

Private Computation Solutions Partner Playbook

Table of Contents

Introduction

[How to use the playbook](#)

Requirements

[Private Computation Products overview](#)

Step 1: Private Computation Infrastructure Setup (1 Hour)

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

Step 3: Private Computation Environment Setup (30 Minutes)

[Verify infra completeness and connectedness](#)

[Data Ingestion](#)

Step 4: Private Computation Runs

Prerequisites

[Step 1: Check config before running computation \(5 mins\)](#)

[Step 2: \(optional\) automate diagnostic data sharing with Meta](#)

Private Lift

[Step 1: Run Private Lift Computation \(15 mins\)](#)

[Step 2: View Private Lift Results](#)

Private Attribution

[Step 0: Preparation](#)

[Step 1: Run Private Attribution Computation \(30 mins\)](#)

[Step 2: View Private Attribution Results](#)

Appendix

[A1: Semi-auto data ingestion/preparation](#)

[A2: How to set “canary” tier](#)

[A3: Configure an AWS IAM user with minimal permissions for future computation after initial infra deployment](#)

[A4: Data Migration](#)

[A5: Ad-hoc system diagnosis](#)

[A6: Sharing diagnostic data with Meta](#)

[Manual sharing with Meta](#)

Automatic sharing with Meta

- [A7: How to enable Multi-key for Private Lift](#)
- [A8: \[FYI\] New Requirements on Graph API Access Token Permissions are Enforced](#)
- [A9: Advanced setting on infrastructure deployment page, on modal stepper “get VPC details from meta”](#)
- [A10: How to retry a failed VPC peering connection during deployment.](#)
- [A13: Private Computation Infrastructure upgrade guideline and questions.](#)
- [A14: Ensure that the logging permission exists](#)

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 products. 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 Private Computation Infrastructure 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: Private Computation Infrastructure Setup (1 Hour)

Configure AWS and Install Private Computation Infrastructure

1. Login to AWS console.
2. Click [here](#) to go to the Cloudformation stack creation page.
3. Select the AWS region where you want to install Private Computation Infrastructure.
4. Fill the subdomain (example: pl.your-domain.com), admin email, password and proceed with stack creation.
5. Select at least **t2.xlarge** instance type.
6. The creation process takes approximately 5 minutes, after which two outputs will be written to the "Outputs" tab for the new AWS instance. You will know it is done when the Status says: CREATE_COMPLETE in the Stack info Tab.
7. Once complete, go to the "Outputs" tab.
8. Take note of the IP under CallToAction, as you'll need to add this as an A and AAAA name record for the subdomain you specified in step 4.
9. Take note of the ConversionsApiGatewayInstanceURL, which you'll need to provision the Private Computation Infrastructure UI.

Configure Your DNS

Set up your DNS by accessing your DNS provider.

1. Add the IP address generated under CallToAction in the previous step as an A name record for the subdomain you're using.
2. Add the subdomain you've defined in your Cloudformation step during the "**Configure AWS and Install**" process above.
3. Steps for this will vary depending on your partner, so please refer to their documentation if you have any problems completing this step.
4. Save your configuration.

Provisioning

Next, refer to the ConversionsApiGatewayInstanceURL, and click on the URL.

1. A new webpage appears titled Conversions API Gateway is provisioning. This URL indicates that Private Computation Infrastructure is being installed.
2. Setup takes approximately 30 minutes, and then the Private Computation Infrastructure UI will be accessible.
3. You will get the message "provisioning finished" once setup is done.

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

You will need this token to perform various tasks during Private Computation deployment and while using Private Lift and Private Attribution products.

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

The screenshot shows the 'Admin Apps' section of the Meta for Developers website. It lists several apps, each with a thumbnail icon, App ID, Type, and Business name. The apps listed are: markapitestapp (App ID: I, Type: Business), Test Business App (App ID: I, Type: Business), marketingapitestapp (App ID: I, Type: Business, Business: Ashish Kharbanda), Test API (App ID: I, Type: Business), Lead Ads App (App ID: I, Type: Business), Offline_conversions_app (App ID: I, Type: Business), Ashish Coffee Shop (App ID: I, Type: Business), and Test App (App ID: I, Type: Business). Each app entry includes a 'Data Use Checkup' status (e.g., 'Past due'), a 'Last checkup' date, and an 'Administrator' role. A search bar at the top right and a 'Create App' button are also visible.

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

The screenshot shows the 'Basic' settings page for the app 'markapitestapp'. The left sidebar shows navigation options like Dashboard, Settings (Basic selected), Roles, Alerts (with 9 notifications), App Review, Products, and Facebook Login. The main area displays basic app information: App ID (markapitestapp), App Mode (Development), App type (Business), and App secret (redacted). Other fields include Display name (markapitestapp), Namespace, App domains, Contact email, Privacy Policy URL, Terms of Service URL, App icon, and Category (Choose a category). A 'Show' button is present next to the App secret field.

- 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 <App Secret> 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**.

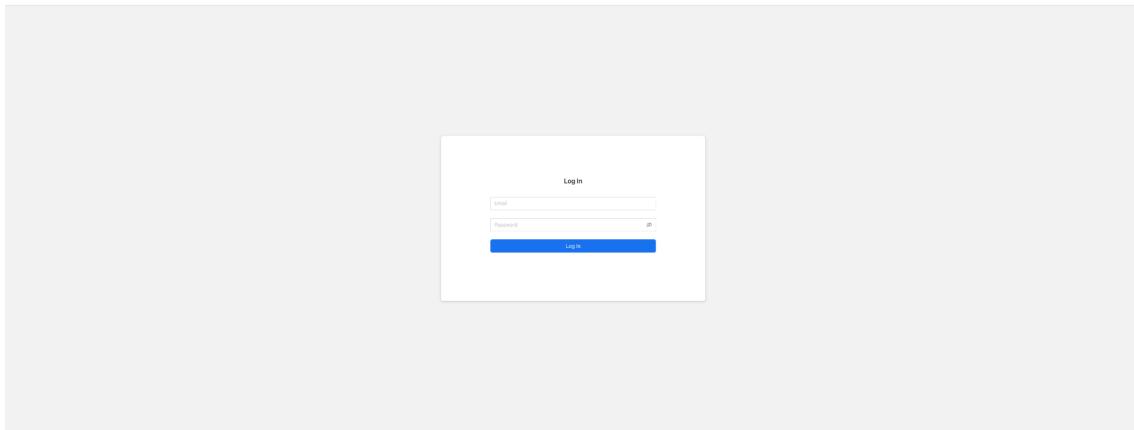
~~private_computation_access~~

Add a Permission

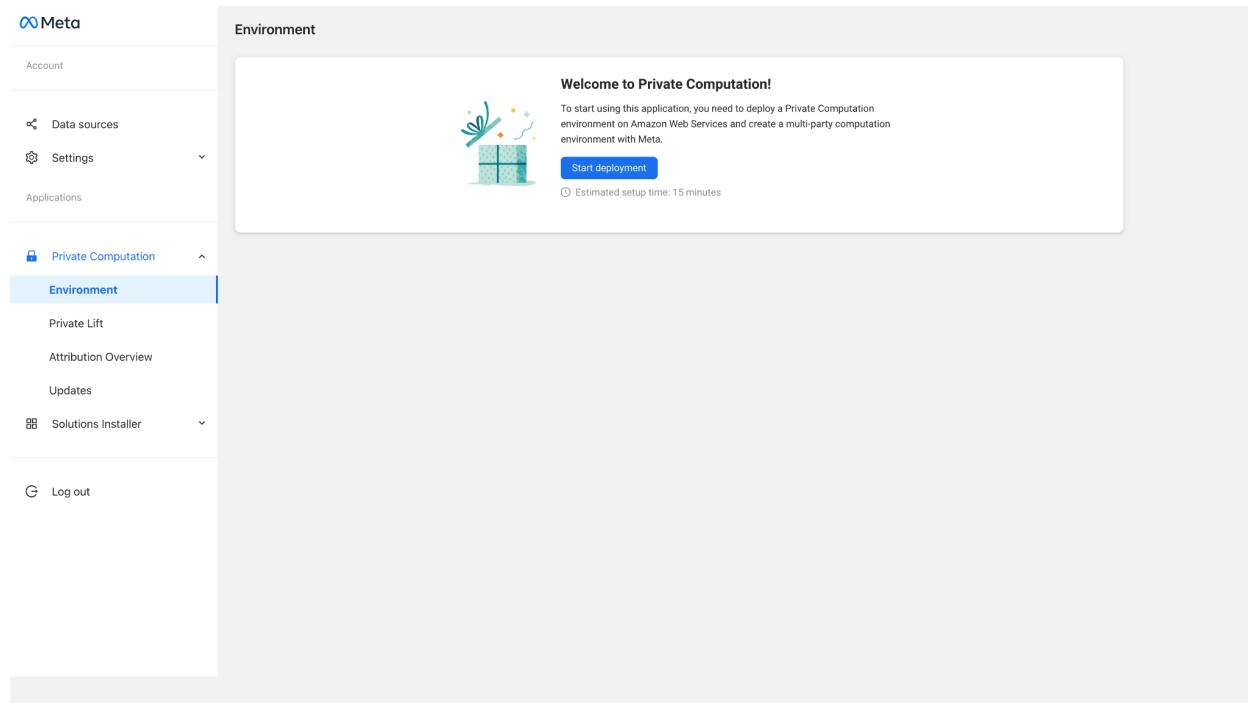
- Click on Submit Button and copy the access_token received in the response.
 - Please carefully store this token.

Step 3: Private Computation Environment Setup (30 Minutes)

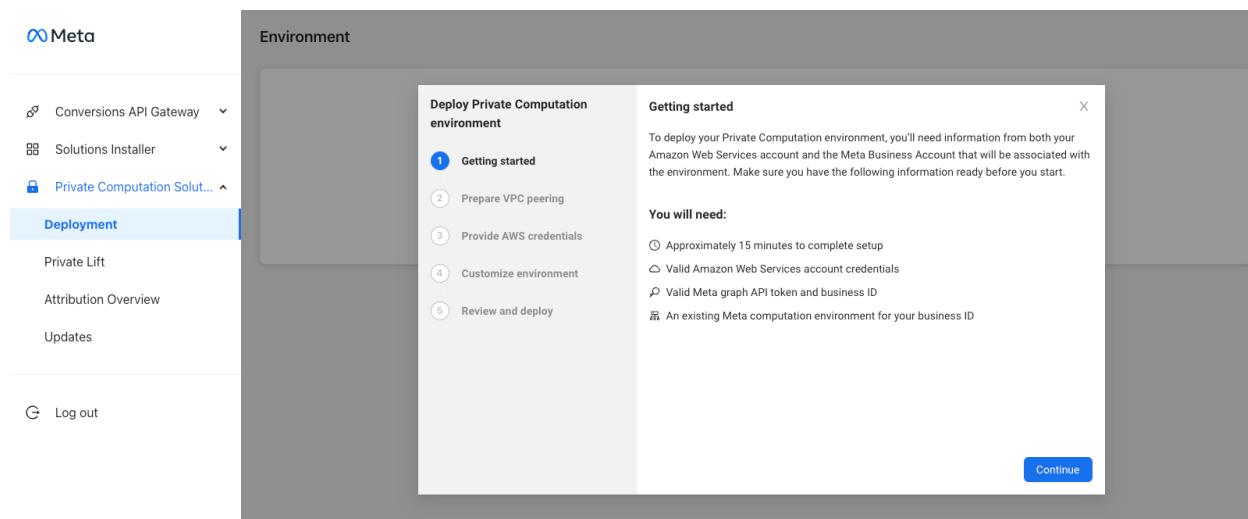
- Navigate to <https://<private-computation-infrastructure.instance.url>/hub/ui>. You should see the following window:



- 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:

Deploy Private Computation environment

1 Getting started

2 **Prepare VPC peering**

3 Provide AWS credentials

4 Customize environment

5 Review and deploy

Prepare Virtual Private Cloud (VPC) peering

To create a multi-party computation environment with Meta, we need to fetch VPC details from Meta and connect to your VPC in the same Amazon Web Services region.

Use advanced settings

* **Meta Business ID** ?

* **Meta Graph API access token** ?

Get Meta VPC details

X

Back **Continue**

- 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](#).

Deploy Private Computation environment

- ✓ Getting started
- 2 Prepare VPC peering
- 3 Provide AWS credentials
- 4 Customize environment
- 5 Review and deploy

Prepare Virtual Private Cloud (VPC) peering

To create a multi-party computation environment with Meta, we need to fetch VPC details from Meta and connect to your VPC in the same Amazon Web Services region.

Use advanced settings

* Meta Business ID ?
[REDACTED]

* Meta Graph API access token ?
[REDACTED]

Get Meta VPC details

AWS region: us-west-2
Meta VPC ID: vpc-[REDACTED]

Back
Continue

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

Deploy Private Computation environment

- ✓ Getting started
- ✓ Prepare VPC peering
- 3 Provide AWS credentials
- 4 Customize environment
- 5 Review and deploy

Provide AWS credentials

To deploy Private Computation solutions on your AWS, you need to provide your AWS access keys and account ID. You can find or create your access keys through your [AWS IAM console](#)

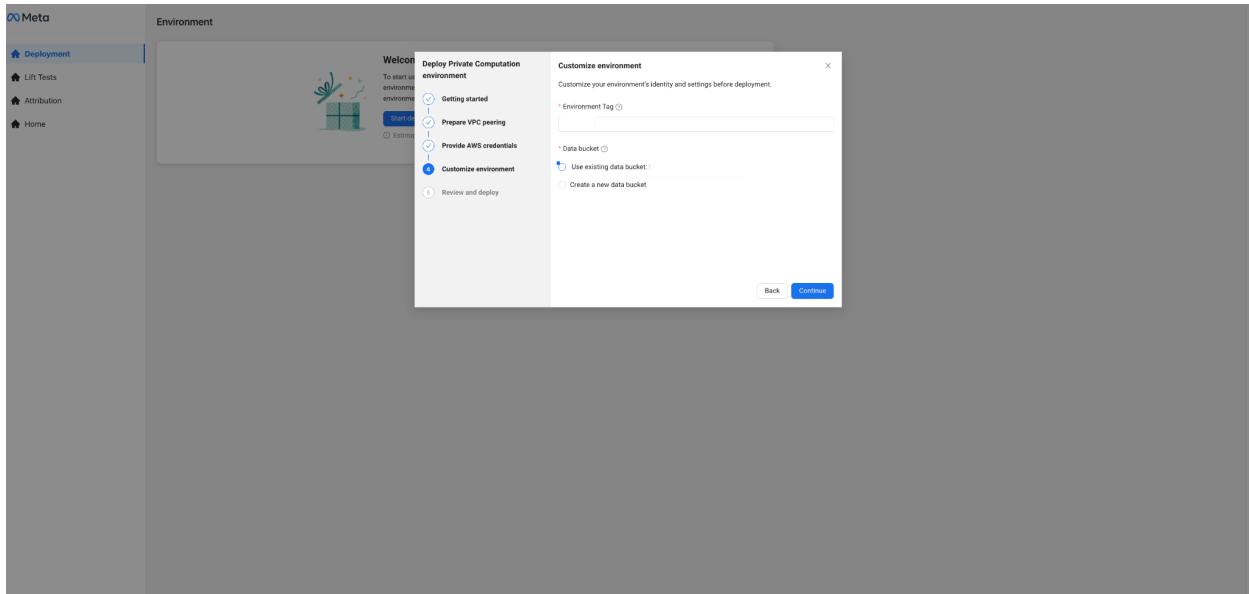
* Amazon Web Services Access Key ID ?
[REDACTED]

* Amazon Web Services Secret Access Key ?
[REDACTED]

* Amazon Web Services Account ID ?
[REDACTED]

Back
Continue

- 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 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:

Deploy Private Computation environment

Customize environment X

Customize your environment's identity and settings before deployment.

* Environment Tag ?

Settings

Consumption of Pixel data i

Manual event upload is enabled by default with semi-automated ingestion infrastructure

Back Continue

Getting started
Prepare VPC peering
Provide AWS credentials
4 Customize environment
Review and deploy

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.

Deploy Private Computation environment X

Getting started
Prepare VPC peering
Provide AWS credentials
4 Customize environment
Review and deploy

Customize environment
Customize your environment's identity and settings before deployment.

* Environment tag i
a-prefilled-tag-the-user-can-change

Settings
 Share diagnostic data with Meta i

[Give feedback](#) [Back](#) [Continue](#)

Share diagnostic data settings:

- To help clients better troubleshoot issues and improve the product, it's recommended to opt-in for diagnostic data sharing with Meta. When turned on, diagnostic data sharing will automatically upload logs to Meta within 5 minutes after a calculation run. The diagnostic data will contain: console logs from the study runner (i.e. run coordinator), console logs from the worker containers, logs from the PC data pipeline (Athena, Glue, Kinesis, Lambda). For more details on diagnostic data sharing see [A6: Sharing diagnostic data with Meta](#)
- 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.

The screenshot shows a step-by-step wizard titled "Deploy Private Computation environment". The current step is "Review and deploy", which is the final step before deployment. The left sidebar lists five steps: "Getting started" (checkmark), "Prepare VPC peering" (checkmark), "Provide AWS credentials" (checkmark), "Customize environment" (checkmark), and "Review and deploy" (number 5). The main area displays review details:

- Amazon Web Service region: us-west-2
- Publisher (Meta) VPC ID: vpc-[REDACTED]
- Your Amazon Web Services account ID: [REDACTED]
- Environment tag: [REDACTED]
- Data bucket:
Use existing data bucket: fb-pc-data-[REDACTED]

At the bottom right are "Back" and "Deploy" buttons.

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

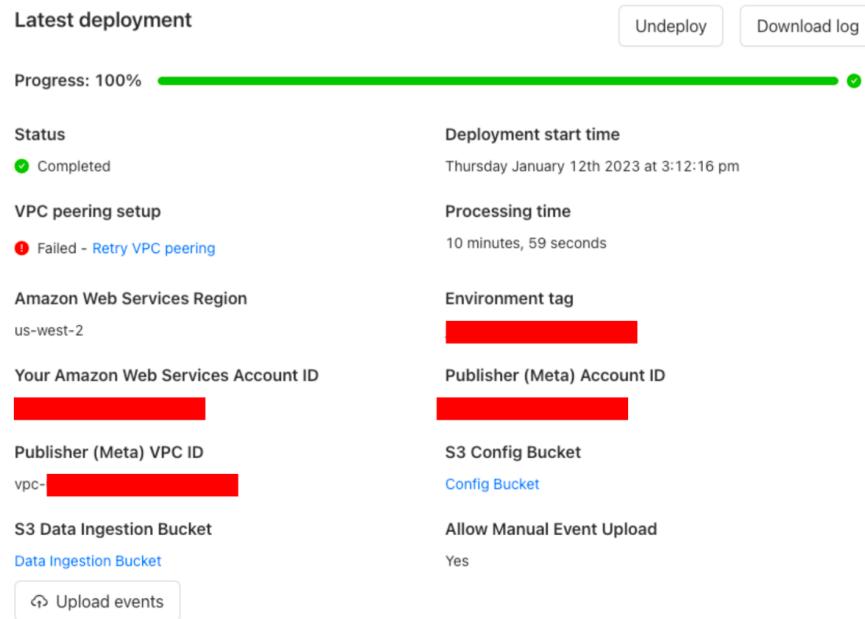
The screenshot shows the Meta Private Computation Environment dashboard. On the left is a sidebar with navigation links: Account, Data sources, Applications, Private Computation (selected), Environment (selected), Private Lift, Private Attribution, Updates, Solutions Installer, and Log out.

The main content area is titled "Environment". It displays the "Latest deploy" status, which is at 100% progress (Completed). Buttons for "Undeploy" and "Download log" are available. Below this, various environment settings are listed with their current status:

- Status: Completed
- VPC peering setup: Completed
- Amazon Web Services Region: us-west-2
- Your Amazon Web Services Account ID: [REDACTED]
- Publisher (Meta) Account ID: [REDACTED]
- Publisher (Meta) VPC ID: [REDACTED]
- S3 Config Bucket: Config Bucket
- S3 Data Ingestion Bucket: Data Ingestion Bucket
 - Upload events button
- Allow Manual Event Upload: No
- Automatic diagnostic data sharing: Disabled, Edit, ⓘ

Below this section is a "System diagnostics" panel with a "Run system diagnostics" button. At the bottom is a "Live Log Streams" panel.

- 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.



- Automatic diagnostic data sharing status:
 - To help clients better troubleshoot issues and improve the product, it's recommended to opt-in for diagnostic data sharing with Meta. ~~it will automatically~~ Then the diagnostic data will be automatically uploaded to Meta within 5 minutes after a failed run. The diagnostic data will contain: console logs from the study runner (i.e. run coordinator), console logs from the worker containers, logs from the PC data pipeline (Athena, Glue, Kinesis, Lambda). For more details on diagnostic data sharing see in [A6: Sharing diagnostic data with Meta](#)
 - You can click the Edit button to bring up the dialog "Share diagnostic data with Meta", and enable or disable the automatic diagnostic data sharing. Your select will apply to the future calculation runs.

Environment

Latest deploy

Progress: 100%  ✓

Status Completed **Deploy start time** 

VPC peering setup Completed **Processing time** 

Amazon Web Services Region us-west-2 **Environment tag** 

Your Amazon Web Services Account ID 

Publisher (Meta) Account ID 

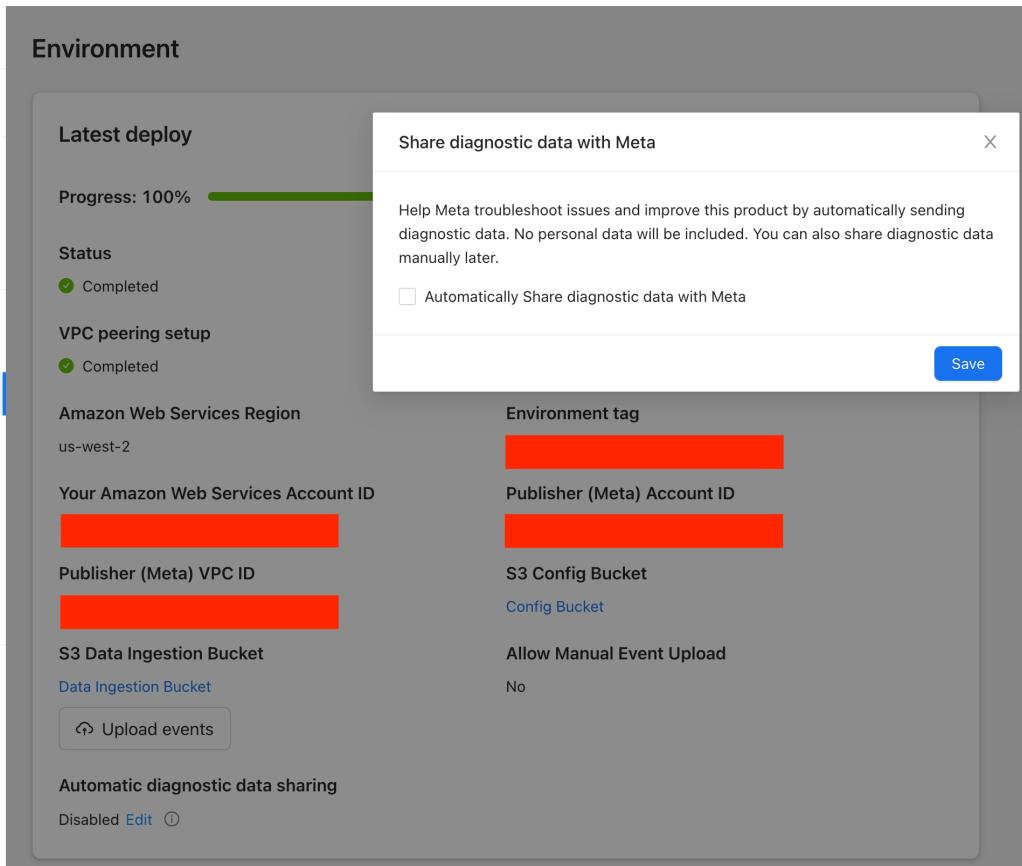
Publisher (Meta) VPC ID 

S3 Config Bucket [Config Bucket](#)

S3 Data Ingestion Bucket [Data Ingestion Bucket](#) **Allow Manual Event Upload** No

[Upload events](#)

Automatic diagnostic data sharing
Disabled [Edit](#) ⓘ



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 Elastic Container Registry (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

Create Data Sources

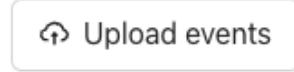
1. Login to the Private Computation Infrastructure UI by following the subdomain defined in your Cloudformation step during the “**Configure AWS and Install**” process above. Use the email and password used while setting up the Cloudformation stack.
2. Go to the “Data sources” page and create a new data source.
3. Data sources are the ways you ingest conversion/signal data and use it for Private Computation. Conversion signals can be sent to data sources through Meta pixel optionally.
4. Conversion signals can also be uploaded to data sources manually.

There are two different ways to ingest your data.

1. Automated data ingestion and computation with pixel:
 - a. Connect to Meta pixel while creating a new data source. This will facilitate automatic data ingestion.
 - 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). Generally, this process 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://<private-computation-infrastructure.instance.url>/hub/pcs/deployment>
 - ii. Click on the 'Upload events' button under the 'S3 Data Ingestion Bucket' section

S3 Data Ingestion Bucket

Data Ingestion Bucket

 [Upload events](#)

- 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)

Upload events for Private Computation

X

Upload CSV files to your data storage: **fb-pc-data-[REDACTED]**. Uploaded events will be processed and available for Private Computation use in about 2 hours.

Event data formatting must correspond the flattened csv of [Meta's Server Event Parameter](#). You can use this [sample file](#) as a template to format your data correctly. Data that is formatted incorrectly will be removed automatically.



Drag file here or click to upload

File size limit of 5GB and estimated upload time is 10-15 minutes.

[Close](#)

- 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. Troubleshooting
 - i. If the file was uploaded but the expected events are missing during a computation run, double check if the semi-automated Glue job is [up to date](#)
 - 1. This issue could be resolved by removing spaces and special characters from your file name. Or, you can redeploy the Private Computation infra if the issue persists after changing the file name.
 - d. 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

Prerequisites

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

1. Open Private Computation Infrastructure Shell:
<https://<private-computation-infrastructure.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": "events_data",  
    "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  
}
```

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 sharing diagnostic data with Meta, for automatically uploading logs to Meta within 5 minutes after a failed run.

You can open the Environment tab and find the setting “Automatic diagnostic data sharing”, then click the Edit button to update the setting.

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.

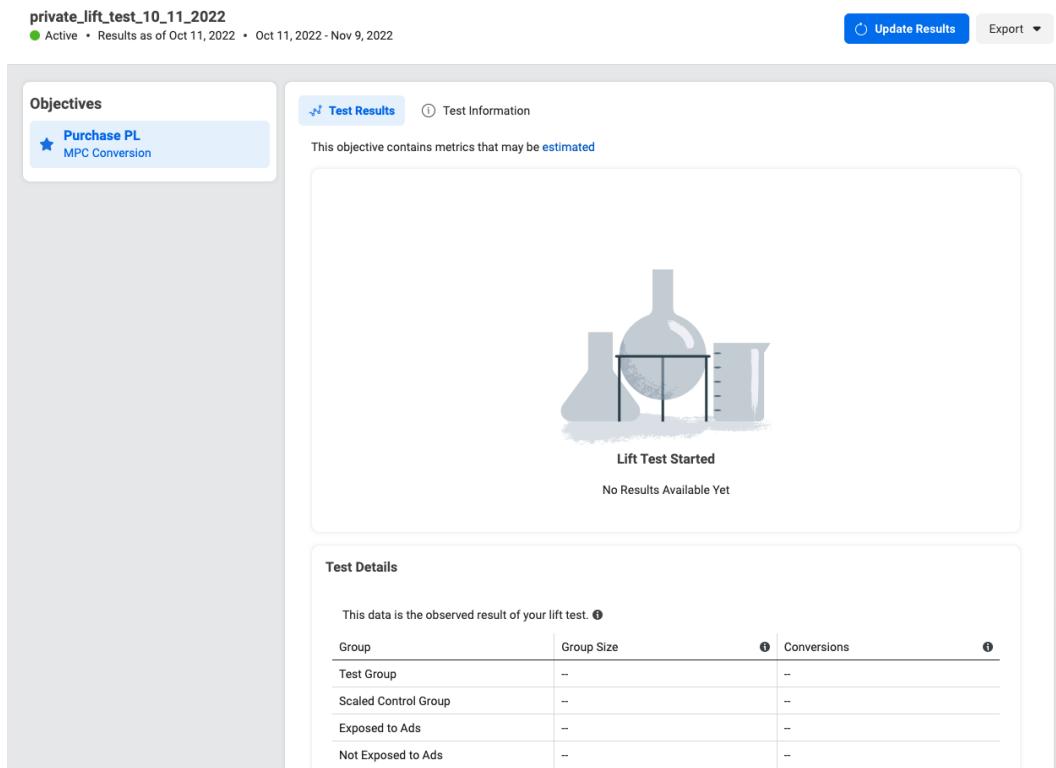
More details can be found in [A6: Sharing diagnostic data with Meta](#).

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>](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



private_lift_test_10_11_2022

Active • Results as of Oct 11, 2022 • Oct 11, 2022 - Nov 9, 2022

Update Results Export

Objectives

Purchase PL
MPC Conversion

Test Results Test Information

This objective contains metrics that may be estimated

Lift Test Started

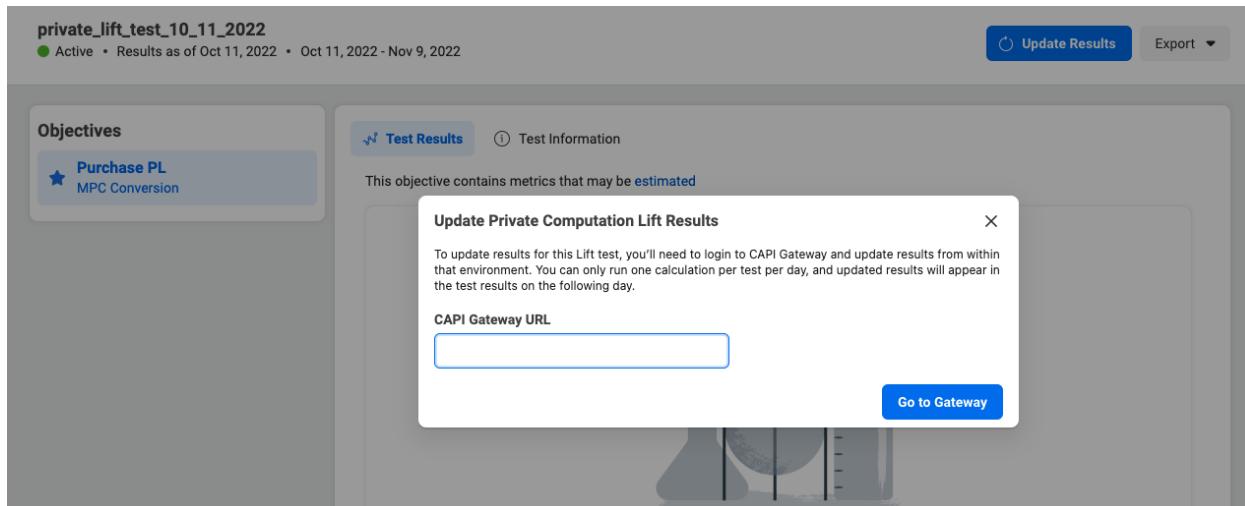
No Results Available Yet

Test Details

This data is the observed result of your lift test.

Group	Group Size	Conversions
Test Group	--	--
Scaled Control Group	--	--
Exposed to Ads	--	--
Not Exposed to Ads	--	--

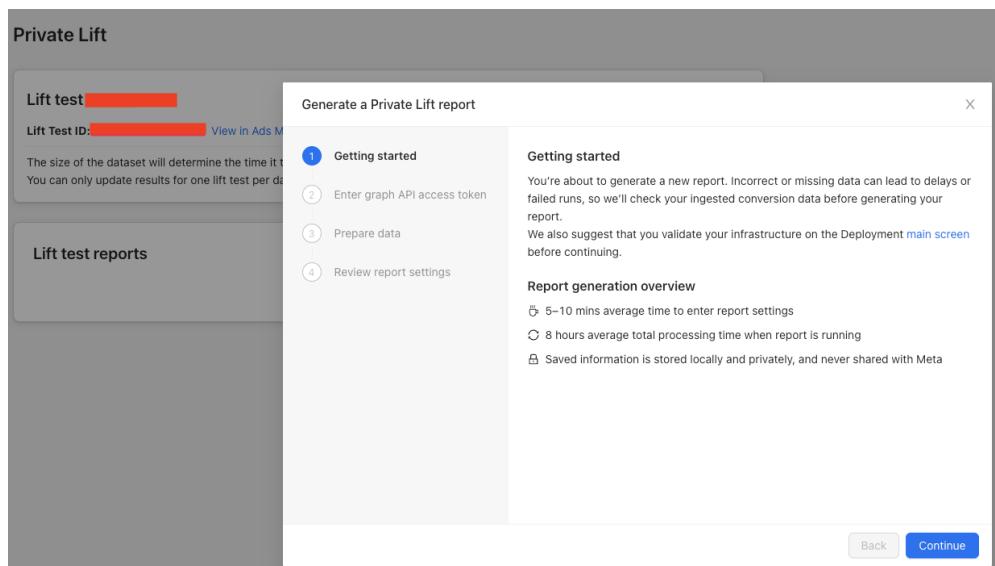
- Click on “Update Results”. A new window will pop-up, enter your Private Computation Infrastructure instance url here and click on “Go to Gateway”.



- You are now re-directed back to Private Computation Infrastructure. To start the computation, click on “Update Results”.
 - Format:
`https://<private-computation-infrastructure.instance.url>/hub/pcs/calculation/<your_ad_study_id>/<your_ad_study_name>`
- Enter the following URL and click on “Go to Gateway”.

The screenshot shows the Meta dashboard with a sidebar on the left. Under 'Private Computation', the 'Private Lift' section is selected. It displays a 'Lift test' entry with a redacted ID, a 'Generate report' button, and a note about generating a lift report. Below this, under 'Lift test reports', it says 'You have no active reporting runs.'

- 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”.

Generate a Private Lift report X

Review report settings

You've completed all necessary steps and you are now ready to generate your report.

Lift test name
Lift Test ID: [REDACTED]
This lift test contains 1 MPC objective

- Graph API access token validated
Validated at 13:45:46 GMT-0800 (Pacific Standard Time)
- Data prepared
 - Objective ID: [REDACTED] data prepared.
Completed at 13:46:01 GMT-0800 (Pacific Standard Time).
Data source ID: [REDACTED]
S3 path: https://fb-pc-data-[REDACTED]s3.us-west-[REDACTED].csv

Back Generate report

- 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 Lift

Lift test [REDACTED]

Lift Test ID [REDACTED] in Ads Manager Experiments

The size of the dataset will determine the time it takes to generate a lift report and associated server costs.
You can only update results for one lift test per day.

Generate report

Lift test reports

You have no active reporting runs.

Report History

Run started on Nov 7, 2022 6:25 PM
Run ended on Nov 7, 2022 7:19 PM
Run time: 0 hours 54 minutes

View results

✓ Computation succeeded.
The result will be available within 48 hours after the computation is finished. Detailed computation logs are archived and uploaded to S3 path: s3://fb-pc-data-[REDACTED]logging/.

Private Attribution

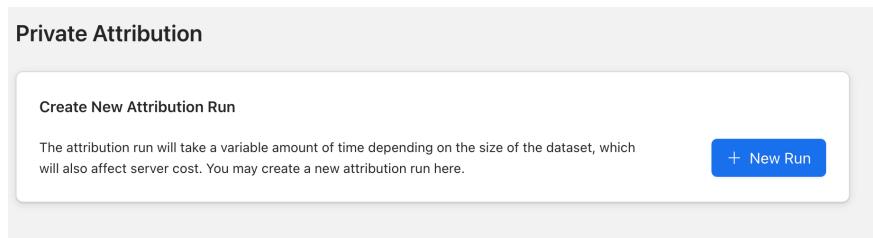
Step 0: Preparation

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 “Private Computation” → “Private Attribution” 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.

A screenshot of a detailed "Create a new attribution run" dialog. It includes fields for "Graph API Access Token" (with placeholder "Please enter Graph API Access Token"), "Dataset ID" (with placeholder "Please enter Dataset Id"), and "Select available date" (with a dropdown menu). Below these are fields for "Data source ID" (placeholder "Please enter data source Id") and "Event type" (placeholder "Please enter event type", with a note "Event type is case sensitive e.g. AddPaymentInfo"). A "Start data preparation" button is located below these fields. At the bottom, there is an "Advanced settings" section, a "Reset" button, and a "Submit" button.

- 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” button.

Private Attribution

Create New Attribution Run

The attribution run will take a variable amount of time to complete. The time it takes will also affect server cost. You may create multiple runs simultaneously.

Create a new attribution run

* Graph API Access Token
[REDACTED]

* Dataset ID
[REDACTED] ✓

Load Available Date

* Select available date
[REDACTED]

* Data source ID
[REDACTED]

* Event type ⓘ
[REDACTED]

Event type is case sensitive e.g. AddPaymentInfo

Start data preparation

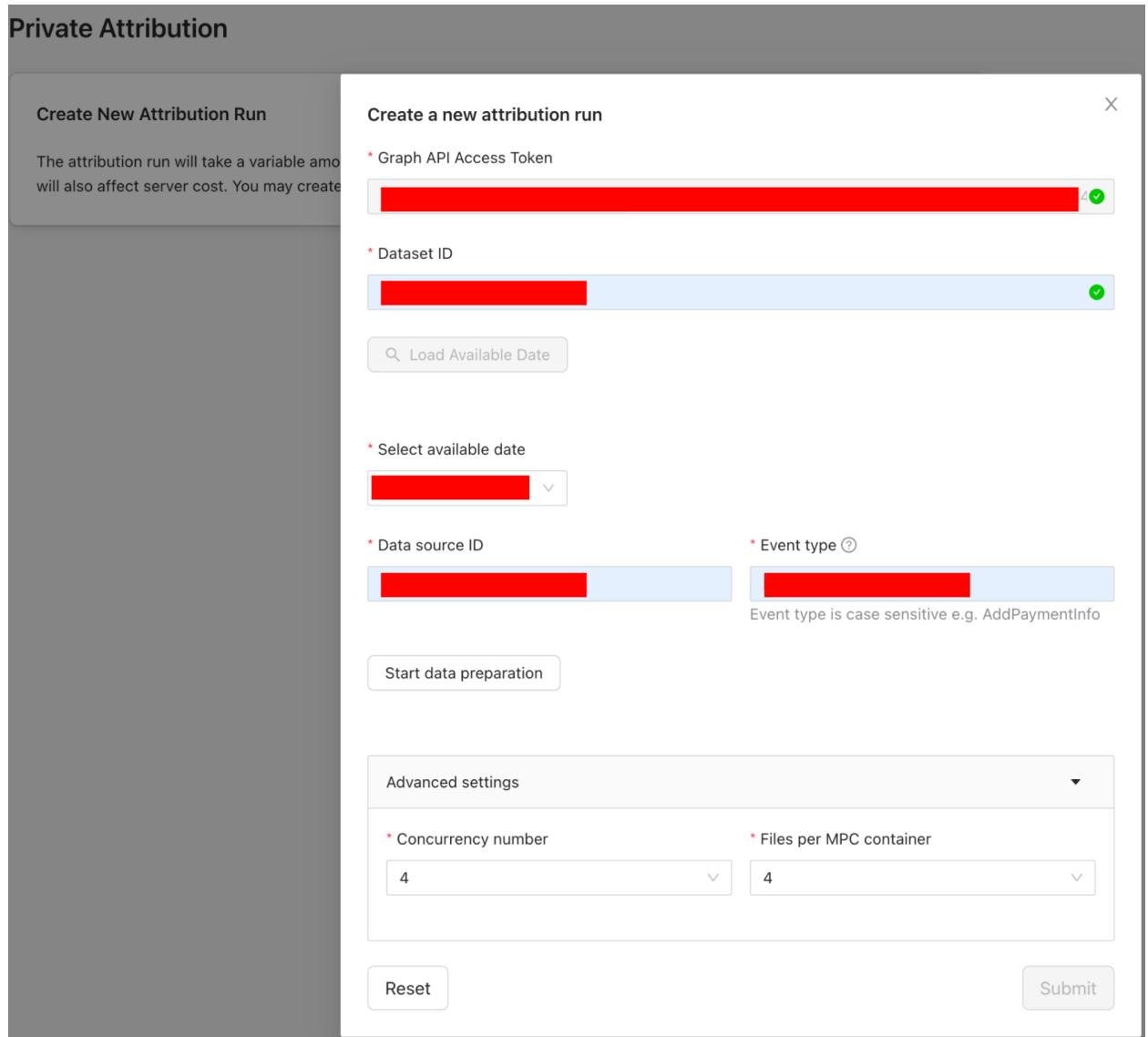
Advanced settings

* Concurrency number
4

* Files per MPC container
4

Reset

Submit



- Wait until it shows “Succeeded”.

The screenshot shows a user interface for data preparation. At the top left is a button labeled "Start data preparation". Below it is a section titled "Data Preparation Status:" with a blue information icon. A table follows, with the first row containing columns for "Created at" (redacted), "Updated at" (redacted), and "Status". The "Status" column contains a green dot followed by the text "Succeeded" and has a red arrow pointing to it. The second row contains columns for "S3 path" (redacted) and "Errors", with the text "No errors detected" in the "Errors" column.

Created at	Updated at	Status
██████████	██████████	• Succeeded
S3 path	Errors	
██████████	No errors detected	

- Click the “Submit” button at the bottom to proceed.
- 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

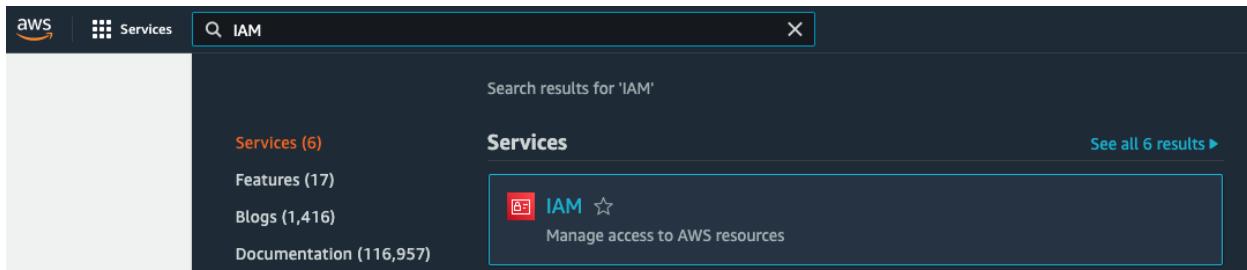
Sometimes Meta would like you to run on the “canary” tier. Here is how you can set it up.

1. Open Private Computation Infrastructure Shell: <https://<capig.instance.url>/hub/shell>
2. Run the following update commands:

a. config write CloudResources /IMAGE_TAG canary

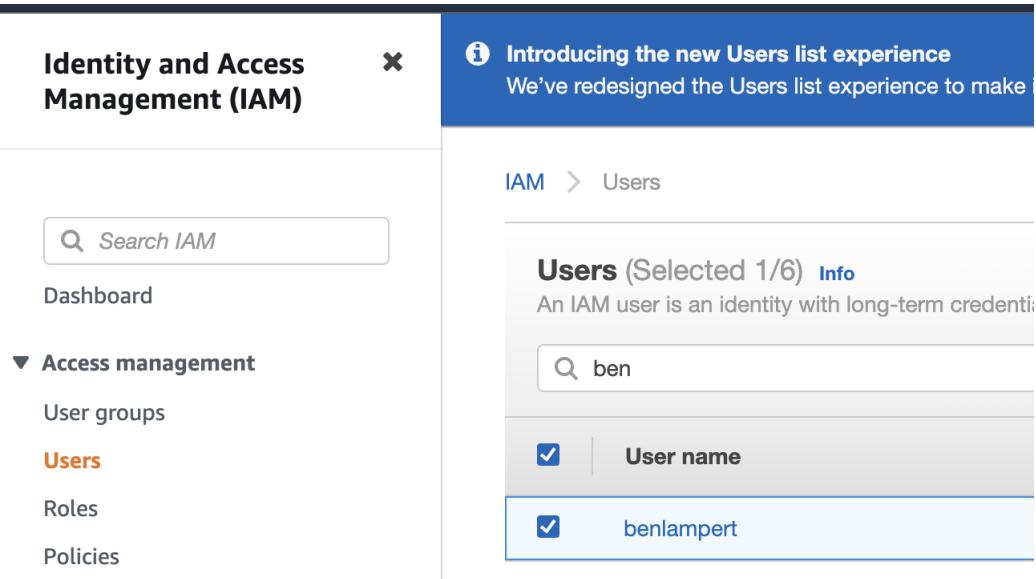
A3: Configure an AWS IAM user with minimal permissions for future computation after initial infra deployment

a. Open your AWS account and enter IAM component



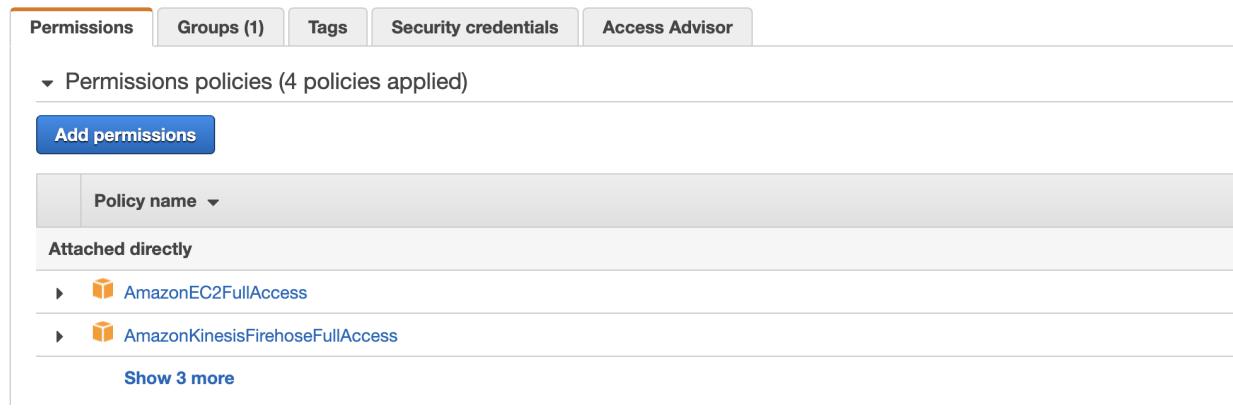
The screenshot shows the AWS search interface with 'IAM' typed into the search bar. Below the search bar, there's a sidebar with links to 'Services (6)', 'Features (17)', 'Blogs (1,416)', and 'Documentation (116,957)'. The main search results area shows a single result for 'IAM' with a star icon and the description 'Manage access to AWS resources'. A blue banner at the top right says 'See all 6 results ▶'.

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



The screenshot shows the 'Identity and Access Management (IAM)' dashboard. On the left, there's a sidebar with 'Access management' expanded, showing 'User groups', 'Users' (which is selected), 'Roles', and 'Policies'. The main area is titled 'Users (Selected 1/6) Info' and shows a search bar with 'ben'. Below it, there are two entries: 'User name' with 'ben' checked and 'User name' with 'benlampert' checked. The URL in the browser is 'IAM > Users'.

c. Enter into user page and click "add permissions"



The screenshot shows the 'Permissions' tab for a user. It has tabs for 'Permissions', 'Groups (1)', 'Tags', 'Security credentials', and 'Access Advisor'. Under 'Permissions', it says 'Permissions policies (4 policies applied)'. There's a large blue button labeled 'Add permissions'. Below it, there's a table with a header 'Policy name ▾' and a section 'Attached directly' containing two items: 'AmazonEC2FullAccess' and 'AmazonKinesisFirehoseFullAccess'. There's also a link 'Show 3 more'.

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

Add permissions to benlampert

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy

Filter policies Showing 12 results

Policy name	Type	Used as
fb-pc-policy-b1e2e-3baz	Customer managed	None

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

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:

Latest deploy

Progress: 100% ✓

Status	Deploy start time
Completed	[Redacted]
VPC peering setup	Processing time
Completed	[Redacted]
Amazon Web Services Region	Environment tag
us-west-2	[Redacted]
Your Amazon Web Services Account ID	Publisher (Meta) Account ID
[Redacted]	[Redacted]
Publisher (Meta) VPC ID	S3 Config Bucket
[Redacted]	Config Bucket
S3 Data Ingestion Bucket	Allow Manual Event Upload
Data Ingestion Bucket	No
<input type="button" value="Upload events"/>	
Automatic diagnostic data sharing Disabled Edit ⓘ	

System diagnostics

Environment diagnostics

Environment diagnostics can help you detect and troubleshoot issues with your Private Computation environment.

[Run system diagnostics](#)

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

System diagnostics

System diagnostics can help you detect and troubleshoot issues with Private Computation deployment.

[Run system diagnostics](#)

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](#).

In Meta, the diagnostics data will be kept for no longer than 30 days, and will be access controlled.

Limitations for automatic collection of diagnostic data:

- Currently only for Private Lift on top of the Private Computation Infrastructure 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_1
2/. 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 run launch, logs collection and uploading.

Usually sharing the first part of logs data is sufficient for Meta support engineers to investigate any issue related to a study run. In the computation UI, if you see a clickable link "Share Diagnostic Data with Meta" under a calculation run status, you can click it to trigger the logs upload. Then the logs upload should be finished within 5 minutes.

In case the second part of logs data is required for investigation, the advertiser engineer has to manually download them from S3 buckets, and share with Meta support engineers.

Automatic sharing with Meta

You can opt in for automatic logs upload in the following stages:

1) For new deployment of CAPIG: there is "Share diagnostic data settings" setting in the "Customize environment" step. You can make your choice on whether to automatically share the diagnostic data or not.

2) For upgrade and new deployment of CAPIG: you will see the one-time popup dialog when refreshing any page (except the “Updates” page) within the Private Computation Solution app. You can make your choice on whether to automatically share the diagnostic data or not, and save the setting.

3) After deployment of CAPIG: you can open the Environment tab within the Private Computation Solution app, and find the setting “Automatic diagnostic data sharing”. You can click the Edit button to update your choice, and save the setting.

By default, logs are not automatically uploaded after a successful run. You can manually click the “Share Diagnostic Data with Meta” link under a successful run to upload the corresponding logs. Or you can contact Meta support engineers to change the CAPIG config to automatically upload the logs after every successful run.

Logs upload status per computation run and troubleshooting:

- In normal cases, the logs upload status shows one of the following: a) “Diagnostic Data was shared with Meta at <timestamp with timezone>” (non-clickable). This means success of logs upload. b) “Share Diagnostics Data with Meta” (clickable), when the automatic sharing is not enabled, or the computation run succeeded.
- The status is “Error Sharing Diagnostic Data with Meta” with clickable “Retry”: the logs upload had failed after multiple internal attempts. You can click the Retry button and then confirm in the dialog, to try again. When investigation is needed, you can contact Meta support engineers for help if logs upload keeps failing after multiple retries.
- The status is missing: the computation run failed in early preparation stages of the calculation run, and no logs have been collected automatically. When investigation is needed, Meta support engineers might still ask you to manually retrieve logs from specific S3 buckets or CloudWatch locations, and then share with Meta.

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 Private Computation Infrastructure Shell:
<https://<private-computation-infrastructure.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

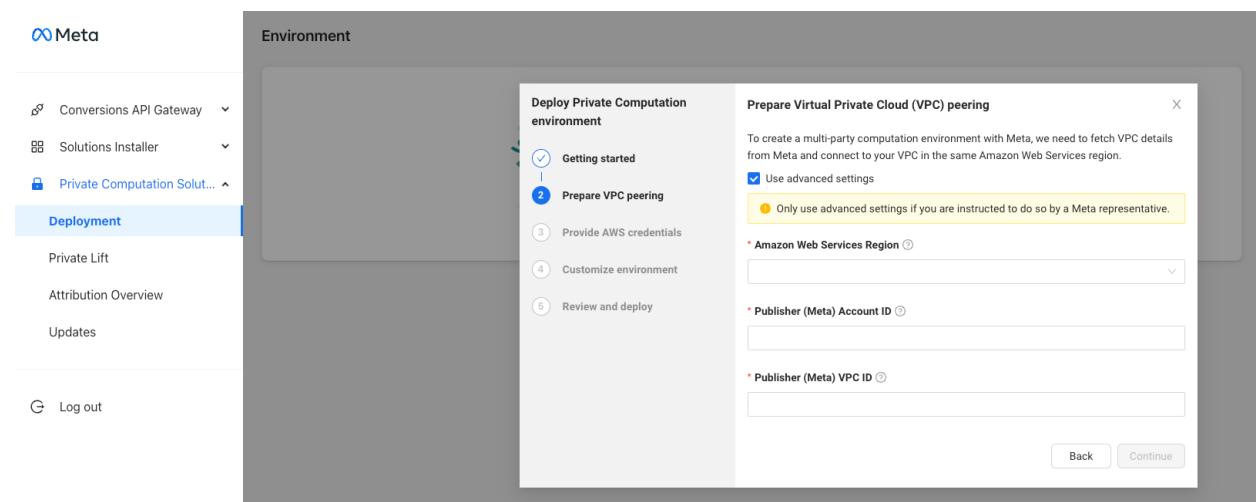
A8: [FYI] New Requirements on Graph API Access Token Permissions are Enforced

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.

Latest Deployment

[Undeploy](#)[Download Log](#)

Progress: 100% ✓

Status	Deployment Start Time
COMPLETED ✓	Monday November 7th 2022, 12:27:05 pm
VPC peering setup	Total Time Taken
⚠ Failed - Retry VPC peering	9 minutes, 50 seconds
Amazon Web Services Region	Deployment Tag
us-west-2	
Your Amazon Web Services Account ID	Publisher (FB) Account ID
	
Publisher (FB) VPC ID	S3 Config Bucket
vpc- 	Config Bucket
S3 Data Ingestion Bucket	Allow Manual Event Upload
Data Ingestion Bucket	Yes
Upload events	
Allow Pixel Data to Route to Meta	
Yes	

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

Retry Virtual Private Cloud (VPC) peering X

Peer Meta VPC with your VPC to create a multi-party computation environment.

Amazon Web Service region us-west-2	Publisher (Meta) VPC ID vpc-[REDACTED]
--	---

* Meta business ID ?

* Meta Graph API access token ?
 (Copy)

Retry

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:

Upload events for Private Computation

X

Error detected: JOB_NOT_PROVISIONED_ERROR

The Events Loader job was not found. The policy permission may be missing or you may not have selected the semi-automated ingestion infrastructure option during deployment. Please refer to the playbook for information on how to proceed, or contact your Meta representative.

Close

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>"  
    ]  
}
```

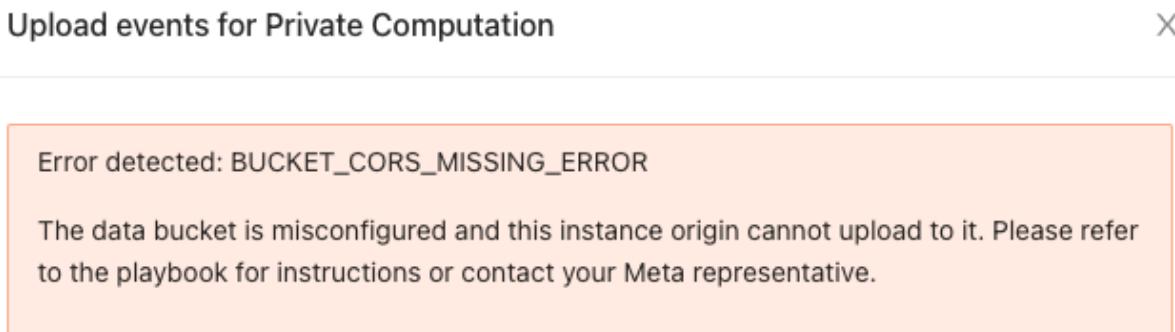
}

- Example of Resource name:
"arn:aws:glue:us-west-2:0123456789:job/glue-ETL-mydeployment-123"
- **Do not** update the 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:

The screenshot shows a user interface for managing Cross-Origin Resource Sharing (CORS) settings. At the top, there's a title 'Cross-origin resource sharing (CORS)' and a note explaining what CORS does. Below that is a large text input area containing JSON code. To the right of the input area is a 'Copy' button with a clipboard icon. An 'Edit' button is located at the top left of the input area.

```
[{
  "AllowedHeaders": [],
  "AllowedMethods": [
    "PUT"
  ],
  "AllowedOrigins": [
    "https://sub.domain.com"
  ],
  "ExposeHeaders": []
}]
```

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

A13: Private Computation Infrastructure upgrade guideline and questions.

Q: What if I have an old instance of Private Computation Infrastructure 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.

Q: After upgrading, I see a popup dialog box showing “Share diagnostic data with Meta”. What should I do?

A: 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 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. More details can be found in [A6: Sharing diagnostic data with Meta](#)

We recommend enabling the checkbox, and then saving. You can also change the setting later in the Environment tab.

If you click the 'X' button to close the dialog without saving your choice, the popup dialog will come up again when you refresh any pages in the Private Computation Solution app.

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