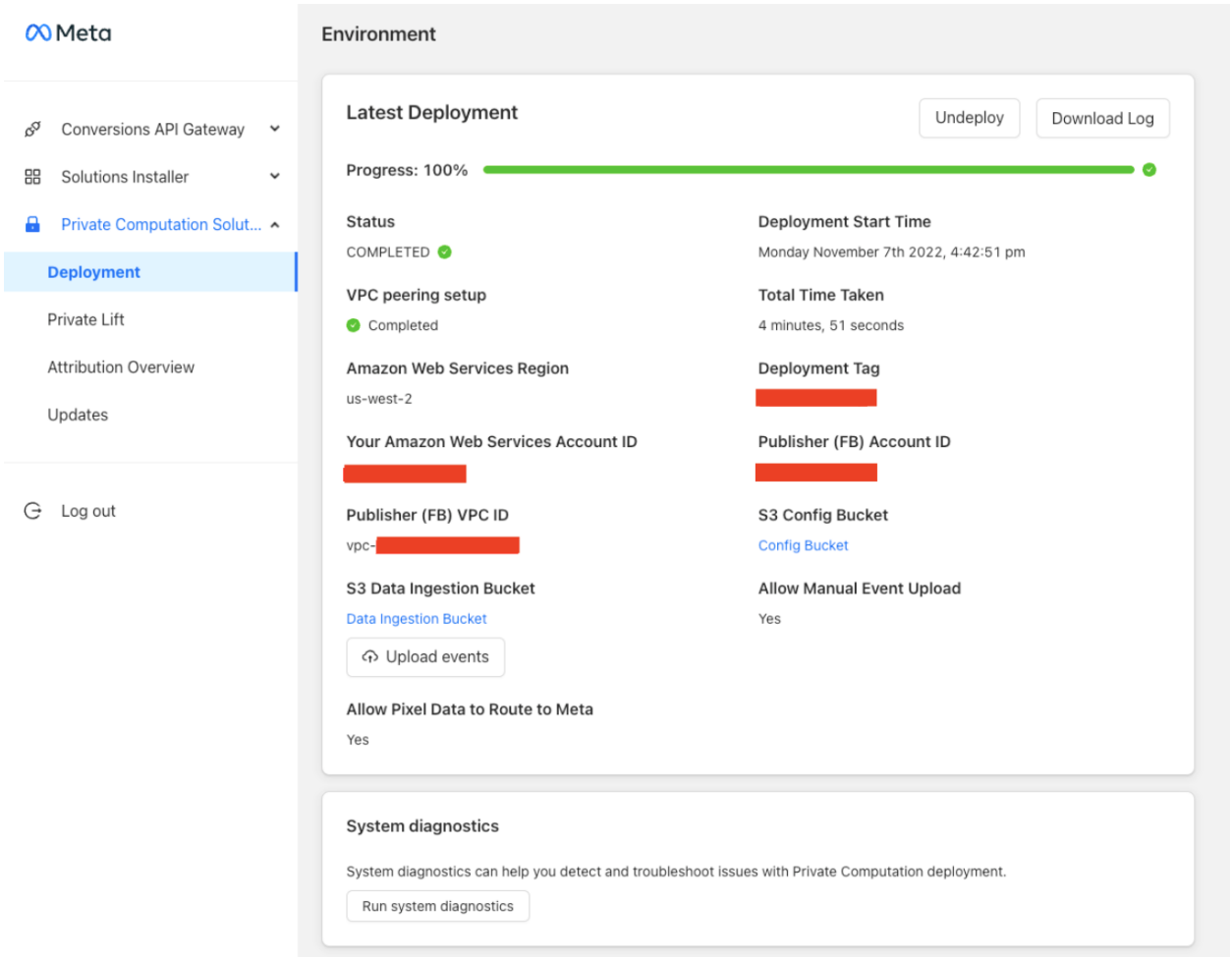


  Data Ingestion Settings:
  - We have enabled Manual event upload pipeline by default. This is required for using the Events Uploader modal.
  - Toggle the checkbox “Consumption of Pixel data” if you don’t wish to send pixel events back to Meta as Conversions API events. If you already have Conversions API integration in place, or if you do not wish to forward your web pixel events as Conversions API events, you can toggle this button to off.

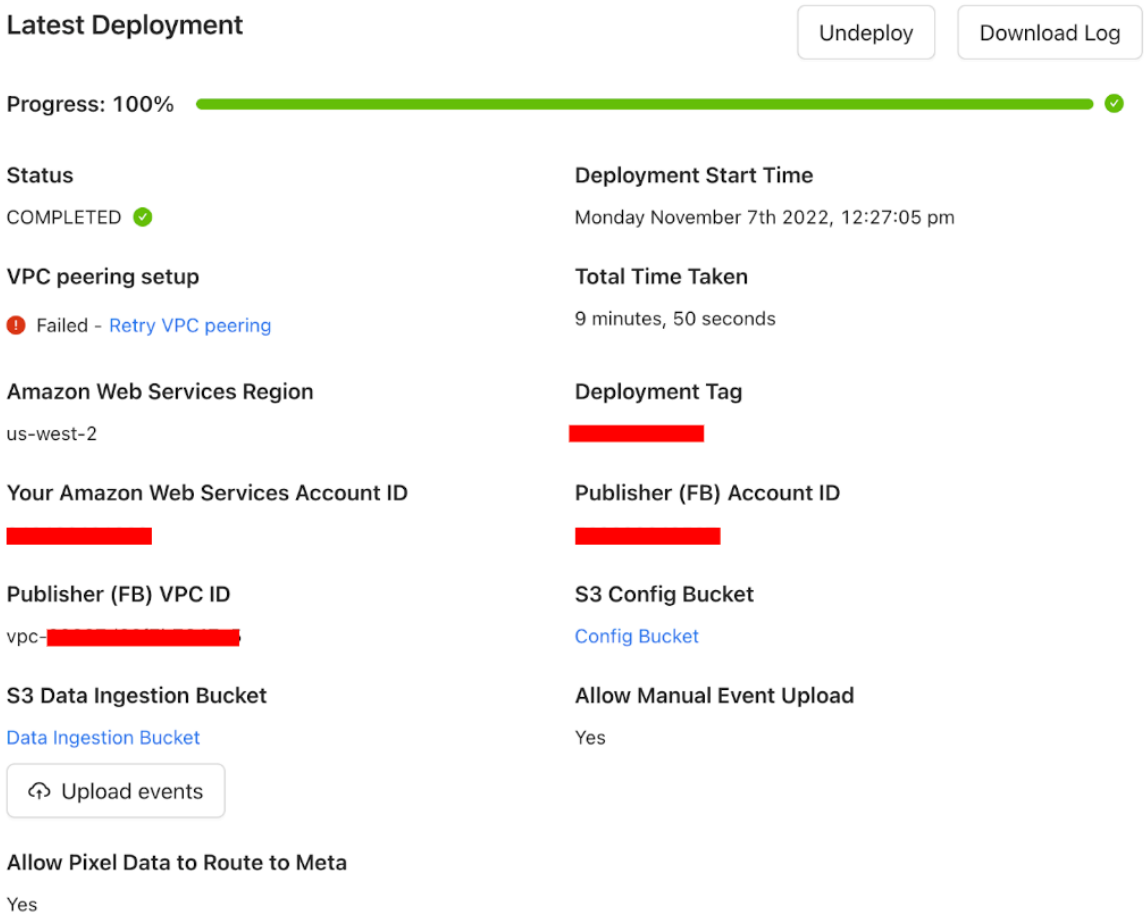
- Next, review and deploy. This is the final step before actual infrastructure deployment starts, Please review the information for a moment before you can click the deploy button.



- About 10 minutes later, you should be able to see the following screen confirming the successful deployment.



- VPC peering status: If the VPC peering has been completed you should see “Completed” status under “VPC peering setup”, and if it’s failed you should see failed status shown below. Please follow the appendix here on how to retry a failed VPC peering connection, and how to proceed forward.



Lastly, please complete below steps:

- Open CAPI API Gateway Shell: `https://<capig.instance.url>/hub/shell`
- Run the following update commands after updating the placeholders (AWS access key and AWS secret key):
    - config write CloudResources /AWS_ACCESS_KEY "<Your aws_access_key_id>"
    - config write CloudResources /AWS_SECRET_KEY "<Your aws_secret_access_key>"
-

Note: The AWS access key and secret key needs either: admin-level access to all AWS services, or a minimal set of necessary permissions (see A3 for more details)

Verify infra completeness and connectedness

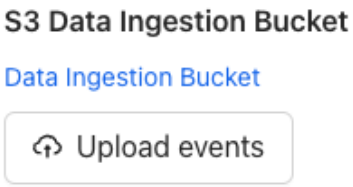
Before moving forward, please

1. Confirm with your Meta POC if they have made changes to routing tables and provided access to ECR repositories. You can follow the instructions to run PCE Validator to ensure the setup is correct.
2. (Recommended) You can run an ad-hoc system diagnosis to validate the cloud infra setup.

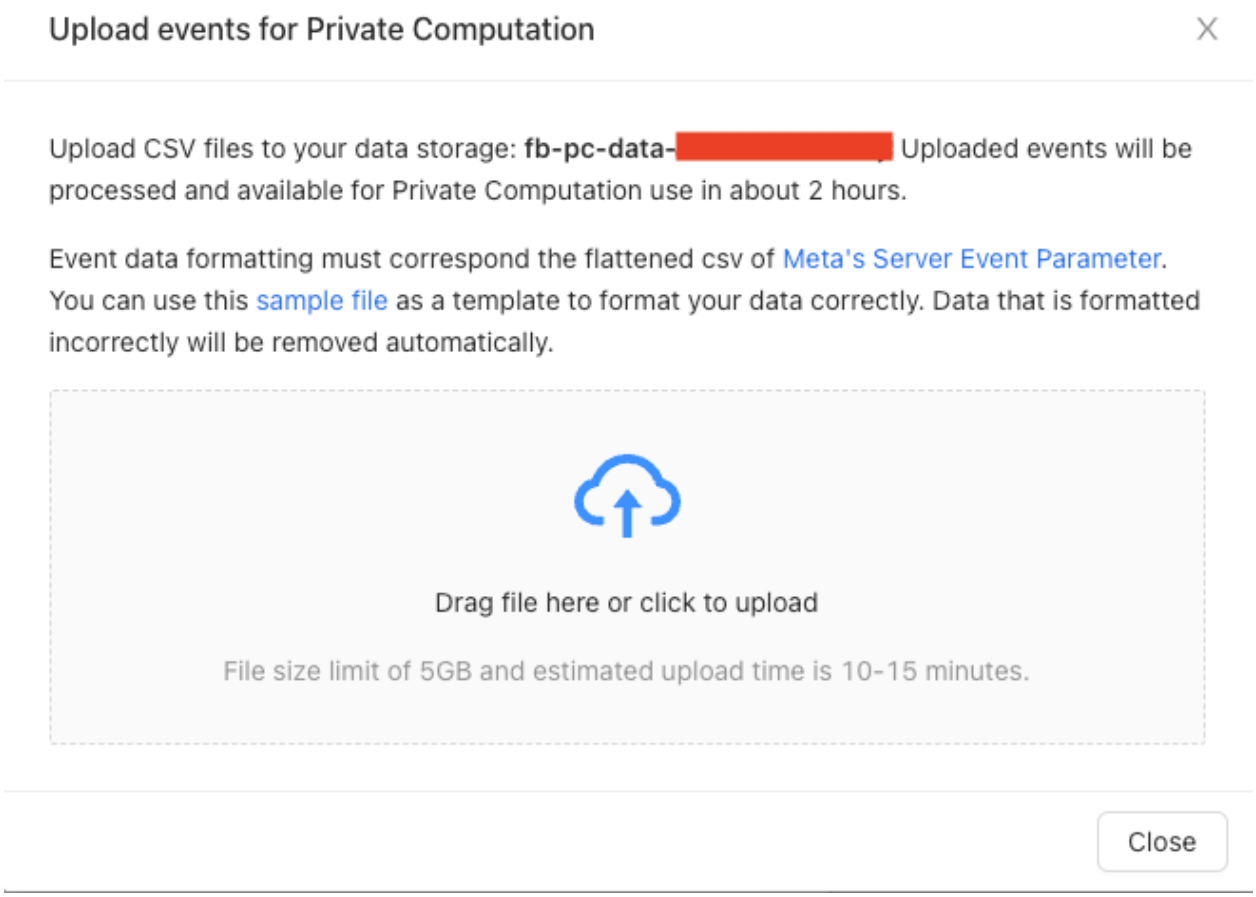
Data Ingestion

There are two different ways to ingest your data.

1. Automated data ingestion and computation:
    a. No further action is required.
    b. Depending on your needs and study setup, different wait time could apply. Your Meta representative will guide you on the exact wait time.
2. Prepare your own conversion data
    a. Using the semi-automated ingestion pipeline (Manual Data Upload). It should take less than 30 mins to ingest multi-month conversion data.
    b. UI option for uploading conversion events data in CSV format
        i. Navigate to the deployment summary page
            1. `https://<capig.instance.url>/hub/pcs/deployment`
        ii. Click on the 'Upload events' button under the 'S3 Data Ingestion Bucket' section



        iii. Prepare your data in the semi-automated events data format (Appendix A1). Open the 'sample file' link for an example of this data format.
            1. Maximum upload size per file: 5GB
        iv. Upload the events files to the upload modal by either selecting or by dropping the file(s)



v. Note - if you see an error, try refreshing the page first and then reopen the uploader modal. If the error persists and you are unable to resolve it, please reach out to your Meta representative.
vi. If you see the 'JOB_NOT_PROVISIONED_ERROR' then please refer to this section for some ideas on how to resolve it.
vii. If you see the 'BUCKET_CORS_MISSING_ERROR' then please refer to this section for some ideas on how to resolve it.

c. S3 API option: Please visit Appendix Semi-auto data ingestion/preparation for more details on uploading events data to s3.

3. (optional) PL Synthetic testing
   a. While we wait for real data to accumulate, you/advertisers can leverage a "synthetic" lift study (e.g., all synthetic, fabricated data on both sides) to test the pipeline E2E (including AWS infra setup, PL binaries correctness and VPC connection with Meta side). It could provide you more streamlined onboarding experiences, enabling the faster feedback loop to flag errors along the pipeline. Please reach out to your Meta representative for more details

Step 4: Private Computation Runs

[WIP] Prerequisites

Step 1: Check config before running computation (5 mins)

1. Open CAPI API Gateway Shell: `https://<capig.instance.url>/hub/shell`
2. Run the following commands
    a. config read Kinesis
        - Expected values:

```
{
  "PUBLISH_TO_KINESIS" : true,
  "BATCH_PUBLISH_PERIOD" : 1000,
  "BATCHING_ENABLED" : true,
  "FIREHOSE_DELIVERY_STREAM_NAME" :
"cb-data-ingestion-stream-<TAG>",
  "AWS_REGION" : "<AWS REGION>"
}
```

For AWS_REGION, it should be lower case and format like “us-west-2”

    b. config read Athena
        - Expected values:

```
{
  "AWS_REGION" : "<AWS REGION>",
  "CATALOG_NAME" : "AwsDataCatalog",
  "DATABASE_NAME" : "mpc-events-db-<TAG>",
  "TABLE_NAME" : "fb_pc_data_<TAG WITH UNDERSCORE>",
  "QUERY_RESULTS_S3_BUCKET_PATH" :
"s3://fb-pc-data-<TAG>/query-results/",
  "ID_FIELDS" : "user_data.device_id,user_data.email"
  "USE_MULTIKEY" : false,
"MULTIKEY_ID_FIELDS":"user_data.device_id|id_device_id,user_data.em
ail|id_email,user_data.processed_client_ip_address|id_ip"
}
```

- For AWS_REGION, it should be lower case and format like “us-west-2”
- For ID_FIELDS
    - If your data only has email PII data. Please update the ID_FIELDS to email only with following command
        - config write Athena /ID_FIELDS "user_data.email"
    - If your data only has device_id PII data. Please update the ID_FIELDS to device_id only with following command
        - config write Athena /ID_FIELDS "user_data.device_id"

    c. config read CloudResources
        - Expected value for a new deployment:

```
{ "AWS_ACCESS_KEY" : "",
  "AWS_SECRET_KEY" : "",
  "AWS_SESSION_TOKEN" : "",
  "CONFIG_FILE_S3" :
"s3://fb-pc-config-<TAG>/config.yml",
  "IMAGE_TAG" : "latest",
"USE_IAM_USER_AUTH" : false
}
```

- Expected value for an older deployment:

```
{ "AWS_ACCESS_KEY" : "<YOUR AWS ACCESS KEY>",
  "AWS_SECRET_KEY" : "<YOUR AWS SECRET KEY>",
  "AWS_SESSION_TOKEN" : "",
  "CONFIG_FILE_S3" :
"s3://fb-pc-config-<TAG>/config.yml",
  "IMAGE_TAG" : "latest",
"USE_IAM_USER_AUTH" : false
}
```

To enable the data ingestion to S3 using CAPIG, please complete below steps:

- Open CAPI API Gateway Shell: `https://<capig.instance.url>/hub/shell`
- Run the following update commands after updating the placeholders (AWS access key and AWS secret key):
  - config write CloudResources /AWS_ACCESS_KEY "<Your aws_access_key_id>"
  - config write CloudResources /AWS_SECRET_KEY "<Your aws_secret_access_key>"

    - **Note**: The AWS access key and secret key needs either: admin-level access to all AWS services, or a minimal set of necessary permissions (see A3 for more details)

Step 2: (optional) automate diagnostic data sharing with Meta

To help clients better troubleshoot issues and improve the product, it's highly recommended to opt-in for diagnostic data sharing with Meta. it will automatically upload logs to Meta within 5 minutes after a completed (either success or failed) run. No customer data (e.g., user identities, pixel events) will be included in the collected diagnostic data, and the retention days is 30-day maximum, with access controlled. Note that:

- Only for the Private Lift. We will add support for Private Attribution later.
- Logs collection won't happen if the computation run failed to start, e.g., due to invalid AWS credentials assigned to config values, failure in input data preparation.

Steps

- Go to CAPI-G Shell: `https://<capig.instance.url>/hub/shell`

- Run the following commands:
  - `config write pclogs /ENABLE_AUTO_UPLOAD true`
  - `config write pclogs /UPLOAD_LOGS_FROM_SUCCESS_RUN true`
- To double check it succeeded:
  - Run the following commands:
    - `config read pclogs`
  - Expected value:

```
{ "ENABLE_AUTO_UPLOAD" : true,
  "UPLOAD_LOGS_FROM_SUCCESS_RUN" : true}
```

- Make sure the following section is in the account policies to enable collection of diagnostic data, and add the section if it does not already exist. Here are the instructions on how to ensure that the required permission exists.

```
{
    "Action": [
        "logs:*"
    ],
    "Effect": "Allow",
    "Resource": "*"
},
```

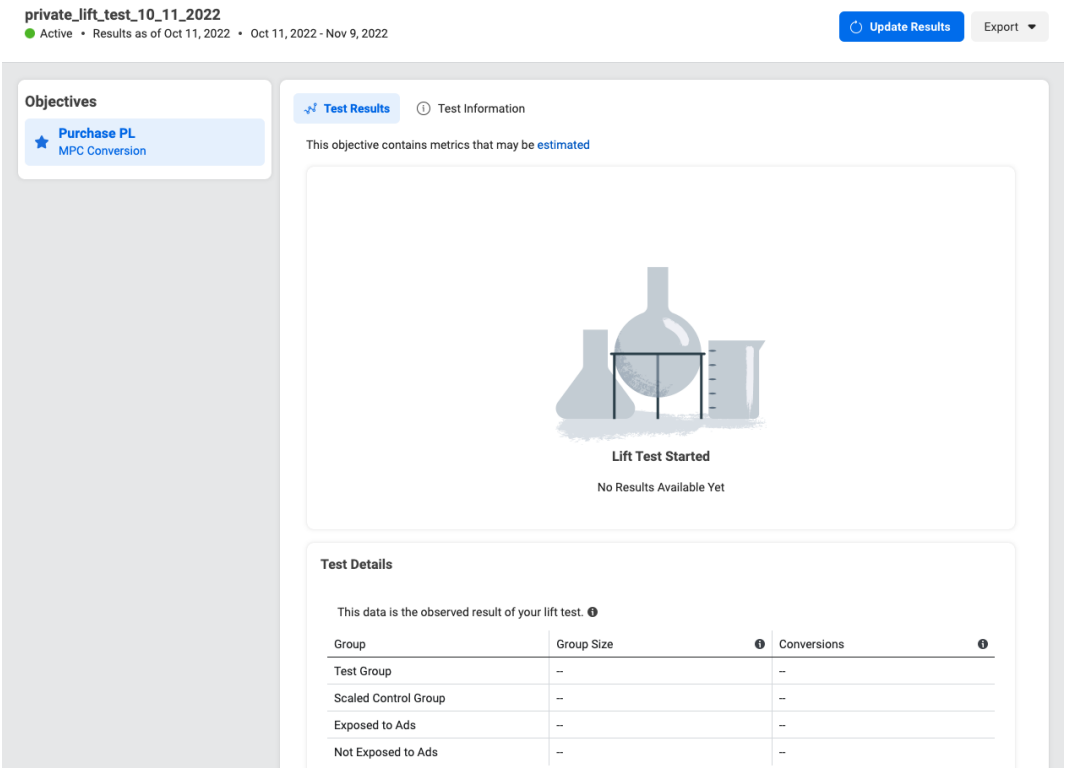
More details can be found in A6: Sharing diagnostic data with Meta.

Now you are ready to use Private Computation products. Follow the section below to run Private Lift, or go to this section to run Private Attribution.

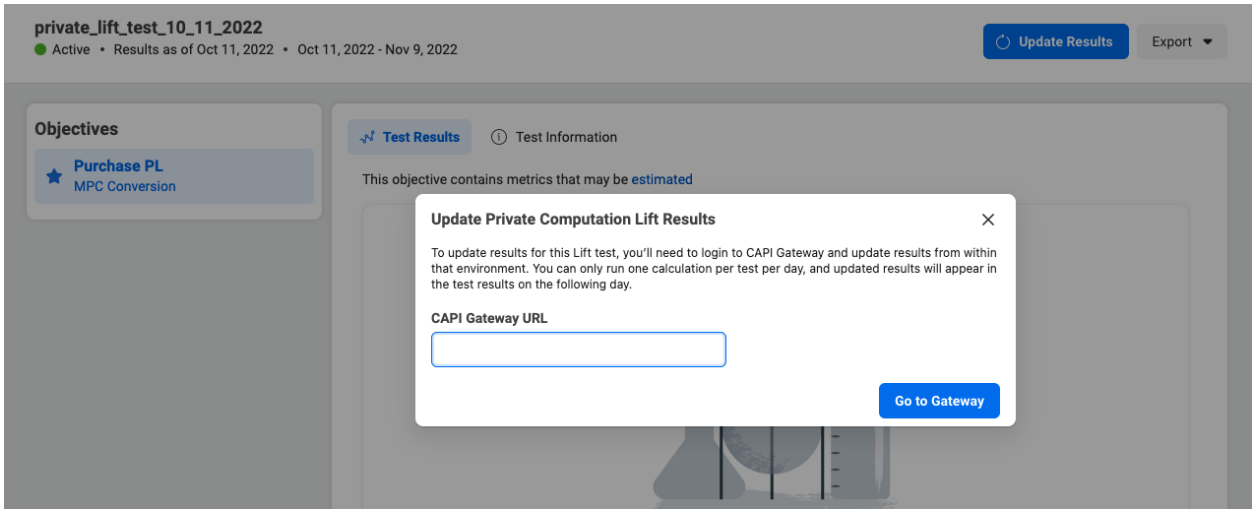
Private Lift

Step 1: Run Private Lift Computation (15 mins)

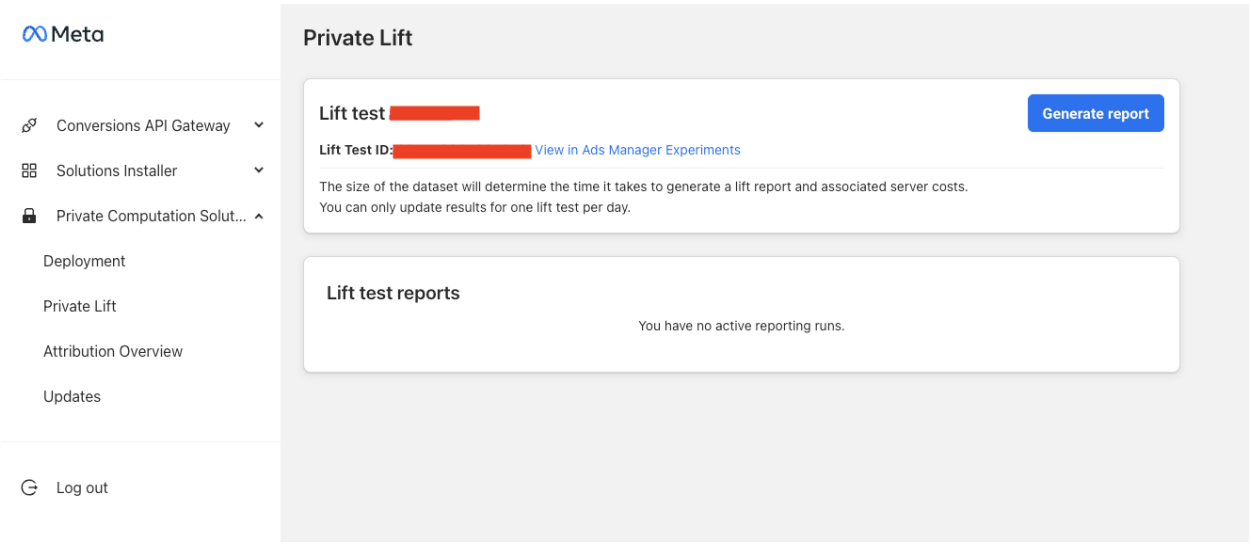
- Go to Lift Report UI (sample URL: `https://business.facebook.com/ads/lift/report/?ad_study_id=<your ad_study_id>`) and select a MPC Conversion objective
  - replace <your ad_study_id> with your own study id



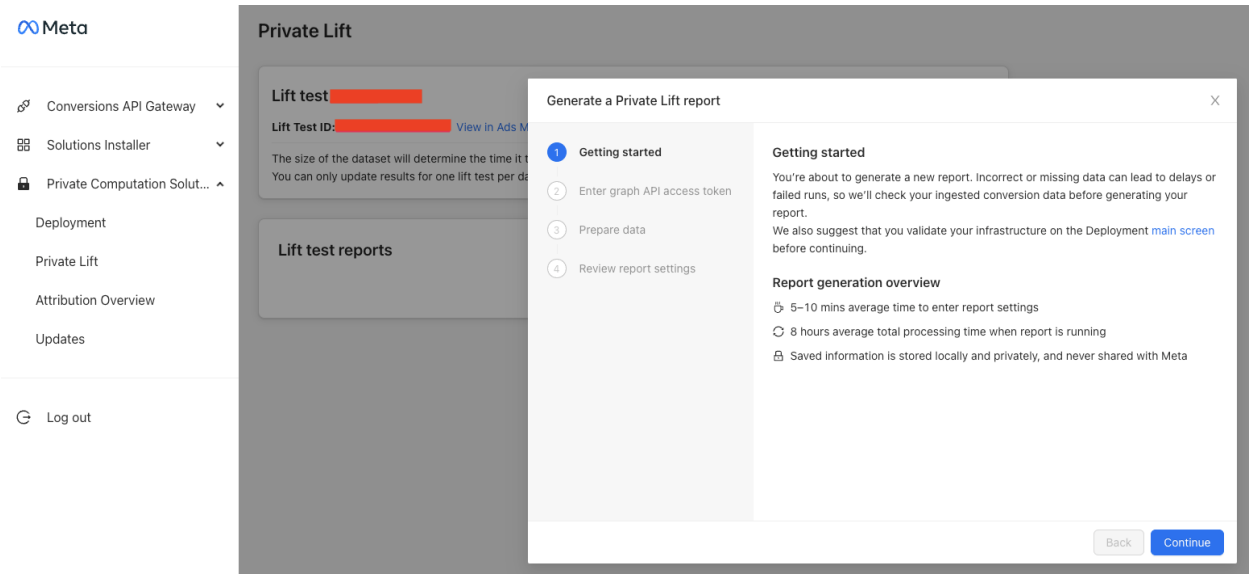
- Click on “Update Results”. A new window will pop-up, enter your Conversions API Gateway instance url here and click on “Go to Gateway”.



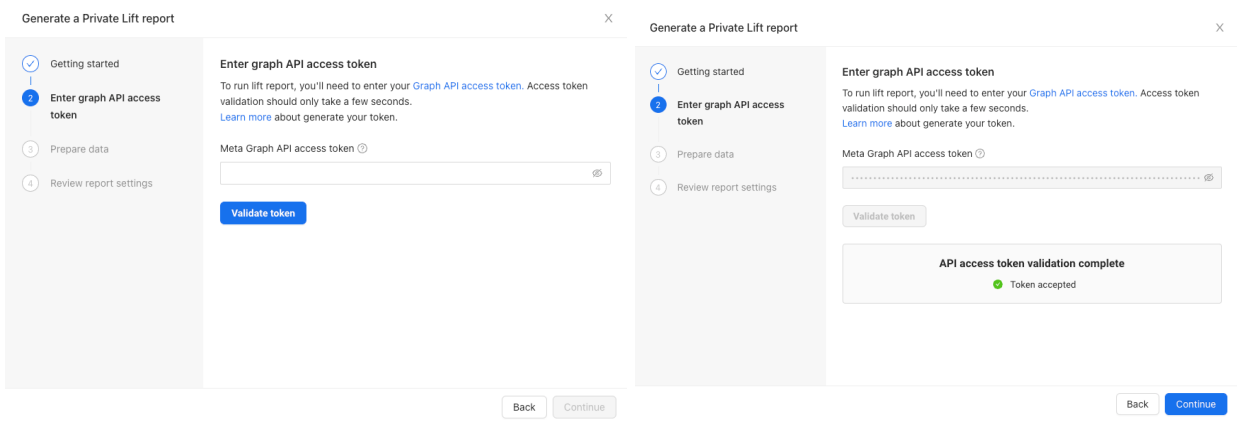
- You are now re-directed back to CAPI Gateway. To start the computation, click on “Update Results”.
  - Format: `https://<capig.instance.url>/hub/pcs/calculation/<your ad_study_id>/<your ad_study_name>`
- Enter the following URL and click on “Go to Gateway”.



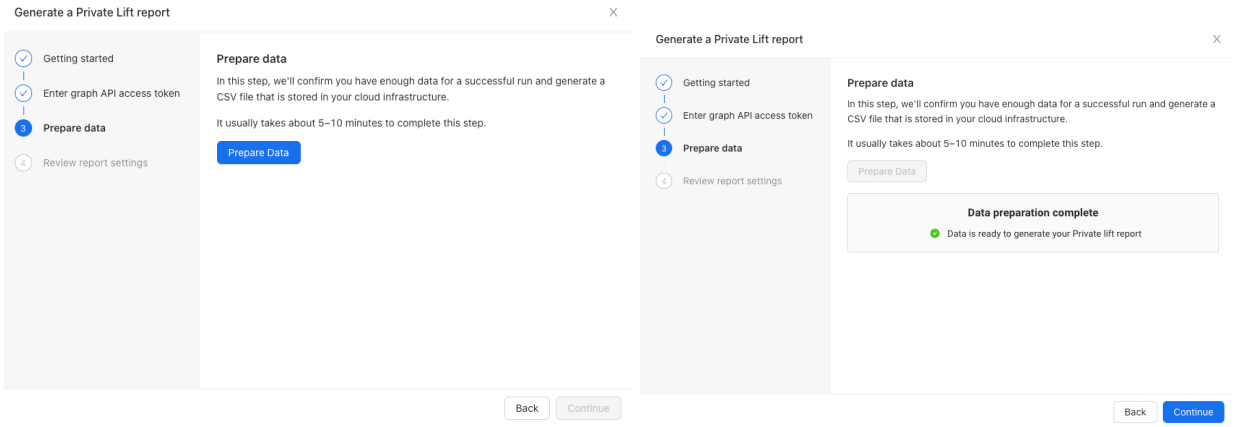
- To proceed, click “Generate report” to launch a pop-up window with instructions and multiple steps to guide you through.



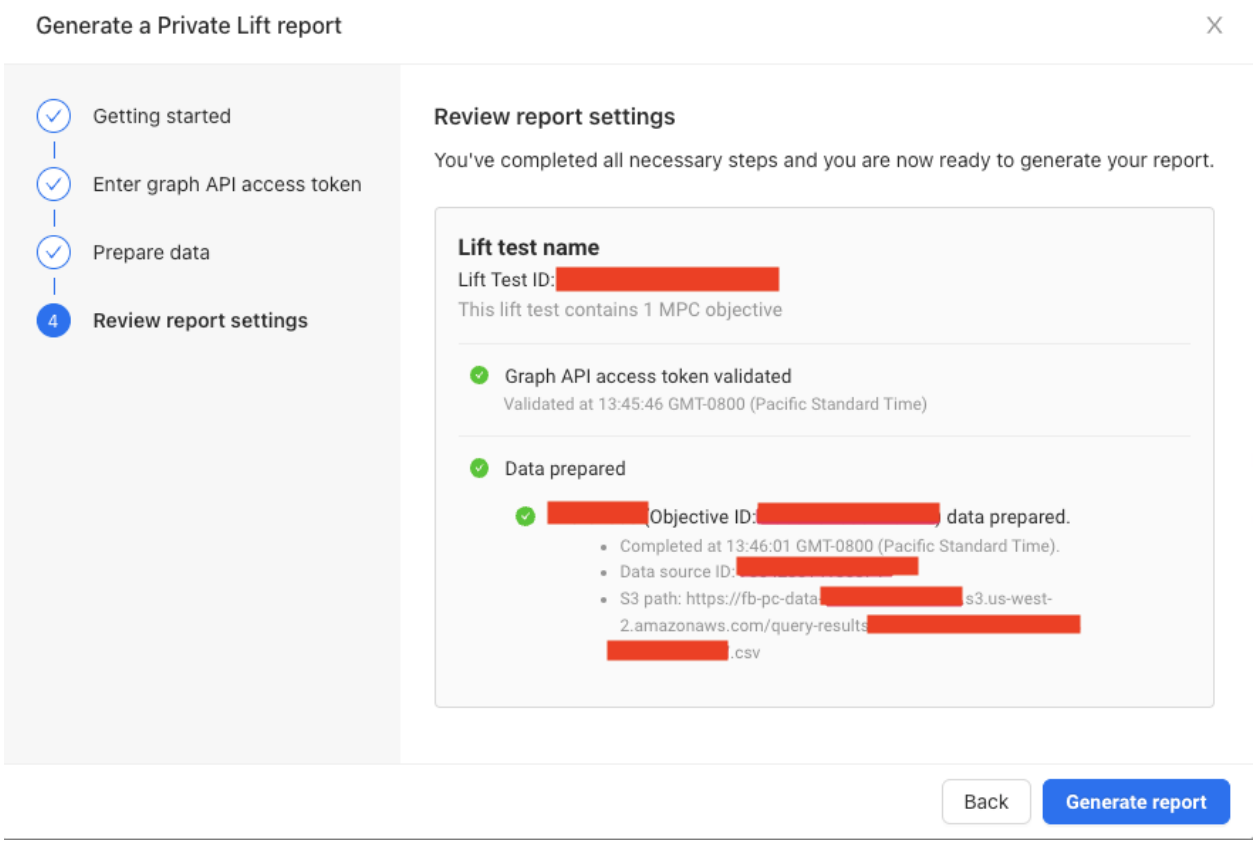
- After reading the instructions and basic info on “Getting started” step, click “Continue” to go to the next step



- Enter the Graph API token generated in Step 2 and click on “Validate token”. Once the token is validated, click “Continue”.



- Click “Prepare Data” to confirm if enough data has been ingested for a successful run, and generate a CSV file that is stored in S3 data ingestion bucket. Once data preparation completes, click “Continue”.

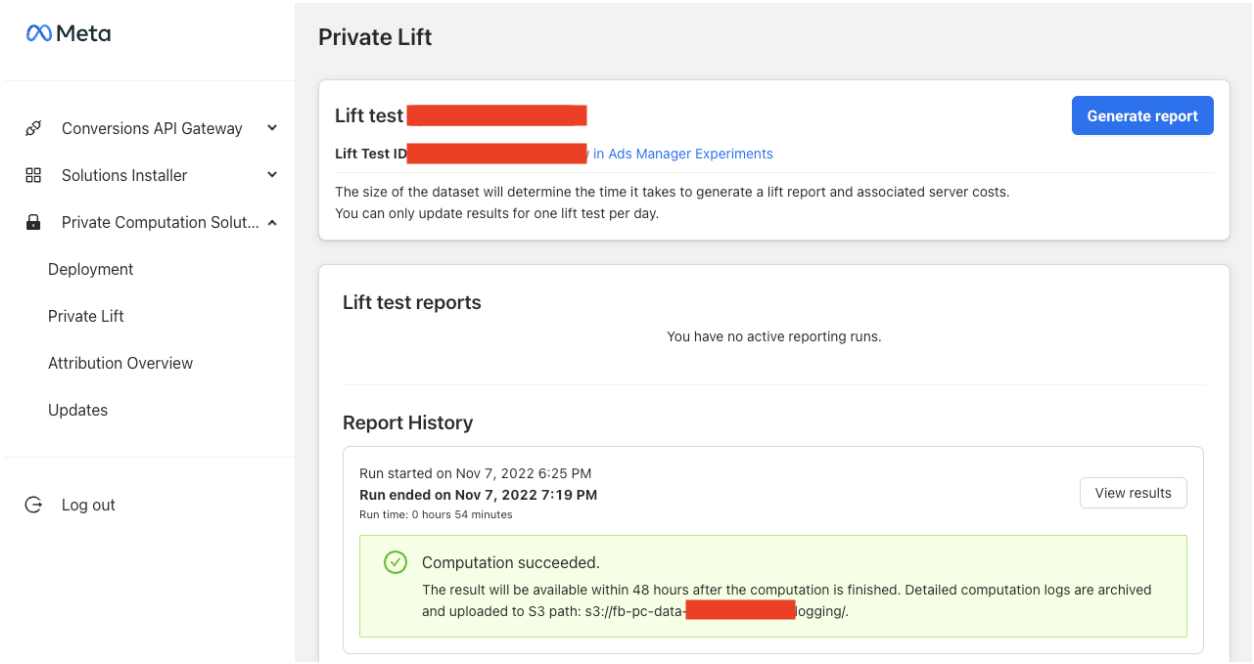


- After reviewing the Private Lift report settings, access token validation, and data preparation results, if everything looks good, click “Generate report” to start generating the Private Lift report. Please note that computation will run for approx 3 - 6 hours (at most 24 hours) before completion.
- Once the computation begins, logs will be printed to output.txt in your S3 bucket under the directory `<data`

bucket>/query-results/fbpcs_instances_<studyId>_<postfix>. This will be a key resource to monitor and use for debugging purposes in case any issue occurs.

Step 2: View Private Lift Results

- After the computation is complete, click on “View Results” to navigate to Lift UI. It can take up to 2 days for the results to populate.



Private Attribution

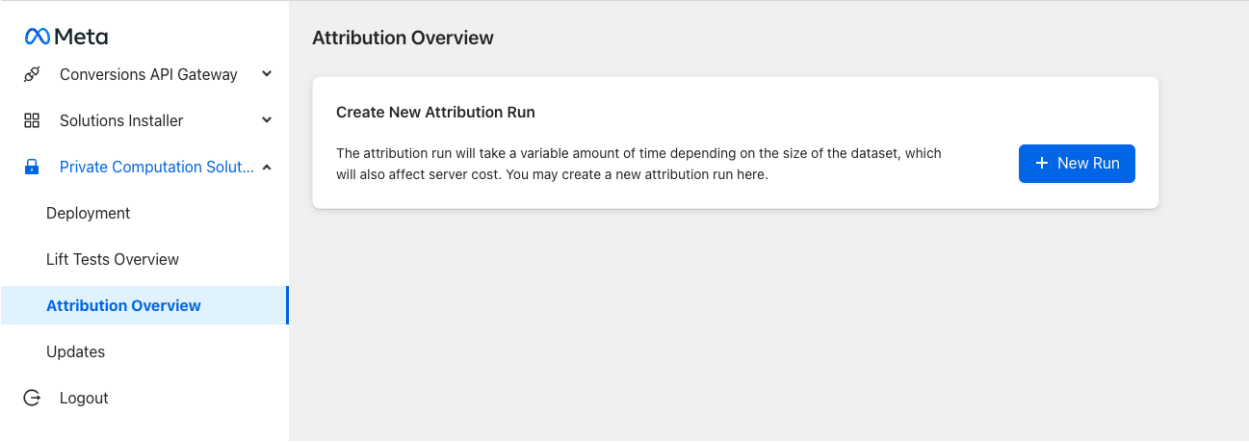
Step 0: Preparation

Request the following information if your Meta representative hasn't provided them to you:

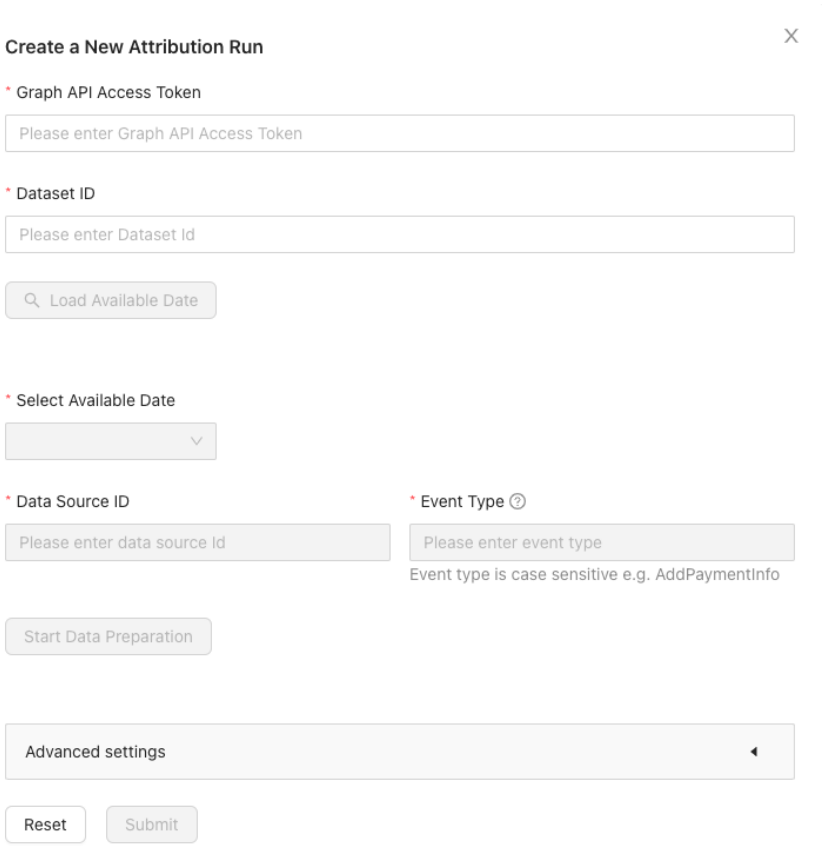
- Dataset Id

Step 1: Run Private Attribution Computation (30 mins)

- Login to your CAPI Gateway instance.
- Navigate to the "Attribution Overview" page.

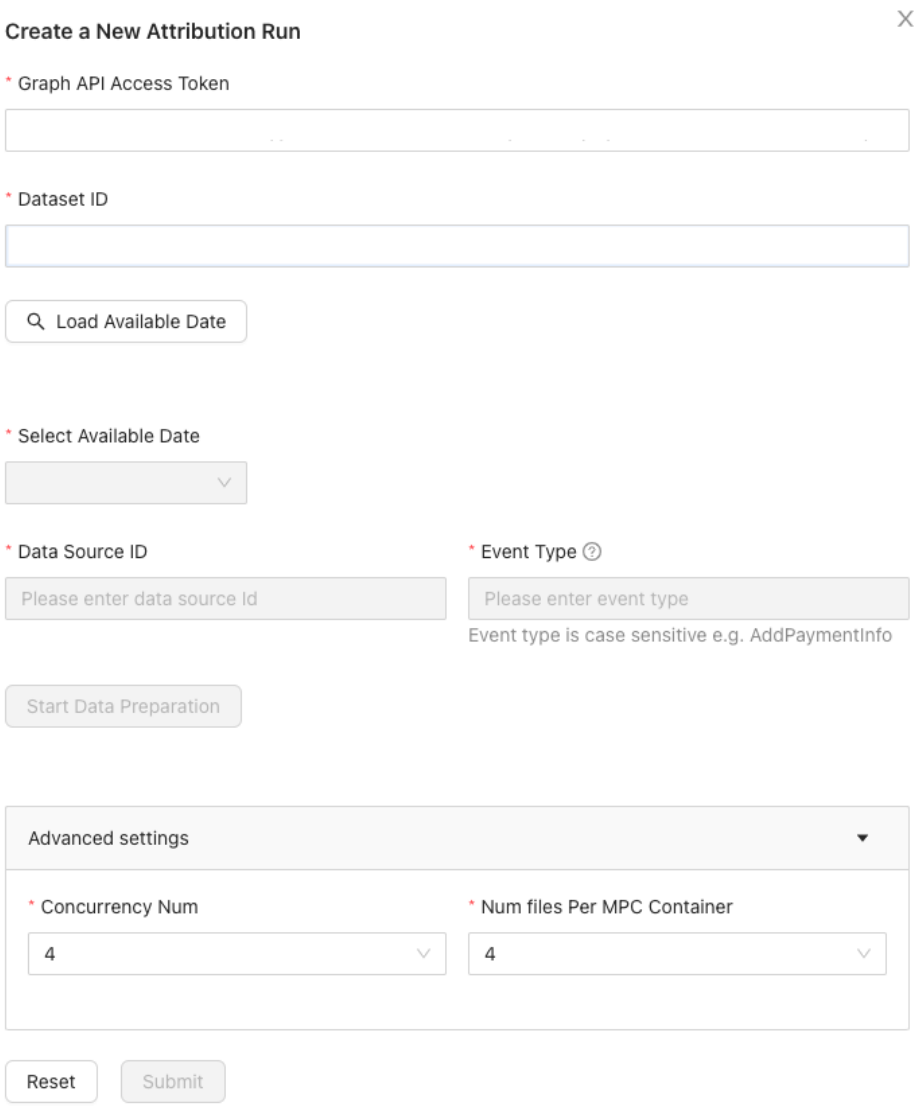


- Click the "New Run" button, and a dialog will appear.



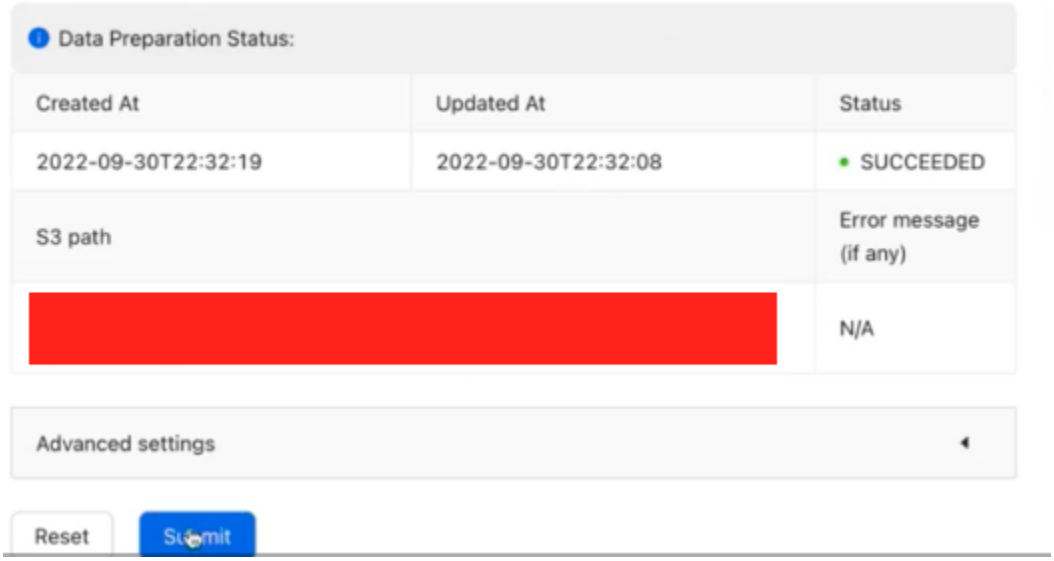
- Enter the Graph API token generated in Step 2, and the Dataset Id you get from your Meta representative.
- Click the “Load Available Date”.
- Choose any date from the “Select Available Date” dropdown list. If you don’t know which one to choose, use the latest one or ask your Meta representative.
- Enter the “Data Source Id” and “Event Type”. If you don’t know what the correct values should be, ask your Meta representative.

- Click the “Start Data Preparation”.



- Wait until it shows “SUCCEEDED”.

- Click the “Submit” button.



- Once the computation begins, logs will be printed to output.txt in your S3 bucket under the directory `<data bucket>/query-results/fbpcs_instances_<dataset id>_<dataset timestamp>_<postfix>.` This will be a key resource to monitor and use for debugging purposes in case any issue occurs.

Step 2: View Private Attribution Results

Ask your Meta representative for the results.

Appendix

A1: Semi-auto data ingestion/preparation

Github URL

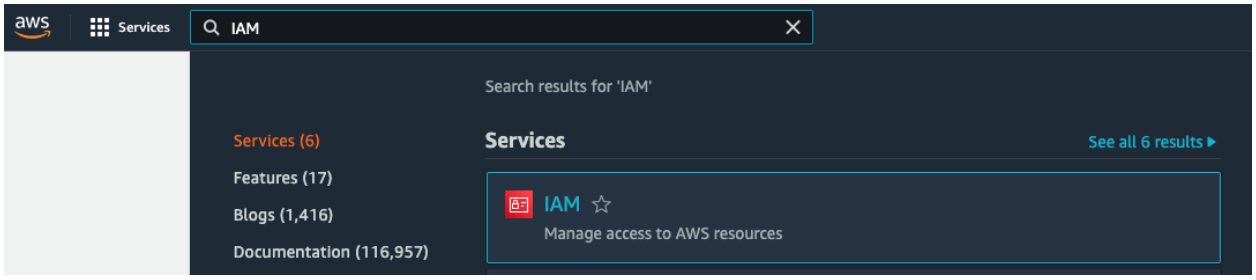
A2: How to set “canary” tier

Sometimes Meta would like you to run on the “canary” tier. Here is how you can set it up.

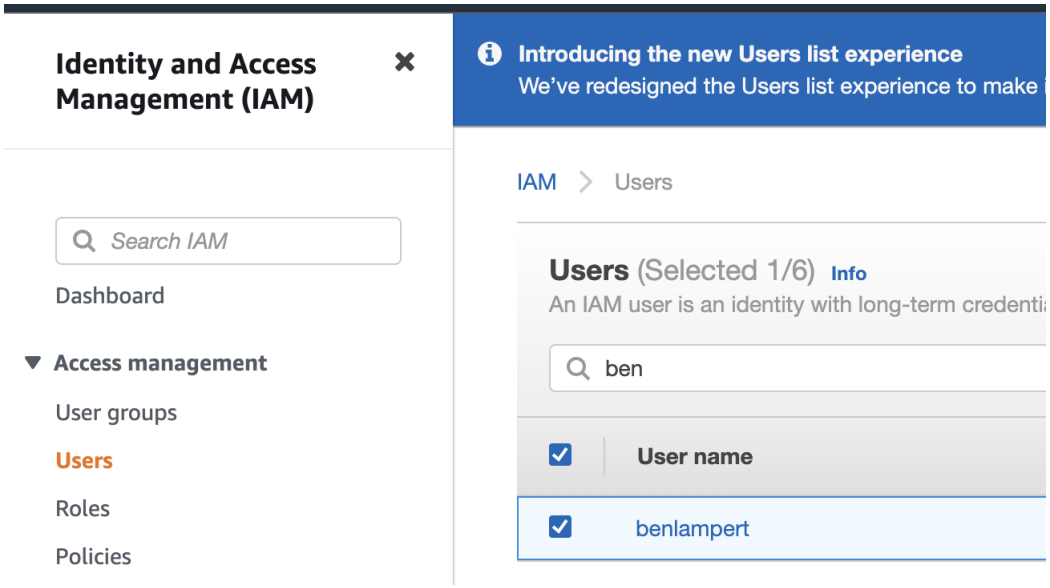
1. Open CAPI API Gateway Shell: `https://<capig.instance.url>/hub/shell`
2. Run the following update commands:
    a. `config write CloudResources /IMAGE_TAG canary`

A3: Configure an AWS IAM user with minimal permissions for future computation after initial infra deployment

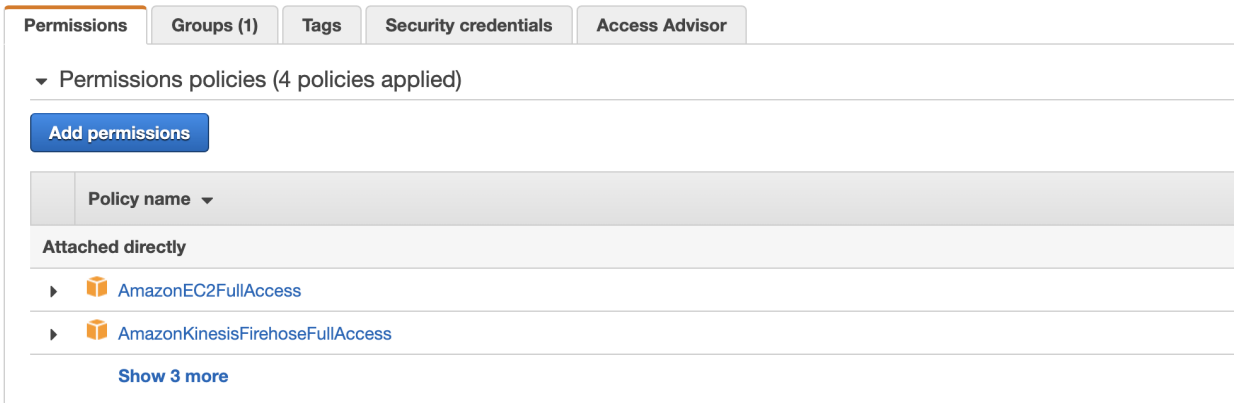
a. Open your AWS account and enter IAM component



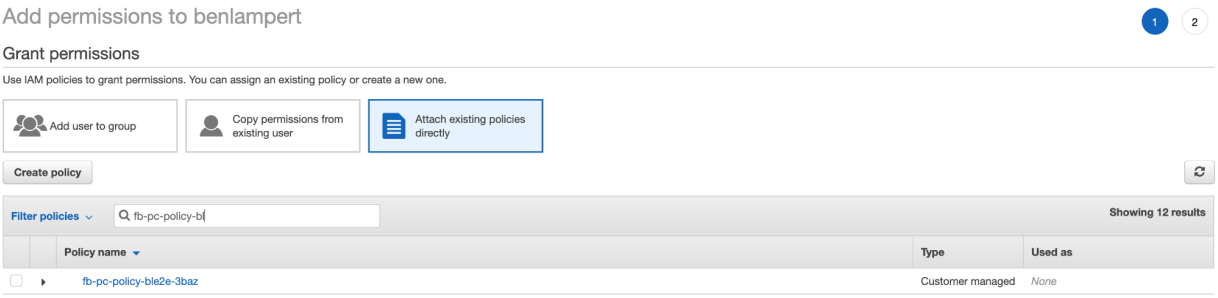
b. Select an AWS IAM user, either create an user or re-use an existing one.



c. Enter into user page and click “add permissions”



d. Attach the “fb-pc-policy-<tag>” policy to the user.



e. Add permission.
f. Then you could generate the access_key and Secret_key for this user to fill in the next step in cloudbridge.

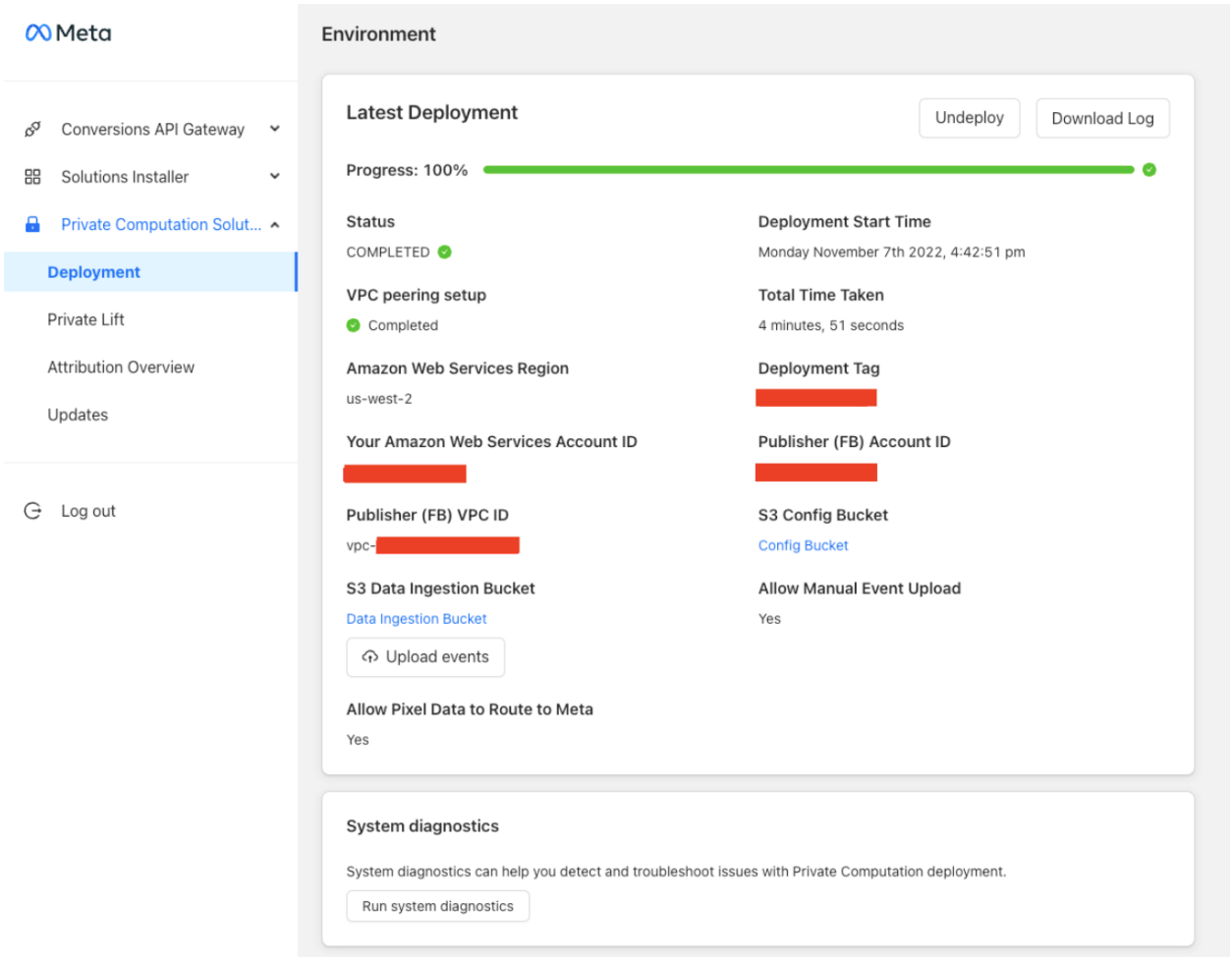
A4: Data Migration

Github URL.

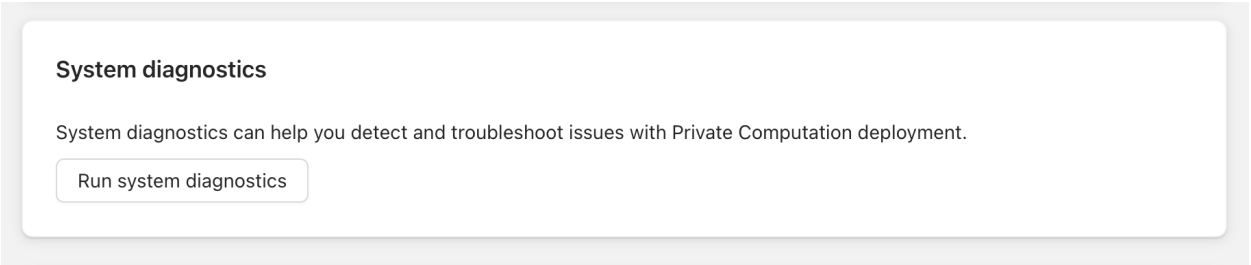
A5: Ad-hoc system diagnosis

System diagnosis is a way to validate your deployed infrastructure completeness and connectedness (including VPC peering) before we kick off any computation runs.

Once you have completed the AWS infrastructure deployment, you shall see a summary page. Under the deployment summary page a new section is added for validating infrastructure dynamically as show below:



Click *run system diagnostics* and provide AWS admin-level credentials to continue.
If the system diagnosis finished successfully, you will see the following result:



In case of any failure, you will see a download button to download logs, which you can then share to Meta to further help in debugging.

A6: Sharing diagnostic data with Meta

For a partner to help Meta troubleshoot issues and improve the product, you can send diagnostic data to Meta either manually or automatically. No customer data (e.g., user identities, pixel events) will be included in the collected diagnostic data.
Limitations for automatic collection of diagnostic data:

- Currently only for Private Lift on top of the CAPI-G Computation UI.
- Logs collection happens at the end of a computation run.
- Logs collection won't happen if the computation run failed to start, e.g., due to invalid AWS credentials assigned to config values, failure in input data preparation.

Manual sharing with Meta

The diagnostic data is always collected automatically after every study run completes (with success or failure), and is saved to two locations in the S3 bucket used for input data, in the advertiser's cloud account:

- In the folder s3://fb-pc-data-<ENVIRONMENT_TAG>/logging/. Log archive file is like logs_20221105T044117.481056Z_study-14827452455_run-12.zip. The archive file contains multiple logs from: output.txt (i.e. coordinator logs), worker containers, data pipeline (Athena, Kinesis, Glue, Crawler).
- In the folder containing result data, e.g. s3://fb-pc-data-<ENVIRONMENT_TAG>/query-results/fbpcs_instances_14827452455_12/. Log files can be: output.txt, job-debug.txt and download_logs_cli.txt. Output.txt is the same as in the above archive file. The other two files help debugging the logs collection and uploading.

Note: make sure the required section to allow access log resources is in the account policies.

When you want to share the diagnostic data to Meta manually, you can download the logs archive file, which is the comprehensive and most helpful, and share with Meta.

Automatic sharing with Meta

To further reduce the support effort by partners and shorten the latency to retrieve the logs from a partner to Meta, Meta allows a partner to opt-in for automatically logs upload. After opt-in, above diagnostic data in the logs archive will be automatically uploaded to Meta within 5 minutes after a completed study run. In Meta, the diagnostics data will be kept for no longer than 30 days, and will be access controlled.

Steps

1. It is highly recommended for a partner to opt-in for automatic logs upload, by entering the following commands in the CAPIG Gateway shell:

- Command: `config write pclogs /ENABLE_AUTO_UPLOAD true`

    - This is the primary config. Its config value is false by default, and no diagnostic data is uploaded automatically. True value means the diagnostic data after a failure study run will be automatically uploaded to Meta. This will greatly help the Meta engineers to diagnose failed study runs.
- Command: `config write pclogs /UPLOAD_LOGS_FROM_SUCCESS_RUN true`
    - This is a secondary config. Its config value is false by default, and no diagnostic data is uploaded automatically after a successful study run. True value means the diagnostic data after a successful study run will be uploaded automatically, if the above primary config has true value. This will help the Meta engineers to diagnose successful study runs that produced unexpected results, i.e., bad content from seemingly successful computation.

The partner can change the above config values at any time. The changed config values only affect uploading of diagnostic data from future computation runs, and do not impact the completed computation runs.

A7: How to enable Multi-key for Private Lift

To improve the performance and quality of matching, you can enable the multi-key feature (expected to 8 percent match rate increase, only support private lift at this moment).

1. Open CAPI API Gateway Shell: `https://<capig.instance.url>/hub/shell`
2. Run the following update commands:

    b. `config write Athena /USE_MULTIKEY true`

To disable (disabled by default) the multi-key feature, repeat the steps above but replace true with false: `config write Athena /USE_MULTIKEY true`

A8: [FYI] New Requirements on Graph API Access Token Permissions are Enforced

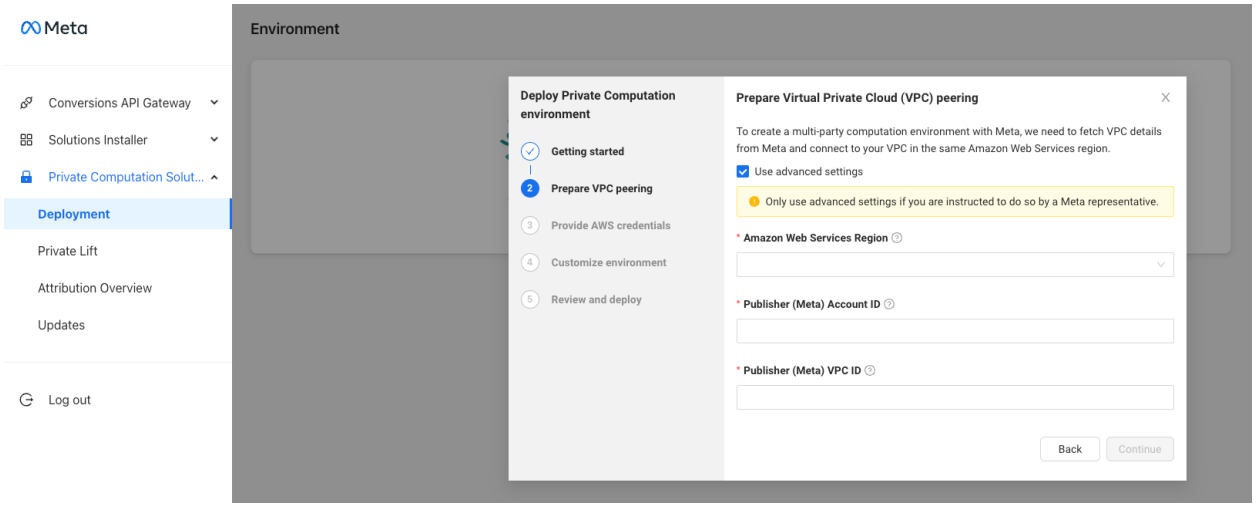
Back in June 2022, we updated instructions for generating GraphAPI token in Step 2. The consolidated list of permissions (required for both PL and PA) are: ads_management, ads_read, business_management, AND private_computation_access. We recommend you to cross-check the access token permission list, to ensure it has the full set of desired permission scopes. Here are the steps:

- Go to Access Token Debugger: https://developers.facebook.com/tools/debug/accesstoken
- Place access token in use into the input box, then click “Debug”
- Verify if all required permissions (ads_management, ads_read and business_management, and private_computation_access) are listed in “Scopes”.

- If yes, no actions needed;
- If not, we'd recommend asking the advertiser to re-generate the long-lived access token per instructions in Step 2. Once the new access token is ready, it should be good to go!

A9: Advanced setting on infrastructure deployment page , on modal stepper "get VPC details from meta"

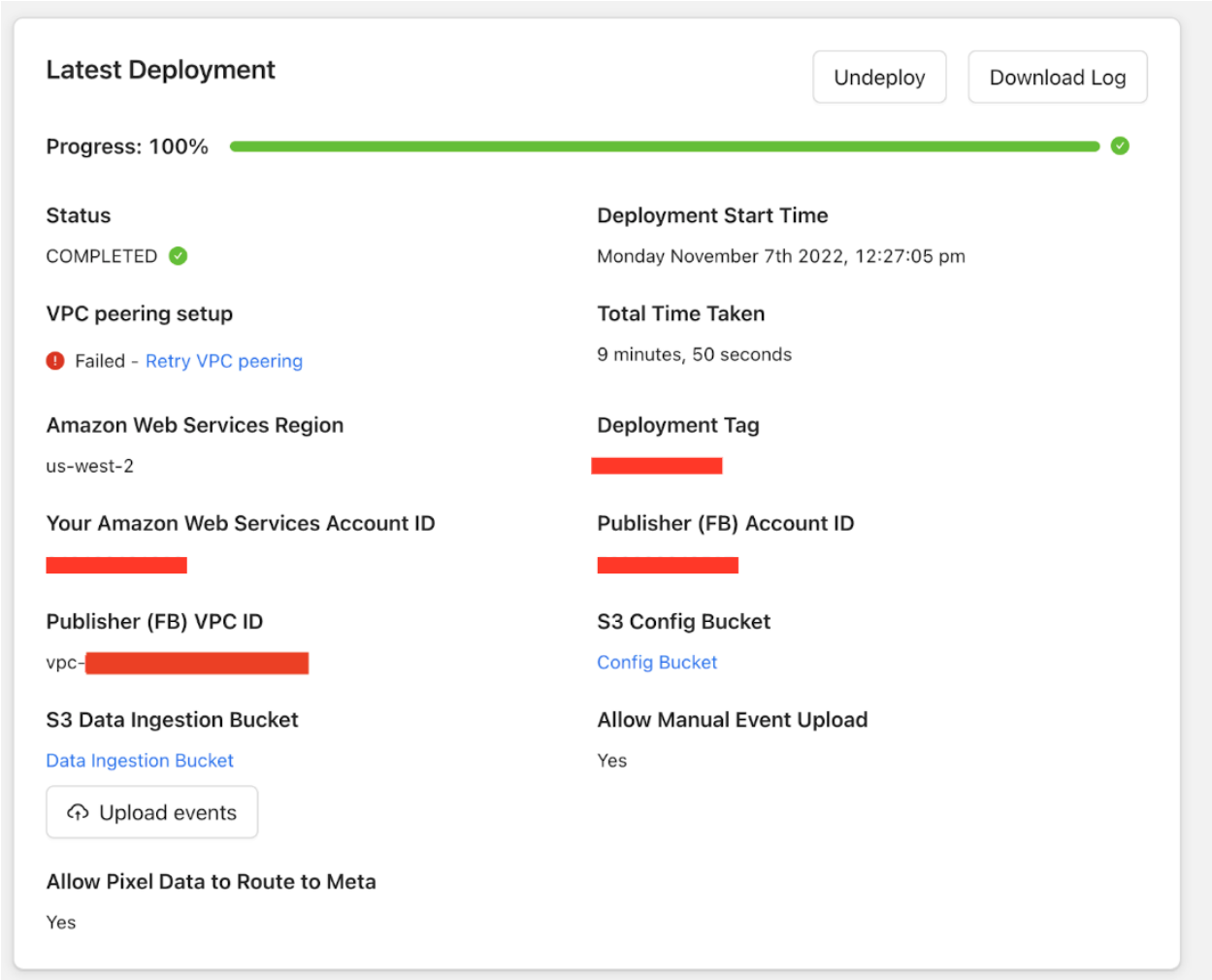
While deploying, if instructed by a meta representative ,you can use advance option in getting VPC details as shown below:



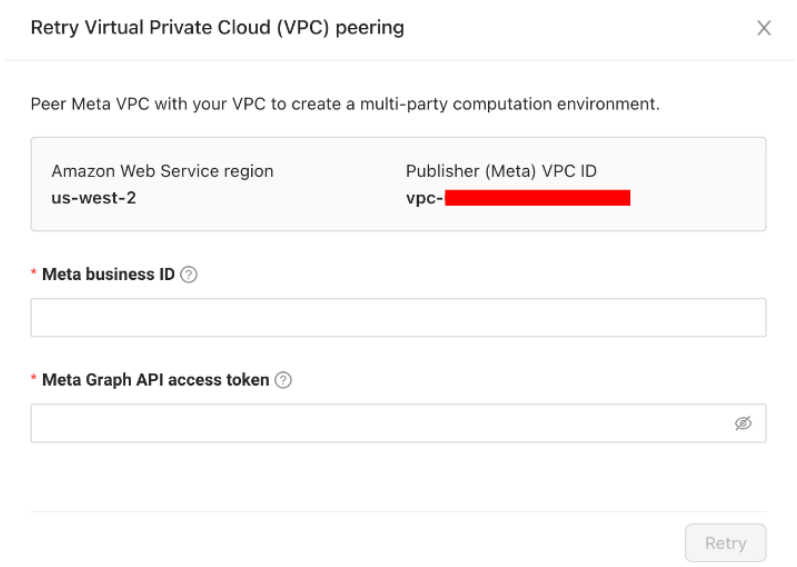
- Please fill in the required fields and click on Next:
  Amazon Web Services Region: This is the AWS region where the resources would be deployed. It should be the same as the region used for Conversions API Gateway deployment. (This region should also match the META side AWS region)
  Publisher (Meta) Account ID: Meta AWS Account number that is provided by META representative.
  Publisher (Meta) VPC ID: Meta VPC ID that is provided by a META representative.

A10: How to retry a failed VPC peering connection during deployment.

In case of a failed VPC peering connection during infrastructure deployment, you should see a screen like below.



- Please click on the “retry VPC peering” button. You should see a pop-up window like below:

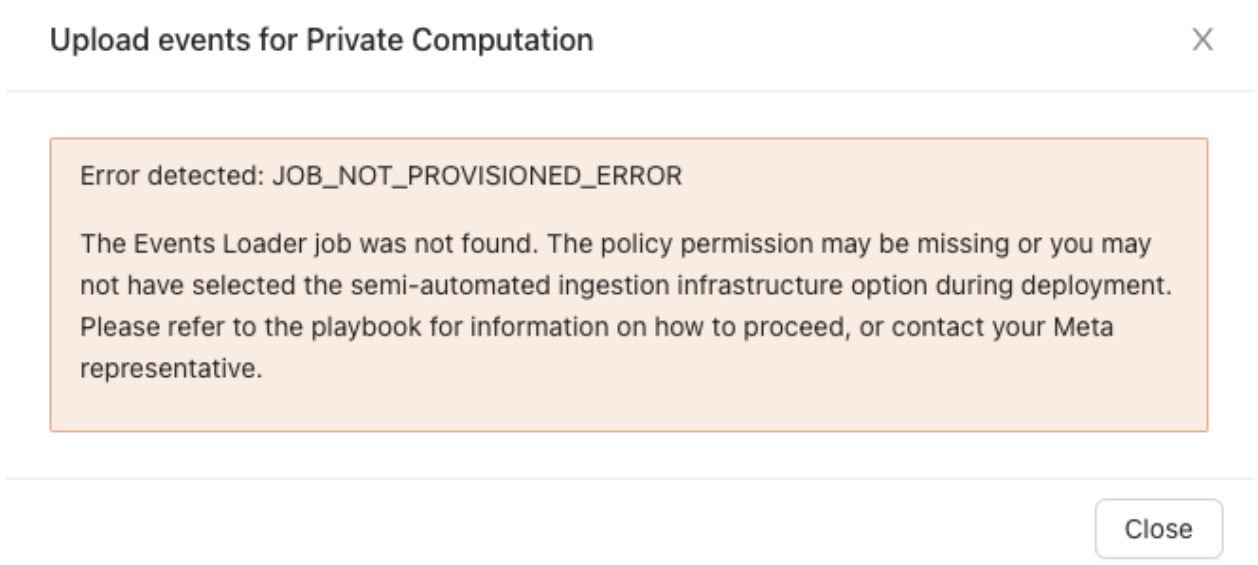


Please input your business ID and graph API token obtained from Step 2 which you would have obtained earlier, and click on retry.

If the VPC peering status still shows as failed, please contact META representative to further assist you.

A11: How to resolve the JOB_NOT_PROVISIONED_ERROR in the Events Uploader modal

If you see an error message that looks like this:



Then check the following:

1. Go to the IAM AWS services page
2. Click on ‘Policies’
3. Search for the deployed policy
    a. It should look like `fb-pc-policy-<deploy_tag>`
4. Click on ‘{} JSON’
5. Check if the policy has permission to access the ‘glue-ETL-<deploy_tag>’ resource.
    - Search for glue-ETL on that page
6. The allowed Resource should look like the following:
    - "arn:aws:glue:us-west-2:0123456789:job/glue-ETL-deploytag123"

If this glue-ETL resource permission is missing, then:

7. Click on “Edit policy” -> “JSON”
8. Add this JSON block next to the other Statements (first replace the <> sections with your own deployment values)

```
        {
            "Effect": "Allow",
            "Action": [
                "glue:Get*",
                "glue:BatchGet*",
                "glue:List*",
                "glue:QuerySchemaVersionMetadata",
                "glue:CheckSchemaVersionValidity",
                "glue:SearchTables"
            ],
            "Resource": [

"arn:aws:glue:<region>:<your_AWS_account_id>:job/glue-ETL-<deploy_tag>"
            ]
```

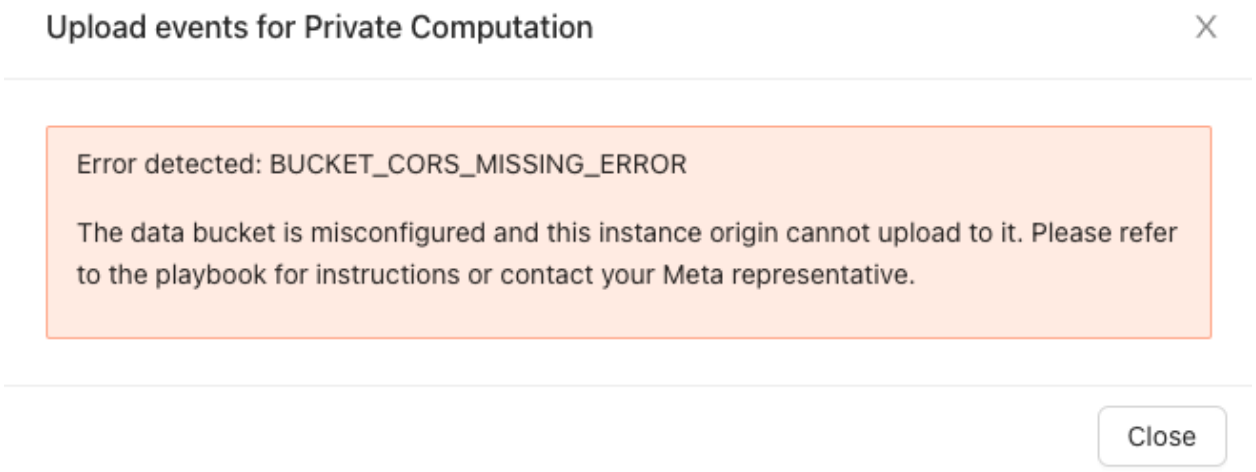
```
        }
```

- Example of Resource name: "arn:aws:glue:**us-west-2**:**0123456789**:job/glue-ETL-**mydeployment-123**"
- **Do not** update the the existing “glue:*” statement. Instead, add a new section with the above block.

9. Save the updated policy
10. Refresh the Deployment UI page
11. Open the Uploader modal and check if the problem has been resolved

A12: How to resolve the BUCKET_CORS_MISSING_ERROR in the Events Uploader modal

If you see this error in the Uploader modal:



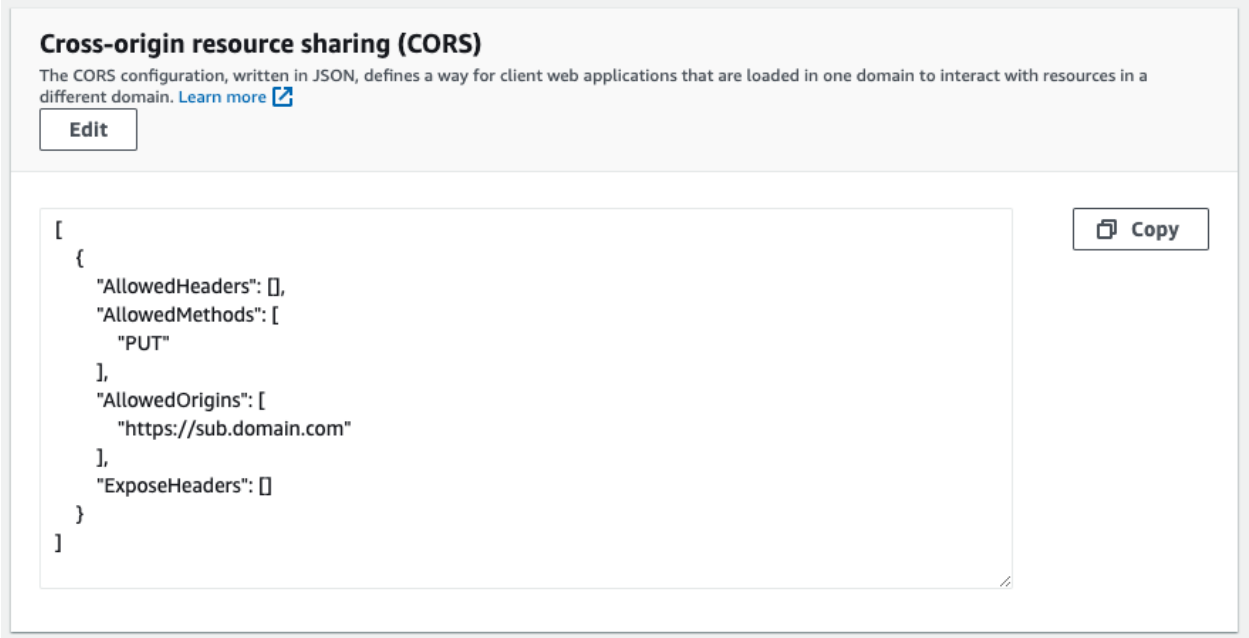
Then follow these instructions to resolve the error:

1. Open the Deployment UI at /hub/pcs/deployment
2. Click on the ‘Data Ingestion Bucket’ link
3. Click on the ‘Permissions’ tab
4. Scroll down to the ‘Cross-origin resource sharing (CORS)’ section
5. Click on ‘Edit’
6. Paste this block into the CORS config section
    a. Update the AllowedOrigins to be your EC2 instance’s domain name

```
[
    {
        "AllowedHeaders": [],
        "AllowedMethods": [
            "PUT"
        ],
```

```
        "AllowedOrigins": [
            "https://<sub.domain.com>"
        ],
        "ExposeHeaders": []
    }
]
```

7. Click on ‘Save changes’
    - It should look similar to this:



8. Refresh the Deployment UI page
9. Open the Uploader modal and check if the problem has been resolved

A13 capi-g upgrade guideline and questions.

Q: What if I have an old instance of CAPIG where I have deployed infrastructure already, should I still see a VPC peering status on the deployment UI?

A: No, you would not. The previous VPC peering connection status will just get carry forward. So if the previous VPC peering connection was in pending state, either you could contact META representative to accept the connection request from META manually, or un-deploy and redeploy the infrastructure to avail the latest auto VPC peering feature.

A14: Ensure that the logging permission exists

Check the following:

1. Go to the IAM AWS services page
2. Click on ‘Policies’
3. Search for the deployed policy
    a. It should look like `fb-pc-policy-<deploy_tag>`
    b. Click on ‘{} JSON’
4. Check if the policy has AWS CloudWatch permissions.
    - Search for logs:* on that page
5. If the logs:* permission is missing, continue to the next step
    a. Otherwise, if the below permission already exists on the policy, then no further setup is required
6. Click on “Edit policy” -> “JSON”
7. Add this JSON block next to the other Statements

```
{
    "Action": [
        "logs:*"
    ],
    "Effect": "Allow",
    "Resource": "*"
},
```

8. Save the updated policy