User Guide

AWS Clean Rooms

AWS Clean Rooms: User Guide

Table of Contents

What is AWS Clean Rooms?

AWS Clean Rooms helps you and your partners analyze and collaborate on your collective datasets to gain new insights without revealing underlying data to one another. You can use AWS Clean Rooms, a secure collaboration workspace, to create your own clean rooms in minutes, and start analyzing your collective datasets with just a few steps. You can choose the partners with whom you want to collaborate, select their datasets, and configure restrictions for participants.

With AWS Clean Rooms, you can collaborate with thousands of companies already using AWS. Collaboration doesn't require moving data out of AWS or loading it into another platform. When you run queries, AWS Clean Rooms reads data from its original location and applies built-in analysis rules to help you maintain control over their data.

AWS Clean Rooms provides built-in data access controls and audit support controls that you can configure. These controls include:

- Analysis rules to restrict SQL queries and provide output constraints
- Cryptographic Computing for Clean Rooms to keep data encrypted, even as queries are processed, to comply with stringent data handling policies
- Query logs to review queries and help support audits
- Differential privacy to protect against user-identification attempts. AWS Clean Rooms Differential Privacy is a fully-managed capability that protects the privacy of your users with mathematically-backed techniques and intuitive controls that you can apply in a few clicks.
- AWS Clean Rooms ML to allow two parties to identify similar users in their data without the need to share their data with each other. The first party creates and configures a lookalike model from their training data. The second party brings their seed data to a collaboration and creates a lookalike segment that resembles the training data.

The following video explains more about AWS Clean Rooms.

AWS Clean Rooms

Are you a first-time AWS Clean Rooms user?

If you are a first-time user of AWS Clean Rooms, we recommend that you begin by reading the following sections:

How AWS Clean Rooms works

The following workflow assumes that:

- The collaboration member has already uploaded their data tables to Amazon S3 and created an AWS Glue table.
- (Optional) For encrypted data tables only, the collaboration member has already prepared encrypted data tables using the C3R encryption client.

In summary, the workflow for AWS Clean Rooms is as follows:

1. The collaboration creator does the following tasks:
   - Creates a collaboration.
   - Invites one or more members to the collaboration.
   - Assigns abilities to members, such as the member who can query and the member who can receive results.

     If the collaboration creator is also the member who can receive results, they specify the query results destination and format. They also provide a service role Amazon Resource Name (ARN) to write the results to the query results destination.
   - Configures which member is responsible for paying for query compute costs in the collaboration.
2. The invited member joins the collaboration by creating a membership resource.

   If the invited member is the member who can receive results, they specify the query results destination and format. They also provide a service role ARN to write to the query results destination.

   If the invited member is the member who is responsible to pay for query compute costs, they accept their payment responsibilities before joining the collaboration.

3. The member configures an existing AWS Glue table for use in AWS Clean Rooms. (This step can be done before or after joining a collaboration, unless using Cryptographic Computing for Clean Rooms.)

   > **Note**
   >
   > AWS Clean Rooms supports AWS Glue tables. For more information about getting your data in AWS Glue, see Step 3: Upload your data table to Amazon S3.

   1. The member names the configured table and chooses which columns to use in the collaboration.
   2. The member configures one of the following analysis rules to the configured table:
      - Aggregation analysis rule or list analysis rule – To control the type of analysis that can be run on the table.
      - Custom analysis rule – To allow a specific set of pre-approved queries or a specific set of accounts that can provide queries that use your data. Allows the member to turn on differential privacy to protect against user-identification attempts.

      > **Note**
      >
      > The member can configure the analysis rule any time before they associate their configured tables with the collaboration.

4. The member associates their configured tables with the collaboration and gives AWS Clean Rooms a service role to access their AWS Glue tables.

   > **Note**
   >
   > This service role has permissions to the tables. The service role is assumable only by AWS Clean Rooms to run allowed queries on behalf of the member who can query. No collaboration members (other than the data owner) have access to the underlying tables in the collaboration. The data owner can turn on differential privacy to make their tables available for querying by other members.

5. The member who can query runs SQL queries on the configured tables.

Queries can only be run if the member who is responsible to pay for query compute costs has joined the collaboration as an active member.

The analysis rules and output constraints are enforced automatically. AWS Clean Rooms only returns the results that comply with the analysis rules defined in Step 3.b.

For queries on encrypted data, the member who can receive results receives the encrypted output from AWS Clean Rooms that must be decrypted (see Step 8).

6. The member who can receive results reviews the results in either the AWS Clean Rooms console or in the Amazon S3 bucket that they specified.
7. The member paying for query compute costs is charged for the queries run in the collaboration.
8. (Optional) For encrypted data tables only, the member who can receive results decrypts the query results by running the C3R encryption client in the decrypt mode.

Related services

The following AWS services are related to AWS Clean Rooms:

- **Amazon S3**

  Collaboration members can store data that they bring into AWS Clean Rooms in Amazon S3.

  For more information, see the following topics:

  Preparing data tables for queries in AWS Clean Rooms

  What Is Amazon S3? in the *Amazon Simple Storage Service User Guide*
- **AWS Glue**

  Collaboration members can create AWS Glue tables from their data in Amazon S3 for use in AWS Clean Rooms.

  For more information, see the following topics:

  Preparing data tables for queries in AWS Clean Rooms

  What is AWS Glue? in the *AWS Glue Developer Guide*

- **AWS CloudFormation**

  Create the following resources in AWS CloudFormation: collaborations, configured tables, configured table associations, and memberships

  For more information, see Creating AWS Clean Rooms resources with AWS CloudFormation.
- **AWS CloudTrail**

  Use AWS Clean Rooms with CloudTrail logs to enhance your analysis of AWS service activity.

  For more information, see Logging AWS Clean Rooms API calls using AWS CloudTrail.

Accessing AWS Clean Rooms

You can access AWS Clean Rooms through the following options:

- Directly through the AWS Clean Rooms console at https://console.aws.amazon.com/cleanrooms/.
- Programmatically through the AWS Clean Rooms API. For more information, see the *AWS Clean Rooms API Reference*.

Pricing for AWS Clean Rooms

For pricing information, see AWS Clean Rooms Pricing.

Billing for AWS Clean Rooms

AWS Clean Rooms gives the collaboration creator the ability to configure which member is paying for query compute costs in the collaboration.

In most cases, the member who can query and the member paying for query compute costs are the same. However, if the member who can query and the member paying for query compute costs are different, then, when the member who can query runs queries against their own membership resource, the membership resource of the member paying for query compute costs is billed.

The member paying for query compute costs doesn't see any event for queries being run in their CloudTrail Event history because the payer is neither the one running the queries nor the owner of the resource against which the queries are run. However, the payer does see bills generated on their membership resource for all queries run by the member who can run queries in the collaboration.

For more information about how to create a collaboration and configure the member paying for query compute costs, see Create a collaboration.

Analysis rules in AWS Clean Rooms

As part of enabling a table to use in AWS Clean Rooms for collaboration analysis, the collaboration member must configure an *analysis rule*.

An analysis rule is a privacy-enhancing control that each data owner sets up on a configured table. An analysis rule determines how the configured table can be analyzed.

The analysis rule is an account-level control on the configured table (an account-level resource) and is enforced in any collaboration where the configured table is associated. If there is no analysis rule configured, the configured table can be associated to collaborations but it can’t be queried. Queries can only reference configured tables with the same analysis rule type.

To configure an analysis rule, you first select a type of analysis and then specify the analysis rule. For both steps, you should consider the use case you want to enable and how you want to protect your underlying data.

AWS Clean Rooms enforces the more restrictive controls across all configured tables referenced in a query.

The following examples illustrate the restrictive controls.

Example Restrictive control: Output constraint

- Collaborator A has an output constraint on the identifier column of 100.
- Collaborator B has an output constraint on the identifier column of 150.

  An aggregation query that references both configured tables requires at least 150 distinct values of identifier within an output row for it to be displayed in the query output. The query output doesn't indicate that results are removed because of the output constraint.

Example Restrictive control: Analysis template not approved

- Collaborator A has allowed an analysis template with a query that references configured tables from Collaborator A and Collaborator B in their custom analysis rule.
- Collaborator B hasn't allowed the analysis template.

  Because Collaborator B hasn't allowed the analysis template, the member who can query can’t run that analysis template.

Analysis rule types

There are three types of analysis rules: aggregation, list and custom. The following tables compare the analysis rule types. Each type has a separate section that describes specifying the analysis rule.

The following tables show a comparison summary of the analysis rule types.

Supported use cases

The following tables show a comparison summary of the supported use cases for each analysis rule type.

Use case	Aggregation	List	Custom
Supported analyses	Queries that aggregate statistics using COUNT, SUM, and AVG functions along optional dimensions	Queries that output row-level lists of the overlap between multiple tables	Any custom analysis as long as the analysis template or the analysis creator have been reviewed and allowed
Common use cases	Segment analysis, measurement, attribution	Enrichment, segment building	First-touch attribution, incremental analyses, audience discovery
SQL constructs	• JOIN statement	• JOIN statement	Majority of SQL functions

Use case	Aggregation	List	Custom
	s: INNER JOIN • Aggregate functions: COUNT/ COUNT DISTINCT, SUM/ SUM DISTINCT, and AVG • Scalar functions: Limited subset	s: INNER JOIN • Scalar functions: None	and SQL constructs available with the SELECT command
Subqueries and common table expressions (CTEs)	No	No	Yes
Analysis templates	No	No	Yes

Supported controls

The following tables show a comparison summary of how each analysis rule type protects your underlying data.

Control	Aggregation	List	Custom
Control mechanism	Control how data in the table can be used in a query *(For example, allow COUNT and SUM of column hashed_email.)*	Control how data in the table can be used in a query *(For example, allow use of column hashed_email only for joining.)*	Control what queries are allowed to run on the table *(For example, allow only queries defined in analysis templates "Custom query 1".)*
Built-in privacy enhancing techniques	• Blind match • Aggregation required • Min aggregation threshold >= • 2 Pre-defined query structure	• Blind match • Overlap required • Pre-defined query structure	Differential privacy

Control	Aggregation	List	Custom
Review query before it can be run	No	No	Yes, using analysis templates

For more information about the analysis rules that are available in AWS Clean Rooms, see the following topics.

- Aggregation analysis rule
- List analysis rule
- Custom analysis rule in AWS Clean Rooms

Aggregation analysis rule

In AWS Clean Rooms, an *aggregation analysis rule* generates aggregate statistics using COUNT, SUM, and/or AVG functions along optional dimensions. When the aggregation analysis rule is added to a configured table, it enables the member who can query to run queries on the configured table.

The aggregation analysis rule supports uses cases such as campaign planning, media reach, frequency measurement, and attribution.

The supported query structure and syntax are defined in Aggregation query structure and syntax.

The parameters of the analysis rule, defined in Aggregation analysis rule - query controls, include query controls and query results controls. Its query controls include the ability to require that a configured table is joined to at least one configured table owned by the member who can query, either directly or transitively. This requirement allows you to ensure that the query is run on the intersection (INNER JOIN) of your table and theirs.

Aggregation query structure and syntax

Queries on tables that have an aggregation analysis rule must adhere to the following syntax.

```
    --select_aggregate_function_expression
SELECT
aggregation_function(column_name) [[AS] column_alias ] [, ...]

    --select_grouping_column_expression
  [, {column_name|scalar_function(arguments)} [[AS] column_alias ]][, ...]

--table_expression
FROM table_name [[AS] table_alias ]
  [[INNER] JOIN table_name [[AS] table_alias] ON join_condition] [...]

--where_expression
[WHERE where_condition]

--group_by_expression
[GROUP BY {column_name|scalar_function(arguments)}, ...]]

--having_expression
[HAVING having_condition]

--order_by_expression
[ORDER BY {column_name|scalar_function(arguments)} [{ASC|DESC}]] [,...]]
```

The following table explains each expression listed in the preceding syntax.

Expression	Definition	Examples
select_aggregate_function_expression	A comma-separated list containing the following expressions: • `select_aggregation_function_expression` • `select_aggregate_expression`	`SELECT SUM(PRICE), user_segment`

Expression	Definition	Examples
	Note There must be at least one `select_aggregation_function_expression` in the `select_aggregate_expression` .	

select_aggregation_function_expression	One or more supported aggregation functions applied to one or more columns. Only columns are allowed as arguments of aggregation functions.
	
	Note
	There must be at least one `select_aggregation_function_expression` in the `select_aggregate_expression` .
		`AVG(PRICE)`
		
		`COUNT(DISTINCT user_id)`

<table>
<tr><th>Expression</th><th>Definition</th><th>Examples</th></tr>
<tr><td><code>select_grouping_column_expression</code></td><td>An expression that can contain any expression using the following:
Table column names
Supported scalar functions
String literals
Numerical literals
Note
<code>select_aggregate_expression</code> can alias columns with or without the AS parameter. For more information, see the AWS Clean Rooms SQL Reference.</td><td><code>TRUNC(timestampColumn)</code>

<code>UPPER(campaignName)</code></td></tr>
</table>

<table>
<tr><th>Expression</th><th>Definition</th><th>Examples</th></tr>
<tr><td>table_expression</td><td>A table, or join of tables, connecting join conditional expressions with <code>join_condition</code>.

<code>join_condition</code> returns a Boolean.

The <code>table_expression</code> supports:
A specific JOIN type (INNER JOIN)
The equality comparison condition within a <code>join_condition</code> (=)
Logical operators (AND, OR).</td><td><pre>FROM consumer_table
INNER JOIN provider_table
ON
consumer_table.identifier1 = provider_table.identifier1
AND
consumer_table.identifier2 = provider_table.identifier2</pre></td></tr>
</table>

Expression	Definition	Examples

where_expression	A conditional expression that returns a Boolean. It may be comprised of the following:
	• Table column names
	• Supported scalar functions
	• Mathematical operators
	• String literals
	• Numerical literals
	
	Supported comparison conditions are (`=, >, <, <=, >=, <>, !=, NOT, IN, NOT IN, LIKE, IS NULL, IS NOT NULL`).
	
	Supported logical operators are (`AND, OR`).
	
	The `where_expression` is optional.
		`WHERE where_condition`
		
		`WHERE price > 100`
		
		`WHERE TRUNC(timestampColumn) = '1/1/2022'`
		
		`WHERE timestampColumn = timestampColumn2 - 14`

group_by_expression	A comma-separated list of expressions that match the requirements for the `select_grouping_column_expression` .	`GROUP BY TRUNC(timestampColumn), UPPER(campaignName), segment`

Expression	Definition	Examples
having_expression	A conditional expression that returns an Boolean. They have a supported aggregation function applied to a single column (for example, `SUM(price)`) and are compared to a numerical literal. Supported conditions are (`=, >, <, <=, >=, <>, !=`). Supported logical operators are (`AND, OR`). The `having_expression` is optional.	`HAVING SUM(SALES) > 500`

Expression	Definition	Examples
order_by_expression	A comma-separated list of expressions that is compatible with the same requirements defined in `select_aggregate_expression` defined earlier. The `order_by_expression` is optional. **Note** `order_by_expression` permits ASC and DESC parameters. For more information, see ASC DESC parameters in the [AWS Clean Rooms SQL Reference]().	`ORDER BY SUM(SALES), UPPER(campaignName)`

For aggregation query structure and syntax, be aware of the following:

- SQL commands other than SELECT are not supported.
- Sub-queries and common table expressions (for example, WITH) are not supported.
- Operators that combine multiple queries (for example, UNION) are not supported.
- TOP, LIMIT, and OFFSET parameters are not supported.

Aggregation analysis rule - query controls

With aggregation query controls, you can control how the columns in your table are used to query the table. For example, you can control which column is used for joining, which column can be counted, or which column can be used in WHERE statements.

The following sections explain each control.

Topics

- Aggregation controls
- Join controls
- Dimension controls
- Scalar functions

Aggregation controls

By using *aggregation controls*, you can define which aggregation functions to allow, and what columns they must to be applied to. Aggregation functions can be used in the SELECT, HAVING, and ORDER BY expressions.

Control	Definition	Usage
`aggregateColumns`	Columns of configured table columns you allow for use within aggregation functions.	`aggregateColumns` can be used inside an aggregati on function in the SELECT, HAVING, and ORDER BY expressions. Some `aggregateColumns` can also be categorized as a `joinColumn` (defined later). Given `aggregateColumn` can't also be categorized as a `dimensionColumn` (defined later).
`function`	The COUNT, SUM, and AVG functions you allow for use on top of the `aggregate Columns` .	`function` can be applied to an `aggregateColumns` that is associated to it.

Join controls

A `JOIN` clause is used to combine rows from two or more tables, based on a related column between them.

You can use *Join controls* to control how your table can be joined to other tables in the `table_expression`. AWS Clean Rooms only supports INNER JOIN. INNER JOIN statements can only use columns that have been explicitly categorized as a `joinColumn` in your analysis rule, subject to the controls that you define.

The INNER JOIN must operate on a `joinColumn` from your configured table and a `joinColumn` from another configured table in the collaboration. You decide which columns from your table can be used as `joinColumn`.

Each match condition within the ON clause is required to use the equality comparison condition (=) between two columns.

Multiple match conditions within an ON clauses can be:

- Combined using the `AND` logical operator
- Separated using the `OR` logical operator

> **Note**
>
> All JOIN match conditions must match one row from each side of the JOIN. All conditionals connected by an `OR` or an `AND` logical operator must adhere to this requirement as well.

The following is an example of a query with an `AND` logical operator.

```
SELECT some_col, other_col
FROM table1
    JOIN table2
    ON table1.id = table2.id AND table1.name = table2.name
```

The following is an example of a query with an `OR` logical operator.

```
SELECT some_col, other_col
FROM table1
    JOIN table2
```

```
    ON table1.id = table2.id OR table1.name = table2.name
```

Control	Definition	Usage
`joinColumns`	The columns (if any) that you want to allow the member who can query to use in the INNER JOIN statement.	A specific `joinColumn` can also be categorized as a `aggregateColumn` (see Aggregation controls). The same column can't be used both as `joinColumn` and `dimensionColumns` (see later). Unless it has also been categorized as an `aggregate Column` , a `joinColumn` can't be used in any other parts of the query other than the INNER JOIN.
`joinRequired`	Control whether you require an INNER JOIN with a configured table from the member who can query.	If you enable this parameter , an INNER JOIN is required. If you don't enable this parameter, an INNER JOIN is optional. Assuming you enable this parameter, the member who can query is required to include a table they own in the INNER JOIN. They must JOIN your table with theirs, either directly or transitively (that is, join their table to another table, which itself is joined to your table).

Following is an example of transitivity.

```
ON
my_table.identifer = third_party_table.identifier
....
ON
third_party_table.identifier = member_who_can_query_table.id
```

> **Note**
>
> The member who can query can also use the `joinRequired` parameter. In that case, the query must join their table with at least one other table.

Dimension controls

Dimension controls control the column along which the aggregation columns can be filtered, grouped, or aggregated.

Control	Definition	Usage
`dimensionColumns`	The columns (if any) that you allow the member who can query to use in SELECT, WHERE, GROUP BY, and ORDER BY.	A `dimensionColumn` can be used in SELECT (`select_grouping_column_expression`), WHERE, GROUP BY, and ORDER BY. The same column can't be both a `dimensionColumn` , a `joinColumn` , and/or an `aggregateColumn` .

Scalar functions

Scalar functions control which scalar functions can be used on dimension columns.

Control	Definition	Usage
`scalarFunctions`	The scalar functions that can be used on `dimension Columns` in the query.	Specifies the scalar functions (if any) that you allow (for example, CAST) to be applied on `dimensionColumns` . Scalar functions can't be used on top of other functions or within other functions. Arguments of scalar functions can be columns, string literals, or numeric literals.

The following scalar functions are supported:

- Math functions – ABS, CEILING, FLOOR, LOG, LN, ROUND, SQRT
- Data type formatting functions – CAST, CONVERT, TO_CHAR, TO_DATE, TO_NUMBER, TO_TIMESTAMP
- String functions – LOWER, UPPER, TRIM, RTRIM, SUBSTRING
  - For RTRIM, custom character sets to trim aren't allowed.
- Conditional expressions – COALESCE
- Date functions – EXTRACT, GETDATE, CURRENT_DATE, DATEADD
- Other functions – TRUNC

For more details, see the AWS Clean Rooms SQL Reference.

Aggregation analysis rule - query results controls

With the aggregation query results controls, you can control which results are returned by specifying one or more conditions that each output row must meet for it to be returned. AWS Clean Rooms supports aggregation constraints in the form of `COUNT (DISTINCT column) >= X`. This form requires that each row aggregates at least X distinct values of a choice from your configured table (for example, a minimum number of distinct `user_id` values). This minimum threshold is automatically enforced, even if the submitted query itself does not use the specified column. They are enforced collectively across each configured table in the query from the configured tables from each member in the collaboration.

Each configured table must have at least one aggregation constraint in their analysis rule. Configured table owners can add multiple `columnName` and associated `minimum` and they are enforced collectively.

Aggregation constraints

Aggregation constraints control which rows in the query results are returned. To be returned, a row must meet the specified minimum number of distinct values in each column specified in the aggregation constraint. This requirement applies even if the column isn't explicitly mentioned in the query or in other parts of the analysis rule.

Control	Definition	Usage
`columnName`	The `aggregateColumn` that is used in the condition that each output row must meet.	Can be any column in the configured table.
`minimum`	The minimum number of distinct values for the associated `aggregateColumn` that the output row must have (for example, COUNT DISTINCT) for it to be returned in the query results.	The `minimum` must be at least value of 2.

Aggregation analysis rule structure

The following example shows a predefined structure for an aggregation analysis rule.

In the following example, *`MyTable`* refers to your data table. You can replace each *`user input placeholder`* with your own information.

```
{
  "aggregateColumns": [
    {
      "columnNames": [MyTable column names], "function": [Allowed Agg Functions]
```

```
    },
  ],
  "joinRequired": ["QUERY_RUNNER"],
  "joinColumns": [MyTable column names],
  "dimensionColumns": [MyTable column names],
  "scalarFunctions": [Allowed Scalar functions],
  "outputConstraints": [
    {
      "columnName": [MyTable column names], "minimum": [Numeric value]
    },
  ]
}
```

Aggregation analysis rule - example

The following example demonstrates how two companies can collaborate in AWS Clean Rooms using aggregation analysis.

Company A has customer and sales data. Company A is interested in understanding product return activity. Company B is one of Company A's retailers and has returns data. Company B also has segment attributes on customers that are useful to Company A (for example, purchased related products, uses customer service from the retailer). Company B doesn't want to provide row-level customer return data and attribute information. Company B only wants to enable a set of queries for Company A to obtain aggregate statistics on overlapping customers at a minimum aggregation threshold.

Company A and Company B decide to collaborate so that Company A can understand product return activity and deliver better products at Company B and other channels.

To create the collaboration and run an aggregation analysis, the companies do the following:

1. Company A creates a collaboration and creates a membership. The collaboration has Company B as another member in the collaboration. Company A enables query logging in the collaboration, and it enables query logging in their account.
2. Company B creates a membership in the collaboration. It enables query logging in its account.
3. Company A creates a sales configured table.
4. Company A adds the following aggregation analysis rule to the sales configured table.

```
{
  "aggregateColumns": [
```

```
    {
      "columnNames": [
        "identifier"
      ],
      "function": "COUNT_DISTINCT"
    },
    {
      "columnNames": [
        "purchases"
      ],
      "function": "AVG"
    },
    {
      "columnNames": [
        "purchases"
      ],
      "function": "SUM"
    }
  ],
  "joinColumns": [
    "hashedemail"
  ],
  "dimensionColumns": [
    "demoseg",
    "purchasedate",
    "productline"
  ],
  "scalarFunctions": [
    "CAST",
    "COALESCE",
    "TRUNC"
  ],
  "outputConstraints": [
    {
      "columnName": "hashedemail",
      "minimum": 2,
      "type": "COUNT_DISTINCT"
    },
  ]
}
```

`aggregateColumns` – Company A wants to count the number of unique customers in the overlap between sales data and returns data. Company A also wants to sum the number of `purchases` made to compare to number of `returns`.

`joinColumns` – Company A wants to use `identifier` to match customers from sales data to customers from returns data. This will help Company A match returns to the right purchases. It also helps Company A segment overlapping customers.

`dimensionColumns` – Company A uses `dimensionColumns` to filter by the specific product, compare purchases and returns over a certain period of time, ensure the return date is after the product date, and help segment overlapping customers.

`scalarFunctions` – Company A selects CAST scalar function to help update data type formats if needed based on the configured table Company A associates to the collaboration. It also adds scalar functions to help formatting columns if needed.

`outputConstraints` – Company A sets minimum output constraints. It doesn't need to constrain the results because the analyst is allowed to see row-level data from their sales table

> **Note**
>
> Company A doesn't include `joinRequired` in the analysis rule. It provides flexibility for their analyst to query the sales table alone.

5. Company B creates a returns configured table.
6. Company B adds the following aggregation analysis rule to the returns configured table.

```
{
  "aggregateColumns": [
    {
      "columnNames": [
        "identifier"
      ],
      "function": "COUNT_DISTINCT"
    },
    {
      "columnNames": [
        "returns"
      ],
```

```
        "function": "AVG"
      },
      {
        "columnNames": [
          "returns"
        ],
        "function": "SUM"
      }
    ],
    "joinColumns": [
      "hashedemail"
    ],
    "joinRequired": [
      "QUERY_RUNNER"
    ],
    "dimensionColumns": [
      "state",
      "popularpurchases",
      "customerserviceuser",
      "productline",
      "returndate"
    ],
    "scalarFunctions": [
      "CAST",
      "LOWER",
      "UPPER",
      "TRUNC"
    ],
    "outputConstraints": [
      {
        "columnName": "hashedemail",
        "minimum": 100,
        "type": "COUNT_DISTINCT"
      },
      {
        "columnName": "producttype",
        "minimum": 2,
        "type": "COUNT_DISTINCT"
      }
    ]
}
```

`aggregateColumns` – Company B enables Company A to sum `returns` to compare to the number of purchases. They have at least one aggregate column because they are enabling an aggregate query.

`joinColumns` – Company B enables Company A to join on `identifier` to match customers from return data to customers from sales data. `identifier` data is particularly sensitive and having it as a `joinColumn` ensures that the data will never be outputted in a query.

`joinRequired` – Company B requires queries on the return data to be overlapped with the sales data. They don't want to enable Company A to query all individuals in their dataset. They also agreed on that restriction in their collaboration agreement.

`dimensionColumns` – Company B enables Company A to filter and group by `state`, `popularpurchases`, and `customerserviceuser` which are unique attributes that could help make the analysis for Company A. Company B enables Company A to use `returndate` to filter output on `returndate` that occurs after `purchasedate`. With this filtering, the output is more accurate for evaluating the impact of the product change.

`scalarFunctions` – Company B enables the following:

- TRUNC for dates
- LOWER and UPPER in case the `producttype` is entered in a different format in their data
- CAST if Company A needs to convert data types in sales to be the same as data types in returns

Company A doesn't enable other scalar functions because they don't believe they are required for queries.

`outputConstraints` – Company B sets minimum output constraints on `hashedemail` to help reduce the ability to re-identify customers. It also adds minimum output constraint on `producttype` to reduce the ability to re-identify specific products that were returned. Certain product types could be more dominant based on dimensions of the output (for example, `state`). Their output constraints will always be enforced regardless of output constraints added by Company A to their data.

7. Company A creates a sales table association to collaboration.
8. Company B creates a returns table association to collaboration.

9. Company A runs queries, such as the following example, to better understand the quantity of returns in Company B as compared to total purchases by location in 2022.

```
SELECT
  companyB.state,
  SUM(companyB.returns),
  COUNT(DISTINCT companyA.hashedemail)
FROM
  sales companyA
  INNER JOIN returns companyB ON companyA.identifier = companyB.identifier
WHERE
  companyA.purchasedate BETWEEN '2022-01-01' AND '2022-12-31' AND
  TRUNC(companyB.returndate) > companyA.purchasedate
GROUP BY
  companyB.state;
```

10. Company A and Company B review query logs. Company B verifies that the query aligns with what was agreed upon in the collaboration agreement.

Troubleshooting aggregation analysis rule issues

Use the information here to help you diagnose and fix common issues when you work with aggregation analysis rules.

Issues

- [My query didn't return any results](#)

My query didn't return any results

This can happen when there are no matching results or when the matching results don't meet one or more minimum aggregation thresholds.

For more information about minimum aggregation thresholds, see [Aggregation analysis rule - example](#).

List analysis rule

In AWS Clean Rooms, a *list analysis rule* outputs row-level lists of the overlap between the configured table that it's added to and the configured tables of the member who can query. The member who can query runs queries that include a list analysis rule.

The list analysis rule type supports uses cases such as enrichment and audience building.

For more information about the predefined query structure and syntax for this analysis rule, see List analysis rule predefined structure.

The parameters of the list analysis rule, defined in List analysis rule - query controls, have query controls. Its query controls include the ability to select the columns that can be listed in the output. The query is required to have at least one join with a configured table from the member who can query, either directly or transitively.

There are no query results controls like there are for the Aggregation analysis rule.

List queries can only use mathematical operators. They can't use other functions (such as aggregation or scalar).

Topics

List query structure and syntax

Queries on tables that have a list analysis rule must adhere to the following syntax.

```
--select_list_expression
SELECT
[TOP number ] DISTINCT column_name [[AS] column_alias ] [, ...]

--table_expression
FROM table_name [[AS] table_alias ]
  [[INNER] JOIN table_name [[AS] table_alias] ON join_condition] [...]
```

```
--where_expression
[WHERE where_condition]

--limit_expression
[LIMIT number]
```

The following table explains each expression listed in the preceding syntax.

Expression	Definition	Examples
select_list_expression	A comma-separated list containing at least one table column name. A `DISTINCT` parameter is required. **Note** The `select_list_expression` can alias columns with or without the `AS` parameter. It also supports the `TOP` parameter. For more information, see the [AWS Clean Rooms SQL Reference](#).	`SELECT DISTINCT segment`
table_expression	A table, or join of tables, with `join_condition` to connect it to `join_condition`. `join_condition` returns a Boolean.	`FROM consumer_table INNER JOIN provider_table ON consumer_table.identifier1 = provider_table.identifier1 AND`

Expression	Definition	Examples
	The `table_expression` supports: • A specific JOIN type (INNER JOIN) • The equality comparison conditions within a `join_condition` (=) • Logical operators (AND, OR).	`consumer_table.identifier2 = provider_table.identifier2`

where_expression	A conditional expression that returns a Boolean. It can be comprised of the following:
	• Table column names
	• Mathematical operators
	• String literals
	• Numerical literals
	
	Supported comparison conditions are (`=, >, <, <=, >=, <>, !=, NOT, IN, NOT IN, LIKE, IS NULL, IS NOT NULL`).
	
	Supported logical operators are (`AND, OR`).
	
	The `where_expression` is optional.
		`WHERE state + '_' + city = 'NY_NYC'`
		
		`WHERE timestampColumn = timestampColumn2 - 14`

Expression	Definition	Examples
limit_expression	This expression must take a positive integer. It can also be interchanged with a TOP parameter. The `limit_expression` is optional.	`LIMIT 100`

For list query structure and syntax, be aware of the following:

- SQL commands other than SELECT are not supported.
- Subqueries and common table expressions (for example, WITH) are not supported
- HAVING, GROUP BY, and ORDER BY clauses are not supported
- OFFSET parameter is not supported

List analysis rule - query controls

With list query controls, you can control how the columns in your table are used to query the table. For example, you can control which column is used for joining, or which column can be used in SELECT statement and WHERE clause.

The following sections explain each control.

Topics

Join controls

With *Join controls*, you can control how your table can be joined to other tables in the **table_expression**. AWS Clean Rooms only supports INNER JOIN. In the list analysis rule, at least one INNER JOIN is required and the member who can query is required to include a table they own in the INNER JOIN. This means they must join your table with theirs, either directly or transitively.

Following is an example of transitivity.

```
ON
my_table.identifer = third_party_table.identifier
....
ON
third_party_table.identifier = member_who_can_query_table.id
```

INNER JOIN statements can only use columns that have been explicitly categorized as a `joinColumn` in your analysis rule.

The INNER JOIN must operate on a `joinColumn` from your configured table and a `joinColumn` from another configured table in the collaboration. You decide which columns from your table can be used as `joinColumn`.

Each match condition within the ON clause is required to use the equality comparison condition (=) between two columns.

Multiple match conditions within an ON clause can be:

- Combined using the `AND` logical operator
- Separated using the `OR` logical operator

> **Note**
>
> All JOIN match conditions must match one row from each side of the JOIN. All conditionals connected by an `OR` or an `AND` logical operator must adhere to this requirement as well.

The following is an example of a query with an `AND` logical operator.

```
SELECT some_col, other_col
FROM table1
    JOIN table2
    ON table1.id = table2.id AND table1.name = table2.name
```

The following is an example of a query with an `OR` logical operator.

```
SELECT some_col, other_col
```

```
FROM table1
    JOIN table2
    ON table1.id = table2.id OR table1.name = table2.name
```

Control	Definition	Usage
`joinColumns`	The columns that you want to allow the member who can query to use in the INNER JOIN statement.	The same column can't be categorized as both a `joinColumn` and `listColumn` (see List controls). `joinColumn` can't be used in any other parts of the query other than INNER JOIN.

List controls

List controls control the columns that can be listed in the query output (that is, used in the SELECT statement) or used to filter results (that is, used in the WHERE statement).

Control	Definition	Usage
`listColumns`	The columns that you allow the member who can query to use in the SELECT and WHERE	A `listColumn` can be used in SELECT and WHERE. The same column can't be used as both a `listColumn` and `joinColumn` .

List analysis rule predefined structure

The following example includes a predefined structure that shows how you complete a list analysis rule.

In the following example, *`MyTable`* refers to your data table. You can replace each *`user input placeholder`* with your own information.

```
{
  "joinColumns": [MyTable column name(s)],
  "listColumns": [MyTable column name(s)],
}
```

List analysis rule - example

The following example demonstrates how two companies can collaborate in AWS Clean Rooms using list analysis.

Company A has customer relationship management (CRM) data. Company A wants to obtain additional segment data on its customers to learn more about their customers and potentially use attributes as input into other analyses. Company B has segment data comprised of unique segment attributes that they created based on their first party data. Company B wants to provide the unique segment attributes to Company A only on customers that are overlapping between their data and Company A data.

The companies decide to collaborate so that Company A can enrich the overlapping data. Company A is the member who can query, and Company B is the contributor.

To create a collaboration and run list analysis in collaboration, the companies do the following:

1. Company A creates a collaboration and creates a membership. The collaboration has Company B as another member on the collaboration. Company A enables query logging in the collaboration, and it enables query logging in its account.
2. Company B creates a membership in the collaboration. It enables query logging in its account.
3. Company A creates a CRM configured table
4. Company A adds the analysis rule to the customer configured table, as shown in the following example.

```
{
  "joinColumns": [
    "identifier1",
    "identifier2"
  ],
  "listColumns": [
    "internalid",
    "segment1",
    "segment2",
    "customercategory"
```

```
  ]
}
```

`joinColumns` – Company A wants to use `hashedemail` and/or `thirdpartyid` (obtained from an identity vendor) to match customers from CRM data to customers from segment data. This will help ensure Company A matches enriched data for the right customers. They have two joinColumns to potentially improve the match rate of the analysis.

`listColumns` – Company A uses `listColumns` to obtain enriched columns beside an `internalid` they use within their own systems. They add `segment1`, `segment2`, and `customercategory` to potentially limit the enrichment to specific segments by using them in filters.

5. Company B creates a segment configured table.
6. Company B adds the analysis rule to the segment configured table.

```
{
  "joinColumns": [
    "identifier2"
  ],
  "listColumns": [
    "segment3",
    "segment4"
  ]
}
```

`joinColumns` – Company B enables Company A to join on `identifier2` to match customers from segment data to CRM data. Company A and Company B worked with the identity vendor to obtain `identifier2` which would match for this collaboration. They didn't add other `joinColumns` because they believed `identifier2` provides the highest and most accurate match rate and other identifiers aren't required for the queries.

`listColumns` – Company B enables Company A to enrich their data with `segment3` and `segment4` attributes which are unique attributes they have created, collected and aligned on (with customer A) to be a part of data enrichment. They want Company A to obtain these segments for the overlap at a row-level because this is a data enrichment collaboration.

7. Company A creates a CRM table association to the collaboration.
8. Company B creates a segment table association to the collaboration.
9. Company A runs queries, such as the following one to enrich overlapping customer data.

```
SELECT companyA.internalid, companyB.segment3, companyB.segment4
INNER JOIN returns companyB
 ON companyA.identifier2 = companyB.identifier2
WHERE companyA.customercategory > 'xxx'
```

10. Company A and Company B review query logs. Company B verifies that the query aligns with what was agreed upon in the collaboration agreement.

Custom analysis rule in AWS Clean Rooms

In AWS Clean Rooms, a *custom analysis rule* is a new type of analysis rule that allows custom queries to be run on the configured table. Custom SQL queries are still restricted to having only the SELECT command but can use more SQL constructs than aggregation and list queries (for example, window functions, OUTER JOIN, CTEs, or subqueries; see the AWS Clean Rooms SQL Reference for a complete list). Custom SQL queries don't have to follow a query structure like aggregation and list queries.

The custom analysis rule supports more advanced use cases than those that can be supported by the aggregation and list analysis rule such as custom attribution analysis, benchmarking, incrementality analysis, and audience discovery. This is in addition to a superset of the use cases supported by aggregation and list analysis rule.

The custom analysis rule also supports differential privacy. Differential privacy is a mathematically-rigorous framework for data privacy protection. For more information, see AWS Clean Rooms Differential Privacy. When you create an analysis template, AWS Clean Rooms Differential Privacy checks the template to determine whether it is compatible with the general-purpose query structure for AWS Clean Rooms Differential Privacy. This validation ensures that you don't create an analysis template that isn't allowed with a differential privacy protected table.

To configure the custom analysis rule, data owners can choose to allow specific custom queries, stored in analysis templates, to run on their configured tables. Data owners review analysis templates before adding them to the allowed analysis control in the custom analysis rule. Analysis templates are available and visible only in the collaboration in which they are created (even if the table is associated to other collaborations) and can be run only by the member who can query in that collaboration.

Alternatively, members can choose to allow other members (query providers) to create queries without review. Members add query providers' accounts the allowed query providers control in the custom analysis rule. If the query provider is the member who can query, they could run any query directly on the configured table. Query providers could also create queries by creating analysis templates. Any queries that have been created by the query providers are automatically allowed to run on the table in all collaborations in which the AWS account is present and the table is associated.

Data owners can only allow analysis templates or accounts to create queries, not both. If the data owner leaves it empty, the member who can query can't run queries on the configured table.

Topics

Custom analysis rule predefined structure

The following example includes a predefined structure that shows you how to complete a custom analysis rule with differential privacy turned on. The `userIdentifier` value is the column that uniquely identifies your users, such as *user_id*. When you have two or more tables with differential privacy turned on in a collaboration, AWS Clean Rooms requires you to configure the same column as the user identifier column in both of the analysis rules to maintain a consistent definition of the users across tables.

```
{
  "allowedAnalyses": ["ANY_QUERY"] | string[],
  "allowedAnalysisProviders": [],
  "differentialPrivacy": {
    "columns": [
      {
        "name": "userIdentifier"
      }
    ]
  }
}
```

You can either:

- Add analysis template ARNs to allowed analyses control. In this case, the `allowedAnalysisProviders` control is not included.

```
{
  allowedAnalyses: string[]
}
```

- Add member AWS account IDs to the `allowedAnalysisProviders` control. In this case, you add `ANY_QUERY` to the `allowedAnalyses` control.

```
{
  allowedAnalyses: ["ANY_QUERY"],
  allowedAnalysisProviders: string[]
}
```

Custom analysis rule example

The following example demonstrates how two companies can collaborate in AWS Clean Rooms using the custom analysis rule.

Company A has customer and sales data. Company A is interested in understanding the sales incrementality of an advertising campaign on Company B site. Company B has viewership data and segment attributes that are useful to Company (for example, the device they used when viewing the advertising).

Company A has a specific incrementality query they want to run in the collaboration.

To create a collaboration and run a custom analysis in collaboration, the companies do the following:

1. Company A creates a collaboration and creates a membership. The collaboration has Company B as another member on the collaboration. Company A enables query logging in the collaboration, and it enables query logging in its account.
2. Company B creates a membership in the collaboration. It enables query logging in its account.
3. Company A creates a CRM configured table
4. Company A adds empty custom analysis rule to sales configured table.
5. Company A associates sales configured table to the collaboration.
6. Company B creates viewership configured table.
7. Company B adds an empty custom analysis rule to the viewership configured table.
8. Company B associates viewership configured table to the collaboration.

9. Company A views the sales table and viewership table associated to the collaboration and creates analysis template, adding the incrementality query and parameter for campaign month.

```
{
    "analysisParameters": [
    {
        "defaultValue": ""
        "type": "DATE"
        "name": "campaign_month"
    }
    ],
    "description": "Monthly incrementality query using sales and viewership data"
    "format": "SQL"
    "name": "Incrementality analysis"
    "source":
        "WITH labeleddata AS
        (
        SELECT hashedemail, deviceid, purchases, unitprice, purchasedate,
        CASE
            WHEN testvalue IN ('value1', 'value2', 'value3') THEN 0
            ELSE 1
        END AS testgroup
        FROM viewershipdata
        )
        SELECT labeleddata.purchases, provider.impressions
        FROM labeleddata
        INNER JOIN salesdata
          ON labeleddata.hashedemail = provider.hashedemail
        WHERE MONTH(labeleddata.purchasedate) > :campaignmonth
        AND testgroup = :group
       "
}
```

10. Company A adds their account (for example, 444455556666) to the allowed analysis provider control in the custom analysis rule. They use the allowed analysis provider control because they want to allow any queries they create to run on their sales configured table.

```
{
  "allowedAnalyses": [
    "ANY_QUERY"
  ],
  "allowedAnalysisProviders": [
    "444455556666"
```

```
  ]
}
```

11. Company B sees the created analysis template in the collaboration and reviews its contents including the query string and parameter.
12. Company B determines that the analysis template achieves the incrementality use case and meets their privacy requirements for how their viewership configured table can be queried.
13. Company B adds the analysis template ARN to the allowed analysis control in the custom analysis rule of the viewership table. They use the allowed analysis control because they only want to allow the incrementality query to run on their viewership configured table.

```
{
  "allowedAnalyses": [
    "arn:aws:cleanrooms:us-east-1:111122223333:membership/41327cc4-bbf0-43f1-b70c-a160dddceb08/analysistemplate/1ff1bf9d-781c-418d-a6ac-2b80c09d6292"
  ]
}
```

14. Company A runs the analysis template and uses the parameter value `05-01-2023`.

Custom analysis rule with differential privacy

In AWS Clean Rooms, the custom analysis rule supports differential privacy. Differential privacy is a mathematically-rigorous framework for data privacy protection that helps you protect your data against re-identification attempts.

Differential privacy supports aggregate analysis such as ad campaign planning, post-ad-campaign measurement, benchmarking in a financial institution consortium, and A/B testing for healthcare research.

The supported query structure and syntax are defined in Query structure and syntax.

Custom analysis rule with differential privacy example

Consider the custom analysis rule example presented in the previous section. This example demonstrates how you can use differential privacy to protect your data against re-identification attempts while allowing your partner to learn business-critical insights from your data. Assume that Company B, who has the viewership data, wants to protect their data using differential privacy. To complete the differential privacy setup, Company B completes the following steps:

1. Company B turns on differential privacy while adding custom analysis rule to the viewership configured table. Company B selects `viewershipdata.hashedemail` as the user identifier column.
2. Company B adds a differential privacy policy in the collaboration to make their viewership data table available for querying. Company B selects the default policy to quickly complete the setup.

Company A, who wants to understand the sales incrementality of an advertising campaign on Company B's site, runs the analysis template. Because the query is compatible with the general-purpose query structure of AWS Clean Rooms Differential Privacy, the query runs successfully.

Query structure and syntax

Queries containing at least one table that have the differential privacy turned on must adhere to the following syntax.

```
query_statement:
    [cte, ...] final_select

 cte:
    WITH sub_query AS (
       inner_select
       [ UNION | INTERSECT | UNION_ALL | EXCEPT/MINUS ]
       [ inner_select ]
    )

 inner_select:
     SELECT [user_id_column, ] expression [, ...]
     FROM table_reference [, ...]
     [ WHERE condition ]
     [ GROUP BY user_id_column[, expression] [, ...] ]
     [ HAVING condition ]

 final_select:
     SELECT [expression, ...] | COUNT | COUNT_DISTINCT | SUM | AVG | STDDEV
     FROM table_reference [, ...]
     [ WHERE condition ]
     [ GROUP BY expression [, ...] ]
     [ HAVING COUNT | COUNT_DISTINCT | SUM | AVG | STDDEV | condition ]
     [ ORDER BY column_list ASC | DESC ]
     [ OFFSET literal ]
     [ LIMIT literal ]
```

```
expression:
   column_name [, ...] | expression AS alias | aggregation_functions |
window_functions_on_user_id | scalar_function | CASE | column_name math_expression [,
expression]

window_functions_on_user_id:
   function () OVER (PARTITION BY user_id_column, [column_name] [ORDER BY column_list
ASC|DESC])
```

> **Note**
>
> For differential privacy query structure and syntax, be aware of the following:
>
> - Sub-queries are not supported.
> - Common Table Expressions (CTEs) should emit the user identifier column if a table or CTE involve data protected by differential privacy. Filters, groupings, and aggregations should be done at the user level.
> - Final_select allows COUNT DISTINCT, COUNT, SUM, AVG, and STDDEV aggregate functions.

For more details about which SQL keywords are supported for differential privacy, see [SQL capabilities of AWS Clean Rooms Differential Privacy](#).

AWS Clean Rooms Differential Privacy

AWS Clean Rooms Differential Privacy helps you protect the privacy of your users with a mathematically-backed technique that is implemented with intuitive controls in a few clicks. As a fully managed capability, no prior differential privacy experience is needed to help you prevent the re-identification of your users. AWS Clean Rooms automatically adds a carefully calibrated amount of noise to query results at runtime in order to help protect your individual-level data.

AWS Clean Rooms Differential Privacy supports a wide range of analytical queries and is a good fit for a wide variety of use cases, where a small amount of error in the query results will not compromise the usefulness of your analysis. With it, your partners can generate business-critical insights about advertising campaigns, investment decisions, clinical research, and more, all without requiring any additional setup from your partners.

AWS Clean Rooms Differential Privacy protects against overflow or invalid cast errors that make use of scalar functions or math operator symbols in a malicious manner.

For more information about AWS Clean Rooms Differential Privacy, see the following topics.

Topics

Differential privacy

Differential privacy allows only aggregated insights and obfuscates the contribution of any individual's data in those insights. Differential privacy protects the collaboration data from the member who can receive results learning about a specific individual. Without differential privacy, the member who can receive results can attempt to infer individual user data by adding or removing records about an individual and observing the difference in query results.

When differential privacy is turned on, a specified amount of noise is added to the query results to obfuscate the contribution of individual users. If the member who can receive results tries to observe the difference in query results after removing records about an individual from their dataset, the variability in the query result helps prevent the identification of the individual's data. AWS Clean Rooms Differential Privacy uses the SampCert sampler, a proven correct sampler implementation developed by AWS.

How Differential Privacy in AWS Clean Rooms works

The workflow to turn on differential privacy in AWS Clean Rooms requires the following additional steps when completing the workflow for AWS Clean Rooms:

1. You turn on differential privacy when adding a custom analysis rule.
2. You configure the differential privacy policy for the collaboration to make your data tables protected with differential privacy available for querying.

After you complete these steps, the member who can query can start running queries on differential privacy protected data. AWS Clean Rooms returns results that comply with the differential privacy policy. AWS Clean Rooms Differential Privacy tracks the estimated number of remaining queries that you can run, similar to the gas gauge in a car that shows you the car's current fuel level. The number of queries that the member who can query can run is limited by the **Privacy budget** and **Noise added per query** parameters that are set in the Differential privacy policy.

Considerations

When using differential privacy in AWS Clean Rooms, consider the following:

- The member who can receive results can't use differential privacy. They will configure a custom analysis rule with differential privacy turned off for their configured tables.
- The member who can query can't join tables from two or more data providers when both have differential privacy turned on.

Differential privacy policy

The differential privacy policy controls how many aggregation functions the member who can query is allowed to run in a collaboration. The **Privacy budget** defines a common, finite resource that is applied all tables in a collaboration. The **Noise added per query** governs the rate at which the privacy budget is depleted.

A differential privacy policy is required to make your differential privacy protected tables available for querying. This is a one-time step in a collaboration and includes two inputs:

- **Privacy budget** – Quantified in terms of epsilon, the privacy budget controls the level of privacy protection. It is a common, finite resource that is applied for all of your tables protected with differential privacy in the collaboration, because the goal is to preserve the privacy of your users whose information can be present in multiple tables.

  The **Privacy budget** is consumed every time a query is run on your tables. When the privacy budget is fully exhausted, the collaboration member who can query can't run additional queries until it is increased or refreshed. By setting a larger privacy budget, the member who can receive results can reduce their uncertainty about individuals within the data. Choose a privacy budget that balances your collaboration requirements against your privacy needs and after consulting with business decision-makers.

  You can select **Refresh privacy budget monthly** to automatically create a new privacy budget each calendar month, if you plan to regularly bring new data into the collaboration. Choosing this option allows arbitrary amounts of information to be revealed about rows of the data when repeatedly queried across refreshes. Avoid choosing this if the same rows will be repeatedly queried between privacy budget refreshes.
- **Noise added per query** is measured in terms of the number of users whose contributions you want to obscure. This value governs the rate at which the privacy budget is depleted. A larger noise value reduces the rate at which the privacy budget is depleted, and therefore allows more queries to be run on your data. However, this should be balanced against releasing less accurate data insights. Consider the desired accuracy for collaboration insights when setting this value.

You can use the default differential privacy policy to quickly complete the setup or customize your differential privacy policy as per your use case. AWS Clean Rooms Differential Privacy provides intuitive controls to configure the policy. AWS Clean Rooms Differential Privacy lets you preview the utility in terms of the number of aggregations possible across all queries on your data and estimate how many queries can be run in a data collaboration.

You can use the interactive examples to understand how different values of **Privacy budget** and **Noise added per query** would impact the results for different types of SQL queries. In general, you need to balance your privacy needs against the number of queries you want to permit and the accuracy of those queries. A smaller **Privacy budget** or larger **Noise added per query** can better protect user privacy, but provides less meaningful insights to your collaboration partners.

If you increase the **Privacy budget** while keeping the **Noise added per query** parameter the same, the member who can query can run more aggregations on your tables in the collaboration. You can increase the **Privacy budget** any time during the collaboration. If you decrease the **Privacy budget** while keeping the **Noise added per query** parameter the same, the member who can query can run fewer aggregations. You can't decrease the **Privacy budget** after the member who can query has started analyzing your data.

If you increase the **Noise added per query** while keeping the **Privacy budget** input the same, the member who can query can run more aggregations on your tables in the collaboration. If you decrease the **Noise added per query** while keeping the **Privacy budget** input the same, the member who can query can run fewer aggregations. You can increase or decrease the **Noise added per query** any time during the collaboration.

The differential privacy policy is managed by the privacy budget template API actions.

SQL capabilities of AWS Clean Rooms Differential Privacy

AWS Clean Rooms Differential Privacy uses a general-purpose query structure to support complex SQL queries. Custom analysis templates are validated against this structure to ensure that they can run on tables protected by differential privacy. The following table indicates which functions are supported. See [Query structure and syntax]() for more information.

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Aggregate functions	• ANY_VALUE function • APPROXIMATE PERCENTILE_DISC function • AVG function • COUNT and COUNT DISTINCT functions • LISTAGG function	Supported with the condition that CTEs using differential privacy protected tables must result in data with user-level records. You should write the SELECT expression in those CTEs using `SELECT	Supported aggregations: AVG, COUNT, COUNT DISTINCT, STDDEV, and SUM.

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
	• MAX function • MEDIAN function • MIN function • PERCENTILE_CONT function • STDDEV_SAMP and STDDEV_POP functions • SUM and SUM DISTINCT functions • VAR_SAMP and VAR_POP functions	`userIdentifierColumn...'` format.	
CTEs	WITH clause, WITH clause subquery	Supported with the condition that CTEs using differential privacy protected tables must result in data with user-level records. You should write the SELECT expression in those CTEs using `` `SELECT userIdentifierColumn...' `` format.	N/A
Subqueries	SELECT list subquery, FROM clause subquery, WHERE clause subquery	Not supported. Subqueries in the query that references a table with differential privacy turned on are not supported. Rewrite your subqueries as Common Table Expressions (CTEs).	

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Join clauses	• INNER JOIN • LEFT JOIN • RIGHT JOIN • FULL JOIN • [JOIN] OR operator • CROSS JOIN	Supported with the condition that only JOIN functions that are equi-joins on user identifier columns are supported and are mandatory when querying two or more tables with differential privacy turned on. Ensure that the mandatory equi-join conditions are correct. Confirm that the table owner has configured the same user identifier column in all tables so that the definition of a user remains consistent across tables. CROSS JOIN functions are not supported when combining two or more relations with differential privacy turned on.	
Set operators	UNION, UNION ALL, INTERSECT, EXCEPT | MINUS (these are synonyms)	All are supported	Not supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Window functions	Aggregate functions • AVG window function • COUNT window function • CUME_DIST window function • DENSE_RANK window function • FIRST_VALUE window function • LAG window function • LAST_VALUE window function • LEAD window function • MAX window functions • MEDIAN window functions • MIN window functions • NTH_VALUE window function • RATIO_TO_REPORT window function • STDDEV_SAMP and STDDEV_POP window function	All are supported with the condition that the user identifie r column in the window function's partition clause is required when you query a relation with differential privacy turned on.	Not supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
	(STDDEV_SAMP and STDDEV are synonyms) • SUM window functions • VAR_SAMP and VAR_POP window functions (VAR_SAMP and VARIANCE are synonyms) Ranking functions • DENSE_RANK window function • NTILE window function • PERCENT_RANK window function • RANK window function • ROW_NUMBER window function		

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Conditional expressions	• CASE condition expression • COALESCE expression • GREATEST and LEAST functions • NVL and COALESCE functions • NVL2 function • NULLIF function	All are supported	All are supported
Conditions	• Comparison condition • Logical conditions • Pattern-matching conditions • BETWEEN range conditions • Null condition	`EXISTS` and `IN` cannot be used because they require subqueries. All others are supported.	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Date-time functions	• Date and time functions in transactions • Concatenation operator • ADD_MONTHS functions • CONVERT_TIMEZONE function • CURRENT_DATE function • DATEADD function • DATEDIFF function • DATE_PART functions • DATE_TRUNC function • EXTRACT function • GETDATE function • TIMEOFDAY functions • TO_TIMESTAMP function • Date parts for date or timestamp functions	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
String functions	• || (concatenation) operator • BTRIM function • CHAR_LENGTH function • CHARACTER _LENGTH function • CHARINDEX function • CONCAT function • LEFT and RIGHT functions • LEN function • LENGTH function • LOWER function • LPAD and RPAD functions • LTRIM function • POSITION functions • REGEXP_COUNT function • REGEXP_INSTR function • REGEXP_REPLACE function • REGEXP_SUBSTR function • REPEAT function • REPLACE function	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
	• REPLICATE function • REVERSE function • RTRIM function • SOUNDEX function • SPLIT_PART function • STRPOS function • SUBSTRING function • TEXTLEN function • TRANSLATE function • TRIM functions • UPPER function		
Data type formatting functions	• CAST function • TO_CHAR • TO_DATE function • TO_NUMBER • Datetime format strings • Numeric format strings	All are supported	All are supported
Hash functions	• MD5 function • SHA function • SHA1 function • SHA2 function • MURMUR3_32_HASH	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Mathematical operator symbols	+, -, *, /, %, and @	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Math functions	• ABS function • ACOS function • ASIN function • ATAN function • ATAN2 function • CBRT function • CEILING (or CEIL) function • COS function • COT function • DEGREES function • DEXP function • LTRIM function • DLOG1 function • DLOG10 function • EXP function • FLOOR function • LN function • LOG function • MOD function • PI function • POWER function • RADIANS function • RANDOM function • ROUND function • SIGN function • SIN function • SQRT functions	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
	• TRUNC function		
SUPER type information functions	• DECIMAL_PRECISION function • DECIMAL_SCALE function • IS_ARRAY function • IS_BIGINT function • IS_CHAR function • IS_DECIMAL function • IS_FLOAT function • IS_INTEGER function • IS_OBJECT function • IS_SCALAR function • IS_SMALLINT function • IS_VARCHAR function • JSON_TYPEOF function	All are supported	All are supported
VARBYTE functions	• FROM_HEX function • FROM_VARBYTE function • TO_HEX function • TO_VARBYTE function	All are supported	All are supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
JSON	• CAN_JSON_PARSE function • JSON_EXTRACT_ARRAY_ELEMENT_TEXT function • JSON_EXTRACT_PATH_TEXT function • JSON_PARSE function • JSON_SERIALIZE function • JSON_SERALIZE_TO_VARBYTE function	All are supported	All are supported
Array functions	• array function • array_concat function • array_flatten function • get_array_length function • split_to_array function • subarray function	Not supported	Not supported
Extended GROUP BY	GROUPING SETS, ROLLUP, CUBE	Not supported	Not supported

Short name	SQL constructs	Common table expressions (CTEs)	Final SELECT clause
Sort operation	ORDER BY	Supported with the condition that an ORDER BY clause is only supported in a window function's partition clause when querying tables with differential privacy turned on.	Supported
Row limits	LIMIT, OFFSET	Not supported in CTEs using different ial privacy protected tables	All are supported
Table and column aliasing		Supported	Supported
Math functions on aggregate functions		Supported	Supported
Scalar functions within aggregate functions		Supported	Supported

Common alternatives for unsupported SQL constructs

Category	SQL construct	Alternative
Window functions	• LISTAGG • PERCENTILE_CONT • PERCENTILE_DISC	You can use the equivalen t aggregate function with GROUP BY.

Category	SQL construct	Alternative
Mathematical operator symbols	• $column ||/ 2 • $column |/ 2 • $column ^ 2	• CBRT • SQRT • POWER($column, 2)
Scalar functions	• SYSDATE • $column::integer • convert(type, $column)	• CURRENT_DATE • CAST $column AS integer • CAST $column AS type
Literals	INTERVAL ‘1 SECOND'	INTERVAL '1' SECOND
Row limiting	TOP n	LIMIT n
Join	• USING • NATURAL	ON clause should explicitly contain a join criterion.

Differential Privacy query tips and examples

AWS Clean Rooms Differential Privacy uses a general-purpose query structure to support a wide variety of SQL constructs such as Common Table Expressions (CTEs) for data preparation and commonly used aggregate functions such as `COUNT`, or `SUM`. In order to obfuscate the contribution of any possible user in your data by adding noise to aggregate query results at run-time, AWS Clean Rooms Differential Privacy requires that aggregate functions in the final `SELECT statement` are run on user-level data.

The following example uses two tables named `socialco_impressions` and `socialco_users` from a media publisher who wants to protect data using differential privacy while collaborating with an athletic brand with `athletic_brand_sales` data. The media publisher has configured the `user_id` column as the user identifier column while enabling differential privacy in AWS Clean Rooms. The advertiser does not need differential privacy protection and wants to run a query using CTEs on combined data. Since their CTE uses differential privacy protected tables, the advertiser includes the user identifier column from those protected tables in the list of CTE columns and joins the protected tables on the user identifier column.

```
WITH matches_table AS(
     SELECT si.user_id, si.campaign_id, s.sale_id, s.sale_price
```

```
    FROM socialco_impressions si
    JOIN socialco_users su
        ON su.user_id = si.user_id
    JOIN athletic_brand_sales s
        ON s.emailsha256 = su.emailsha256
    WHERE s.timestamp > si.timestamp

UNION ALL

    SELECT si.user_id, si.campaign_id, s.sale_id, s.sale_price
    FROM socialco_impressions si
    JOIN socialco_users su
        ON su.user_id = si.user_id
    JOIN athletic_brand_sales s
        ON s.phonesha256 = su.phonesha256
    WHERE s.timestamp > si.timestamp
)

SELECT COUNT (DISTINCT user_id) as unique_users
FROM matches_table
GROUP BY campaign_id
ORDER BY COUNT (DISTINCT user_id) DESC
LIMIT 5
```

Similarly, if you want to run window functions on differential privacy protected data tables, you must include the user identifier column in the `PARTITION BY` clause.

```
ROW_NUMBER() OVER (PARTITION BY conversion_id, user_id ORDER BY match_type, match_age)
 AS row
```

Limitations of AWS Clean Rooms Differential Privacy

AWS Clean Rooms Differential Privacy does not address the following situations:

1. AWS Clean Rooms Differential Privacy does not address timing attacks. For example, these attacks are possible in scenarios where an individual user contributes a large number of rows and adding or removing this user significantly changes the query computation time.
2. AWS Clean Rooms Differential Privacy does not guarantee differential privacy when a SQL query can result in overflow or invalid cast errors at run time due to the use of certain SQL constructs. The following table is a list of some, but not all, SQL constructs that may produce run-time

errors and should be verified in analysis templates. We recommend that you approve analysis templates that minimize the chances of such run-time errors and periodically review query logs to determine if the queries align with the collaboration agreement.

The following SQL constructs are vulnerable to overflow errors:

- Aggregate functions - AVG, LISTAVG, PERCENTILE_COUNT, PERCENTILE_DISC, SUM/ SUM_DISTINCT
- Data type formatting functions - TO_TIMESTAMP, TO_DATE
- Date and time functions - ADD_MONTHS, DATEADD, DATEDIFF
- Math functions - +, -, *, /, POWER
- String functions - ||, CONCAT, REPEAT, REPLICATE
- Window functions - AVG, LISTAGG, PERCENTILE_COUNT, PERCENTILE_DISC, RATIO_TO_REPORT, SUM

The CAST data type formatting function is vulnerable to invalid cast errors.

AWS Clean Rooms ML

AWS Clean Rooms ML

AWS Clean Rooms ML provides a privacy-preserving method for two parties to identify similar users in their data without the need to share their data with each other. The first party brings the training data to AWS Clean Rooms so that they can create and configure a lookalike model and associate it with a collaboration. The second party then brings their seed data to AWS Clean Rooms and generates a lookalike segment that resembles the training data.

For a more detailed explanation of how this works, see Cross-account jobs.

- *Training data provider* – The party that contributes the training data, creates and configures a lookalike model, and then associates that lookalike model with a collaboration.
- *Seed data provider* – The party that contributes the seed data, generates a lookalike segment, and exports their lookalike segment.
- *Training data* – The training data provider's data, which is used to generate a lookalike model. The training data is used to measure similarity in user behaviors.

  The training data must contain a user ID, item ID, and timestamp column. Optionally, the training data can contain other interactions as numerical or categorical features. Examples of interactions are a list of videos watched, items purchased, or articles read.
- *Seed data* – The seed data provider's data, which is used to create a lookalike segment. The lookalike segment output is a set of users from the training data that most closely resembles the seed users.
- *Lookalike model* – A machine learning model of the training data that is used to find similar users in other datasets.

  When using the API, the term *audience model* is used equivalently to lookalike model. For example, you use the CreateAudienceModel API to create a lookalike model.
- *Lookalike segment* – A subset of the training data that most closely resembles the seed data.

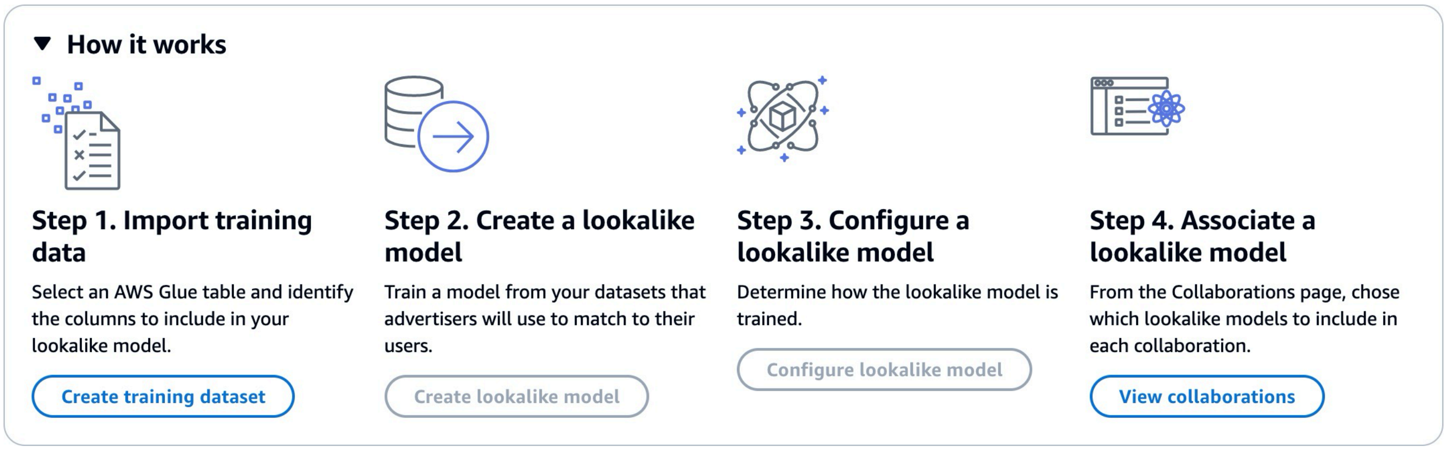
  When using the API, you create a lookalike segment with the StartAudienceGenerationJob API.

The training data provider's data is never shared with the seed data provider and the seed data provider's data is never shared with the training data provider. The lookalike segment output is shared with the training data provider, but never the seed data provider.

For more information about lookalike models, see the following topics.

Topics

How AWS Clean Rooms ML works



Clean Rooms ML requires that two parties, a training data provider and a seed data provider, work sequentially in AWS Clean Rooms to bring their data into a collaboration. This is the workflow that the training data provider must complete first:

1. The training data provider's data must be stored in a AWS Glue data catalog table of user-item interactions. At a minimum, the training data must contain a user ID column, interaction ID column, and a timestamp column.
2. The training data provider registers the training data with AWS Clean Rooms.
3. The training data provider creates a lookalike model that can be shared with multiple seed data providers. The lookalike model is a deep neural network that can take up to 24 hours to train. It is not automatically retrained and we recommend that you retrain the model weekly.
4. The training data provider configures the lookalike model, including whether to share relevance metrics and the Amazon S3 location of the output segments. The training data provider can create multiple configured lookalike models from a single lookalike model.
5. The training data provider associates the configured audience model to a collaboration that is shared with a seed data provider.

This is the workflow that the seed data provider must complete next:

1. The seed data provider's data must be stored in an Amazon S3 bucket.
2. The seed data provider opens the collaboration that they share with the training data provider.
3. The seed data provider creates a lookalike segment from the Clean Rooms ML tab of the collaboration page.
4. The seed data provider can evaluate the relevance metrics, if they were shared, and export the lookalike segment for use outside AWS Clean Rooms.

Privacy protections of AWS Clean Rooms ML

Clean Rooms ML is designed to reduce the risk of *membership inference attacks* where the training data provider can learn who is in the seed data and the seed data provider can learn who is in the training data. Several steps are taken to prevent this attack.

First, seed data providers do not directly observe the Clean Rooms ML output and training data providers can never observe the seed data. Seed data providers can choose to include the seed data in the output segment.

Next, the lookalike model is created from a random sample of the training data. This sample includes a significant number of users that do not match the seed audience. This process makes it harder to determine whether a user was not in the data, which is another avenue for membership inference.

Further, multiple seed customers can be used for every parameter of seed-specific lookalike model training. This limits how much the model can overfit, and thus how much can be inferred about a user. As a result, we recommend that the minimum size of the seed data is 500 users.

Finally, user-level metrics are never provided to training data providers, which eliminates another avenue for a membership inference attack.

Training data requirements for Clean Rooms ML

To successfully create a lookalike model, your training data must meet the following requirements:

- The training data must be in Parquet, CSV, or JSON format.

- Your training data must be cataloged in AWS Glue. For more information, see Getting started with the AWS Glue Data Catalog in the AWS Glue Developer Guide. We recommend using AWS Glue crawlers to create your tables because the schema is inferred automatically.
- The Amazon S3 bucket that contains the training data and seed data is in the same AWS region as your other Clean Rooms ML resources.
- The training data must contain at least 100,000 unique user IDs with at least two item interactions each.
- The training data must contain at least 1 million records.

The following table describes the supported data types for each data field in the training data.

Field type	Supported data types	Required	Description
USER_ID	string, int, bigint	Yes	A unique identifier for each user in the dataset. It should be a non-PII (Personally Identifiable Information) value. This might be a hashed identifier or a customer ID.
ITEM_ID	string, int, bigint	Yes	A unique identifier for each item a user

Field type	Supported data types	Required	Description
			interacts with.
TIMESTAMP	bigint, int, timestamp	Yes	The time when a user interacted with the item. Values must be in the Unix epoch time in seconds format.

Field type	Supported data types	Required	Description
CATEGORICAL_FEATURE	string, int, float, bigint, double, boolean, array	No	Captures categorical data related to the user or the item. This can include things like an event type (click, purchase, etc), user demographics (age group, gender - anonymized), user location (city, country - anonymized), item category (clothing, electronics, etc), or item brand.

Field type	Supported data types	Required	Description
NUMERICAL_FEATURE	double, float, int, bigint	No	Captures numerical data related to the user or the item. This can include things like user purchase history (total amount spent), item price, number of times an item is visited, or user ratings for items.

Here is an example of a valid training data set in CSV format

```
USER_ID,ITEM_ID,TIMESTAMP, EVENT_TYPE(CATEGORICAL FEATURE),EVENT_VALUE (NUMERICAL FEATURE)
196,242,881250949,click,15
186,302,891717742,click,13
22,377,878887116,click,10
244,51,880606923,click,20
166,346,886397596,click,10
```

Seed data requirements for Clean Rooms ML

The seed data must meet the following requirements:

- The seed data must be in JSON lines format with a list of User IDs.
- The seed size should be between 25 and 500,000 unique user IDs.
- The minimum number of seed users must match the minimum matching seed size value that was specified when you created the configured audience model.

Here is an example of a valid training data set in CSV format

```
      {"user_id": "abc"}
{"user_id": "def"}
{"user_id": "ghijkl"}
{"user_id": "123"}
{"user_id": "456"}
{"user_id": "7890"}
```

AWS Clean Rooms ML model evaluation metrics

Clean Rooms ML computes the *recall* and *relevance score* to determine how well your model performs. Recall compares the similarity between the lookalike data and training data. The relevance score is used to decide how large the audience should be, not whether the model is well-performing.

Recall is an unbiased measure of how similar the lookalike segment is to the training data. Recall is the percentage of the most similar users (by default, the most similar 20%) from a sample of the training data that are included in the seed audience by the audience generation job. Values range from 0-1, larger values indicate a better audience. A recall value approximately equal to the maximum bin percentage indicates that the audience model is equivalent to random selection.

We consider this a better evaluation metric than accuracy, precision, and F1 scores because Clean Rooms ML does not have accurately labeled true negative users when building its model.

Segment-level *relevance score* is a measure of similarity with values ranging from -1 (least similar) to 1 (most similar). Clean Rooms ML computes a set of relevance scores for various segment sizes to help you determine the best segment size for your data. Relevance scores monotonically decrease as the segment size increases, thus as the segment size increases it can be less similar to the seed data. When the segment-level relevance score reaches 0, the model predicts that all users in the lookalike segment are from the same distribution as the seed data. Increasing the output size is likely to include users in the lookalike segment that are not from the same distribution as the seed data.

Relevance scores are normalized within a single campaign and should not be used to compare across campaigns. Relevancy scores should not be used as a single-sourced evidence for any business outcome because those are impacted by multiple complex factors in addition to relevance, such as inventory quality, inventory type, timing of advertising, and so on.

Relevance scores should not be used to judge the quality of the seed, but rather if it can be increased or decreased. Consider the following examples:

- All positive scores – This indicates that there are more output users that are predicted as similar than are included in the lookalike segment. This is common for seed data that is part of a large market, such as everybody who has bought toothpaste in the past month. We recommend looking at smaller seed data, such as everybody who has bought toothpaste more than once in the past month.
- All negatives scores or negative for your desired lookalike segment size – This indicates that Clean Rooms ML predicts there are not enough similar users in the desired lookalike segment size. This can be because the seed data is too specific or the market is too small. We recommend either applying fewer filters to the seed data or widening the market. For example, if the original seed data was customers that bought a stroller and car seat, you could expand the market to customers that bought multiple baby products.

Training data providers determine whether the relevance scores are exposed and the bucket bins where relevance scores are computed.

Working with AWS Clean Rooms ML

A *lookalike model* is a model of a training data provider's data that allows a seed data provider to create a lookalike segment of training data provider's data that most closely resembles their seed data. To create a lookalike model that can be used in a collaboration, you must import your training data, create a lookalike model, configure that lookalike model, and then associate it to a collaboration.

After the training data provider is done creating the ML model, the seed data provider can create and export the seed segment.

Topics

- Working with lookalike models (training data provider)
- Working with lookalike segments (seed data provider)
- Next steps

Working with lookalike models (training data provider)

Import training data

Before you create a lookalike model, you must specify the AWS Glue table that contains the training data. Clean Rooms ML does not store a copy of this data, just metadata that allows it to access the data.

To import training data in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **ML Modeling**.
3. On the **Training datasets** tab, choose **Create training dataset**.
4. Enter a **Name** and optional **Description**.
5. For **Data source**, choose your AWS Glue table:

   a. Choose the **Database** that you want to configure from the dropdown list.

   b. Choose the **Training data source** by selecting the **Database** and **Table** that you want to configure from the dropdown lists.

> **Note**
>
> To verify that this is the correct table, do either one of the following:
>
> - Choose **View in AWS Glue**.
> - Turn on **View schema** to view the schema.

6. For **Training details**, choose the **User identifier column**, **Item identifier column**, and **Timestamp column** from your data. The training data must contain these three columns. You can also select any other columns that you want to include in the training data.

   The data in the **Timestamp column** must be in the Unix epoch time in seconds format.

7. In **Service access**, you must specify a service role that can access your data and provide a KMS key if your data is encrypted. Choose **Create and use a new service role** and Clean Rooms ML will automatically create a service role and add the necessary permissions policy. Choose **Use an existing service role** and enter it in the **Service role name** field if you have a specific service role that you want to use.

   If your data is encrypted, enter your KMS key in the **AWS KMS key** field, or click **Create an AWS KMS key** to generate a new KMS key.

8. If you want to enable **Tags** for the training dataset, choose **Add new tag** and then enter the **Key** and **Value** pair.
9. Choose **Create training dataset**.

For the corresponding API action, see CreateTrainingDataset.

Create a lookalike model

After you have created a training dataset, you are ready to create a lookalike model. You can create many lookalike models from a single training dataset.

You must create a default database in your AWS Glue Data Catalog or include the `glue:createDatabase` permission in the provided role.

To create a lookalike model in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **ML Modeling**.
3. On the **Lookalike models** tab, choose **Create lookalike model**.
4. For **Create lookalike model**, for **Lookalike model details**:

   a. Enter a **Name** and optional **Description**.
   b. Choose the **Training dataset** that you want to model from the dropdown list.

   c. Enter an optional **Training window**.

5. If you want to enable custom encryption settings for the lookalike model, choose **Customize encryption settings** and then enter the KMS key.
6. If you want to enable **Tags** for the lookalike model, choose **Add new tag** and then enter the **Key** and **Value** pair.
7. Choose **Create lookalike model**.

For the corresponding API action, see CreateAudienceModel.

Configure a lookalike model

After you have created a lookalike model, you are ready to configure it for use in a collaboration. You can create multiple configured lookalike models from a single lookalike model.

To configure a lookalike model in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **ML Modeling**.
3. On the **Configured lookalike models** tab, choose **Configure lookalike model**.
4. For **Configure lookalike model**, for **Configured lookalike model details**:
   a. Enter a **Name** and optional **Description**.
   b. Choose the **Lookalike model** that you want to configure from the dropdown list.
   c. Choose the **Minimum matching seed size** that you want. This is the minimum number of users in the seed data provider's data that overlap with users in the training data. This value must be greater than 0.
5. For **Metrics to share with other members**, choose whether you want the seed data provider in your collaboration to receive model metrics, including relevance scores.
6. For **Lookalike segment destination location**, enter the Amazon S3 bucket where lookalike segment is exported. This bucket must be located in the same region as your other resources.
7. For **Service access**, choose the **Existing service role name** that will be used to access this table.
8. Choose **Configure Lookalike Model**.

9. If you want to enable **Tags** for the configured table resource, choose **Add new tag** and then enter the **Key** and **Value** pair.

For the corresponding API action, see CreateConfiguredAudienceModel.

Associate a configured lookalike model

After you have configured a lookalike model, you can associate it to a collaboration.

To associate a configured lookalike model in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. On the **With active membership** tab, choose a collaboration.
4. On the **ML Modeling** tab, choose **Associate lookalike model**.
5. For **Associate configured lookalike model**, for **Associate lookalike model details**:
   a. Enter a **Name** for the associated configured audience model.
   b. Enter a **Description** of the table.

      The description helps differentiate between other associated configured audience models with similar names.
6. For **Configured lookalike model**, choose a configured lookalike model from the dropdown list.
7. Choose **Associate**.

For the corresponding API action, see CreateConfiguredAudienceModelAssociation.

Update a configured lookalike model

After you have associated a configured a lookalike model, you can update it to change information such as the name, metrics to share, or output Amazon S3 location.

To update an associated configured lookalike model in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **ML modeling**.

3. On the **Configured lookalike models** tab, choose a configured lookalike model and select **Edit**.
4. For **Configure lookalike model**, for **Configured lookalike model details**:

   a. Choose the **Lookalike model** that you want configured from the dropdown list.

   b. Choose the **Minimum matching seed size** that you want. This is the minimum number of users in the seed data provider's data that overlap with users in the training data. This value must be greater than 0.
5. For **Metrics to share with other members**, choose whether you want the seed data provider in your collaboration to receive model metrics, including relevance scores.
6. For **Lookalike segment destination location**, enter the Amazon S3 bucket where lookalike segment is exported. This bucket must be located in the same region as your other resources.
7. For **Service access**, choose the **Existing service role name** that will be used to access this table.
8. For **Advanced bin size configuration**, choose how you want to configure the audience bin sizes.
9. Choose **Save changes**.

For the corresponding API action, see UpdateConfiguredAudienceModel.

Working with lookalike segments (seed data provider)

Create a lookalike segment

A lookalike segment is a subset of the training data that most closely resembles the seed data.

To create a lookalike segment in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. On the **With active membership** tab, choose a collaboration.
4. On the **ML Modeling** tab, choose **Create lookalike segment**.
5. For **Create lookalike segment**, for **Lookalike segment details** enter a **Name** and optional **Description**.
6. For **Seed profiles**, choose the **Amazon S3 input source** where your seed data is stored.

7. For **Service access**, choose the **Existing service role name** that will be used to access this table.
8. If you want to enable **Tags** for the training dataset, choose **Add new tag** and then enter the **Key** and **Value** pair.
9. Choose **Create lookalike segment**.

For the corresponding API action, see StartAudienceGenerationJob.

Export a lookalike segment

After you have created a lookalike segment, you can export that data to an Amazon S3 bucket.

To export a lookalike segment in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. On the **With active membership** tab, choose a collaboration.
4. On the **ML Modeling** tab, select a lookalike segment and choose **Export**.
5. For **Export lookalike model**, for **Export lookalike model details** enter a **Name** and optional **Description**.
6. For **Segment size**, choose the size you want for the exported segment.
7. Choose **Export**.

For the corresponding API action, see StartAudienceExportJob.

Next steps

Now that you have created a lookalike model and exported a seed segment, you are ready to:

- Manage AWS Clean Rooms

Cryptographic Computing for Clean Rooms

Cryptographic Computing for Clean Rooms (C3R) is a capability in AWS Clean Rooms that can be used in addition to analysis rules. With C3R, organizations can bring sensitive data together to derive new insights from data analytics while cryptographically limiting what can be learned by any party in the process. C3R can be used by two or more parties that want to collaborate with their sensitive data but are required to only use encrypted data in the cloud.

The C3R encryption client is a client-side encryption tool that you can use to encrypt your data for use with AWS Clean Rooms. When you use the C3R encryption client, data remains cryptographically protected while in use in an AWS Clean Rooms collaboration. As with a regular AWS Clean Rooms collaboration, the input data is relational database tables, and the computation is expressed as a SQL query. However, C3R only supports a limited subset of SQL queries on encrypted data.

Specifically, C3R supports SQL JOIN and SELECT statements on cryptographically protected data. Each column in the input table can be used in exactly one of the following SQL statement types:

- Columns that are cryptographically protected for use in JOIN statements are called **fingerprint columns**.
- Columns that are cryptographically protected for use in SELECT statements are called **sealed columns**.
- Columns that are not cryptographically protected for use in JOIN or SELECT statements are called **cleartext columns**.

In some cases, GROUP BY statements are supported on fingerprint columns. For more information, see Fingerprint columns. Currently, C3R doesn't support the use of other SQL constructs on encrypted data, such as WHERE clauses or aggregate functions like SUM and AVERAGE, even if they would otherwise be allowed by the relevant analysis rules.

C3R is designed to protect data in individual cells of a table. Using the default configuration for C3R, the underlying data that a customer makes available to third parties through a collaboration remains encrypted while the content is in use within AWS Clean Rooms. C3R uses industry standard AES-GCM encryption for all sealed columns and an industry standard pseudorandom function, known as a Hash-based Message Authentication Code (HMAC), for protecting fingerprint columns.

Although C3R encrypts the data in your tables, the following information might still be able to be inferred:

- Information about the tables themselves, including the number of columns, column names, and the number of rows in your table.
- As with most standard forms of encryption, C3R doesn't try to hide the length of encrypted values. C3R does offer the ability to pad encrypted values to hide the exact length of cleartexts. However, an upper bound on the length of the cleartexts in each column could still be revealed to another party.
- Logging-level information, such as when a particular row was added to an encrypted C3R table.

For more information about C3R, see the following topics.

Topics

- Considerations when using Cryptographic Computing for Clean Rooms
- Supported file and data types in Cryptographic Computing for Clean Rooms
- Column names in Cryptographic Computing for Clean Rooms
- Column types in Cryptographic Computing for Clean Rooms
- Cryptographic computing parameters
- Optional flags in Cryptographic Computing for Clean Rooms
- Queries with Cryptographic Computing for Clean Rooms
- Guidelines for the C3R encryption client

Considerations when using Cryptographic Computing for Clean Rooms

Cryptographic Computing for Clean Rooms (C3R) seeks to maximize data protection. However, some use cases might benefit from lower levels of data protection in exchange for additional functionality. You can make these specific tradeoffs by modifying C3R from its most secure configuration. As the customer, you should be aware of these tradeoffs and determine if they are appropriate for your use case. Tradeoffs to consider include the following:

Topics

- Allowing mixed cleartext and encrypted data in your tables

- Allowing repeated values in fingerprint columns
- Loosening restrictions on how fingerprint columns are named
- Determining how NULL values are represented

For more information about how to set parameters for these scenarios, see Cryptographic computing parameters.

Allowing mixed cleartext and encrypted data in your tables

Having all data be client-side encrypted provides maximum data protection. However, this limits certain kinds of queries (for example, the SUM aggregate function). The risk of allowing cleartext data is that it's feasible that anyone with access to the encrypted tables could infer some information about encrypted values. This could be done by performing a statistical analysis on the cleartext and associated data.

For example, imagine you had the columns of `City` and `State`. The `City` column is cleartext and the `State` column is encrypted. When you see the value `Chicago` in the `City` column, that helps you determine with high probability that the `State` is `Illinois`. In contrast, if one column is `City` and the other column is `EmailAddress`, a cleartext `City` is unlikely to reveal anything about an encrypted `EmailAddress`.

For more information about the parameter for this scenario, see Allow cleartext columns parameter.

Allowing repeated values in fingerprint columns

For the most secure approach, we assume that any fingerprint column contains exactly one instance of a variable. No item can be repeated in a fingerprint column. The C3R encryption client maps these cleartext values into unique values that are indistinguishable from random values. Therefore, it's impossible to infer information about the cleartext from these random values.

The risk of repeated values in a fingerprint column is that repeated values will result in repeated random-looking values. Thus, anyone who has access to the encrypted tables could, in theory, perform a statistical analysis of the fingerprint columns that might reveal information about cleartext values.

Again, suppose the fingerprint column is `State`, and every row of the table corresponds to a US household. By doing a frequency analysis, one could infer which state is `California` and which is `Wyoming` with high probability. This inference is possible because `California` has many more residents than `Wyoming`. In contrast, say the fingerprint column is on a household identifier and each household appeared in the database between 1 and 4 times in a database of millions of entries. It's unlikely that a frequency analysis would reveal any useful information.

For more information about the parameter for this scenario, see Allow duplicates parameter.

Loosening restrictions on how fingerprint columns are named

By default, we assume that when two tables are joined using encrypted fingerprint columns, those columns have the same name in each table. The technical reason for this result is that, by default, we derive a different cryptographic key for encrypting each fingerprint column. That key is derived from a combination of the shared secret key for the collaboration and the column name. If we try to join two columns with different column names, we derive different keys and we can't compute a valid join.

To address this issue, you can turn off the feature that derives keys from each column name. Then, the C3R encryption client uses a single derived key for all fingerprint columns. The risk is that another kind of frequency analysis can be done that might reveal information.

Let's use the `City` and `State` example again. If we derive the same random values for each fingerprint column (by not incorporating the column name). `New York` has the same random value in the `City` and `State` columns. New York is one of a few cities in the US where the `City` name is the same as the `State` name. In contrast, if your dataset has completely different values in each column, no information is leaked.

For more information about the parameter for this scenario, see Allow JOIN of columns with different names parameter.

Determining how NULL values are represented

The option available to you is whether to process cryptographically (encrypt and HMAC) NULL values like any other value. If you don't process NULL values like any other value, information might be revealed.

For example, suppose that NULL in the `Middle Name` column in the cleartext indicates people without middle names. If you don't encrypt those values, you leak which rows in the encrypted table are used for people without middle names. That information might be an identifying signal for some people in some populations. But if you do cryptographically process NULL values, certain

SQL queries act differently. For example, GROUP BY clauses will not group fingerprint NULL values in fingerprint columns together.

For more information about the parameter for this scenario, see Preserve NULL values parameter.

Supported file and data types in Cryptographic Computing for Clean Rooms

The C3R encryption client recognizes the following file types:

- CSV files
- Parquet files

You can use the `--fileFormat` flag in the C3R encryption client to specify a file format explicitly. When explicitly specified, file format is not determined by file extension.

Topics

CSV files

A file with a .csv extension is assumed to be CSV formatted and contain UTF-8 encoded text. The C3R encryption client treats all values as strings.

Supported properties in .csv files

The C3R encryption client requires that .csv files have the following properties:

- Might or might not contain an initial header row that uniquely names each column.
- Comma-delimited. (Currently, custom delimiters are not supported.)
- UTF-8 encoded text.

White space trimming from .csv entries

Both leading and trailing white space is trimmed from .csv entries.

Custom NULL encoding for a .csv file

A .csv file can use custom NULL encoding.

With the C3R encryption client, you can specify custom encodings for NULL entries in the input data by using the `--csvInputNULLValue=<csv-input-null>` flag. The C3R encryption client can use custom encodings in the generated output file for NULL entries by using the `--csvOutputNULLValue=<csv-output-null>` flag.

> **Note**
>
> A NULL entry is considered to be *lacking* content, specifically in the context of a richer tabular format like an SQL table. Although .csv doesn't explicitly support this characterization for historical reasons, it's a common convention to consider an empty entry that contains only white space to be NULL. Therefore, that's the default behavior of the C3R encryption client and it can be customized as needed.

How .csv entries are interpreted by C3R

The following table provides examples of how .csv entries are marshalled (cleartext to cleartext for clarity) based on the values (if any) that are provided for the `--csvInputNULLValue=<csv-input-null>` and `--csvOutputNULLValue=<csv-output-null>` flags. Leading and trailing white space outside of quotes is trimmed before C3R interprets any value's meaning.

`<csv-input-null>`	`<csv-output-null>`	Input entry	Output entry
None	None	`,AnyProduct,`	`,AnyProduct,`
None	None	`, AnyProduct ,`	`,AnyProduct,`
None	None	`,"AnyProduct",`	`,AnyProduct,`
None	None	`, "AnyProduct" ,`	`,AnyProduct,`
None	None	`,,`	`,,`
None	None	`, ,`	`,,`

`<csv-input-null>`	**`<csv-output-null>`**	**Input entry**	**Output entry**
None	None	`,"",`	`,,`
None	None	`," ",`	`," ",`
None	None	`, " " ,`	`," ",`
`"AnyProduct"`	`"NULL"`	`,AnyProduct,`	`,NULL,`
`"AnyProduct"`	`"NULL"`	`, AnyProduct ,`	`,NULL,`
`"AnyProduct"`	`"NULL"`	`,"AnyProduct",`	`,NULL,`
`"AnyProduct"`	`"NULL"`	`, "AnyProduct" ,`	`,NULL,`
None	`"NULL"`	`,,`	`,NULL,`
None	`"NULL"`	`, ,`	`,NULL,`
None	`"NULL"`	`,"",`	`,NULL,`
None	`"NULL"`	`," ",`	`," ",`
None	`"NULL"`	`, " " ,`	`," ",`
`""`	`"NULL"`	`,,`	`,NULL,`
`""`	`"NULL"`	`, ,`	`,NULL,`
`""`	`"NULL"`	`,"",`	`,"",`
`""`	`"NULL"`	`," ",`	`," ",`
`""`	`"NULL"`	`, " " ,`	`," ",`
`"\"\""`	`"NULL"`	`,,`	`,,`
`"\"\""`	`"NULL"`	`, ,`	`,,`

<csv-input-null>	**<csv-output-null>**	**Input entry**	**Output entry**
`"\"\""`	`"NULL"`	`,"",`	`,NULL,`
`"\"\""`	`"NULL"`	`," ",`	`," ",`
`"\"\""`	`"NULL"`	`, " " ,`	`," ",`

CSV file without headers

The source .csv file doesn't need to have headers in the first row that uniquely name each column. However, a .csv file without a header row requires a positional encryption schema. The positional encryption schema is required instead of the typical mapped schema that's used for both .csv files with a header row and Parquet files.

A positional encryption schema specifies output columns by position instead of by name. A mapped encryption schema maps source column names to target column names. For more information, including a detailed discussion and examples of both schema formats, see Mapped and positional table schemas.

Parquet files

A file with a .parquet extension is assumed to be in the Apache Parquet format.

Supported Parquet data types

The C3R encryption client can process any non-complex (that is, primitive type) data in a Parquet file that represents a data type supported by AWS Clean Rooms.

However, only string columns can be used for sealed columns.

The following Parquet data types are supported:

- `Binary` primitive type with the following logical annotations:
  - None if the `--parquetBinaryAsString` is set (`STRING` data type)
  - `Decimal(scale, precision)` (`DECIMAL` data type)
  - `String` (`STRING` data type)
- `Boolean` primitive data type with no logical annotation (`BOOLEAN` data type)

- `Double` primitive data type with no logical annotation (`DOUBLE` data type)
- `Fixed_Len_Binary_Array` primitive type with the `Decimal(scale, precision)` logical annotation (`DECIMAL` data type)
- `Float` primitive data type with no logical annotation (`FLOAT` data type)
- `Int32` primitive type with the following logical annotations:
  - None (`INT` data type)
  - `Date` (`DATE` data type)
  - `Decimal(scale, precision)` (`DECIMAL` data type)
  - `Int(16, true)` (`SMALLINT` data type)
  - `Int(32, true)` (`INT` data type)
- `Int64` primitive data type with the following logical annotations:
  - None (`BIGINT` data type)
  - `Decimal(scale, precision)` (`DECIMAL` data type)
  - `Int(64, true)` (`BIGINT` data type)
  - `Timestamp(isUTCAdjusted, TimeUnit.MILLIS)` (`TIMESTAMP` data type)
  - `Timestamp(isUTCAdjusted, TimeUnit.MICROS)` (`TIMESTAMP` data type)
  - `Timestamp(isUTCAdjusted, TimeUnit.NANOS)` (`TIMESTAMP` data type)

Encrypting non-string values

Currently, only string values are supported for sealed columns.

For .csv files, the C3R encryption client treats all values as UTF-8 encoded text and makes no attempt to interpret them differently before encryption.

For fingerprint columns, types are grouped into equivalence classes. An equivalence class is a set of data types that can be unambiguously compared for equality via a representative data type.

Equivalence classes allow identical fingerprints to be assigned to the same semantic value regardless of the original representation. However, the same value in two equivalence classes will not result in the same fingerprint column.

For example, the `INTEGRAL` value `42` will be assigned the same fingerprint regardless of whether it was originally an `SMALLINT`, `INT`, or `BIGINT`. Also, the `INTEGRAL` value `0` will never match the `BOOLEAN` value `FALSE` (which is represented by the value `0`).

The following equivalence classes and corresponding AWS Clean Rooms data types are supported by fingerprint columns:

Equivalence class	Supported AWS Clean Rooms data type
`BOOLEAN`	`BOOLEAN`
`DATE`	`DATE`
`INTEGRAL`	`BIGINT`, `INT`, `SMALLINT`
`STRING`	`CHAR`, `STRING`, `VARCHAR`

Column names in Cryptographic Computing for Clean Rooms

By default, the names of columns are important in Cryptographic Computing for Clean Rooms.

If the value of the **Allow JOIN of columns with different names** parameter is **false**, column names are used during the encryption of fingerprint columns. For this reason, by default, collaborators must coordinate in advance and use the same target column names for data that will use JOIN statements in queries. By default, columns encrypted for JOIN with different names don't successfully JOIN on any values.

If the value of the **Allow JOIN of columns with different names** parameter is **true**, JOIN statements across columns encrypted as fingerprint columns succeed. Encrypting data with this parameter might allow some inference of the cleartext values. For example, if a row has the same Hash-based Message Authentication Code (HMAC) value in both the `City` column and `State` column, the value might be `New York`.

Normalization of column header names

Column header names are normalized by the C3R encryption client. Any leading and trailing white space is removed, and the column name is made lowercase for the transformed output.

Normalization is applied before all other computations, calculations, or other operations which could possibly be impacted by column names. The emitted output file only contains the normalized names.

Column types in Cryptographic Computing for Clean Rooms

This topic provides information about column types in Cryptographic Computing for Clean Rooms.

Topics

- [Fingerprint columns]
- [Sealed columns]
- [Cleartext columns]

Fingerprint columns

Fingerprint columns are columns that are protected cryptographically for use in JOIN statements.

Data from fingerprint columns can't be decrypted. Only data from sealed columns can be decrypted.

Fingerprint columns must only be used in the following SQL clauses and functions:

- JOIN (INNER, OUTER, LEFT, RIGHT, or FULL) against other fingerprint columns:
  - If the value of the `allowJoinsOnColumnsWithDifferentNames` parameter is set to `false`, both fingerprint columns of the JOIN must also have the same name.
- `SELECT COUNT()`
- `SELECT COUNT(DISTINCT )`
- `GROUP BY` (Only use if the collaboration has set the value of the `preserveNulls` parameter to `true`.)

Queries that violate these constraints might yield incorrect results.

Sealed columns

Sealed columns are columns that are protected cryptographically for use in SELECT statements.

Sealed columns must only be used in the following SQL clauses and functions:

- `SELECT`
- `SELECT ... AS`
- `SELECT COUNT()`

> **Note**
>
> `SELECT COUNT(DISTINCT )` is not supported.

Queries that violate these constraints might yield incorrect results.

Padding data for a sealed column before encryption

When you specify that a column should be a sealed column, C3R asks you what kind of *padding* to choose. Padding data before encryption is optional. Without padding (a pad type of `none`), the encrypted data's length indicates the size of the cleartext. In some circumstances, the size of the cleartext could expose the plaintext. With padding (a pad type of `fixed` or `max`), all values are first padded to a common size and then encrypted. With padding, the length of the encrypted data provides no information about the original cleartext length, other than giving an upper bound on its size.

If you want padding for a column and the maximal byte length of data in that column is known, use `fixed` padding. Use a `length` value that is at least as large as the byte-length of the longest value in that column.

> **Note**
>
> An error occurs and encryption fails if a value is longer than the provided `length`.

If you want padding for a column and the maximal byte length of data in that column isn't known, use `max` padding. This padding mode pads all data to the length of the longest value plus additional `length` bytes.

> **Note**
>
> You might want to encrypt data in batches, or update your tables with new data periodically. Be aware that `max` padding will pad entries to the length (plus `length` byte) of the longest plaintext entry in a given batch. This means that the ciphertext length may vary from batch to batch. Therefore, if you know the maximum byte-length for a column, then you should use `fixed` instead of `max`.

Cleartext columns

Cleartext columns are columns that aren't protected cryptographically for use in JOIN or SELECT statements.

Cleartext columns can be used in any part of the SQL query.

Cryptographic computing parameters

Cryptographic computing parameters are available for collaborations using Cryptographic Computing for Clean Rooms (C3R) when creating a collaboration. You can create a collaboration using either the AWS Clean Rooms console or the `CreateCollaboration` API operation. In the console, you can set values for the parameters in **Cryptographic computing parameters** after you turn on the **Support cryptographic computing** option. For more information, see the following topics.

Topics

- Allow cleartext columns parameter
- Allow duplicates parameter
- Allow JOIN of columns with different names parameter
- Preserve NULL values parameter

Allow cleartext columns parameter

In the console, you can set the **Allow cleartext columns** parameter when creating a collaboration to specify if cleartext data is allowed in a table with encrypted data.

The following table describes the values for the **Allow cleartext columns** parameter.

Parameter value	Description
No	Cleartext columns aren't allowed in the encrypted table. All data is cryptographically protected.

Parameter value	Description
Yes	Cleartext columns are allowed in the encrypted table. Cleartext columns are not cryptographically protected and are included as cleartext. You should take note of what your rows' cleartext data might reveal about the other data in the table. To run SUM or AVG on specific columns, the columns must be in cleartext.

Using the `CreateCollaboration` API operation, for the `dataEncryptionMetadata` parameter, you can set the value of `allowCleartext` to `true` or `false`. For more information about API operations, see the AWS Clean Rooms API Reference.

Cleartext columns correspond to columns that are classified as cleartext in the table-specific schema. Data in these columns is not encrypted and can be used in any way. Cleartext columns can be useful if the data is not sensitive and/or if more flexibility is needed than an encrypted sealed column or fingerprint column allows.

Allow duplicates parameter

In the console, you can set the **Allow duplicates** parameter when creating a collaboration to specify if columns encrypted for JOIN queries can contain duplicate non-NULL values.

> **Important**
>
> The **Allow duplicates**, **Allow JOIN of columns with different names**, and **Preserve NULL values** parameters have separate but related effects.

The following table describes the values for the **Allow duplicates** parameter.

Parameter value	Description
No	Repeated values are not allowed in a fingerprint column. All values in a single fingerprint column must be unique.
Yes	Repeated values are allowed in a fingerprint column.

Parameter value	Description
	If you need to join columns with repeated values, set this value to **Yes**. When set to **Yes**, frequency patterns appearing within fingerprint columns in the C3R table or results might imply some additional information about the structure of the cleartext data.

Using the `CreateCollaboration` API operation, for the `dataEncryptionMetadata` parameter you can set the value of `allowDuplicates` to `true` or `false`. For more information about API operations, see the AWS Clean Rooms API Reference.

By default, if encrypted data must be used in JOIN queries, the C3R encryption client requires that those columns have no duplicate values. This requirement is an effort to increase data protection. This behavior can help ensure that repeated patterns in the data are not observable. However, if you want to work with encrypted data in JOIN queries and aren't concerned about duplicate values, the **Allow duplicates** parameter can disable this conservative check.

Allow JOIN of columns with different names parameter

In the console, you can set the **Allow JOIN of columns with different names** parameter when creating a collaboration to specify if JOIN statements between columns with different names are supported.

For more information, see Normalization of column header names

The following table describes the values for the **Allow JOIN of columns with different names** parameter.

Parameter value	Description
No	Joins of fingerprint columns with different names are not supported. JOIN statements only provide accurate results on columns that have the same name. **Important** The **No** value provides increased information security but requires collaboration participants to agree

Parameter value	Description
	previously about column names. If two columns have different names when encrypted as fingerprint columns and **Allow JOIN of columns with different names** is set to **No**, JOIN statements on those columns produce no results. This is because no values post-encryption are shared between them.
Yes	Joins of fingerprint columns with different names are supported. For additional flexibility, users can set this value to **Yes**, which allows JOIN statements on columns regardless of their column name. If set to **Yes**, the C3R encryption client doesn't consider the column name when protecting fingerprint columns. As a result, common values across different fingerprint columns are observable in the C3R table. For example, if a row has the same encrypted JOIN value in both a `City` column and a `State` column, it might be reasonable to infer that value is `New York`.

Using the `CreateCollaboration` API operation, for the `dataEncryptionMetadata` parameter, you can set the value of `allowJoinsOnColumnsWithDifferentNames` to `true` or `false`. For more information about API operations, see the AWS Clean Rooms API Reference.

By default, fingerprint column encryption is affected by the `targetHeader` for that column, set in Step 4: Generate an encryption schema for a tabular file . Therefore, the same cleartext value has different encrypted representations in each different fingerprint column that it's encrypted for.

This parameter can be useful at preventing the inference of cleartext values in some cases. For example, seeing the same encrypted value in fingerprint columns `City` and `State` might be used to reasonably infer the value is `New York`. However, this parameter's use requires additional coordination in advance, so that all columns to be joined in queries have shared names.

You can use the **Allow JOIN of columns with different names** parameter to loosen this restriction. When the parameter value is set to `Yes`, it allows any columns encrypted for JOIN to be used together regardless of name.

Preserve NULL values parameter

In the console, you can set the **Preserve NULL values** parameter when creating a collaboration to indicate that there is no value present for that column.

The following table describes the values for the **Preserve NULL values** parameter.

Parameter value	Description
No	NULL values are not preserved. NULL values don't appear as NULL in an encrypted table. NULL values appear as unique random values in a C3R table.
Yes	NULL values are preserved. NULL values appear as NULL in an encrypted table. If you require SQL semantics of NULL values, you can set this value to **Yes**. As a result, NULL entries appear as NULL in the C3R table, regardless of whether the column is encrypted and regardless of the parameter setting for **Allow duplicates**.

Using the `CreateCollaboration` API operation, for the `dataEncryptionMetadata` parameter, you can set the value of `preserveNulls` to `true` or `false`. For more information about API operations, see the AWS Clean Rooms API Reference.

When the **Preserve NULL values** parameter is set to **No** for the collaboration:

1. NULL entries in `cleartext` columns are unchanged.
2. NULL entries in encrypted `fingerprint` columns are encrypted as random values to conceal their contents. Joining on an encrypted column with NULL entries in the cleartextcolumn doesn't produce any matches for any of the NULL entries. No matches are made because they each receive their own, unique random content.
3. NULL entries in encrypted `sealed` columns are encrypted.

When the value of the **Preserve NULL values** parameter is set to **Yes** for the collaboration, NULL entries from all columns remain as NULL regardless of whether the column is encrypted.

The **Preserve NULL values** parameter is useful in scenarios such as data enrichment, where you want to share a lack of information expressed as NULL. The **Preserve NULL values** parameter is also useful in fingerprint or HMAC format if you have NULL values in the column you want to JOIN or GROUP BY.

If the value of the **Allow duplicates** and **Preserve NULL values** parameters is set to **No**, having more than one NULL entry in a fingerprint column produces an error and stops encryption. If the value of either parameter is set to **Yes**, no such error occurs.

Optional flags in Cryptographic Computing for Clean Rooms

The following sections describe the optional flags that you can set when you encrypt data using the C3R encryption client for tabular file customization and testing.

Topics

`--csvInputNULLValue` flag

You can use the `--csvInputNULLValue` flag to specify custom encodings for NULL entries in the input data when you encrypt data using the C3R encryption client.

The following table summarizes the usage and parameters of this flag.

Usage	Parameters
Optional. Users can specify custom encodings for NULL entries in the input data.	User-specified encoding of NULL values in the input CSV file

A NULL entry is an entry which is considered to be lacking content, specifically in the context of a richer tabular format like an SQL table. Although .csv doesn't explicitly support this characterization for historical reasons, it's a common convention to consider an empty entry containing only white space to be NULL. Therefore, that's the default behavior of the C3R encryption client and it can be customized as needed.

`--csvOutputNULLValue` flag

You can use the `--csvOutputNULLValue` flag to specify custom encodings for NULL entries in the output data when you [encrypt data](#) using the C3R encryption client.

The following table summarizes the usage and parameters of this flag.

Usage	Parameters
Optional. Users can specify custom encodings in the generated output file for NULL entries.	User-specified encoding of NULL values in the output CSV file

A NULL entry is an entry which is considered to be lacking content, specifically in the context of a richer tabular format like an SQL table. Although .csv doesn't explicitly support this characterization for historical reasons, it's a common convention to consider an empty entry containing only white space to be NULL. Therefore, that's the default behavior of the C3R encryption client and it can be customized as needed.

`--enableStackTraces` flag

When you [encrypt data](#) using the C3R encryption client, use the `--enableStackTraces` flag to provide additional contextual information for error reporting when C3R encounters an error.

AWS doesn't collect errors. If you encounter an error, use the stack trace to troubleshoot the error yourself or send the stack trace to AWS Support for assistance.

The following table summarizes the usage and parameters of this flag.

Usage	Parameters
Optional. Used to provide additional contextual information for error reporting	None

Usage	Parameters
when the C3R encryption client encounters an error.	

`--dryRun` flag

The encrypt and decrypt C3R encryption client commands include an optional `--dryRun` flag. The flag takes all the user-provided arguments and checks them for validity and consistency.

You can use the `--dryRun` flag to check if your schema file is valid and consistent with its corresponding input file.

The following table summarizes the usage and parameters of this flag.

Usage	Parameters
Optional. Causes the C3R encryption client to parse parameters and check files, but performs no encryption or decryption.	None

`--tempDir` flag

You might want to use a temporary directory because encrypted files can sometimes be larger than non-encrypted files, depending on their settings. Datasets must also be encrypted per collaboration to work correctly.

When you encrypt data using C3R, use the `--tempDir` flag to specify the location where temporary files can be created while processing the input.

The following table summarizes the usage and parameters of this flag.

Usage	Parameters
Users can specify the location where temporary files can be created while processin g the input.	Defaults to the system temporary directory.

Queries with Cryptographic Computing for Clean Rooms

This topic provides information about writing queries that use data tables that have been encrypted using Cryptographic Computing for Clean Rooms.

Topics

Queries that branch on NULL

To have a query branch on a NULL statement means to use syntax like `IF x IS NULL THEN 0 ELSE 1`.

Queries can always branch on NULL statements in cleartext columns.

Queries can branch on NULL statements in sealed columns and fingerprint columns only when the value of the **Preserve NULL values** parameter (`preserveNulls`) is set to `true`.

Queries that violate these constraints might yield incorrect results.

Mapping one source column to multiple target columns

One source column can map to multiple target columns. For example, you might want to both JOIN and SELECT on a column.

For more information, see Using the same data for both JOIN and SELECT queries.

Using the same data for both JOIN and SELECT queries

If the data in a column is not sensitive, it can appear in a cleartext target column, which allows it to be used for any purpose.

If data in a column is sensitive and must be used for both JOIN and SELECT queries, map that source column to two target columns in the output file. One column is encrypted with the `type` as a fingerprint column, and one column is encrypted with the `type` as a sealed column. The interactive schema generation of the C3R encryption client suggests header suffixes of `_fingerprint` and `_sealed`. These header suffixes can be a useful convention for differentiating such columns quickly.

Guidelines for the C3R encryption client

The C3R encryption client is a tool that enables organizations to bring sensitive data together to derive new insights from data analytics. The tool cryptographically limits what can be learned by any party and AWS in the process. Although this is vitally important, the process of securing data cryptographically can add significant overhead both in terms of compute and storage resources. Therefore, it is important to understand the tradeoffs of using each setting and how to optimize settings while still maintaining the desired cryptographic assurances. This topic focuses on the performance implications of different settings in the C3R encryption client and schemas.

All C3R encryption client encryption settings provide different cryptographic assurances. The collaboration-level settings are most secure by default. Enabling additional functionality while creating a collaboration weakens privacy guarantees, allowing activities like frequency analysis to be conducted on the ciphertext. For more information about how these settings are used and what their implications are, see *Cryptographic computing*.

Topics

Performance implications for column types

C3R uses three column types: cleartext, fingerprint, and sealed. Each of these column types provide different cryptographic assurances and have different intended uses. In the following sections, the performance implications of the column type are discussed and the performance impact of each setting.

Topics

Cleartext columns

Cleartext columns are not changed from their original format and not cryptographically processed in any way. This column type can't be configured and does not impact storage or compute performance.

Fingerprint columns

Fingerprint columns are meant to be used for joining data across multiple tables. To this end, the resulting ciphertext size must always be the same. However, these columns are impacted by the collaboration-level settings. Fingerprint columns might have varying degrees of impact on the output file size depending on the cleartext contained in the input.

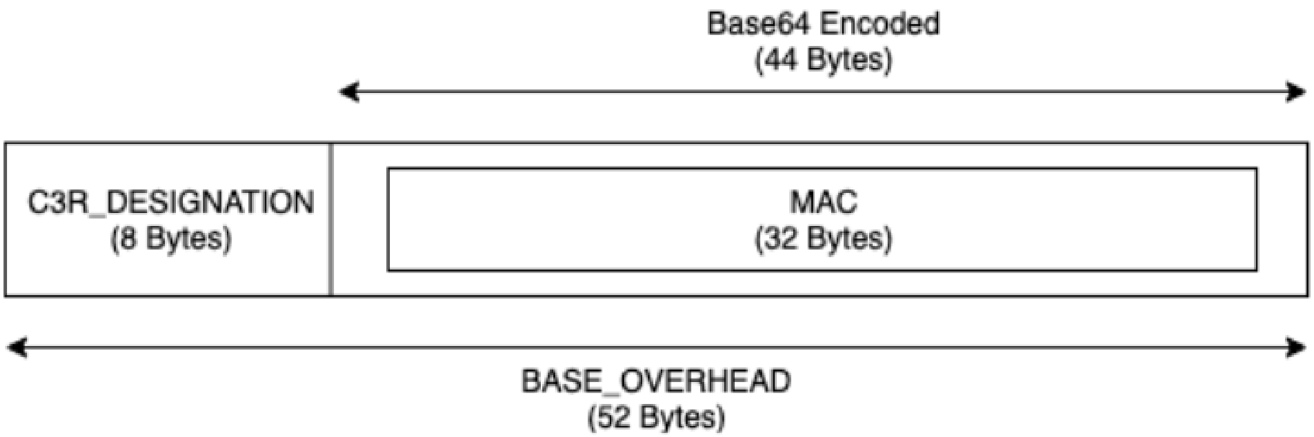
Topics

Base overhead for fingerprint columns

There is a base overhead for fingerprint columns. This overhead is constant and in place of the size of the cleartext bytes.

Data in the fingerprint columns is cryptographically processed through a Hash-based Message Authentication Code (HMAC) function, which turns the data into a 32 byte message authentication code (MAC). This data is then processed through a base64 encoder, adding roughly 33 percent to the byte size. It is pre-pended with an 8 byte C3R designation to designate the type of column that the data belongs to and the client version that produced it. The final result is 52 bytes. This result is then multiplied by the row count to get the total base overhead (use the number of total non-`null` values if `preserveNulls` is set to true).

The following image shows how *`BASE_OVERHEAD = C3R_DESIGNATION + (MAC * 1.33)`*



The output ciphertext in the fingerprint columns will always be 52 bytes. This can be a significant storage decrease if the input cleartext data averages more than 52 bytes (for example, full street addresses). This can be a significant storage increase if the input cleartext data averages less than 52 bytes (for example, customer ages).

Collaboration settings for fingerprint columns

`preserveNulls` setting

When the collaboration-level setting `preserveNulls` is `false` (default), each `null` value is substituted with a unique, random 32 bytes and processed as if it were not `null`. The result is that each `null` value is now 52 bytes. This can add significant storage requirements for tables that contain very sparse data compared to when this setting is `true` and `null` values are passed through as `null`.

If you don't need the privacy assurances of this setting and prefer to retain `null` values within your datasets, enable the `preserveNulls` setting at the time the collaboration is created. The `preserveNulls` setting can't be changed after the collaboration is created.

Example data for a fingerprint column

The following is an example set of input and output data for a fingerprint column with settings to reproduce. Other collaboration-level settings like `allowCleartext` and `allowDuplicates` don't impact the results and can be set as `true` or `false` if trying to reproduce locally.

Example shared secret: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

Example collaboration ID: `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`

allowJoinsOnColumnsWithDifferentNames: `True` This setting doesn't impact performance or storage requirements. However, this setting makes column name choice irrelevant when reproducing the values shown in the following tables.

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`
Input bytes	0
Output bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`
Output	`01:hmac:3lkFjthvV3IUu6mMvFc1a+XAHwgw/ElmOq4p3Yg25kk=`
Deterministic	No
Input bytes	0
Output bytes	52

Example 3

Input	`empty string`
`preserveNulls`	`-`

Output	`01:hmac:oKTgi3Gba+eUb3JteSz 2EMgXUkF1WgM77UP0Ydw5kPQ=`
Deterministic	`Yes`
Input bytes	0
Output bytes	52

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-
Output	`01:hmac:kU/IqwG7FMmzzshr0B9 scomE0UJUEE7j9keTctplGww=`
Deterministic	`Yes`
Input bytes	26
Output bytes	52

Example 5

Input	`abcdefghijklmnopqrstuvwxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ01 23456789`
`preserveNulls`	-
Output	`01:hmac:ks3htnQbw2vdhCRFF6J NzW5LMndJaHG57uvE26mBtSs=`
Deterministic	`Yes`
Input bytes	62

Output bytes	52

Troubleshooting fingerprint columns

Why is the ciphertext in my fingerprint columns several times greater than the size of the cleartext that went into it?

Ciphertext in a fingerprint column is always 52 bytes in length. If your input data were small (for example, the ages of customers), it will show a significant increase in size. This can also happen if the `preserveNulls` setting is set to `false`.

Why is the ciphertext in my fingerprint columns several times smaller than the size of the cleartext that went into it?

Ciphertext in a fingerprint column is always 52 bytes in length. If your input data were large (for example, the full street addresses of customers), it will show a significant decrease in size.

How do I know if I need the cryptographic assurances provided by `preserveNulls`?

Unfortunately, the answer is that it depends. At a minimum, the the section called "Parameters" should be reviewed for how the `preserveNulls` setting is protecting your data. However, we recommend that you reference your organization's data handling requirements and any contracts applicable to the respective collaboration.

Why do I have to incur the overhead of base64?

To allow for compatibility with tabular file formats such as CSV, base64-encoding is necessary. Although some file formats like Parquet might support binary representations of data, it's important that all participants in a collaboration represent data in the same way to ensure proper query results.

Sealed columns

Sealed columns are meant to be used for transferring data between members of a collaboration. The ciphertext in these columns is non-deterministic and has significant impact on both performance and storage based on how the columns are configured. These columns can be configured individually and often have the greatest impact on the performance of the C3R encryption client and the resulting output file size.

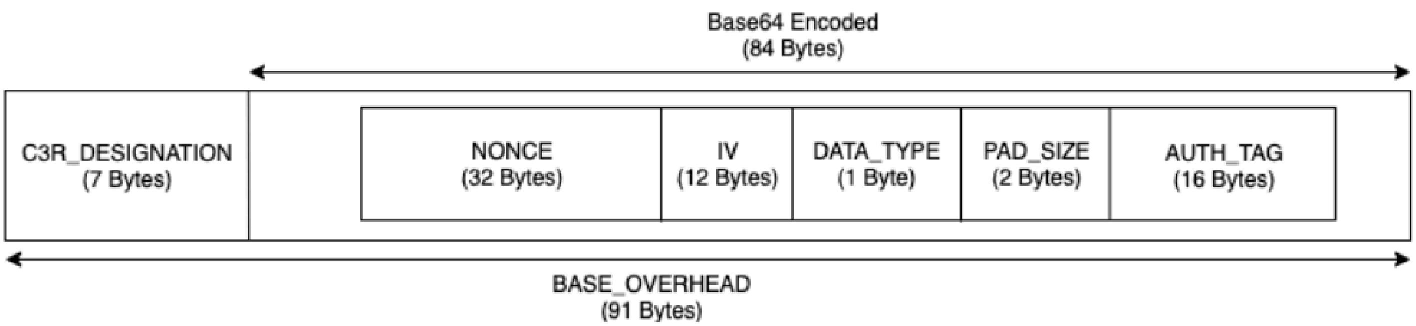
Topics

Base overhead for sealed columns

There is a base overhead for sealed columns. This overhead is constant and in addition to the size of the cleartext and padding (if any) bytes.

Before any encryption, data in the sealed columns is pre-pended with a 1 byte character designating what type of data is contained. If padding is selected, the data is then padded and appended with 2 bytes stating the pad size. After these bytes are added, data is cryptographically processed by using AES-GCM and stored with the IV (12 bytes), nonce (32 bytes), and Auth Tag (16 bytes). This data is then processed through a base64 encoder, adding roughly 33 percent to the byte size. The data is pre-pended with a 7 byte C3R designation to designate what type of column the data belongs to and the client version used to produce it. The result is a final base overhead of 91 bytes. This result can then be multiplied by the row count to get the total base overhead (use the number of total non-null values if `preserveNulls` is set to true).

The following image shows how `BASE_OVERHEAD = C3R_DESIGNATION + ((NONCE + IV + DATA_TYPE + PAD_SIZE + AUTH_TAG) * 1.33)`



Collaboration settings for sealed columns

`preserveNulls` setting

When the collaboration-level setting `preserveNulls` is `false` (default), each `null` value is unique, random 32 bytes and processed as if it were not `null`. The result is that each `null` value is now 91 bytes (more if padded). This can add significant storage requirements for tables that contain very sparse data compared to when this setting is `true` and `null` values are passed through as `null`.

If you don't need the privacy assurances of this setting and prefer to retain `null` values within your datasets, enable the `preserveNulls` setting at the time the collaboration is created. The `preserveNulls` setting can't be changed after the collaboration is created.

Schema settings sealed columns: padding types

Topics

- Pad type of none
- Pad type of fixed
- Pad type of max

Pad type of none

Selecting a pad type of `none` doesn't add any padding to the cleartext and adds no additional overhead to the base overhead described earlier. No padding results in the most space-efficient output size. However, it doesn't provide the same privacy assurances as the `fixed` and `max` padding types. This is because the size of the underlying cleartext is discernible from the size of the ciphertext.

Pad type of `fixed`

Selecting a pad type of `fixed` is a privacy-preserving measure to hide the lengths of the data contained within a column. This is done by padding all the cleartext to the provided `pad_length` before it is encrypted. Any data exceeding that size causes the C3R encryption client to fail.

Given that the padding is added to the cleartext before it is encrypted, AES-GCM has a 1-to-1 mapping of cleartext to ciphertext bytes. The base64 encoding will add 33 percent. The additional storage overhead of the padding can be calculated by subtracting the average length of the cleartext from the value of the `pad_length` and multiplying it by 1.33. The result is the average overhead of padding per record. This result can then be multiplied by the number of rows to get the total padding overhead (use the number of total non-`null` values if `preserveNulls` is set to `true`).

`PADDING_OVERHEAD = (PAD_LENGTH - AVG_CLEARTEXT_LENGTH) * 1.33 * ROW_COUNT`

We recommend that you select the minimum `pad_length` that encompasses the largest value in a column. For example, if the largest value is 50 bytes, a `pad_length` of 50 is sufficient. A value larger than that will only add additional storage overhead.

Fixed padding does not add any significant compute overhead.

Pad type of `max`

Selecting a pad type of `max` is a privacy-preserving measure to hide the lengths of the data contained within a column. This is done by padding all the cleartext to the largest value in the column plus the additional `pad_length` before it is encrypted. Generally, `max` padding provides the same assurances as `fixed` padding for a single dataset while allowing for not knowing the largest cleartext value in the column. However, `max` padding might not provide the same privacy assurances as `fixed` padding across updates because the largest value in the individual datasets might differ.

We recommend that you select an additional `pad_length` of 0 when using `max` padding. This length pads all values to be the same size as the largest value in the column. A value larger than that will only add additional storage overhead.

If the largest cleartext value is known for a given column, we recommend that you use the `fixed` pad type instead. Using `fixed` padding creates consistency across updated datasets. Using `max` padding results in each subset of data being padded to the largest value that was in the subset.

Example data for a sealed column

The following is an example set of input and output data for a sealed column with settings to reproduce. Other collaboration-level settings like `allowCleartext`, `allowJoinsOnColumnsWithDifferentNames`, and `allowDuplicates` don't impact the results and can be set as `true` or `false` if trying to reproduce locally. Although these are the basic settings to reproduce, the sealed column is non-deterministic and values will change every time. The goal is to show the bytes in as compared to the bytes out. The example `pad_length` values were chosen intentionally. They show that `fixed` padding results in the same values as `max` padding with the recommended minimum `pad_length` settings or when additional padding is desired.

Example shared secret: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

Example collaboration ID: `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`

Topics

Pad type of none

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`
Input bytes	0
Output bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSPbNIJfG3iXmu6cbCUrizuV`
Deterministic	No
Input bytes	0
Output bytes	91

Example 3

Input	`empty string`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfstGSPEM6qR8DWC2PB2GMlX41YK`
Deterministic	No
Input bytes	0
Output bytes	91

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6pkx9sGL5VLDQeHzh6DmPpyWNuI=`
Deterministic	No
Input bytes	26
Output bytes	127

Example 5

Input	`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6plwtH/8tRFnn2rF91bcB9G4+n8GiRfJNmqdP4/QOQ3cXb/pbvPcnnohrHIGSX54ua+1/JfcVjc=`
Deterministic	No
Input bytes	62
Output bytes	175

Pad type of `fixed` (Example 1)

In this example, `pad_length` is 62 and largest input is 62 bytes.

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`
Input bytes	0
Output bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcoNpATs0GzbnLkor4L+/aSuA=`
Deterministic	No
Input bytes	0
Output bytes	175

Example 3

Input	`empty string`
`preserveNulls`	`-`
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfstGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcoLB53l07VZpA6OwkuXu29CA=`
Deterministic	No
Input bytes	0
Output bytes	175

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6pkx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcutBAcO+Mb9tuU2KIHH31AWg=`
Deterministic	No
Input bytes	26
Output bytes	175

Example 5

Input	`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6plwtH/8tRFnn2rF91bcB9G4+n8GiRfJNmqdP4/QOQ3cXb/pbvPcnnohrHIGSX54ua+1/JfcVjc=`
Deterministic	No
Input bytes	62

Output bytes	175

Pad type of `fixed` (Example 2)

In this example, `pad_length` is 162 and largest input is 62 bytes.

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`
Input bytes	0
Output bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcnkB0xbLWD7zNdAqQGR0rXoSESdW0I0vpNoGcBfv4cJbG0A3h1DvtkSSVc2B80OOGppzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6uwrmwv/xAySX+xcntotL703aBTBb`

Deterministic	No
Input bytes	0
Output bytes	307

Example 3

Input	`empty string`
`preserveNulls`	-

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfstGSNWfMRp7nSb7S`
	`MX2s3JKLOhK1+7r75Tk+Mx9jy48`
	`Fcg1yOPvBqRSZ7oqy1V3UKfYTLE`
	`Zb/hCz7oaIneVsrcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwv84lVaT9Yd+6oQx65/+gdVT`

Deterministic	No
Input bytes	0
Output bytes	307

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5` `MG5vbmNlMDEyMzQ1Njc4OTBqfRY`

	`Z98t5KU6aWfsteEE1GKEPiRzyh0`
	`h7t6OmWMLTWCvO2ckr6pkx9jy48`
	`Fcg1yOPvBqRSZ7oqy1V3UKfYTLE`
	`Zb/hCz7oaIneVsrcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwtX5Hnl+WyfO6ks3QMaRDGSf`

Deterministic	No
Input bytes	26
Output bytes	307

Example 5

Input	`abcdefghijklmnopqrstuvwxyzA`
	`BCDEFGHIJKLMNOPQRSTUVWXYZ01`
	`23456789`

`preserveNulls`	-

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfsteEE1GKEPiRzyh0`
	`h7t6OmWMLTWCvO2ckr6plwtH/8t`
	`RFnn2rF91bcB9G4+n8GiRfJNmqd`
	`P4/QOQ3cXb/pbvPcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwjkJXQZOgPdeFX9Yr/8alV5i`

Deterministic	No

Input bytes	62
Output bytes	307

Pad type of `max` (Example 1)

In this example, `pad_length` is 0 and largest input is 62 bytes.

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`
Input Bytes	0
Output Bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfssGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcoNpATs0GzbnLkor4L+/aSuA=`
Deterministic	No
Input bytes	0

Output bytes	175

Example 3

Input	`empty string`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfstGSNWfMRp7nSb7SMX2s3JKLOhK1+7r75Tk+Mx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcoLB53l07VZpA6OwkuXu29CA=`
Deterministic	No
Input bytes	0
Output bytes	175

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6pkx9jy48Fcg1yOPvBqRSZ7oqy1V3UKfYTLEZb/hCz7oaIneVsrcutBAcO+Mb9tuU2KIHH31AWg=`
Deterministic	No

Input bytes	26
Output bytes	175

Example 5

Input	`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
`preserveNulls`	-
Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OTBqfRYZ98t5KU6aWfsteEE1GKEPiRzyh0h7t6OmWMLTWCvO2ckr6plwtH/8tRFnn2rF91bcB9G4+n8GiRfJNmqdP4/QOQ3cXb/pbvPcnnohrHIGSX54ua+1/JfcVjc=`
Deterministic	No
Input bytes	62
Output bytes	175

Pad type of `max` (Example 2)

In this example, `pad_length` is 100 and largest input is 62 bytes.

Example 1

Input	`null`
`preserveNulls`	`TRUE`
Output	`null`
Deterministic	`Yes`

Input bytes	0
Output bytes	0

Example 2

Input	`null`
`preserveNulls`	`FALSE`

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfssGSNWfMRp7nSb7S`
	`MX2s3JKLOhK1+7r75Tk+Mx9jy48`
	`Fcg1yOPvBqRSZ7oqy1V3UKfYTLE`
	`Zb/hCz7oaIneVsrcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwv/xAySX+xcntotL703aBTBb`

Deterministic	No
Input bytes	0
Output bytes	307

Example 3

Input	`empty string`
`preserveNulls`	`-`

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfstGSNWfMRp7nSb7S`
	`MX2s3JKLOhK1+7r75Tk+Mx9jy48`

	`Fcg1yOPvBqRSZ7oqy1V3UKfYTLE`
	`Zb/hCz7oaIneVsrcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwv84lVaT9Yd+6oQx65/+gdVT`

Deterministic	No
Input bytes	0
Output bytes	307

Example 4

Input	`abcdefghijklmnopqrstuvwxyz`
`preserveNulls`	-

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfsteEE1GKEPiRzyh0`
	`h7t6OmWMLTWCvO2ckr6pkx9jy48`
	`Fcg1yOPvBqRSZ7oqy1V3UKfYTLE`
	`Zb/hCz7oaIneVsrcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwtX5Hnl+WyfO6ks3QMaRDGSf`

Deterministic	No
Input bytes	26
Output bytes	307

Example 5

Input	`abcdefghijklmnopqrstuvwxyzA`
	`BCDEFGHIJKLMNOPQRSTUVWXYZ01`
	`23456789`

`preserveNulls`	-

Output	`01:enc:bm9uY2UwMTIzNDU2Nzg5`
	`MG5vbmNlMDEyMzQ1Njc4OTBqfRY`
	`Z98t5KU6aWfsteEE1GKEPiRzyh0`
	`h7t6OmWMLTWCvO2ckr6plwtH/8t`
	`RFnn2rF91bcB9G4+n8GiRfJNmqd`
	`P4/QOQ3cXb/pbvPcnkB0xbLWD7z`
	`NdAqQGR0rXoSESdW0I0vpNoGcBf`
	`v4cJbG0A3h1DvtkSSVc2B80OOGp`
	`pzdDqhrUVN5wFNyn8vgfPMqDaeJk5bn`
	`+8o4WtG/ClipNcjDXvXVtK4vfCohcCA6`
	`uwrmwjkJXQZOgPdeFX9Yr/8alV5i`

Deterministic	No
Input bytes	62
Output bytes	307

Troubleshooting sealed columns

Why is the ciphertext in my sealed columns several times greater than the size of the cleartext that went into it?

This depends on several factors. For one, ciphertext in a Cleartext column is always at least 91 bytes in length. If your input data were small (for example, the ages of customers), it will show a significant increase in size. Second, if `preserveNulls` were set to `false` and your input data contained a lot of `null` values, each of those `null` values will have been turned into 91 bytes of ciphertext. Finally, if you use padding, by definition bytes are added to the cleartext data before it is encrypted.

Most of my data in a sealed column is really small, and I need to use padding. Can I just remove the big values and process them separately to save space?

We don't recommend that you remove large values and process them separately. Doing so changes the privacy assurances that the C3R encryption client is providing. As a threat model, assume that an observer can see both encrypted datasets. If the observer sees that one subset of data has a column padded significantly more or less than another subset, they can make inferences on the size of the data in each subset. For example, assume a `fullName` column is padded to a total of 40 bytes in one file and is padded to 800 bytes in another file. An observer might be able to assume that one dataset contains the world's longest name747 bytes).

Do I need to provide extra padding when using the `max` padding type?

No. When using `max` padding, we recommend that the `pad_length`, also known as the additional padding *beyond* the largest value in the column, is set to 0.

Can I just pick a large `pad_length` when using `fixed` padding to avoid worrying if the largest value will fit?

Yes, but the large pad length is inefficient and uses more storage than necessary. We recommend that you to check to see how large the largest value is and set the `pad_length` to that value.

How do I know if I need the cryptographic assurances provided by `preserveNulls`?

Unfortunately, the answer is that it depends. At a minimum, the [Cryptographic Computing for Clean Rooms]() should be reviewed for how the `preserveNulls` setting is protecting your data. However, we recommend that you reference your organization's data handling requirements and any contracts applicable to the respective collaboration.

Why do I have to incur the overhead of base64?

To allow for compatibility with tabular file formats such as CSV, base64 encoding is necessary. Although some file formats like Parquet might support binary representations of data, it's important that all participants in a collaboration represent data in the same way to ensure proper query results.

Troubleshooting unanticipated increases in ciphertext size

Let's say that you encrypted your data, and the size of the resulting data is surprisingly large. The following steps can help you identify where the size increase occurred and what, if any, actions you can take.

Identifying where the size increase occurred

Before you can troubleshoot why your encrypted data is significantly larger than your cleartext data, you must first identify where the increase in size is. Cleartext columns can safely be ignored because they are unchanged. Look at the remaining fingerprint and sealed columns, and choose one that appears significant.

Identifying the reason the size increase occurred

A fingerprint column or a sealed column might contribute to the size increase.

Topics

- Is the size increase coming from a fingerprint column?
- Is the size increase coming from a sealed column?

Is the size increase coming from a fingerprint column?

If the column that's most contributing to the increase in storage is a fingerprint column, this is likely because the cleartext data is small (for example, customer age). Each resulting fingerprint ciphertext is 52 bytes in length. Unfortunately, nothing can be done about this issue on a column-by-column basis. For more information, see Base overhead for fingerprint columns for details about this column, including how it impacts storage requirements.

The other possible cause of size increase in a fingerprint column is the collaboration setting, `preserveNulls`. If the collaboration setting for `preserveNulls` is disabled (the default setting), all `null` values in fingerprint columns will have become 52 bytes of ciphertext. There is nothing that can be done for this in the current collaboration. The `preserveNulls` setting is set at the time a collaboration is created and all collaborators must use the same setting to ensure correct query results. For more information about the `preserveNulls` setting and how enabling it impacts the privacy assurances of your data, see *Cryptographic computing*.

Is the size increase coming from a sealed column?

If the column that's most contributing to the increase in storage is a sealed column, there are a few details that could contribute to the size increase.

If the cleartext data is small (for example, customer age), each resulting sealed ciphertext is at least 91 bytes in length. Unfortunately, nothing can be done about this issue. For more information, see

Base overhead for sealed columns for details about this column, including how it impacts storage requirements.

The second primary cause for storage increase in sealed columns is padding. Padding adds extra bytes to the cleartext before it's encrypted to hide the size of individual values in a dataset. We recommend that you set padding to the minimum possible value for your dataset. At a minimum, `pad_length` for `fixed` padding must be set to encompass the largest possible value in the column. Any higher setting than that doesn't add additional privacy assurances. For example, if you know the largest possible value in a column can be 50 bytes, we recommend that you set the `pad_length` to 50 bytes. However, if the sealed column is using `max` padding, we recommend that you set the `pad_length` to 0 bytes. This is because `max` padding is referring to the *additional* padding beyond the largest value in the column.

The final possible cause of size increase in a sealed column is the collaboration setting, `preserveNulls`. If the collaboration setting for `preserveNulls` is disabled (the default setting), all `null` values in sealed columns will have become 91 bytes of ciphertext. There is nothing that can be done for this in the current collaboration. The `preserveNulls` setting is set at the time a collaboration is created, and all collaborators must use the same setting to ensure correct query results. For more information about this setting does and how enabling it impacts the privacy assurances of your data, see *Cryptographic computing*.

Query logging in AWS Clean Rooms

Query logging is a feature in AWS Clean Rooms. When you create a collaboration and turn on **Query logging**, members can store query logs relevant to them in Amazon CloudWatch Logs.

With query logs, members can determine if the queries comply with the analysis rules and align with the collaboration agreement. In addition, query logs help support audits.

When the **Query logging** option is turned on in the AWS Clean Rooms console, the query logs include the following:

- `analysisRule` – The analysis rule for the configured table.
- `analysisTemplateArn` – The analysis template that was run (appears depending on analysis rule).
- `collaborationId` – The unique identifier for collaboration in which the query was run.
- `configuredTableID` – The unique identifier for configured table referenced in the query.
- `directQueryAnalysisRulePolicy.custom.allowedAnalysis` – The analysis template allowed to run on configured table (appears depending on analysis rule).
- `directQueryAnalysisRulePolicy.v1.custom.allowedAnalysisProviders` – The query providers allowed to create query (appears depending on analysis rule).
- `eventID` – The unique identifier for the query run. After August 31, 2023, the unique identifier is the same as the `protectedQueryID`.
- `eventTimestamp` – The query run time.
- `parameters.parametervalue` – The parameter values (appears depending on the query text).
- `queryText` – The SQL definition of query run. If there are parameters, they are labelled as `:parametervalue`.
- `queryValidationErrors` – The query errors at query validation.
- `schemaName` – The name of configured table association referenced in the query.

Receiving query logs

You don't need to perform any actions outside of AWS Clean Rooms to set up query logs. AWS Clean Rooms creates log groups for collaborations after each collaboration member creates a membership.

Members who can query, members who can receive results, and members whose configuration tables are referenced in the query will receive a query log.

The member who can query and member who can receive results will receive query logs for each configured table that is referenced in the query. If they don’t own the configured table, they won't be able to view the configured table ID (`configuredTableID`).

If a member has multiple configured table associations referenced in the query, they will receive a query log for each configured table.

Logs are created for queries that contain unsupported and supported SQL in AWS Clean Rooms. For more details, see the [AWS Clean Rooms SQL Reference](#).

Logs are also created when queries reference configured tables that are not associated to the collaboration.

Logs are not created for incorrect SQL in AWS Clean Rooms.

Query logs don't indicate that a query was successful and query output was delivered. They confirm that a query was submitted by the member who can query. Query logs also confirm that the query contains supported SQL in AWS Clean Rooms and references configured tables associated to the collaboration.

Example

For example, a log isn't produced if the query was cancelled after AWS Clean Rooms validated its compliance with analysis rules and during query processing.

If you delete the log group, you must re-create the log group manually with the same log group name (collaboration ID of the collaboration). Or, you can turn the logging off and on in your membership.

For more information about how to turn on query logging, see Creating a collaboration in AWS Clean Rooms.

For more information about Amazon CloudWatch Logs, see the Amazon CloudWatch Logs User Guide.

Using query logs

We recommend that members periodically take the following actions:

- To verify that the queries match the use cases or queries that were agreed upon for the collaboration, review the queries that are run in the collaboration.

  For more information about how to view recent queries, see Viewing recent queries.
- To verify that the configured table columns match what was agreed upon for the collaboration, review the configured table columns that are used in collaboration members' analysis rules and in queries.

  For more information about how to view the configured columns, see Viewing tables and analysis rules.

Setting up AWS Clean Rooms

The following topics explain how to set up AWS Clean Rooms.

Topics

Sign up for AWS

Before you can use any AWS service, including AWS Clean Rooms, you must sign up for AWS.

If you don't have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.
3. When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to an administrative user, and use only the root user to perform tasks that require root user access.

Set up service roles for AWS Clean Rooms

Topics

Create an administrator user

To use AWS Clean Rooms, you need to create an administrator user for yourself and add the administrator user to an administrators group.

To create an administrator user, choose one of the following options.

Choose one way to manage your administrator	To	By	You can also
In IAM Identity Center (Recommended)	Use short-term credentials to access AWS. This aligns with the security best practices. For information about best practices, see Security best practices in IAM in the *IAM User Guide*.	Following the instructions in Getting started in the *AWS IAM Identity Center User Guide*.	Configure programmatic access by Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*.
In IAM (Not recommended)	Use long-term credentials to access AWS.	Following the instructions in Creating your first IAM admin user and user group in the *IAM User Guide*.	Configure programmatic access by Managing access keys for IAM users in the *IAM User Guide*.

Create an IAM role for a collaboration member

A member is an AWS customer who is a participant in a collaboration.

To create an IAM role for a collaboration member

1. Follow the [Creating a role to delegate permissions to an IAM user] procedure in the *AWS Identity and Access Management User Guide*.
2. For the **Create policy** step, select the **JSON** tab in the **Policy editor**, and then add policies depending on the abilities granted to the collaboration member.

   AWS Clean Rooms offers the following managed policies based on common use cases:

If you want to …	Then use …
View the resources and metadata	AWS managed policy: `AWSCleanRoomsReadOnlyAccess`
Query	AWS managed policy: `AWSCleanRoomsFullAccess`
Query and receive results	AWS managed policy: `AWSCleanRoomsFullAccess`
Manage collaboration resources but do not query	AWS managed policy: `AWSCleanRoomsFullAccessNoQuerying`

   For information about the different managed policies offered by AWS Clean Rooms, see AWS managed policies for AWS Clean Rooms

Create a service role to read data

AWS Clean Rooms uses a service role to read the data.

There are two ways to create this service role:

If …	Then
You have the necessary IAM permissions to create a service role	Use the AWS Clean Rooms console to create a service role.

If ...	Then
You don't have `iam:CreateRole` , `iam:CreatePolicy` and `iam:AttachRolePolicy` permissions or You want to create the IAM roles manually	Do one of the following: • Use the following procedure to create a service role. • Ask your administrator to create the service role using the following procedure.

To create a service role to read data

> **Note**
>
> You or your IAM administrator should only follow this procedure if you don't have the necessary permissions to create a service role using the AWS Clean Rooms console.

1. Follow the Creating a role using custom trust policies (console) procedure in the *AWS Identity and Access Management User Guide*.
2. Use the following custom trust policy according to the Creating a role using custom trust policies (console) procedure.

   > **Note**
   >
   > If you want to ensure that the role can only be used in the context of a certain collaboration membership, you can scope down the trust policy further. For more information, see Cross-service confused deputy prevention.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RoleTrustPolicyForCleanRoomsService",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "cleanrooms.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

3. Use the following permissions policy according to the [Creating a role using custom trust policies (console)](#) procedure.

> **Note**
>
> The following example policy supports the permissions needed to read AWS Glue metadata and its corresponding Amazon S3 data. However, you might need to modify this policy depending on how you've set up your S3 data. For instance, if you have set up a custom KMS key for your S3 data, you may need to amend this policy with additional AWS KMS permissions.
> Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "NecessaryGluePermissions",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition"
            ],
            "Resource": [
                "arn:aws:glue:aws-region:accountId:database/database",
                "arn:aws:glue:aws-region:accountId:table/table",
                "arn:aws:glue:aws-region:accountId:catalog"
            ]
```

```
        },
    {
            "Effect": "Allow",
            "Action": [
                "glue:GetSchema",
                "glue:GetSchemaVersion"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "NecessaryS3BucketPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "s3BucketOwnerAccountId"
                    ]
                }
            }
        },
        {
            "Sid": "NecessaryS3ObjectPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3::bucket/prefix/*"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "s3BucketOwnerAccountId"
                    ]
                }
```

```
            }
        }

    ]
}
```

4. Replace each *placeholder* with your own information.
5. Continue to follow the [Creating a role using custom trust policies (console)](#) procedure to create the role.

Create a service role to receive results

> **Note**
>
> If you are the member who can only receive results (in the console, **Your member abilities** is only **Receive results**), follow this procedure.
> If you are a member who can both query and receive results (in the console, **Your member abilities** is both **Query** and **Receive results**), you can skip this procedure.

For collaboration members who can only receive results, AWS Clean Rooms uses a service role to write results of the queried data in the collaboration to the specified Amazon S3 bucket.

There are two ways to create this service role:

If ...	Then
You have the necessary IAM permissions to create a service role	Use the AWS Clean Rooms console to create a service role.
You don't have `iam:CreateRole` , `iam:CreatePolicy` and `iam:AttachRolePolicy` permissions or You want to create the IAM roles manually	Do one of the following: • Use the following procedure to create a service role. • Ask your administrator to create the service role using the following procedure.

To create a service role to receive results

Note

You or your IAM administrator should only follow this procedure if you don't have the necessary permissions to create a service role using the AWS Clean Rooms console.

1. Follow the Creating a role using custom trust policies (console) procedure in the *AWS Identity and Access Management User Guide.*
2. Use the following custom trust policy according to the Creating a role using custom trust policies (console) procedure.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIfExternalIdMatches",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "sts:ExternalId":
 "arn:aws:*:region:*:dbuser:*/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa*"
                }
            }
        },
        {
            "Sid": "AllowIfSourceArnMatches",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ForAnyValue:ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cleanrooms:us-east-1:555555555555:membership/
a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
```

```
                    ]

                }
            }
        }
    ]
}
```

3. Use the following permissions policy according to the [Creating a role using custom trust policies (console)](#) procedure.

> **Note**
>
> The following example policy supports the permissions needed to read AWS Glue metadata and its corresponding Amazon S3 data. However, you might need to modify this policy depending on how you've set up your S3 data.
> Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket_name"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount":"accountId"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::bucket_name/optional_key_prefix/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount":"accountId"
                }
            }
        }
    ]
}
```

4. Replace each *placeholder* with your own information:

   - *region* – The name of the AWS Region. For example, **us-east-1**.
   - *a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa* – The **Membership ID** of the member who can query. The **Membership ID** can be found on the **Details** tab of the collaboration. This ensures that AWS Clean Rooms is assuming the role only when this member runs the analysis in this collaboration.
   - *arn:aws:cleanrooms:us-east-1:555555555555:membership/a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa* – The single **Membership ARN** of the member who can query. The **Membership ARN** can be found on the **Details** tab of the collaboration. This ensures AWS Clean Rooms is assuming the role only when this member runs the analysis in this collaboration.
   - *bucket_name* – The **Amazon Resource Name (ARN)** of the S3 bucket. The **Amazon Resource Name (ARN)** can be found on the **Properties** tab of the bucket in Amazon S3.
   - *accountId* – The AWS account ID in which the S3 bucket is located.

     bucket_name/optional_key_prefix – The **Amazon Resource Name (ARN)** of the results destination in S3. The **Amazon Resource Name (ARN)** can be found on the **Properties** tab of the bucket in Amazon S3.

5. Continue to follow the [Creating a role using custom trust policies (console)] procedure to create the role.

Set up service roles for AWS Clean Rooms ML

Topics

- [Create a service role to read training data]
- [Create a service role to write a lookalike segment]
- [Create a service role to read seed data]

Create a service role to read training data

AWS Clean Rooms uses a service role to read training data. You can create this role using the console if you have the necessary IAM permissions. If you don't have `CreateRole` permissions, ask your administrator to create the service role.

To create a service role to train a dataset

1. Sign in to the IAM console (https://console.aws.amazon.com/iam/) with your administrator account.
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. In the **Policy editor**, select the **JSON** tab, and then copy and paste the following policy.

> **Note**
>
> The following example policy supports the permissions needed to read AWS Glue metadata and its corresponding Amazon S3 data. However, you might need to modify this policy depending on how you've set up your S3 data. This policy doesn't include a KMS key to decrypt data.
> Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetTable",
```

```
                "glue:GetTables",
                "glue:GetPartitions",
                "glue:GetPartition",
                "glue:BatchGetPartition",
                "glue:GetUserDefinedFunctions"
            ],
            "Resource": [
                "arn:aws:glue:region:accountId:database/databases",
                "arn:aws:glue:region:accountId:table/databases/tables",
                "arn:aws:glue:region:accountId:catalog",
                "arn:aws:glue:region:accountId:database/default"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase"
            ],
            "Resource": [
                "arn:aws:glue:region:accountId:database/default"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucket"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::bucketFolders/*"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        }
    ]
}
```

If you need to use a KMS key to decrypt data, add this AWS KMS statement to the previous template:

```
{
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
            ],
            "Resource": [
                "arn:aws:kms:region:accountId:key/keyId"
            ],
            "Condition": {
                "ArnLike": {
                        "kms:EncryptionContext:aws:s3:arn":
 "arn:aws:s3:::bucketFolders*"
                }
            }
        }
    ]
}
```

5. Choose **Next**.
6. For **Review and create**, enter a **Policy name** and **Description**, and review the **Summary**.
7. Choose **Create policy**.

   You have created a policy for AWS Clean Rooms.
8. Under **Access management**, choose **Roles**.

With **Roles**, you can create short-term credentials, which is recommended for increased security. You can also choose **Users** to create long-term credentials.

9. Choose **Create role**.
10. In the **Create role** wizard, for **Trusted entity type**, choose **Custom trust policy**.
11. Copy and paste the following custom trust policy into the JSON editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms-ml.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEqualsIfExists": {
                    "aws:SourceAccount": ["accountId"]
                },
                "StringLikeIfExists": {
                    "aws:SourceArn": "arn:aws:cleanrooms-
ml:region:account:training-dataset/*"
                }
            }
        }
    ]
}
```

The `SourceAccount` is always your AWS account. The `SourceArn` can be limited to a specific training dataset, but only after that dataset is created. Because you can't pre-know the training dataset ARN, the wildcard is specified here.

12. Choose **Next** and under **Add permissions**, enter the name of the policy you just created. (You might need to reload the page.)
13. Select the check box next to the name of the policy you created, and then choose **Next**.
14. For **Name, review, and create**, enter the **Role name** and **Description**.

> **Note**
>
> The **Role name** must match the pattern in the `passRole` permissions granted to the member who can query and receive results and member roles.

a. Review **Select trusted entities**, and edit if necessary.

b. Review the permissions in **Add permissions**, and edit if necessary.

c. Review the **Tags**, and add tags if necessary.

d. Choose **Create role**.

15. The service role for AWS Clean Rooms has been created.

Create a service role to write a lookalike segment

AWS Clean Rooms uses a service role to write lookalike segments to a bucket. You can create this role using the console if you have the necessary IAM permissions. If you don't have `CreateRole` permissions, ask your administrator to create the service role.

To create a service role to write a lookalike segment

1. Sign in to the IAM console (https://console.aws.amazon.com/iam/) with your administrator account.
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. In the **Policy editor**, select the **JSON** tab, and then copy and paste the following policy.

> **Note**
>
> The following example policy supports the permissions needed to read AWS Glue metadata and its corresponding Amazon S3 data. However, you might need to modify this policy depending on how you've set up your S3 data. This policy doesn't include a KMS key to decrypt data.
> Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::buckets"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucketFolders/*"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        }
  ]
}
```

If you need to use a KMS key to encrypt data, add this AWS KMS statement to the template:

```
{
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt*",
            ],
            "Resource": [
                "arn:aws:kms:region:accountId:key/keyId"
            ],
            "Condition": {
                "ArnLike": {
                        "kms:EncryptionContext:aws:s3:arn":
 "arn:aws:s3:::bucketFolders*"
                }
            }
        }
  ]
}
```

If you need to use a KMS key to decrypt data, add this AWS KMS statement to the template:

```
{
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:region:accountId:key/keyId"
            ],
            "Condition": {
                "ArnLike": {
                        "kms:EncryptionContext:aws:s3:arn":
 "arn:aws:s3:::bucketFolders*"
                }
            }
        }
  ]
}
```

5. Choose **Next**.

6. For **Review and create**, enter a **Policy name** and **Description**, and review the **Summary**.
7. Choose **Create policy**.

   You have created a policy for AWS Clean Rooms.
8. Under **Access management**, choose **Roles**.

   With **Roles**, you can create short-term credentials, which is recommended for increased security. You can also choose **Users** to create long-term credentials.
9. Choose **Create role**.
10. In the **Create role** wizard, for **Trusted entity type**, choose **Custom trust policy**.
11. Copy and paste the following custom trust policy into the JSON editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms-ml.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEqualsIfExists": {
                    "aws:SourceAccount": ["accountId"]

                },
                "StringLikeIfExists": {
                    "aws:SourceArn": "arn:aws:cleanrooms-
ml:region:account:configured-audience-model/*"
                }
            }
        }
    ]
}
```

   The `SourceAccount` is always your AWS account. The `SourceArn` can be limited to a specific training dataset, but only after that dataset is created. Because you can't pre-know the training dataset ARN, the wildcard is specified here.
12. Choose **Next**.

13. Select the check box next to the name of the policy you created, and then choose **Next**.
14. For **Name, review, and create**, enter the **Role name** and **Description**.

> **Note**
>
> The **Role name** must match the pattern in the `passRole` permissions granted to the member who can query and receive results and member roles.

   a. Review **Select trusted entities**, and edit if necessary.
   b. Review the permissions in **Add permissions**, and edit if necessary.
   c. Review the **Tags**, and add tags if necessary.
   d. Choose **Create role**.

15. The service role for AWS Clean Rooms has been created.

Create a service role to read seed data

AWS Clean Rooms uses a service role to read seed data. You can create this role using the console if you have the necessary IAM permissions. If you don't have `CreateRole` permissions, ask your administrator to create the service role.

To create a service role to read seed data

1. Sign in to the IAM console (https://console.aws.amazon.com/iam/) with your administrator account.
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. In the **Policy editor**, select the **JSON** tab, and then copy and paste the following policy.

> **Note**
>
> The following example policy supports the permissions needed to read AWS Glue metadata and its corresponding Amazon S3 data. However, you might need to modify this policy depending on how you've set up your S3 data. This policy doesn't include a KMS key to decrypt data.

Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
            ],
            "Resource": [
                "arn:aws:s3:::buckets"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucketFolders/*"
            ],
            "Condition":{
                "StringEquals":{
                    "s3:ResourceAccount":[
                        "accountId"
                    ]
                }
            }
        }
  ]
}
```

If you need to use a KMS key to decrypt data, add this AWS KMS statement to the template:

```
{
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:region:accountId:key/keyId"
            ],
            "Condition": {
                "ArnLike": {
                        "kms:EncryptionContext:aws:s3:arn":
 "arn:aws:s3:::bucketFolders*"
                }
            }
        }
  ]
}
```

5. Choose **Next**.
6. For **Review and create**, enter a **Policy name** and **Description**, and review the **Summary**.
7. Choose **Create policy**.

   You have created a policy for AWS Clean Rooms.
8. Under **Access management**, choose **Roles**.

   With **Roles**, you can create short-term credentials, which is recommended for increased security. You can also choose **Users** to create long-term credentials.
9. Choose **Create role**.
10. In the **Create role** wizard, for **Trusted entity type**, choose **Custom trust policy**.
11. Copy and paste the following custom trust policy into the JSON editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeRole",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "cleanrooms-ml.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEqualsIfExists": {
                    "aws:SourceAccount": ["accountId"]

                },
                "StringLikeIfExists": {
                    "aws:SourceArn": "arn:aws:cleanrooms-
ml:region:account:audience-generation-job/*"
                }
            }
        }
    ]
}
```

The `SourceAccount` is always your AWS account. The `SourceArn` can be limited to a specific training dataset, but only after that dataset is created. Because you can't pre-know the training dataset ARN, the wildcard is specified here.

12. Choose **Next**.
13. Select the check box next to the name of the policy you created, and then choose **Next**.
14. For **Name, review, and create**, enter the **Role name** and **Description**.

> **Note**
>
> The **Role name** must match the pattern in the `passRole` permissions granted to the member who can query and receive results and member roles.

    a. Review **Select trusted entities**, and edit if necessary.
    b. Review the permissions in **Add permissions**, and edit if necessary.
    c. Review the **Tags**, and add tags if necessary.
    d. Choose **Create role**.
15. The service role for AWS Clean Rooms has been created.

Creating a collaboration in AWS Clean Rooms

A *collaboration* is a secure logical boundary in AWS Clean Rooms in which members can perform SQL queries on configured tables.

Any member in AWS Clean Rooms can create a collaboration.

The collaboration creator can designate a single member to query and receive results. However, the collaboration creator might want to prevent the member who can query from having access to the query results. In that case, the collaboration creator can designate one member to who can query and another member who can receive results.

In most cases, the member who can query is also the member paying for query compute costs. However, the collaboration creator can configure a different member to be responsible for paying for the query compute costs.

For information about how to create a collaboration using the AWS SDKs, see the *AWS Clean Rooms API Reference*.

Topics

Create a collaboration

Before you begin, make sure that you have completed the following prerequisites:

- You have the name and AWS account ID for each member that you want to invite to the collaboration.
- You have permission to share the name and AWS account ID for each member with all members of the collaboration.

> **Note**
>
> You can't add more members after the collaboration is created.

To create a collaboration using the AWS Clean Rooms console

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with the AWS account that will function as the collaboration creator.
2. In the left navigation pane, choose **Collaborations**.
3. In the upper right corner, choose **Create collaboration**.
4. For **Step 1: Define collaboration**, do the following:

   a. For **Details**, enter the **Name** and **Description** of the collaboration.

      This information will be visible to collaboration members who are invited to participate in the collaboration. The **Name** and **Description** helps them understand what the collaboration is in reference to.

   b. For **Members**:

      i. For **Member 1: You**, enter your **Member display name** as you want it to appear for the collaboration.

         > **Note**
         >
         > Your AWS account ID is included automatically for **Member AWS account ID**.

      ii. For **Member 2**, enter the **Member display name** and **Member AWS account ID** for the member that you want to invite to the collaboration.

         The **Member display name** and **Member AWS account ID** will be visible to everyone invited to the collaboration. After you enter and save the values for these fields, they are not editable.

         > **Note**
         >
         > You must inform the collaboration member that their **Member AWS account ID** and **Member display name** will be visible to all invited and active collaborators in the collaboration.

      iii. If you want to add another member, choose **Add another member**. Then enter the **Member display name** and **Member AWS account ID** for each member who can contribute data that you want to invite to the collaboration.

c. For **Member abilities**, choose one of the following,

If you want to ...	Then ...
Query the data in the collabora tion and receive the results	1. Choose yourself as the member who can **Run queries**. 2. Leave the default setting of the member who can **Receive results** is the **Same as who runs queries**.
Query the data in the collabora tion and assign a different member to receive results	1. Choose yourself as the member who can **Run queries**. 2. Select the member who can **Receive results** from the dropdown list.
Receive the results of the query in the collaboration and assign a different member to query the data	1. Select the member who can **Run queries** from the dropdown list. 2. Choose yourself as member who can **Receive results** from the dropdown list.
Create and manage the collaboration, assign a different member to query the data, and assign a different member to receive results	1. Select the member who can **Run queries** from the dropdown list. 2. Select the member who can **Receive results** from the dropdown list.

d. For **Payment configuration**, choose one of the following:

If you want to ...	Then ...
Assign the member who can **Run queries** to be the member who pays for the query compute costs	Leave the default setting of the member who will **Pay for queries** is the **Same as who runs queries**.
Assign a different member to pay for the query compute costs	Select the member who will **Pay for queries** from the dropdown list.

e. If you want to enable **Query logging**, select the **Support query logging for this collaboration** check box.

f. If you want to enable the **Cryptographic computing** capability, select the **Support cryptographic computing in this collaboration** check box and choose the following **Cryptographic computing parameters**:

   - **Allow cleartext columns**

     Choose **No** if you don't want cleartext columns allowed in the encrypted table.

     Choose **Yes** if you want cleartext columns allowed in the encrypted table.

     To run SUM or AVG on certain columns, the columns must be in cleartext.

   - **Allow duplicates**

     Choose **No** if you don't want duplicate entries allowed in a fingerprint column.

     Choose **Yes** if you want duplicate entries allowed in a fingerprint column.

   - **Allow JOIN of columns with different names**

     Choose **No** if you don't want to join fingerprint columns with different names.

     Choose **Yes** if you want to join fingerprint columns with different names.

   - **Preserve NULL values**

     Choose **No** if you don't want to preserve NULL values. NULL values won't appear as NULL in an encrypted table.

     Choose **Yes** if you want to preserve NULL values. NULL values will appear as NULL in an encrypted table.

   For more information about **Cryptographic computing parameters**, see [Cryptographic computing parameters](#).

   For more information about how to encrypt your data for use in AWS Clean Rooms, see [Preparing encrypted data tables with Cryptographic Computing for Clean Rooms](#).

> **Note**
>
> Verify these configurations carefully before completing the next step. After you create the collaboration, you can only edit the collaboration name, description, and whether the query logs are stored in Amazon CloudWatch Logs.

g. If you want to enable **Tags** for the collaboration resource, choose **Add new tag** and then enter the **Key** and **Value** pair.

h. Choose **Next**.

5. For **Step 2: Configure membership**, do the following:

a. Choose one option:

If you choose...	Then ...
Yes, join by creating membership now	Both the collaboration and your membership are created. Your status in the collaboration is active.
No, I will create a membership later	Only the collaboration is created. Your status in the collaboration is inactive.

b. If you are the member who can **Receive results**, under **Query results settings defaults**, choose one option:

If you ...	Then ...
Keep the **Set default settings now** check box selected. (It is selected by default.)	1. For the **Results destination in Amazon S3**, enter the Amazon S3 destination. 2. For the query **Result format**, choose either **CSV** or **PARQUET**.
Clear the **Set default settings now** check box	Only the collaboration is created. Your status in the collaboration is inactive.

c. If you chose to enable **Query logging** in step 4.e, choose one of the following options for **Log storage in Amazon CloudWatch Logs**:

If you choose...	Then ...
Turn on	The query logs relevant to you are stored in Amazon CloudWatch Logs. Each member can receive only logs for queries that they initiated or that contain their data. The member who can receive results also receives logs for all queries run in a collaboration, even if their data is not accessed in a query.
Turn off	The query logs relevant to you aren't stored in your Amazon CloudWatch Logs account.

Note

After you turn on **Query logging**, it can take a few minutes for log storage to be set up and start receiving logs in Amazon CloudWatch Logs. During this brief period, the member who can query might run queries that don't actually send logs.

d. If you want to enable **Tags** for the membership resource, choose **Add new tag** and then enter the **Key** and **Value** pair.

e. If you are the member who is **Paying for queries**, indicate your acceptance by selecting the **I agree to pay for the query compute costs in this collaboration** check box.

Note

You must select this check box to proceed.

For more information about how pricing is calculated, see Pricing for AWS Clean Rooms.

If you are the member paying for query compute costs but not the member who can query, it is recommended that you use AWS Budgets to configure a budget for AWS Clean Rooms and receive notifications once the maximum budget has been reached. For more information about setting up a budget, see Managing your costs with AWS Budgets in the *AWS Cost Management User Guide*. For more information about setting up notifications, see Creating an Amazon SNS topic for budget notifications in the *AWS Cost Management User Guide*. If the maximum budget has been reached, you can contact the member who can run queries or leave the collaboration. If you leave the collaboration, no more queries will be allowed to run, and therefore you will no longer be billed for query compute costs.

f. Choose **Next**.

6. For **Step 3: Review and create**, do the following:

a. Review the selections that you made for the previous steps and edit if necessary.

b. Choose one of the following:

If you have chosen to...	Then choose...
Create a membership with the collaboration (**Yes, join by creating membership now**)	**Create collaboration and membership**
Create the collaboration, and not to create a membership at this time (**No, I will create a membership later**)	**Create collaboration**

After your collaboration has been created successfully, you can see the collaboration details page under **Collaborations**.

Next steps

You are now ready to:

- Prepare your data table to be queried in AWS Clean Rooms. (Optional if you want to query your own data.)
- Associate the configured table to your collaboration. (Optional if you want to query your own data.)
- Configure an analysis rule for the configured table. (Optional if you want to query your own data.)
- Create a membership and join a collaboration.
- Manage your collaboration.

Creating a membership and joining a collaboration

A *membership* is a resource that is created when a member joins a collaboration in AWS Clean Rooms.

You can join a collaboration as a member who can query data, member who can receive results of a query, or both. You can also join a collaboration as a member paying for query compute costs. All members can contribute data.

For information about how to create a membership and join a collaboration using the AWS SDKs, see the AWS Clean Rooms API Reference.

Topics

Create a membership and join a collaboration

To create a membership and join a collaboration

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your member AWS account.
2. In the left navigation pane, choose **Collaborations**.
3. On the **Available to join** tab, for **Collaborations available to join**, choose the **Name** of the collaboration.
4. On the collaboration details page, view the collaboration details, including your member details and a list of the other members.

   Verify that the AWS account IDs for each member of the collaboration are the ones with whom you intend to enter in to the collaboration.
5. Choose **Create membership**.
6. On the **Create membership** page, in the **Overview**, view the **Collaboration name**, **Collaboration description**, AWS account ID of the **Collaboration creator**, **Your member abilities**, and the AWS account ID of the member who will **Pay for queries**.
7. If the collaboration creator has chosen to enable **Query logging**, choose one of the following options for **Log storage in Amazon CloudWatch Logs**:

If you choose...	Then ...
Turn on	The query logs relevant to you are stored in Amazon CloudWatch Logs. Each member can receive only logs for queries that they initiated or that contain their data. The member who can receive results also receives logs for all queries run in a collaboration, even if their data isn't accessed in a query.
Turn off	The query logs relevant to you aren't stored in your Amazon CloudWatch Logs account.

> **Note**
>
> After you turn on **Query logging**, it can take a few minutes for log storage to be set up and start receiving logs in Amazon CloudWatch Logs. During this brief period, the member who can query might run queries that don’t actually send logs.

8. If **Your member abilities** includes **Receive results**:

   a. For **Query results settings**,

      i. Specify the **Results destination in Amazon S3** by entering the S3 destination or choose **Browse S3** to select from a list of available S3 buckets.

         Example

         For example: **`s3://bucket/prefix`**

      ii. Choose the **Result format** (either **CSV** or **PARQUET**).

   b. For **Service access**, choose to either **Create and use a new service role** or **Use an existing service role**.

> **Note**
>
> You must either select an existing service role or have permissions to create a new one. For more information, see Create a service role to receive results.

9. If you want to enable **Tags** for the membership resource, choose **Add new tag** and then enter the **Key** and **Value** pair.
10. If the collaboration creator has designated you as the member who will **Pay for queries**, indicate your acceptance by selecting the **I agree to pay for the query compute costs in this collaboration** check box.

    > **Note**
    >
    > You must select this check box to proceed.
    > For more information about how pricing is calculated, see Pricing for AWS Clean Rooms.

    If you are the member paying for query compute costs but not the member who can query, it is recommended that you use AWS Budgets to configure a budget for AWS Clean Rooms and receive notifications once the maximum budget has been reached. For more information about setting up a budget, see Managing your costs with AWS Budgets in the *AWS Cost Management User Guide*. For more information about setting up notifications, see Creating an Amazon SNS topic for budget notifications in the *AWS Cost Management User Guide*. If the maximum budget has been reached, you can contact the member who can run queries or leave the collaboration. If you leave the collaboration, no more queries will be allowed to run, and therefore you will no longer be billed for query compute costs.
11. If you are sure that you want to create a membership and join the collaboration, choose **Create membership**.

You are given read access to the collaboration metadata. This includes information such as the display name and description of the collaboration, in addition to all the names and AWS account IDs of other members.

For information about how to leave a collaboration, see Leaving a collaboration.

Next steps

You are now ready to:

- Prepare your data table to be queried in AWS Clean Rooms. (Optional if you want to query your own data.)
- Associate the configured table to your collaboration.
- Configure an analysis rule for the configured table.

Preparing data tables for queries in AWS Clean Rooms

> **Note**
>
> Preparing data tables can take place before or after you have joined a collaboration. After a table is prepared, you can reuse it across multiple collaborations as long as your privacy needs for that table are the same.

As a member in the collaboration, you must prepare your data tables before they can be queried in AWS Clean Rooms by the collaboration member who can query .

If your use case doesn't require you to bring your own data, you can skip this procedure.

If your data tables are already cataloged in AWS Glue, skip to Creating a configured table in AWS Clean Rooms.

Preparing your data tables involves the following steps:

- Step 1: Complete the prerequisites
- Step 2: (Optional) Prepare your data for cryptographic computing
- Step 3: Upload your data table to Amazon S3
- Step 4: Create an AWS Glue table
- Next steps

For more information about the data formats that you can use for queries, see Data formats for AWS Clean Rooms.

Step 1: Complete the prerequisites

To prepare your data tables for use with AWS Clean Rooms, you must complete the following prerequisites:

- Your datasets must be saved as one of the supported data formats for AWS Clean Rooms.
- Your data tables must be cataloged in AWS Glue and use the supported data types for AWS Clean Rooms.

- All of your data tables must be stored in Amazon Simple Storage Service (Amazon S3) in the same AWS Region in which the collaboration was created.
- The AWS Glue Data Catalog must be in the same Region in which the collaboration was created.
- The AWS Glue Data Catalog must be in the same AWS account as the membership.
- The Amazon S3 bucket can't be registered with AWS Lake Formation.
- The collaboration creator has set up a collaboration in AWS Clean Rooms. For more information, see Creating a collaboration in AWS Clean Rooms.
- The collaboration creator has sent the collaboration ID to you as a participant in the collaboration.

Step 2: (Optional) Prepare your data for cryptographic computing

(Optional) If you're using cryptographic computing and your data table contains sensitive information that you want to encrypt, you must encrypt the data table using the C3R encryption client.

To prepare your data for cryptographic computing, follow the procedures in Preparing encrypted data tables with Cryptographic Computing for Clean Rooms.

Step 3: Upload your data table to Amazon S3

> **Note**
>
> If you intend to use encrypted data tables in the collaboration, you must first encrypt the data for cryptographic computing before you upload your data table to Amazon S3. For more information, see Preparing encrypted data tables with Cryptographic Computing for Clean Rooms.

To upload your data table to Amazon S3

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose **Buckets**, and then choose a bucket where you want to store your data table.

3. Choose **Upload**, and then follow the prompts.
4. Choose the **Objects** tab to view the prefix where your data is stored. Make a note of the name of the folder.

   You can select the folder to view the data.

Step 4: Create an AWS Glue table

If you already have an AWS Glue data table, you can skip this step.

In this step, you set up a crawler in AWS Glue that crawls all the files in your S3 bucket and creates an AWS Glue table. For more information, see Defining crawlers in AWS Glue in the *AWS Glue User Guide*.

For more information about supported AWS Glue Data Catalog data types, see Supported data types.

> **Note**
>
> AWS Clean Rooms doesn't currently support S3 buckets registered with AWS Lake Formation.

The following procedure describes how to create an AWS Glue table. If you want to use an encrypted AWS Glue Data Catalog object with an AWS Key Management Service (AWS KMS) key, you need to configure the KMS key permissions policy to allow access to that encrypted table. For more information, see Setting up encryption in AWS Glue in the *AWS Glue Developer Guide*.

To create an AWS Glue table

1. Follow the Working with crawlers on the AWS Glue console procedure in the *AWS Glue User Guide*.
2. Make a note of the AWS Glue database name and AWS Glue table name.

Next steps

Now that you have prepared your data tables, you are ready to:

- Create a configured table
- Create an ML model

Data formats for AWS Clean Rooms

The datasets that you use for queries in AWS Clean Rooms are commonly the same types of datasets that you use for other applications. For example, the same types of datasets are used with Amazon Athena, Amazon EMR, Amazon Redshift Spectrum, and Amazon QuickSight. You can query the data in its original format directly from Amazon Simple Storage Service (Amazon S3).

To query data, the datasets must be in a format that AWS Clean Rooms supports. The Amazon S3 bucket with the datasets and the AWS Clean Rooms cluster must be in the same AWS Region.

Supported data formats

AWS Clean Rooms supports the following structured formats:

- Apache Iceberg tables
- Parquet
- RCFile
- TextFile
- SequenceFile
- RegexSerde
- OpenCSV
- AVRO
- JSON

> **Note**
>
> A `timestamp` value in a text file must be in the format `yyyy-MM-dd HH:mm:ss.SSSSSS`. For example: `2017-05-01 11:30:59.000000`.

We recommend using a columnar storage file format, such as Apache Parquet. With a columnar storage file format, you can minimize data transfer out of Amazon S3 by selecting only the columns that you need. For optimal performance, large objects should be split into 100mb–1gb objects.

Supported data types

For an optimal experience with AWS Clean Rooms, all of your data must be cataloged in AWS Glue. For more information, see the section titled Getting started with the AWS Glue Data Catalog in the *AWS Glue Developer Guide*.

AWS Clean Rooms supports the following AWS Glue Data Catalog data types:

- bigint
- boolean
- char
- date
- decimal
- double
- float
- int
- Nested data types such as:
  - array
  - map
  - struct
- smallint
- string
- timestamp
- varchar

AWS Clean Rooms does not support:

- binary
- interval

File compression types for AWS Clean Rooms

To reduce storage space, improve performance, and minimize costs, we strongly recommend that you compress your datasets.

AWS Clean Rooms recognizes file compression types based on the file extension and supports the compression types and extensions shown in the following table.

Compression algorithm	File extension
GZIP	.gz
Bzip2	.bz2
Snappy	.snappy

You can apply compression at different levels. Most commonly, you compress a whole file or compress individual blocks within a file. Compressing columnar formats at the file level doesn't yield performance benefits.

Server-side encryption for AWS Clean Rooms

Note

Server-side encryption does not replace cryptographic computing for those use cases that require it.

AWS Clean Rooms transparently decrypts datasets that are encrypted using the following encryption options:

- **SSE-S3 –** Server-side encryption using an AES-256 encryption key managed by Amazon S3
- **SSE-KMS –** Server-side encryption with keys managed by AWS Key Management Service

To use SSE-S3, the AWS Clean Rooms service role used to associate the configured table to the collaboration must have KMS-decrypt permissions. To use SSE-KMS, the KMS key policy must also allow the AWS Clean Rooms service role to decrypt.

AWS Clean Rooms doesn't support Amazon S3 client-side encryption. For more information about server-side encryption, see [Protecting data using server-side encryption]() in the *Amazon Simple Storage Service User Guide*.

Using Apache Iceberg tables in AWS Clean Rooms

Apache Iceberg is an open source table format for data lakes. AWS Clean Rooms can use the statistics stored in Apache Iceberg metadata to optimize query plans and reduce file scans during clean room query processing. For more information, see the [Apache Iceberg]() documentation.

Consider the following when using AWS Clean Rooms with Iceberg tables:

- **Tables within the AWS Glue Data Catalog only** – Apache Iceberg tables must be defined in the AWS Glue Data Catalog based on the [open source glue catalog implementation]().
- **Parquet file format** – AWS Clean Rooms only supports Iceberg tables in the Parquet data file format.
- **GZIP and Snappy compression** – AWS Clean Rooms supports Parquet with GZIP and Snappy compression.
- **Iceberg versions** – AWS Clean Rooms supports running queries against version 1 and version 2 Iceberg tables.
- **Partitions** – You don't need to manually add partitions for your Apache Iceberg tables in AWS Glue. AWS Clean Rooms detects new partitions in Apache Iceberg tables automatically and no manual operation is needed to update partitions in the table definition. Iceberg partitions appear as regular columns in the AWS Clean Rooms table schema and not separately as a partition key in the configured table schema.
- **Limitations**
  - **New Iceberg tables only**

    Apache Iceberg tables converted from Apache Parquet tables are not supported.
  - **Time travel queries**

    AWS Clean Rooms does not support time travel queries with Apache Iceberg tables.
  - **Athena engine version 2**

    Iceberg tables created with Athena engine version 2 are not supported.
  - **File formats**

Avro and Optimized Row Columnar (ORC) file formats are not supported.

- **Compression**

  Zstandard (Zstd) compression for Parquet is not supported.

Supported data types for Iceberg tables

AWS Clean Rooms can query Iceberg tables that contain the following data types:

- boolean
- date
- decimal
- double
- float
- int
- list
- long
- map
- string
- struct
- timestamp without time zone

For more information about Iceberg data types, see the Schemas for Iceberg in the Apache Iceberg documentation.

Preparing encrypted data tables with Cryptographic Computing for Clean Rooms

Cryptographic Computing for Clean Rooms (C3R) is a capability in AWS Clean Rooms. You can use C3R to limit cryptographically what can be learned by any party and AWS in an AWS Clean Rooms collaboration.

You can encrypt the data table using the C3R encryption client, a client-side encryption tool, before uploading the data table to Amazon Simple Storage Service (Amazon S3).

For more information, see Cryptographic Computing for Clean Rooms.

Preparing encrypted data tables with C3R involves the following steps:

Steps

Step 1: Complete the prerequisites

To prepare your data tables for use with C3R, you must complete the following prerequisites:

- You can access the Cryptographic Computing for Clean Rooms repository on GitHub:

  https://github.com/aws/c3r
- You have set up AWS credentials to use the C3R encryption client. These credentials are used by C3R encryption client for read-only API calls to AWS Clean Rooms to retrieve collaboration

metadata. For more information, see Configuring the AWS CLI in the *AWS Command Line Interface User Guide for Version 2*.

- You have Java Runtime Environment (JRE) 11 or later installed on your machine.
  - The recommended Java Runtime Environment, Amazon Corretto 11 or higher, can be downloaded from https://aws.amazon.com/corretto.
  - The Java Development Kit (JDK) includes a corresponding JRE of the same version. However, the additional capabilities of the JDK are not needed for running the Cryptographic Computing for Clean Rooms (C3R) encryption client.
- Your tabular data files (.csv) or Parquet files (.parquet) are saved locally.
- You or another member in the collaboration has the ability to create a shared secret key. For more information, see Step 5: Create a shared secret key.
- The collaboration creator has created a collaboration in AWS Clean Rooms with **Cryptographic computing** enabled for the collaboration. For more information, see Creating a collaboration in AWS Clean Rooms.
- The collaboration creator has sent the collaboration ID to you as a participant in the collaboration. The collaboration Amazon Resource Name (ARN) is included in the invitation that is sent, which contains the collaboration ID.

Step 2: Download the C3R encryption client

To download the C3R encryption client from GitHub

1. Go to the Cryptographic Computing for Clean Rooms AWS GitHub repository: https://github.com/aws/c3r
2. Select and download the files.

   The source code, licenses, and related material can be cloned or downloaded as a .zip file from the GitHub repository's landing page. (See the **Code** button at the top-right of the repository's content list).

   The latest signed C3R encryption client Java Executable File (that is, the command line interface application) is on the **Releases** page of the GitHub repository.

   The C3R encryption client package for Apache Spark (`c3r-cli-spark`) is a version of the c3r-cli that must be submitted as a job to a running Apache Spark server. For more information, see Running C3R on Apache Spark.

(Optional) Step 3: View available commands in the C3R encryption client

Use this procedure to familiarize yourself with the available commands in the C3R encryption client.

To view all of the available commands in the C3R encryption client

1. From a command line interface (CLI), navigate to the folder that contains the downloaded c3r-cli.jar file.
2. Run the following command: `java -jar c3r-cli.jar`
3. View the list of available commands and options.

Step 4: Generate an encryption schema for a tabular file

To encrypt data, an encryption schema describing how the data will be used is required. This section describes how the C3R encryption client assists in generating an encryption schema for a CSV file with a header row or a Parquet file.

You only need to do this once per file. After the schema exists, it can be re-used to encrypt the same file (or any file with identical column names). If the column names or desired encryption schema changes, you must update the schema file. For more information, see [(Optional) Create a schema (advanced users)](#).

> **Important**
>
> It is paramount that all collaborating parties use the same shared secret key. Collaborating parties should also coordinate column names to match if they will be JOINed or otherwise compared for equality in queries. Otherwise, the SQL queries might produce unexpected or incorrect results. However, this is not necessary if the collaboration creator enabled the `allowJoinsOnColumnsWithDifferentNames` encryption setting during collaboration creation. For more information about encryption-relevant settings, see [Cryptographic computing parameters](#).

When run in schema mode, the C3R encryption client goes through the input file column by column, prompting you if and how that column should be treated. If the file contains many columns that aren't wanted for the encrypted output, the interactive schema generation might become tedious because you must skip each undesired column. To avoid this, you could manually write a schema, or create a simplified version of the input file featuring only the wanted columns. Then, the interactive schema generator could be run on that reduced file. The C3R encryption client outputs information about the schema file and asks you how the source columns should be included or encrypted (if at all) in the target output.

For each source column in the input file, you are prompted for:

1. How many target columns should be generated
2. How each target column should be encrypted (if at all)
3. The name of each target column
4. How data should be padded before encryption if the column is being encrypted as a sealed column

> **Note**
>
> When you encrypt data for a column that has been encrypted as a sealed column, you must determine which data needs padding. The C3R encryption client suggests a default padding during schema generation that pads all entries in a column to the same length.
> When determining the length for `fixed`, note that padding is in bytes, not bits.

The following is a decision table for creating the schema.

Schema decision table

Decision	Number of target columns from source column <‘name-of-column’> ?	Target column type: [c] cleartext, [f] fingerprint, or [s] sealed ?	Target column headername <default 'name-of-column'>	Add suffix <suffix> to header to indicate how it was encrypted, [y] yes or [n] no <default 'yes'>	<‘name-of-column_sealed’> padding type: [n] one, [f] fixed, or [m] max <default ’max’>
Leave the column unencrypted.	1	**c**	Not applicable	Not applicable	Not applicable
Encrypt the column as a fingerprint column.	1	**f**	Choose default or enter a new header name.	Enter `y` to choose default (`_fingerprint`) or enter `n`.	Not applicable
Encrypt the column as a sealed column.	1	**s**	Choose default or enter a new header name.	Enter `y` to choose default (`_sealed`) or enter `n`.	Choose padding type . For more information, see [(Optional) Create a schema (advanced users)]().

Decision	Number of target columns from source column <'name-of-column'> ?	Target column type: [c] cleartext , [f] fingerprint, or [s] sealed ?	Target column headername <default 'name-of-column'>	Add suffix <suffix> to header to indicate how it was encrypted, [y] yes or [n] no <default 'yes'>	<'name-of-column_sealed'> padding type: [n] one, [f] fixed, or [m] max <default 'max'>
Encrypt the column as both fingerprint and sealed.	2	Enter first target column: **f** . Enter second target column: **s**.	Choose the target headers for each target column.	Enter y to choose default or enter n **.**	Choose padding type (for sealed columns only). For more information, see [(Optional) Create a schema (advanced users)]().

The following are two examples of how to create encryption schemas. The exact content of your interaction depends on the input file and the responses that you provide.

Examples

- Example: Generate an encryption schema for a fingerprint column and a cleartext column
- Example: Generate an encryption schema with sealed, fingerprint, and cleartext columns

Example: Generate an encryption schema for a fingerprint column and a cleartext column

In this example, for `ads.csv`, there are only two columns: `username` and `ad_variant`. For these columns, we want the following:

- For the `username` column to be encrypted as a `fingerprint` column
- For the `ad_variant` column to be a `cleartext` column

To generate an encryption schema for a fingerprint column and a cleartext column

1. (*Optional*) To ensure the c3r-cli.jar file and file to be encrypted are present:

   a. Navigate to the desired directory and run `ls` (if using a Mac or Unix/Linux) or `dir` if using Windows).

   b. View the list of tabular data files (for example, .csv) and choose a file to encrypt.

      In this example, `ads.csv` is the file that we want to encrypt.

2. From the CLI, run the following command to create a schema interactively.

```
java -jar c3r-cli.jar schema ads.csv --interactive --output=ads.json
```

> **Note**
>
> - You can run `java --jar PATH/TO/c3r-cli.jar`. Or, if you have added `PATH/TO/c3r-cli.jar` to your CLASSPATH environment variable, you can also run the class name. The C3R encryption client will look in the CLASSPATH to find it (for example, `java com.amazon.psion.cli.Main`).
> - The `--interactive` flag selects the interactive mode for developing the schema. This walks the user through a wizard for creating the schema. Users with advanced skills can create their own schema JSON without using the wizard. For more information, see [(Optional) Create a schema (advanced users)](#).
> - The `--output` flag sets an output name. If you don't include the `--output` flag, the C3R encryption client tries to pick a default output name (such as `<input>.out.csv` or for the schema, `<input>.json`).

3. For `Number of target columns from source column 'username'?`, enter **1** and then press **Enter**.
4. For `Target column type: [c]leartext, [f]ingerprint, or [s]ealed?`, enter **f** and then press **Enter**.
5. For `Target column headername <default 'username'>`, press **Enter**.

   The default name '`username`' is used.
6. For `Add suffix '_fingerprint' to header to indicate how it was encrypted, [y]es or [n]o <default 'yes'>`, enter **y** and then press **Enter**.

   > **Note**
   >
   > The interactive mode suggests suffixes to add to the encrypted column headers (`_fingerprint` for fingerprint columns and `_sealed` for sealed columns). The suffixes might be helpful when you're performing tasks such as uploading data to AWS services or creating AWS Clean Rooms collaborations. These suffixes can help indicate what can be done with the encrypted data in each column. For example, things will not work if you encrypt a column as a sealed column (`_sealed`) and try to JOIN on it or try the reverse.

7. For `Number of target columns from source column 'ad_variant'?`, enter **1** and then press **Enter**.
8. For `Target column type: [c]leartext, [f]ingerprint, or [s]ealed?`, enter **c** and then press **Enter**.
9. For `Target column headername <default 'username'>`, press **Enter**.

   The default name '`ad_variant`' is used.

   The schema is written to a new file called `ads.json`.

   > **Note**
   >
   > You can view the schema by opening it in any text editor, such as Notepad on Windows or TextEdit on macOS.

10. You are now ready to [encrypt data](#).

Example: Generate an encryption schema with sealed, fingerprint, and cleartext columns

In this example, for `sales.csv`, there are three columns: `username` , `purchased`, and `product`. For these columns, we want the following:

- For the `product` column to be a `sealed` column
- For the `username` column to be encrypted as a `fingerprint` column
- For the `purchased` column to be a `cleartext` column

To generate an encryption schema with sealed, fingerprint, and cleartext columns

1. (*Optional*) To ensure the c3r-cli.jar file and file to be encrypted are present:

   a. Navigate to the desired directory and run `ls` (if using a Mac or Unix/Linux) or `dir` if using Windows).

   b. View the list of tabular data files (.csv) and choose a file to encrypt.

      In this example, `sales.csv` is the file that we want to encrypt.

2. From the CLI, run the following command to create a schema interactively.

   ```
   java -jar c3r-cli.jar schema sales.csv --interactive --
   output=sales.json
   ```

   > **Note**
   >
   > - The `--interactive` flag selects the interactive mode for developing the schema. This walks the user through a guided workflow for creating the schema.
   > - If you are an advanced user, you can create your own schema JSON without using the guided workflow. For more information, see [(Optional) Create a schema (advanced users)](#).
   > - For .csv files with no column headers, see the `--noHeaders` flag for the schema command available in the CLI.

- The --output flag sets an output name. If you don't include the --output flag, the C3R encryption client tries to pick a default output name (such as <input>.out or for the schema, <input>.json).

3. For Number of target columns from source column ‘username’?, enter **1** and then press **Enter**.
4. For Target column type: [c]leartext, [f]ingerprint, or [s]ealed?, enter **f** and then press **Enter**.
5. For Target column headername <default 'username'>, press **Enter**.

   The default name ‘username’ is used.
6. For Add suffix '_fingerprint' to header to indicate how it was encrypted, [y]es or [n]o <default 'yes'>, enter **y** and then press **Enter**.
7. For Number of target columns from source column ‘purchased’?, enter **1** and then press **Enter**.
8. For Target column type: [c]leartext, [f]ingerprint, or [s]ealed?, enter **c** and then press **Enter**.
9. For Target column headername <default 'purchased'>, press **Enter**.

   The default name ‘purchased’ is used.
10. For Number of target columns from source column ‘product’?, enter **1** and then press **Enter**.
11. For Target column type: [c]leartext, [f]ingerprint, or [s]ealed?, enter **s** and then press **Enter**.
12. For Target column headername <default 'product'>, press **Enter**.

    The default name ‘product’ is used.
13. For ‘product_sealed’ padding type: [n]one, [f]ixed, or [m]ax <default ’max’?>, press **Enter** to choose the default.
14. For Byte-length beyond max length to pad cleartext to in ‘product_sealed’ <default ‘0’>? press **Enter** to choose the default.

    The schema is written to a new file called sales.json.
15. You are now ready to encrypt data.

Step 5: Create a shared secret key

To encrypt the data tables, the collaboration participants must agree upon and securely share a shared secret key.

The shared secret key must be at least 256-bits (32 bytes). You can specify a larger key, but it won't give you any additional security.

> **Important**
>
> Remember, the key and collaboration ID used for encryption and decryption must be identical for all collaboration participants.

The following sections provide examples of console commands for generating a shared secret key saved as `secret.key` in the respective terminal's current working directory.

Topics

Example: Key generation using OpenSSL

For a common general purpose cryptography library, run the following command to create a shared secret key.

```
openssl rand 32 > secret.key
```

If you're using Windows and don't have OpenSSL installed, you can generate keys using the example described in Example: Key generation on Windows using PowerShell.

Example: Key generation on Windows using PowerShell

For PowerShell, a terminal application available on Windows, run the following command to create a shared secret key.

```
$bs = New-Object Byte[](32);
[Security.Cryptography.RandomNumberGenerator]::Create().GetBytes($bs); Set-
Content 'secret.key' -Encoding Byte -Value $bs
```

Step 6: Store the shared secret key in an environment variable

An environment variable is a convenient and extensible way for users to provide a secret key from various key stores like AWS Secrets Manager and pass it to the C3R encryption client.

The C3R encryption client can use keys stored in AWS services if you use the AWS CLI to store those keys in the relevant environment variable. For example, the C3R encryption client can use a key from AWS Secrets Manager. For more information, see Create and manage secrets with AWS Secrets Manager in the *AWS Secrets Manager User Guide*.

> **Note**
>
> However, before you use an AWS service such as AWS Secrets Manager to hold your C3R keys, verify that your use case permits it. Certain use cases might require that the key be withheld from AWS. This is to ensure that the encrypted data and the key are never held by the same third party.

The only requirements for a shared secret key are that the shared secret key is base64-encoded and stored in the environment variable `C3R_SHARED_SECRET`.

The following sections describe the console commands for converting a `secret.key` file to base64 and storing it as an environment variable. The `secret.key` file could have been generated from any of the commands listed in Step 5: Create a shared secret key and is only an example source.

Store key in an environment variable on Windows using PowerShell

To convert to base64 and set the environment variable on Windows using PowerShell, run the following command.

```
$Bytes=[IO.File]::ReadAllBytes((Get-Location).ToString()+'\secret.key');
 $env:C3R_SHARED_SECRET=[Convert]::ToBase64String($Bytes)
```

Store key in an environment variable on Linux or macOS

To convert to base64 and set the environment variable on Linux or macOS, run the following command.

```
export C3R_SHARED_SECRET="$(cat secret.key | base64)"
```

Step 7: Encrypt data

To perform this step, you must acquire the AWS Clean Rooms collaboration ID and the shared secret key. For more information, see the Prerequisites.

In the following example, we run the encryption on `ads.csv`, using the schema that we created called `ads.json`.

To encrypt data

1. Store the shared secret key for the collaboration in Step 6: Store the shared secret key in an environment variable.
2. From the command line, enter the following command.

   ```
   java -jar c3r-cli.jar encrypt <name of input .csv file> --schema=<name of schema .json file> --id=<collaboration id> --output=<name of output.csv file> <optional flags>
   ```
3. For *<name of input .csv file>*, enter the name of the input .csv file.
4. For `schema=`, enter the name of the .json encryption schema file.
5. For `id=`, enter the collaboration ID.
6. For `output=`, enter the name of the output file (for example, `ads-output.csv`).
7. Include any of the command line flags described in Cryptographic computing parameters and Optional flags in Cryptographic Computing for Clean Rooms.
8. Run the command.

In the example for `ads.csv`, we run the following command.

java -jar c3r-cli.jar encrypt *ads.csv* --schema=*ads.json* --id=*123e4567-e89b-42d3-a456-556642440000* --output=*ads-output.csv*

In the example for **sales.csv**, we run the following command.

java -jar c3r-cli.jar encrypt *sales.csv* --schema=*sales.json* --id=*123e4567-e89b-42d3-a456-556642440000*

> **Note**
>
> In this example, we don't specify an output file name (`--output=sales-output.csv`). As a result, the default output file name `name-of-file.out.csv` was generated.

You are now ready to verify the encrypted data.

Step 8: Verify data encryption

To verify that the data was encrypted

1. View the encrypted data file (for example, `sales-output.csv`).
2. Verify the following columns:
   a. Column **1** – Encrypted (for example, `username_fingerprint`).

      For the fingerprint columns (HMAC), after the version and type prefix (for example, `01:hmac:`), there are 44 characters of base64-encoded data.
   b. Column **2** – Not encrypted (for example, `purchased`).
   c. Column **3** – Encrypted (for example, `product_sealed`).

      For encrypted (SELECT) columns, the length of the cleartext plus any padding after the version and type prefix (for example, `01:enc:`) is directly proportional to the length of the cleartext that was encrypted. That is, the length is the size of the input plus approximately 33 percent overhead because of the encoding.

You are now ready to:

1. Upload the encrypted data to S3.
2. Create an AWS Glue table.
3. Create a configured table in AWS Clean Rooms.

The C3R encryption client will create temporary files that don't contain unencrypted data (unless that data would also be unencrypted in the final output). However, some encrypted values might not be padded properly. Fingerprint columns might contain duplicate values, even if the collaboration setting `allowRepeatedFingerprintValue` is `false`. This issues occurs because the temporary file is written before proper padding lengths and duplicate-removal properties are checked.

If the C3R encryption client fails or is interrupted during encryption, it might stop after writing the temporary file but before checking these properties and deleting the temporary files. Therefore, these temporary files might still be on disk. If this is the case, the contents in these files doesn't protect the plaintext data to the same levels that the output does. In particular, these temporary files might reveal plaintext data to statistical analyses that would not work against the final output. The user should delete these files (particularly a SQLite database) to prevent these files from falling into unauthorized hands.

(Optional) Create a schema (advanced users)

Creating a schema manually is for advanced users.

The following is a description of the JSON schema file format for input files with or without column headers. Advanced users can directly write or modify the schema if desired.

> **Note**
>
> The C3R encryption client can assist you in making a schema through either the interactive process described in Example: Generate an encryption schema with sealed, fingerprint, and cleartext columns or through the creation of a stub template.

Mapped and positional table schemas

The following section describes two kinds of table schemas:

- **Mapped table schema –** This schema is used for encrypting .csv files with a header row and Apache Parquet files.
- **Positional table schema –** This schema is used for encrypting .csv files without a header row.

The C3R encryption client can encrypt a tabular file for a collaboration. To do this, it must have a corresponding schema file that specifies how the encrypted output should be derived from the input.

The C3R encryption client can help generate a schema for an `INPUT` file by running the C3R encryption client schema command at the command line. An example of a command is `java -jar c3r-cli.jar schema --interactive INPUT`.

The schema specifies the following information:

1. Which source columns map to which transformed columns in the output file through their header names (mapped schemas) or position (positional schemas)
2. Which target columns are to remain cleartext
3. Which target columns are to be encrypted for SELECT queries
4. Which target columns are to be encrypted for JOIN queries

This information is encoded in a table-specific JSON schema file, which consists of a single object whose `headerRow` field is a Boolean value. The value must be `true` for Parquet files and .csv files with a header row, and `false` otherwise.

Mapped table schema

The mapped schema has the following shape.

```
{
  "headerRow": true,
  "columns": [
    {
      "sourceHeader": STRING,
      "targetHeader": STRING,
      "type": TYPE,
      "pad": PAD
    },
    ...
  ]
}
```

If `headerRow` is `true`, the next field in the object is `columns`, which contains an array of column schemas that map source headers to target headers (that is, JSON objects describing what the output columns should contain).

- `sourceHeader` – The `STRING` header name of the source column that the data is derived from.

> **Note**
>
> The same source column can be used for multiple target columns.
> A column from the input file not listed as a `sourceHeader` anywhere in the schema doesn't appear in the output file.

- `targetHeader` – The `STRING` header name of the corresponding column in the output file.

> **Note**
>
> This field is optional for mapped schemas. If this field is omitted, the `sourceHeader` is re-used for the header name in the output. Either `_fingerprint` or `_sealed` is appended if the output column is a fingerprint column or sealed column respectively.

- `type` – The `TYPE` of the target column in the output file. That is, one of `cleartext`, `sealed`, or `fingerprint` depending on how the column will be used in the collaboration.
- `pad` – A field of a column schema object that is only present when the `TYPE` is `sealed`. Its corresponding value of `PAD` is an object that describes how the data should be padded before it's encrypted.

```
{
  "type": PAD_TYPE,
  "length": INT
}
```

  To specify pre-encryption padding, `type` and `length` are used as follows:

  - `PAD_TYPE` as none – No padding will be applied to the column's data and the `length` field is not applicable (that is, omitted).
  - `PAD_TYPE` as `fixed` – The column's data is padded to the specified `length` of bytes.
  - `PAD_TYPE` as `max` – The column's data is padded to the size of the longest value's byte length plus an additional `length` bytes.

The following is an example mapped schema, with a column of each type.

```
{
  "headerRow": true,
```

```
  "columns": [
    {
      "sourceHeader": "FullName",
      "targetHeader": "name",
      "type": "cleartext"
    },
    {
      "sourceHeader": "City",
      "targetHeader": "city_sealed",
      "type": "sealed",
      "pad": {
        "type": "max",
        "length": 16
      }
    },
    {
      "sourceHeader": "PhoneNumber",
      "targetHeader": "phone_number_fingerprint",
      "type": "fingerprint"
    },
    {
      "sourceHeader": "PhoneNumber",
      "targetHeader": "phone_number_sealed",
      "type": "sealed",
      "pad": {
        "type": "fixed",
        "length": 20
      }
    }
  ]
}
```

As a more complex example, the following is an example .csv file with headers.

```
FirstName,LastName,Address,City,State,PhoneNumber,Title,Level,Notes
Jorge,Souza,12345 Mills Rd,Anytown,SC,703-555-1234,CEO,10,
Paulo,Santos,0 Street,Anytown,MD,404-555-111,CIO,9,This is a really long note that
 could really be a paragraph
Mateo,Jackson,1 Two St,Anytown,NY,304-555-1324,COO,9,""
Terry,Whitlock4 N St,Anytown,VA,407-555-8888,EA,7,Secret notes
Diego,Ramirez,9 Hollows Rd,Anytown,VA,407-555-1222,SDE I,4,null
John,Doe,8 Hollows Rd,Anytown,VA,407-555-4321,SDE I,4,Jane's younger brother
Jane,Doe,8 Hollows Rd,Anytown,VA,407-555-4322,SDE II,5,John's older sister
```

In the following mapped schema example, the columns `FirstName` and `LastName` are `cleartext` columns. The `State` column is encrypted as a `fingerprint` column and as a `sealed` column with a padding of none. The remaining columns are omitted.

```
{
  "headerRow": true,
  "columns": [
    {
      "sourceHeader": "FirstName",
      "targetHeader": "GivenName",
      "type": "cleartext"
    },
    {
      "sourceHeader": "LastName",
      "targetHeader": "Surname",
      "type": "cleartext"
    },
    {
      "sourceHeader": "State",
      "targetHeader": "State_Join",
      "type": "fingerprint"
    },
    {
      "sourceHeader": "State",
      "targetHeader": "State",
      "type": "sealed",
      "pad": {
        "type": "none"
      }
    }
  ]
}
```

The following is the .csv file that results from the mapped schema.

```
givenname,surname,state_fingerprint,state
John,Doe,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv
+1Mk=,01:enc:FQ3n3Ahv9BQQNWQGcugeHzHYzEZE1vapHa2Uu4SRgSAtZ3qObjPA4TcsHt
+BOkMKBcnHWI13BeGG/SBqmj7vKpI=
Paulo,Santos,01:hmac:CHF4eIrtTNgAooU9v4h9Qjc
+txBnMidQTjdjWuaDTTA=,01:enc:KZ5n5GtaXACco65AXk48BQO2durDNR2ULc4YxmMC8NaZZKKJiksU1IwFadAvV4iBQ1
Mateo,Jackson,01:hmac:iIRnjfNBzryusIJ1w35lgNzeY1RQ1bSfq6PDHW8Xrbk=,01:enc:mLKpS5HIOSgphdEsrzhEd
eN9nBO2gAbIygt4OFn4LalYn9Xyj/XUWXlmn8zFe2T4kyDTD8kGOvpQEUGxAUFk=
```

```
Diego,Ramirez,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:rmZhT98Zm
+IIGw1UTjMIJP4IrW/AAltBLMXcHvnYfRgmWP623VFQ6aUnhsb2MDqEw4G5Uwg5rKKZepUxx5uKbfk=
Jorge,Souza,01:hmac:3BxJdXiFFyZ8HBbYNqqEhBVqhNOd7s2ZiKUe7QiTyo8=,01:enc:vVaqWC1VRbhvkf8gnuR7q0z
Terry,Whitlock01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:3c9VEWbODO/
xbQjdGuccLvI7oZTBdPU+SyrJIyr2kudfAxbuMQ2uRdU/q7rbgyJjxZS8M2U35ILJf/lDgTyg7cM=
Jane,Doe,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:9RWv46YLveykeNZ/
G0NdlYFg+AVdOnu05hHyAYTQkPLHnyX+0/jbzD/g9ZT8GCgVE9aB5bV4ooJIXHGBVMXcjrQ=
```

Positional table schema

The positional schema has the following shape.

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": STRING,
        "type": TYPE,
        "pad": PAD
      },
      {
        "targetHeader": STRING,
        "type": TYPE,
        "pad": PAD
      }
    ],
    [],
    ...
  ]
}
```

If `headerRow` is `false`, the next field in the object is `columns`, which contains an array of entries. Each entry is itself an array of zero or more positional column schemas (no `sourceHeader` field), which are JSON objects describing what the output should contain.

- `sourceHeader` – The `STRING` header name of the source column that the data is derived from.

> **Note**
>
> This field must be omitted in positional schemas. In positional schemas, the source column is inferred by the column's corresponding index in the schema file.

- `targetHeader` – The `STRING` header name of the corresponding column in the output file.

> **Note**
>
> This field is required for positional schemas.

- `type` – The `TYPE` of the target column in the output file. That is, one of `cleartext`, `sealed`, or `fingerprint` depending on how the column will be used in the collaboration.
- `pad` – A field of a column schema object that is only present when the `TYPE` is `sealed`. Its corresponding value of `PAD` is an object that describes how the data should be padded before it's encrypted.

```
{
  "type": PAD_TYPE,
  "length": INT
}
```

To specify pre-encryption padding, `type` and `length` are used as follows:

- `PAD_TYPE` as none – No padding will be applied to the column's data and the `length` field is not applicable (that is, omitted).
- `PAD_TYPE` as `fixed` – The column's data is padded to the specified `length` of bytes.
- `PAD_TYPE` as `max` – The column's data is padded to the size of the longest value's byte length plus an additional `length` bytes.

> **Note**
>
> `fixed` is useful if you know ahead of time of an upper bound on the byte size of the column's data. An error is raised if any data in that column is longer than the specified `length`.
> `max` is convenient when the exact size of input data is unknown because it works regardless of the data's size. However, `max` requires additional processing time because

it encrypts the data twice. `max` encrypts the data once when read in to the temporary file and once after the longest data entry in the column is known.
Also, the length of the longest value isn't saved between invocations of the client. If you plan to encrypt your data in batches, or to encrypt new data periodically, be aware that the resulting ciphertext-lengths might vary among batches.

The following is an example of a positional schema.

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": "name",
        "type": "cleartext"
      }
    ],
    [
      {
        "targetHeader": "city_sealed",
        "type": "sealed",
        "pad": {
          "type": "max",
          "length": 16
        }
      }
    ],
    [
      {
        "targetHeader": "phone_number_fingerprint",
        "type": "fingerprint"
      },
      {
        "targetHeader": "phone_number_sealed",
        "type": "sealed",
        "pad": {
          "type": "fixed",
          "length": 20
        }
      }
    ]
```

```
  ]
}
```

As a complex example, the following is an example .csv file if it didn't have the first row with the headers.

```
Jorge,Souza,12345 Mills Rd,Anytown,SC, 703 -555 -1234,CEO, 10,
Paulo,Santos, 0 Street,Anytown,MD, 404-555-111,CIO, 9,This is a really long note that
 could really be a paragraph
Mateo,Jackson, 1 Two St,Anytown,NY, 304-555-1324,COO, 9, ""
Terry,Whitlock, 4 N St,Anytown,VA, 407-555-8888,EA, 7,Secret notes
Diego,Ramirez, 9 Hollows Rd,Anytown,VA, 407-555-1222,SDE I, 4,null
John,Doe, 8 Hollows Rd,Anytown,VA, 407-555-4321,SDE I, 4,Jane's younger brother
Jane,Doe, 8 Hollows Rd,Anytown,VA, 407-555-4322,SDE II, 5,John's older sister
```

The positional schema has the following form.

```
{
  "headerRow": false,
  "columns": [
    [
      {
        "targetHeader": "GivenName",
        "type": "cleartext"
      }
    ],
    [
      {
        "targetHeader": "Surname",
        "type": "cleartext"
      }
    ],
    [],
    [],
    [
      {
        "targetHeader": "State_Join",
        "type": "fingerprint"
      },
      {
        "targetHeader": "State",
        "type": "sealed",
        "pad": {
```

```
          "type": "none"
        }
      }
    ],
    [],
    [],
    [],
    []
  ]
}
```

The preceding schema produces the following output file with a header row containing the specified target headers.

```
givenname,surname,state_fingerprint,state
Mateo,Jackson,01:hmac:iIRnjfNBzryusIJ1w35lgNzeY1RQ1bSfq6PDHW8Xrbk=,01:enc:ENS6QD3cMVl9vQEGfe9MN
Q8m/Y5SA89dJwKpT5rGPp8e36h6klwDoslpFzGvU0=
Jorge,Souza,01:hmac:3BxJdXiFFyZ8HBbYNqqEhBVqhNOd7s2ZiKUe7QiTyo8=,01:enc:LKo0zirq2+
+XEIIIMNRjAsGMdyWUDwYaum0B+IFP+rUf1BNeZDJjtFe1Z+zbZfXQWwJy52Rt7HqvAb2WIK1oMmk=
Paulo,Santos,01:hmac:CHF4eIrtTNgAooU9v4h9Qjc
+txBnMidQTjdjWuaDTTA=,01:enc:MyQKyWxJ9kvK1xDQQtXlUNwv3F+yrBRr0xrUY/1BGg5KFgOn9pK+MZ7g
+ZNqZEPcPz4lht1u0t/wbTaqzOCLXFQ=
Jane,Doe,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:Pd8sbITBfb0/
ttUB4svVsgoYkDfnDvgkvxzeci0Yxq54rLSwccy1o3/B50C3cpkkn56dovCwzgmmPNwrmCmYtb4=
Terry,Whitlock01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv
+1Mk=,01:enc:Qmtzu3B3GAXKh2KkRYTiEAaMopYedsSdF2e/
ADUiBQ9kv2CxKPzWyYTD3ztmKPMka19dHre5VhUHNpO3O+j1AQ8=
Diego,Ramirez,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:ysdg
+GHKdeZrS/geBIooOEPLHG68MsWpx1dh3xjb+fG5rmFmqUcJLNuuYBHhHAlxchM2WVeV1fmHkBX3mvZNvkc=
John,Doe,01:hmac:UK8s8Cn/WR2JO/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:9uX0wZuO7kAPAx
+Hf6uvQownkWqFSKtWS7gQIJSe5aXFquKWCK6yZN0X5Ea2N3bn03Uj1kh0agDWoiP9FRZGJA4=
```

Creating a configured table in AWS Clean Rooms

A *configured table* is a reference to an existing table in the AWS Glue Data Catalog. It contains an analysis rule that determines how the data can be queried in AWS Clean Rooms. Configured tables can be associated to one or more collaborations. For more information about AWS Glue, see the *AWS Glue Developer Guide*.

Use the statistic generation provided by AWS Glue to compute column-level statistics for AWS Glue Data Catalog tables. Once AWS Glue generates statistics for tables in the Data Catalog, Amazon Redshift Spectrum automatically uses those statistics to optimize the query plan. For more information about computing column-level statistics using AWS Glue, see *Working with column statistics Guide*.

Create a configured table

In this step, you create a configured table in AWS Clean Rooms to use in the collaboration.

To create a configured table in AWS Clean Rooms

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. In the upper right corner, choose **Configure new table**.
4. For **Configure new table**, for **Choose AWS Glue table**:

   a. Choose the **Database** that you want to configure from the dropdown list.

   b. Choose the **Table** that you want to configure from the dropdown list.

   Note

   To verify that this is the correct table, do either one of the following:

   - Choose **View in AWS Glue**.
   - Turn on **View schema** to view the schema.

5. For **Columns allowed in collaborations**, choose either **All columns** or **Custom list**.

If you choose...	Then ...
All columns	All columns are allowed for use in AWS Clean Rooms (subject to analysis rules).
Custom list	Choose one or more columns that you want to allow from the **Specify allowed columns** dropdown list.

6. For **Configured table details**,

   a. Enter a **Name** for the configured table.

      You can use the default name or rename this table.

   b. Enter a **Description** of the table.

      The description helps differentiate between other configured tables with similar names.

   c. If you want to enable **Tags** for the configured table resource, choose **Add new tag** and then enter the **Key** and **Value** pair.

7. Choose **Configure new table**.

Next steps

Now that you have created a configured table, you are ready to:

- Configure an analysis rule to the configured table
- Associate the configured table to a collaboration

Configuring an analysis rule to a configured table

The following sections describe how to configure an analysis rule to your configured table. By defining the analysis rules, you can authorize the member who can query to run queries that match a specific analysis rule supported by AWS Clean Rooms.

AWS Clean Rooms supports the following types of analysis rules: aggregation, list, and custom.

There can be only one analysis rule per configured table.

> **Important**
>
> If you are using Cryptographic Computing for Clean Rooms and have encrypted data tables in the collaboration, the analysis rule you add to the encrypted configured table should be consistent with how the data was encrypted. For example, if you encrypted the data for SELECT (aggregation analysis rule), you should not add the analysis rule for JOIN (list analysis rule).

To gain an understanding of the types of analysis rules that are available in AWS Clean Rooms, see Analysis rules in AWS Clean Rooms.

For more information about the aggregation analysis rule, see Aggregation analysis rule.

For more information about the list analysis rule, see List analysis rule.

For more information about the custom analysis rule, see Custom analysis rule in AWS Clean Rooms.

After you have reviewed and understood these sections, you can perform the following procedures:

Topics

Configuring an aggregation analysis rule to a table (guided flow)

The aggregation analysis rule allows queries that aggregate statistics without revealing row-level information using COUNT, SUM, and AVG functions along optional dimensions.

This procedure describes the process of adding an aggregation analysis rule to your configured table by using the **Guided flow** option in the AWS Clean Rooms console.

To add the aggregation analysis rule to a table (guided flow)

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table.
4. On the configured table detail page, choose **Configure analysis rule**.
5. Under **Step 1: Choose type**, under **Type**, leave the **Aggregation** option selected by default.
6. Under **Creation method**, select **Guided flow**, and then choose **Next**.
7. Under **Step 2: Specify query controls**, for **Aggregate functions**:

    a. Choose an **Aggregate function** from the dropdown:

        - **COUNT**
        - **COUNT DISTINCT**
        - **SUM**
        - **SUM DISTINCT**
        - **AVG**

    b. Choose which columns can be used in the **Aggregate function** from the **Columns** dropdown.

    c. (Optional) Choose **Add another function** to add another aggregate function and associate one or more columns to that function.

    > **Note**
    >
    > At least one aggregate function is required.

d. (Optional) Choose **Remove** to remove an aggregate function.

8. For **Join controls**,

a. Choose one option for **Allow table to be queried by itself**:

If you choose...	Then ...
No, only overlap can be queried	The table can be queried only when joined to a table owned by the member who can query.
Yes	The table can be queried by itself or when joined to other tables.

b. Under **Specify join columns**, choose the columns that you want to allow to be used in the INNER JOIN statement.

This is *optional* if you have selected **Yes** in the previous step.

c. Under **Specify allowed operators for matching**, choose which, if any, operators can be used for matching on multiple join columns. If you select two or more JOIN columns, one of these operators is required.

If you choose...	Then ...
AND	You can include `AND` in the `INNER JOIN` match conditions to join one column to another column between tables.
OR	You can include `OR` in the `INNER JOIN` match conditions to combine multiple column matches between tables. This logical operator is useful for obtaining a higher match rate.

9. *(Optional)* For **Dimension controls**, in the **Specify dimension columns** dropdown, choose which columns you want to allow to be used in the SELECT statement, and the WHERE, GROUP BY, and ORDER BY parts of the query.

> **Note**
>
> Aggregate function or join columns can't be used as **Dimension** columns.

10. For **Scalar functions**, choose one option for **Which scalar functions do you want to allow?**

If you choose...	Then ...
All currently supported by AWS Clean Rooms	You allow all scalar functions currently supported by AWS Clean Rooms. • You can choose **View list** to see the entire list of **Scalar functions supported in AWS Clean Rooms**.
A custom list	You can customize which scalar functions to allow. • Choose one or more options from the **Specify allowed scalar functions** dropdown.
None	You don't want to allow any scalar functions .

For more information, see Scalar functions.

11. Choose **Next**.
12. Under **Step 3: Specify query results controls**, for **Aggregation constraints**:

    a. Select the dropdown list for each **Column name**.

    b. Select the dropdown list for each **Minimum number of distinct values** that must be met for each output row to be returned, after the COUNT DISTINCT function is applied to it.

    c. Choose **Add constraint** to add more aggregation constraints.

    d. (Optional) Choose **Remove** to remove an aggregation constraint.
13. Choose **Next**.

14. Under **Step 4: Review and configure**, review the selections you've made for the previous steps, edit if necessary, and then choose **Configure analysis rule**.

You see a confirmation message that you've successfully configured an aggregation analysis rule to the table.

Configuring a list analysis rule to a table (guided flow)

The list analysis rule allows queries that output row-level lists of the overlap between the associated table and a table of the member who can query.

This procedure describes the process of adding the list analysis rule to your configured table using the **Guided flow** option in the AWS Clean Rooms console.

To add a list analysis rule to a table (guided flow)

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table.
4. On the configured table detail page, choose **Configure analysis rule**.
5. Under **Step 1: Choose type**, under **Type**, choose the **List** option.
6. Under **Creation method**, select **Guided flow**, and then choose **Next**.
7. Under **Step 2: Specify query controls**, for **Join controls**:

    a. Under **Specify join columns**, choose the columns that you want to allow to be used in the INNER JOIN statement.

    b. Under **Specify allowed operators for matching**, choose which, if any, operators can be used for matching on multiple join columns. If you select two or more JOIN columns, one of these operators is required.

If you choose...	Then ...
AND	You can include AND in the `INNER JOIN` match conditions to join one column to another column between tables.

If you choose...	Then ...
OR	You can include `OR` in the `INNER JOIN` match conditions to combine multiple column matches between tables. This logical operator is useful for obtaining a higher match rate.

8. *(Optional)* For **List controls**, in the **Specify list columns** dropdown, choose which columns you want to allow to be used in the query output (that is, used in the SELECT statement), or used to filter results (that is, the WHERE statement).
9. Choose **Next**.
10. Under **Step 3: Review and configure**, review the selections you've made for the previous steps, edit if necessary, and then choose **Configure analysis rule**.

You see a confirmation message that you've successfully configured a list analysis rule for the table.

Configuring a custom analysis rule to a table (guided flow)

The custom analysis rule enables custom SQL queries on a configured table. The custom analysis rule is required if using analysis templates or differential privacy.

This procedure describes the process of adding the custom analysis rule to your configured table using the **Guided flow** option in the AWS Clean Rooms console.

To add a custom analysis rule to a table (guided flow)

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table.
4. On the configured table detail page, choose **Configure analysis rule**.
5. Under **Step 1: Choose type**, under **Type**, choose the **Custom** option.
6. Under **Creation method**, select **Guided flow**, and then choose **Next**.

7. Under **Step 2: Set differential privacy**, determine whether you want differential privacy turned on or off. Differential privacy is a mathematically-proven technique to protect your data from re-identification attacks.

   a. For **Differential privacy**:

If you...	Then choose ...
Have user-level data and you want protection against re-identification attempts	**Turn on**
Do not have user-level data or do not need protection against re-identification attempts	**Turn off**

   b. If you have chosen to **Turn on** differential privacy, select the **User identifier column** that contains the unique identifier of your users, such as `user_id` column, whose privacy you want to protect. If you want to turn on differential privacy for two or more tables in a collaboration, you must configure the same column as the **User identifier column** in both analysis rules to maintain a consistent definition of users across tables. In case of a misconfiguration, the member who can query receives an error message that there are two columns to choose from in order to compute the number of user contributions (for example, the number of ad impressions made by a user) while running the query.

   c. Choose **Next**.

8. Under **Step 3: Specify query controls**,

   a. For **Control type**:

If you want to ...	Then choose ...
Review each new analysis template before it's run on your configured table	**Review each new analysis before it is allowed to be run on this table**
Let any analysis template or direct query be performed on your configured table	**Allow any queries created by specific collaborators to run without review on this table**

b. Choose one of the following:

If you've chosen ...	Then ...
Review each new analysis before it is allowed to be run on this table	Under **Analysis templates allowed to be run**, choose **Add analysis template**, and then choose the appropriate **Collaboration** and the **Analysis template** from the dropdown lists.
Allow any queries created by specific collaborators to run without review on this table	Under **AWS accounts allowed to create any query**, choose **Add AWS account**, and then choose the appropriate **AWS account ID**.

9. Choose **Next**.
10. Under **Step 4: Review and configure**, review the selections you've made for the previous steps, edit if necessary, and then choose **Configure analysis rule**.

You see a confirmation message that you've successfully configured a custom analysis rule for the table.

Configuring analysis rule to a table (JSON editor)

The following procedure shows how to add an analysis rule to a table using the **JSON editor** option in the AWS Clean Rooms console.

To configure an aggregation, list, or custom analysis rule to a table (JSON editor)

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table.
4. On the configured table detail page, choose **Configure analysis rule**.
5. Under **Step 1: Choose type**, under **Type**, choose either the **Aggregation**, **List**, or **Custom** option.

6. Under **Creation method**, select **JSON editor**, and then choose **Next**.
7. Under **Step 2: Specify controls**, you can choose to insert a query structure (**Insert template**) or insert a file (**Import from file**).

If you choose...	Then ...
Insert template	1. Specify the parameters for the selected analysis rule in the **Analysis rule definition**. 2. You can press **Ctrl** + **Spacebar** to enable auto-complete. For more information about aggregation analysis rule parameters, see Aggregation analysis rule - query controls. For more information about list analysis rule parameters, see List analysis rule - query controls.
Import from file	1. Select your JSON file from your local drive. 2. Choose **Open**. The **Analysis rule definition** displays the analysis rule from the uploaded file.

8. Choose **Next**.
9. Under **Step 3: Review and configure**, review the selections you've made for the previous steps, edit if necessary, and then choose **Configure analysis rule**.

You receive a confirmation message that you've successfully configured an analysis rule for the table.

Next steps

Now that you configured an analysis rule to your configured table, you are ready to:

- Associate a configured table to a collaboration
- Query the data tables (as a member who can query)

Associating a configured table to a collaboration

After you have created a configured table and added an analysis rule to it, you can associate it to a collaboration.

> **Important**
>
> Before you associate the configured AWS Glue tables to the collaboration, the AWS Glue table location must point to an Amazon Simple Storage Service (Amazon S3) folder and not to a single file. You can verify this location by viewing the table in the AWS Glue console at https://console.aws.amazon.com/glue/.

> **Note**
>
> If you have configured encryption in AWS Glue and created a service role, you must give that role access to use AWS KMS keys to decrypt AWS Glue tables.
> If you associated a configured table that is backed by an AWS KMS-encrypted Amazon S3 dataset, you must give the role access to use the KMS key to decrypt Amazon S3 data.
> For more information, see Setting up encryption in AWS Glue in the *AWS Glue Developer Guide*.

The following topics describe how to associate a configured table to a collaboration using the AWS Clean Rooms console:

Topics

For information about how to associate your configured tables to the collaboration using the AWS SDKs, see the *AWS Clean Rooms API Reference*.

Associate a configured table from the configured table detail page

To associate AWS Glue tables to the collaboration from the configured table detail page

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table.
4. On the configured table detail page, choose **Associate to collaboration**.
5. For the **Associate table to collaboration** dialog box, choose the **Collaboration** from the dropdown list.
6. Choose **Choose collaboration**.

   On the **Associate table** page, the name of the configured table you chose appears under the **Choose configured table** section.
7. For **Choose configured table**, do the following:

If you want to...	Then ...
Configure a new table	Choose **Configure table** and follow the prompts on the **Configure table** page.
View the schema and analysis rule for the configured table	Turn on **View schema and analysis rule**.

8. Specify the **Service access** permissions by selecting either **Create and use a new service role** or **Use an existing service role**.

If you choose...	Then ...
Create and use a new service role	• AWS Clean Rooms creates a service role with the required policy for this table. • The default **Service role name** is `cleanrooms-<timestamp>`

<table>
<tr><th>If you choose...</th><th>Then ...</th></tr>
<tr><td></td><td>You must have permissions to create roles and attach policies.
If your input data is encrypted, you can select This data is encrypted with a KMS key and then enter an AWS KMS key that will be used to decrypt your data input.</td></tr>
<tr><td>Use an existing service role</td><td>Choose an Existing service role name from the dropdown list.

The list of roles are displayed if you have permissions to list roles.

If you don't have permissions to list roles, you can enter the Amazon Resource Name (ARN) of the role that you want to use.
View the service role by choosing the View in IAM external link.

If there are no existing service roles, the option to Use an existing service role is unavailable.

By default, AWS Clean Rooms does not attempt to update the existing role policy to add necessary permissions.
(Optional) Select the Add a pre-confi gured policy with necessary permissio ns to this role check box to add attach necessary permissions to the role. You must have permissions to modify roles and create policies.</td></tr>
</table>

> **Note**
>
> - AWS Clean Rooms requires permissions to query according to the analysis rules. For more information about permissions for AWS Clean Rooms, see AWS managed policies for AWS Clean Rooms.
> - If the role doesn't have sufficient permissions for AWS Clean Rooms, you receive an error message stating that the role doesn't have sufficient permissions for AWS Clean Rooms. The role policy must be added before proceeding.
> - If you can't modify the role policy, you receive an error message stating that AWS Clean Rooms could not find the policy for the service role.

9. If you want to enable **Tags** for the configured table association resource, choose **Add new tag** and then enter the **Key** and **Value** pair.
10. Choose **Associate table**.

Associate a configured table from the collaboration detail page

To associate AWS Glue tables to the collaboration from the collaboration detail page

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Tables** tab, choose **Associate table**.
5. For **Choose configured table**, do the following:

If you want to...	Then ...
Choose an existing configured table	Choose the **Configured table name** that you want to associate with the collaboration from the dropdown list.

If you want to...	Then ...
Configure a new table	Choose **Configure table** and follow the prompts on the **Configure table** page.
View the schema and analysis rule for the configured table	Turn on **View schema and analysis rule**.

6. For **Table association details**,

    a. Enter a **Name** for the associated table.

        You can use the default name or rename this table.

    b. (Optional) Enter a **Description** of the table.

        The description helps with writing queries.

7. Specify the **Service access** permissions by selecting either **Create and use a new service role** or **Use an existing service role**.

If you choose...	Then ...
Create and use a new service role	• AWS Clean Rooms creates a service role with the required policy for this table. • The default **Service role name** is cleanrooms-<timestamp> . • You must have permissions to create roles and attach policies. • If your input data is encrypted, you can select **This data is encrypted with a KMS key** and then enter an AWS KMS key that will be used to decrypt your data input.
Use an existing service role	1. Choose an **Existing service role name** from the dropdown list. The list of roles are displayed if you have permissions to list roles.

If you choose...	Then ...
	If you don't have permissions to list roles, you can enter the Amazon Resource Name (ARN) of the role that you want to use. 2. View the service role by choosing the **View in IAM** external link. If there are no existing service roles, the option to **Use an existing service role** is unavailable. By default, AWS Clean Rooms does not attempt to update the existing role policy to add necessary permissions. 3. (Optional) Select the **Add a pre-confi gured policy with necessary permissio ns to this role** check box to add attach necessary permissions to the role. You must have permissions to modify roles and create policies.

Note

- AWS Clean Rooms requires permissions to query according to the analysis rules. For more information about permissions for AWS Clean Rooms, see [AWS managed policies for AWS Clean Rooms](#).
- If the role doesn’t have sufficient permissions for AWS Clean Rooms, you receive an error message stating that the role doesn't have sufficient permissions for AWS Clean Rooms. The role policy must be added before proceeding.
- If you can’t modify the role policy, you receive an error message stating that AWS Clean Rooms could not find the policy for the service role.

8. If you want to enable **Tags** for the configured table association resource, choose **Add new tag** and then enter the **Key** and **Value** pair.
9. Choose **Associate table**.

Next steps

Now that you associated your configured data table to the collaboration, you are ready to:

- Edit the collaboration, if you're the collaboration creator
- Query the data tables (as a member who can query)

Configuring differential privacy policy

This procedure describes the process of configuring the differential privacy policy in a collaboration by using the **Guided flow** option in the AWS Clean Rooms console. This is a one-time step for all tables with differential privacy protection.

To configure differential privacy settings (guided flow)

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Tables** tab of the collaboration page, choose **Configure differential privacy policy**.
5. On the **Configure differential privacy policy** page, choose values for the following properties:

   - **Privacy budget**
   - **Refresh privacy budget monthly**
   - **Noise added per query**

   You can use the default values or enter custom values that support your specific use case. After choosing values for **Privacy budget** and **Noise added per query**, you can preview the resulting utility in terms of the number of aggregations that are possible across all queries on your data.
6. Choose **Configure**.

You'll see a confirmation message that you've successfully configured the differential privacy policy for the collaboration.

Next steps

Now that you configured differential privacy, you are ready to:

- Query the data tables (as a member who can query)
- Manage the collaboration (if you're the collaboration creator)

Working with analysis templates

Analysis templates work with the Custom analysis rule in AWS Clean Rooms. With an analysis template, you can define parameters to help you reuse the same query. AWS Clean Rooms supports a subset of parameterization with literal values.

Analysis templates are collaboration-specific. For each collaboration, members can only see the queries in that collaboration. If you plan to use differential privacy in a collaboration, you should make sure that your analysis templates are compatible with the general-purpose query structure of AWS Clean Rooms Differential Privacy.

Topics

Creating an analysis template

For information about how to create an analysis template using the AWS SDKs, see the AWS Clean Rooms API Reference.

To create an analysis template using the AWS Clean Rooms console

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with the AWS account that will function as the collaboration creator.
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Templates** tab, go to the **Analysis templates created by you** section.
5. Choose **Create analysis template**.
6. On the **Create analysis template** page, for **Details**, enter a **Name** and an optional **Description**.
7. For **Tables**, view the configured tables associated with the collaboration.
8. For **Definition**,

    a. Enter the definition for the analysis template.

    b. Choose **Import from** to import a definition.

c. (*Optional*) Specify a parameter in the SQL editor by entering a colon (`:`) in front of the parameter name.

For example:

```
WHERE table1.date + :date_period > table1.date
```

9. If you added parameters previously, under **Parameters – optional**, for each **Parameter name**, choose the **Type** and **Default value** (optional).
10. If you want to enable **Tags** for the configured table resource, choose **Add new tag** and then enter the **Key** and **Value** pair.
11. Choose **Create**.

You are now ready to:

- Inform your collaboration member that they can Review an analysis template. (Optional if you want to query your own data.)

Reviewing an analysis template

After a collaboration member has created an analysis template, you can review and approve it. After the analysis template and approved, it can in a query in AWS Clean Rooms.

To review an analysis template using the AWS Clean Rooms console

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with the AWS account that will function as the collaboration creator.
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Templates** tab, go to the **Analysis templates created by other members** section.
5. Choose the analysis template that has the **Can run status** of **No requires your review**.
6. Choose **Review**.
7. Review the analysis rule **Overview**, **Definition**, and **Parameters** (if any).
8. Review the configured tables listed under **Tables referenced in definition**.

   The **Status** next to each table will read **Template not allowed**.

9. Choose a table.

If you	Then choose
Approve the analysis template	**template on table**. Confirm your approval by choosing .
Don't approve the analysis template	**Disallow**

You are now ready to use the analysis template to query the data tables (as a member who can query).

Querying configured tables using an analysis template

This procedure demonstrates how to use an analysis template in the AWS Clean Rooms console to query configured tables with the **Custom** analysis rule.

To use an analysis template to query configured tables with the Custom analysis rule

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Query**.
4. On the **Queries** tab, under **Tables**, view the tables and their associated analysis rule type (**Custom analysis rule**).

   > **Note**
   >
   > If you don't see the tables that you expect in the list, it might be for the following reasons:
   >
   > - The tables haven't been associated.
   > - The tables don't have an analysis rule configured.

5. Under the **Analysis** section, select the analysis template from the dropdown list.
6. Enter the value of the parameters from the analysis template you want to use in the query. The value must be in the parameter's specified data type. You can use different values each

time you run the analysis template. Empty or NULL values for the parameter aren't supported. Using parameters in LIMIT clause is also not supported.

7. Choose **Run**.

> **Note**
>
> You can't run the query if the member who can receive results hasn't configured the query results settings.

8. Continue to adjust parameters and run your query again, or choose the **+** button to start a new query in a new tab.

Querying data in a collaboration

As the member who can query, you can do one of the following:

- Build a SQL query manually using the SQL code editor.
- Use the **Analysis builder UI** to build a query without having to write SQL code.
- Use an approved analysis template.

When the member who can query runs a SQL query on the tables in the collaboration, AWS Clean Rooms assumes the relevant roles to access the tables on their behalf. AWS Clean Rooms applies the analysis rules as necessary to the input query and its output.

AWS Clean Rooms supports SQL queries that can be different than other query engines. For specifications, see the AWS Clean Rooms SQL Reference. If you want to run queries on data tables protected with differential privacy, you should ensure that your queries are compatible with the general-purpose query structure of AWS Clean Rooms Differential Privacy.

> **Note**
>
> When using Cryptographic Computing for Clean Rooms, not all SQL operations generate valid results. For example, you can conduct a COUNT on an encrypted column but conducting a SUM on encrypted numbers leads to errors. In addition, queries might also yield incorrect results. For example, queries that SUM sealed columns produce errors. However, a GROUP BY query over sealed columns seems to succeed but produces different groups than those produced by a GROUP BY query over the cleartext.

The following topics explain how to query data in a collaboration using the AWS Clean Rooms console.

Topics

For information about how to query data or view queries by calling the AWS Clean Rooms `StartProtectedQuery` API operation directly or by using the AWS SDKs, see the AWS Clean Rooms API Reference.

For information about query logging, see Query logging in AWS Clean Rooms.

Note

If you run a query on encrypted data tables, the results from the encrypted columns are encrypted.

For information about receiving query results, see Receiving query results.

Using the SQL code editor

As a member who can query, you can build a query manually by writing SQL code in the SQL code editor. The SQL code editor is located in the **Analysis** section of the **Queries** tab in the AWS Clean Rooms console.

The SQL code editor is displayed by default. If you want to use the analysis builder to build queries, see Using the analysis builder.

Important

If you start writing a SQL query in the code editor and then turn on the **Analysis builder UI**, your query isn't saved.

AWS Clean Rooms supports many SQL commands, functions, and conditions. For more information, see the AWS Clean Rooms SQL Reference.

Tip

If a scheduled maintenance occurs while a query is running, the query is terminated and rolled back. You must restart the query.

To build the query manually using the SQL code editor

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Query**.
4. On the **Queries** tab, go to the **Analysis** section.

> **Note**
>
> The **Analysis** section only displays if the member who can receive results and the member who is responsible to pay for query compute costs have joined the collaboration as an active member.

5. On the **Queries** tab, under **Tables**, view the list of tables and their associated analysis rule type (**Aggregation analysis rule**, **List analysis rule**, or **Custom analysis rule**).

> **Note**
>
> If you don’t see the tables that you expect in the list, it might be for the following reasons:
>
> - The tables haven't been associated.
> - The tables don't have an analysis rule configured.

6. (Optional) To view the table's schema and analysis rule controls, expand the table by selecting the plus sign icon (**+**).
7. Build the query by typing the query into the SQL code editor.

(Optional) If you want to use an example query	(Optional) If you want to insert column names or functions
1. Select the three vertical dots next to the table. 2. Under **Insert in editor**, choose **Example query**.	1. Select the three vertical dots next to a column. 2. Under **Insert in editor**, choose **Column name**.

(Optional) If you want to use an example query	(Optional) If you want to insert column names or functions
Note Inserting an **Example query** appends the query already in the editor. The query example appears. All of the tables listed under **Tables** are included in the query. 3. Edit the placeholder values in the query.	3. To manually insert a function that is permitted on a column, select the three vertical dots next to a column, select **Insert in editor,** and then select the name of the permitted function (such as INNER JOIN, SUM, SUM DISTINCT, or COUNT). 4. Press **Ctrl** + **Space** to view the table schemas in the code editor. **Note** Members who can query can view and use the partition columns in each configured table association. Ensure the partition column is labeled as a partition column in the AWS Glue table underlying the configured table. 5. Edit the placeholder values in the query.

8. Choose **Run**.

> **Note**
>
> You can't run the query if the member who can receive results hasn't configured the query results settings.

9. Continue to adjust parameters and run your query again, or choose the **+** button to start a new query in a new tab.

> **Note**
>
> AWS Clean Rooms aims to provide clear error messaging. If an error message doesn't have enough details to help you troubleshoot, contact the account team. Provide them with a description of how the error occurred and the error message (including any identifiers). For more information, see Troubleshooting AWS Clean Rooms.

Using the analysis builder

You can use the analysis builder to build queries without having to write SQL code. With the analysis builder, you can build a query for a collaboration that has:

- A single table that uses the aggregation analysis rule with no JOIN required
- Two tables (one from each member) that both use the aggregation analysis rule
- Two tables (one from each member) that both use the list analysis rule
- Two tables (one from each member) that both use the aggregation analysis rule and two tables (one from each member) that both use the list analysis rule

If you want to manually write SQL queries, see Using the SQL code editor.

The analysis builder appears as the **Analysis builder UI** option in the **Analysis** section of the **Queries** tab in the AWS Clean Rooms console.

> **Important**
>
> If you turn on the **Analysis builder UI**, start building a query in the analysis builder, and then turn off the **Analysis builder UI**, your query isn't saved.

> **Tip**
>
> If a scheduled maintenance occurs while a query is running, the query is terminated and rolled back. You must restart the query.

The following topics explain how to use the analysis builder.

Topics

- Use the analysis builder to query a single table (aggregation)
- Use the analysis builder to query two tables (aggregation or list)

Use the analysis builder to query a single table (aggregation)

This procedure demonstrates how to use the **Analysis builder UI** in the AWS Clean Rooms console to build a query. The query is for a collaboration that has a single table that uses the aggregation analysis rule with no JOIN required.

To use the analysis builder to query a single table

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Query**.
4. On the **Queries** tab, under **Tables**, view the table and its associated analysis rule type. (The analysis rule type should be the **Aggregation analysis rule**.)

   > **Note**
   >
   > If you don't see the table you expect, it might be for the following reasons:
   >
   > - The table hasn't been associated.
   > - The table doesn't have an analysis rule configured.

5. Under the **Analysis** section, turn on **Analysis builder UI**.
6. Build a query.

If you want to see all of the aggregation metrics, skip to step 9.

a. For **Choose metrics**, review the aggregate metrics that have been preselected by default and remove any metric if needed.

b. (Optional) For **Add segments – optional**, choose one or more parameters.

> **Note**
>
> **Add segments – optional** is only displayed if dimensions are specified for the table.

c. (Optional) For **Add filters – optional**, choose **Add filter**, and then choose a **Parameter**, operator, and **Value**.

   To add more filters, choose **Add another filter**.

   To remove a filter, choose **Remove**.

> **Note**
>
> ORDER BY is not supported for aggregation queries.
> Only the AND operator is supported in filters.

d. (Optional) For **Add description – optional**, enter a description to help identify the query in the list of queries.

7. Expand **Preview SQL code**.

   a. View the SQL code that is generated from the analysis builder.

   b. To copy the SQL code, choose **Copy**.

   c. To edit the SQL code, choose **Edit in SQL code editor**.

8. Choose **Run**.

> **Note**
>
> You can't run the query if the member who can receive results hasn’t configured the query results settings.

9. Continue to adjust parameters and run your query again, or choose the **+** button to start a new query in a new tab.

> **Note**
>
> AWS Clean Rooms aims to provide clear error messaging. If an error message doesn't have enough details to help you troubleshoot, contact the account team. Provide them with a description of how the error occurred and the error message (including any identifiers). For more information, see Troubleshooting AWS Clean Rooms.

Use the analysis builder to query two tables (aggregation or list)

This procedure describes how to use the analysis builder in the AWS Clean Rooms console to build a query for a collaboration that has:

- Two tables (one from each member) that both use the aggregation analysis rule
- Two tables (one from each member) that both use the list analysis rule
- Two tables (one from each member) that both use the aggregation analysis rule and two tables (one from each member) that both use the list analysis rule

To use the analysis builder to query two tables

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Query**..
4. On the **Queries** tab, under **Tables**, view the two tables and their associated analysis rule type (**Aggregation analysis rule** or **List analysis rule**).

    > **Note**
    >
    > If you don’t see the tables you expect in the list, it might be for the following reasons:
    >
    > - The tables haven't been associated.
    > - The tables don't have an analysis rule configured.

5. Under the **Analysis** section, turn on **Analysis builder UI**.

6. Build a query.

   If the collaboration contains two tables that use the **Aggregation analysis rule** and two tables that use the **List analysis rule**, first choose **Aggregation** or **List**, and then follow the prompts based on the selected analysis rule.

If the two tables use the aggregation analysis rule	If the two tables use the list analysis rule
1. For **Choose metrics**, review the aggregate metrics that have been preselected by default and remove any metric if needed. 2. For **Match records**, choose one or more records. **Note** When using the analysis builder, you can match only on a single pair of columns. 3. (Optional) For **Add segments – optional**, choose one or more parameters. **Note** **Add segments – optional** is only displayed if dimension s are specified for the table.	1. For **Choose attributes**, review the list attributes that have been preselected by default and remove any metric if needed. 2. For **Match records**, choose one or more records. **Note** When using the analysis builder, you can match only on a single pair of columns. 3. (Optional) For **Add filters – optional**, choose **Add filter**, and then choose a parameter, operator, and value. To add more filters, choose **Add another filter**. To remove a filter, choose **Remove**.

<table>
<tr><th>If the two tables use the aggregation analysis rule</th><th>If the two tables use the list analysis rule</th></tr>
<tr><td>4. (Optional) For Add filters – optional, choose Add filter, and then choose a parameter, operator, and value.

To add more filters, choose Add another filter.

To remove a filter, choose Remove.

Note
ORDER BY is not supported for aggregation queries. Only the AND operator is supported in filters.

5. (Optional) For Add description – optional, enter a description to help identify the query in the list of recent queries.</td><td>Note
LIMIT is not supported for list queries. Only the AND operator is supported in filters.

4. (Optional) For Add description – optional, enter a description to help identify the query in the list of recent queries.</td></tr>
</table>

7. Expand **Preview SQL code**.

   a. View the SQL code that is generated from the analysis builder.

   b. To copy the SQL code, choose **Copy**.

   c. To edit the SQL code, choose **Edit in SQL code editor**.

8. Choose **Run**.

> **Note**
>
> You can't run the query if the member who can receive results hasn’t configured the query results settings

9. Continue to adjust parameters and run your query again, or choose the **+** button to start a new query in a new tab.

> **Note**
>
> AWS Clean Rooms aims to provide clear error messaging. If an error message doesn't have enough details to help you troubleshoot, contact the account team. Provide them with a description of how the error occurred and the error message (including any identifiers). For more information, see Troubleshooting AWS Clean Rooms.

Querying data with differential privacy

In general, writing and running queries does not change when differential privacy is turned on. However, you can't run a query if there is not enough privacy budget remaining. As you run queries and consume the privacy budget, you can see approximately how many aggregations you can run and how that might impact future queries.

To view the impact of differential privacy in a collaboration

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member details** status of **Run queries**.
4. On the **Queries** tab, under **Tables**, view the remaining privacy budget. This is displayed as the estimated number of **aggregation functions remaining** and the **Utility used** (rendered as a percentage).

   > **Note**
   >
   > The estimated number of **aggregate functions remaining** and the percentage of the **Utility used** only display for the member who can query.

5. Choose **View impact** to view how much noise is injected into the results and approximately how many aggregation functions you can run.

Viewing recent queries

You can view the queries that ran in the last 90 days on the **Recent queries** tab.

> **Note**
>
> If your only member ability is **Contribute data**, and you are not the member paying for query compute costs, the **Queries** tab doesn't appear on the console.

To view recent queries

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose a collaboration.
4. On the **Queries** tab, under **Queries**, view the queries that have been run in the last 90 days.
5. To sort recent queries by **Status**, select a status from the **All statuses** dropdown list.

   The statuses are: **Submitted**, **Started**, **Cancelled**, **Success**, **Failed**, and **Timed out**.

Viewing query details

You can view the query details as the member who can run queries or as a member who can receive results.

To view the details of the query

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose a collaboration.
4. On the **Queries** tab, do one of the following:

   - Choose the option button for the specific query you want to view, and then choose **View details**.
   - Choose the **Protected query ID**.

5. On the **Query details** page,

   - If you are the member who can run queries, view the **Query details**, **SQL text** and **Results**.

     You see a message confirming that the query results were delivered to the member who can receive results.
   - If you are the member who can receive results, view the **Query details** and **Results**.

Receiving query results

As a member who can receive results, you can receive the query output from AWS Clean Rooms into the Amazon S3 bucket that you specified when you joined the collaboration.

The following topics explain how to receive query results using the AWS Clean Rooms console.

Topics

For information about how to query data or view queries by calling the AWS Clean Rooms API directly or by using the AWS SDKs, see the AWS Clean Rooms API Reference.

For information about query logging, see Query logging in AWS Clean Rooms.

> **Note**
>
> If you run a query on encrypted data tables, the results from the encrypted columns are encrypted.

Receive query results

The results of the query are located in the **Query results settings defaults** section and the **Queries** section of the **Queries** tab in the AWS Clean Rooms console.

To receive query results

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Receive results**.
4. To receive the query results directly from AWS Clean Rooms, on the **Queries** tab, under **Queries**, under the **Protected query ID** column, select the query.
5. On the **Query details** page, under **Results**, do one of the following:

If you want to...	Then choose...
Copy the results.	**Copy**
Download the results.	**Download** **Note** By default, the downloaded file's name is the correspon ding `Query id` that was displayed when the query was run in AWS Clean Rooms.
View the results in Amazon S3.	**View in Amazon S3** The Amazon S3 console opens in a separate tab.

6. If you're using encrypted data, you can now decrypt the data tables.

   For more information, see Decrypting data tables with the C3R encryption client.

Edit default values for query results settings

As a member who can receive results, you can edit the default values for query results settings in the AWS Clean Rooms console.

To edit the default values for query results settings

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that has **Your member abilities** status of **Receive results**.
4. On the **Queries** tab, under **Query results settings**, choose **Edit**.

5. On the **Edit query results settings defaults** page, modify any of the following, as needed:

   a. Under **Query results settings**, modify the **Results destination in Amazon S3** or the **Result format**.

   b. Under **Service access**, modify the **Method to authorize AWS Clean Rooms** to write to the Amazon S3 bucket and format that you've specified.

   The updated **Query results settings** appear on the collaboration detail page.

Using query output in other AWS services

Query output from AWS Clean Rooms is available on the console (if the console is used to run queries) and downloaded in a specified Amazon S3 bucket. From there, you can use the query output in other AWS services, such as Amazon QuickSight and Amazon SageMaker, depending on how those services use data from Amazon S3.

For more information about Amazon QuickSight, see the Amazon QuickSight Documentation.

For more information about Amazon SageMaker, see the Amazon SageMaker Documentation.

Decrypting data tables with the C3R encryption client

Follow this procedure for collaborations that use Cryptographic Computing for Clean Rooms and the C3R encryption client to encrypt data tables. Use this procedure after you have queried data in the collaboration.

The shared secret key and collaboration ID are required for this procedure.

The member who can receive results decrypts the data using the same shared secret key and collaboration ID that was used to encrypt the data for the collaboration.

> **Note**
>
> AWS Clean Rooms collaborations already limit who can perform and view query results. To perform the decryption, whoever has access to these results needs the same shared secret key and collaboration ID that was used to encrypt the data.

To decrypt an encrypted data table

1. (Optional) View the available commands in the C3R encryption client.
2. (Optional) Navigate to the desired directory and run `ls` (macOS) or `dir` (Windows).
   - Verify that the c3r-cli.jar file and encrypted query results data file are in the desired directory.

   > **Note**
   >
   > If query results are downloaded from the AWS Clean Rooms console interface, they are likely in the **Downloads** folder for your user account. (For example, the **Downloads** folder in your user directory on Windows and macOS.) We recommend that you move the query results file to the same folder as the c3r-cli.jar.
3. Store the shared secret key in the `C3R_SHARED_SECRET` environment variable. For more information, see Step 6: Store the shared secret key in an environment variable.
4. From the AWS Command Line Interface (AWS CLI), run the following command.

java -jar c3r-cli.jar decrypt *<name of input .csv file>* --id=*<collaboration id>* --output=*<output file name>*

5. Replace each *user input placeholder* with your own information:

   a. For `id=`, enter the collaboration ID.

   b. For `output=`, enter the name of the output file (for example, `results-decrypted.csv`).

      If you don't specify an output name, a default name is displayed in the terminal.

   c. View the decrypted data in the specified output file using your preferred CSV or Parquet viewing application (such as Microsoft Excel, a text editor, or other application).

Managing AWS Clean Rooms

The following topics describe how to manage a collaboration, members, and configured tables in AWS Clean Rooms using the AWS Clean Rooms console.

For information about how to manage AWS Clean Rooms using the AWS SDKs, see the *AWS Clean Rooms API Reference*.

Topics

- Managing collaborations in AWS Clean Rooms
- Managing configured tables in AWS Clean Rooms

Managing collaborations in AWS Clean Rooms

The following topics describe how the collaboration creator can manage a collaboration in AWS Clean Rooms using the AWS Clean Rooms console.

For information about how to manage a collaboration using the AWS SDKs, see the *AWS Clean Rooms API Reference*.

Topics

- Editing collaborations
- Deleting collaborations
- Viewing collaborations
- Viewing tables and analysis rules
- Viewing differential privacy usage logs
- Monitoring member status
- Removing a member from a collaboration
- Leaving a collaboration
- Editing configured table associations
- Disassociating configured tables
- Editing a differential privacy policy
- Deleting a differential privacy policy
- Viewing the calculated differential privacy parameters

Editing collaborations

Learn how to edit the different parts of a collaboration.

Topics

Edit collaboration name and description

After you create the collaboration, you can only edit the collaboration name and description.

> **Note**
>
> If you have enabled **Query logging**, you can edit whether the query logs are stored in your Amazon CloudWatch Logs account.

To edit the collaboration name and description

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. On the collaboration detail page, choose **Actions**, and then choose **Edit collaboration.**
5. For **Details**, edit the **Name** and **Description** of the collaboration.
6. Choose **Save changes**.

Edit collaboration tags

As a collaboration creator, after you have created a collaboration, you can manage the tags on the collaboration resource.

To edit the collaboration tags

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. Choose one of the following:

If you are...	Then ...
A member of the collaboration	Choose the **Details** tab.
The collaboration creator but not a member of the collaboration	Scroll down the page to the **Tags** section.

5. For **Collaboration details**, choose **Manage tags**.
6. On the **Manage tags** page, you can do the following:
   - To remove a tag, choose **Remove**.
   - To add a tag, choose **Add new tag**.
   - To save your changes, choose **Save changes**

Edit membership tags

As a collaboration creator, after you have created a collaboration, you can manage the tags on the membership resource.

To edit the membership tags

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. Choose the **Details** tab.
5. For **Membership details**, choose **Manage tags**.
6. On the **Manage membership tags** page, you can do the following:

- To remove a tag, choose **Remove**.
- To add a tag, choose **Add new tag**.
- To save your changes, choose **Save changes**.

Edit associated table tags

As a collaboration creator, after you associate tables to a collaboration, you can manage the tags on the associated table resource.

To edit the associated table tags

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. Choose the **Tables** tab.
5. For **Tables associated by you**, choose a table.
6. On the configured table detail page, for **Tags**, choose **Manage tags**.

   On the **Manage tags** page, you can do the following:

   - To remove a tag, choose **Remove**.
   - To add a tag, choose **Add new tag**.
   - To save your changes, choose **Save changes**.

Edit analysis template tags

As a collaboration creator, after you have created a collaboration, you can manage the tags on the analysis template resource.

To edit the membership tags

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.

3. Choose the collaboration that you created.
4. Choose the **Templates** tab.
5. On the **Analysis templates created by you** section, choose the analysis template.
6. On the analysis template table detail page, scroll down to the **Tags** section.
7. Choose **Manage tags**.
8. On the **Manage tags** page, you can do the following:
   - To remove a tag, choose **Remove**.
   - To add a tag, choose **Add new tag**.
   - To save your changes, choose **Save changes**.

Edit differential privacy policy tags

As a collaboration creator, after you have created a collaboration, you can manage the tags on the analysis template resource.

To edit the membership tags

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that contains the differential privacy policy you want to edit.
4. Choose the **Tables** tab.
5. On the **Tables** tab, choose the **Manage tags**.
6. On the **Manage tags** page, you can do the following:
   - To remove a tag, choose **Remove**.
   - To add a tag, choose **Add new tag**.
   - To save your changes, choose **Save changes**.

Deleting collaborations

As a collaboration creator, you can delete a collaboration that you created.

> **Note**
>
> When you delete a collaboration, you and all members can't run queries, receive results, or contribute data. Each collaboration member continues to have access to their own data as part of their membership.

To delete a collaboration

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you want to delete.
4. Under **Actions**, choose **Delete collaboration**.
5. Confirm the deletion and then choose **Delete**.

Viewing collaborations

As a collaboration creator, you can view all of the collaborations that you created.

To view collaborations

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. On the **Collaborations** page, under **Last used**, view the last 5 collaborations used.
4. On the **With active membership** tab, view the list of **Collaborations with active membership**.

   You can sort by **Name**, the **Membership created date**, and **Your member details**.

   You can use the **Search** bar to search for a collaboration.
5. On the **Available to join** tab, view the list of **Collaborations available to join**.
6. On the **No longer available** tab, view the list of deleted collaborations and **Memberships for collaborations that are no longer available** (removed memberships).

Viewing tables and analysis rules

To view tables associated with the collaboration and the analysis rules

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console] with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. Choose the **Tables** tab.
5. Choose one of the following:
   a. To view your tables associated in the collaboration, for **Tables associated by you**, choose a table (blue text).
   b. To view other tables associated in the collaboration, for **Tables associated by collaborators**, choose a table (blue text).
6. View the table details and analysis rules on the table details page.

Viewing differential privacy usage logs

As a collaboration member who is protecting data with differential privacy, after you have created a collaboration with differential privacy, you can monitor the usage of the privacy budget.

To view how many aggregations were run and how much of the privacy budget was used

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console] with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. Choose the **Tables** tab.
5. Choose **View usage logs** (blue text).
6. View the usage details, including the privacy budget and how much utility was provided.

Monitoring member status

As a collaboration creator, after you have created a collaboration, you can monitor the status of all members on the **Members** tab.

To check the status of a member

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. Choose the **Members** tab.
5. View the **Member status** of each member.

Removing a member from a collaboration

> **Note**
>
> Removing a member also removes all of their associated datasets from the collaboration.

To remove a member from a collaboration

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration that you created.
4. Choose the **Members** tab.
5. Select the option button next to the member to be removed.

    > **Note**
    >
    > A collaboration creator can't choose their own account ID.

6. Choose **Remove**.

7. In the dialog box, confirm the decision to remove the member by typing **`confirm`** in the text input field.

> **Note**
>
> If you remove the member paying for query compute costs, no more queries are allowed to run in the collaboration.

Leaving a collaboration

As a collaboration member, you can leave a collaboration by deleting your membership. If you are the collaboration creator, you can only leave a collaboration by deleting the collaboration.

> **Note**
>
> When you delete your membership, you leave the collaboration and can't re-join it. If you are the member paying for query compute costs and you delete your membership, no more queries are allowed to run.

To leave a collaboration

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. For **With active membership**, choose the collaboration of which you are a member.
4. Choose **Actions**.
5. Choose **Delete membership**.
6. In the dialog box, confirm the decision to leave the collaboration by typing **`confirm`** in the text input field, and then choose **Empty and delete membership**.

   You see a message on the console indicating that the membership was deleted.

   The collaboration creator sees the **Member status** as **Left**.

Editing configured table associations

As a collaboration member, you can edit the configured table associations that you have created.

To edit configured table associations

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. Choose **Tables** tab.
5. For **Tables associated by you**, choose a table.
6. On the table details page, scroll down to view the **Table association details**.
7. Choose **Edit**.
8. On the **Edit configured table associations** page, update the **Description** or the **Service access information**.
9. Choose **Save changes**.

Disassociating configured tables

As a collaboration member, you can disassociate a configured table from the collaboration. This action prevents the member who can query from querying the table.

To disassociate a configured table

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. Choose **Tables** tab.
5. For **Tables associated by you**, select the option button next to the table that you want to disassociate.
6. Choose **Disassociate**.
7. In the dialog box, confirm the decision to disassociate the configured table and prevent the member who can query from querying the table by choosing **Disassociate**.

Editing a differential privacy policy

At any time after configuring the differential privacy policy, you can update it to better reflect your privacy needs.

To edit the differential privacy policy

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Tables** tab of the collaboration page, under **Tables associated by you**, choose **Edit**.
5. On the **Edit differential privacy** page, choose new values for the following properties:

   - **Privacy budget** – Move the slider bar to either increase or decrease the budget at any point during a collaboration. You can't decrease the budget after the member who can query has started querying your data. If the **Privacy budget** is increased, AWS Clean Rooms will continue using the existing budget until it is fully consumed before utilizing the newly added privacy budget.
   - **Noise added per query** – Move the slider bar to either increase or decrease the **Noise added per query** at any point during a collaboration.

   > **Note**
   >
   > You can chose **Interactive examples** to explore how different values of **Privacy budget** and **Noise added per query** affect the number of aggregate functions that you can run.

   You can't change the value of the **Privacy budget refresh**. To change your selection, you must delete the differential privacy policy and create a new one.
6. Choose **Save changes**.

You see a confirmation message that you've successfully edited the differential privacy policy.

Deleting a differential privacy policy

You can delete the differential privacy policy from the **Tables** tab of a collaboration.

To delete the differential privacy policy

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Tables** tab of the collaboration page, next to **Differential privacy policy**, select **Delete**.
5. If you're certain that you want to delete the differential privacy policy, choose **Delete**.

After deleting a differential privacy policy, you can't access the privacy budget usage logs from that policy. Tables with differential privacy turned on can't be queried if the differential privacy policy is deleted.

Viewing the calculated differential privacy parameters

For users with expertise in differential privacy, you can view the calculated differential privacy parameters from the **Queries** tab of a collaboration.

To view the calculated differential privacy parameters

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Collaborations**.
3. Choose the collaboration.
4. On the **Queries** tab, in the **Results** section, select **View calculated differential privacy parameters**.

In the **Calculated differential privacy parameters** table, you can see sensitivity values of aggregate functions, which is defined as the maximum amount by which the result of a function can change if a single user's records are added, removed, or modified. The list includes the following differential privacy parameters:

- **User contribution limit** (UCL) is the maximum number of rows contributed by a user in a SQL query. For example, if you want to count the total number of matched impressions in a specified campaign where each user can have multiple impressions, AWS Clean Rooms Differential Privacy needs to bound the number of impressions of a single user in order to ensure that the differential privacy calculation is accurate. In other words, if any user has more impressions than the bound, then AWS Clean Rooms automatically takes a uniform random sample of that user's impressions as per the computed UCL value and exclude the remaining impressions of that user while executing the query. The UCL value equals to 1 if you are counting the number of unique users. This is because adding, removing, or modifying a single user can change the count of distinct users by at most 1.
- **Minimum value** is the lower bound of an expression used within an aggregate function such as `sum()`. For example, if the expression is a column known as `purchase_value`, minimum value is the lower bound of the column.
- **Maximum value** is the upper bound of an expression used within an aggregate function such as `sum()`. For example, if the expression is a column known as `purchase_value`, maximum value is the upper bound of the column.

In the **Calculated differential privacy parameters** table, you can use these parameters to better understand the total amount of noise in query results. For example, when the configured **Noise added per query** is 30 users and a `COUNT DISTINCT (user_id)` query is run, then AWS Clean Rooms Differential Privacy adds random noise that falls between -30 and 30 with high probability because the sensitivity of `COUNT DISTINCT` is 1. In the case of a `COUNT` query with the same configuration, AWS Clean Rooms Differential Privacy adds statistical noise that is scaled by the user contribution limit because a single user could contribute multiple rows to the query result. In the case of `SUM` query like `SUM (purchase_value)` where all the column values are positive, the total noise is scaled by the user contribution limit times the maximum value. AWS Clean Rooms Differential Privacy automatically computes the sensitivity parameters to perform noise addition at query run-time and depletes the privacy budget. The depletion of privacy budget is required because the sensitivity parameters are data-dependent.

Managing configured tables in AWS Clean Rooms

The following topics describe how to manage configured tables in AWS Clean Rooms using the AWS Clean Rooms console.

For information about how to manage configured tables using the AWS SDKs, see the *AWS Clean Rooms API Reference*.

Topics

Editing configured table details

As a collaboration member, you can edit the configured table details.

To edit configured table details

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table that you created.
4. On the configured table detail page, scroll down to **Configured table details**.
5. Choose **Edit**.
6. Update the **Name** or **Description** of the configured table.
7. Choose **Save changes**.

Editing configured table tags

As a collaboration member, after you have created a configured table, you can manage the tags on the configured table resource on the **Configured tables** tab.

To edit the configured table tags

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table that you created.

4. On the configured table detail page, scroll down to the **Tags** section.
5. Choose **Manage tags**.
6. On the **Manage tags** page, you can do the following:
   - To remove a tag, choose **Remove**.
   - To add a tag, choose **Add new tag**.
   - To save your changes, choose **Save changes**.

Editing configured table analysis rule

To edit the configured table analysis rule

1. Sign in to the AWS Management Console and open the AWS Clean Rooms console with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table that you created.
4. On the configured table detail page, scroll down to either the **Aggregation analysis rule**, **List analysis rule**, or the **Custom analysis rule** section. (Your choice depends on which type of analysis rule you chose for the configured table.)
5. Choose **Edit**.
6. On the **Edit analysis rule** page, you can:
   - Modify the **Analysis rule definition** by:
     - Modifying the JSON editor.
     - Choosing **Import from file** to upload a new analysis rule definition.
   - Preview what members will see in a collaboration by selecting from the following options:
     - **Table view**
     - **JSON**
     - **Example query**
7. Choose **Save changes** to save your changes.

Deleting configured table analysis rule

> **Warning**
>
> This action can't be undone and impacts all related resources.

To delete the configured table analysis rule

1. Sign in to the AWS Management Console and open the [AWS Clean Rooms console](https://console.aws.amazon.com/cleanrooms/home) with your AWS account (if you have not yet done so).
2. In the left navigation pane, choose **Configured tables**.
3. Choose the configured table that you created.
4. On the configured table detail page, scroll down to either the **Aggregation analysis rule**, **List analysis rule**, or the **Custom analysis rule** section. (Your choice depends on which type of analysis rule you chose for the configured table.)
5. Choose **Delete**.
6. If you're certain that you want to delete the analysis rule, choose **Delete**.

Troubleshooting AWS Clean Rooms

This section describes some common issues that might arise when using AWS Clean Rooms and how to fix them.

Issues

One or more tables referenced by the query is not accessible by its associated service role. The table/role owner must grant the service role access to the table.

- Verify that the permissions for the service role are set up as required. For more information, see Setting up AWS Clean Rooms.

One of the underlying datasets has an unsupported file format.

- Ensure that your dataset is in one of the supported file formats:
  - Parquet
  - RCFile
  - TextFile
  - SequenceFile
  - RegexSerde
  - OpenCSV
  - AVRO
  - JSON

  For more information, see Data formats for AWS Clean Rooms.

Query results are not as expected when using Cryptographic Computing for Clean Rooms.

If you are using Cryptographic Computing for Clean Rooms (C3R), verify that your query uses encrypted columns correctly:

- The sealed columns are only used in SELECT clauses.
- The fingerprint columns are only used in JOIN clauses (and GROUP BY clauses under certain conditions).
- That you are only JOINing fingerprint columns with the same name if the collaboration settings require it.

For more information, see *Cryptographic computing* and the section called “Column types”.

Security in AWS Clean Rooms

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Clean Rooms, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using AWS Clean Rooms. It shows you how to configure AWS Clean Rooms to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Clean Rooms resources.

Contents

Data protection in AWS Clean Rooms

The AWS shared responsibility model applies to data protection in AWS Clean Rooms. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS Clean Rooms or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

AWS Clean Rooms always encrypts all service metadata at rest without requiring any additional configuration. This encryption is automatic when you use AWS Clean Rooms.

Clean Rooms ML encrypts all data stored within the service at rest with AWS KMS. If you choose to provide your own KMS key, the contents of your lookalike models and lookalike segment generation jobs are encrypted at rest with your KMS key.

> **Note**
>
> You can use the encryption options in Amazon S3 to protect your data at rest. For more information, see Specifying Amazon S3 encryption in the *Amazon S3 User Guide*.

Encryption in transit

AWS Clean Rooms uses Transport Layer Security (TLS) and client-side encryption for encryption in transit. Communication with AWS Clean Rooms is always done over HTTPS so your data is always encrypted in transit. This includes all data in transit when using Clean Rooms ML.

Encrypting underlying data

For more information about how to encrypt your underlying data, see Cryptographic Computing for Clean Rooms.

Data retention in AWS Clean Rooms

When you create a lookalike model, Clean Rooms ML reads your training data, transforms it into a format suitable for our ML model, and stores the trained model parameters inside Clean Rooms ML. Clean Rooms ML does not retain a copy of your training data. AWS Clean Rooms SQL queries do not retain any of your data after the query has run. Clean Rooms ML then uses the trained model to summarize the behavior of all of your users. Clean Rooms ML stores a user-level data set for each user in your data for as long as your lookalike model is active.

When you start a lookalike segment generation job, Clean Rooms ML reads the seed data, reads the behavior summaries from the associated lookalike model, and creates a lookalike segment that is stored within the AWS Clean Rooms service. Clean Rooms ML does not retain a copy of your seed data. Clean Rooms ML stores the user-level output of the job as long as the job is active.

If you want to remove your lookalike model or lookalike segment generation job data, use the API to delete it. Clean Rooms ML asynchronously deletes all data associated with the model or job. Once this process is complete, Clean Rooms ML deletes the metadata for the model or job and it is no longer visible in the API. Clean Rooms ML retains deleted data for 3 days for disaster recovery prevention. Once the job or model is no longer visible in the API and 3 days have passed, all data associated with the model or job has been permanently deleted.

Best practices for data collaborations in AWS Clean Rooms

This topic describes the best practices for conducting data collaborations in AWS Clean Rooms.

AWS Clean Rooms follows the AWS Shared Responsibility Model. AWS Clean Rooms offers analysis rules that you can configure to strengthen your ability to protect sensitive data in a collaboration. The analysis rules that you configure in AWS Clean Rooms will enforce the restrictions (query controls and query output controls) that you have configured. You are responsible for determining the restrictions and configuring analysis rules accordingly.

Data collaborations might involve more than just your use of AWS Clean Rooms. To help you maximize the benefit of data collaborations, we recommend that you perform the following best practices with your use of AWS Clean Rooms and specifically with analysis rules.

Topics

Best practices with AWS Clean Rooms

You're responsible for assessing the risk of each data collaboration and comparing it to your privacy requirements such as external and internal compliance programs and policies. We recommend that you take additional actions with your use of AWS Clean Rooms. These actions might help further manage risks and help guard against third-party attempts to re-identify your data (for example, differencing attacks or side-channel attacks).

For example, consider conducting due diligence on your other collaborators and enter into legal agreements with them *before* engaging in a collaboration. To monitor the use of your data, also consider adopting other audit mechanisms with your use of AWS Clean Rooms.

Best practices for using analysis rules in AWS Clean Rooms

Analysis rules in AWS Clean Rooms allow you to restrict the queries that can be run by setting query controls on a configured table. For example, you can set a query control for how a configured table can be joined and which columns can be selected. You can also restrict the query output through setting query result controls such as aggregation thresholds on output rows. The service rejects any query and removes rows that don’t comply with the analysis rules set by members on their configured tables in the query.

We recommend the following *10 best practices* for using analysis rules on your configured table:

- Create separate configured tables for separate query use cases (for example, audience planning or attribution). You can create multiple configured tables with the same underlying AWS Glue table.
- Specify columns in the analysis rule (for example, dimension columns, list columns, join columns) that are necessary for queries in a collaboration. This might help mitigate the risk of differencing attacks or enabling other members to reverse engineer your data. Use the **allowlist columns** feature to note other columns that you might want to make queryable in the future. To customize the columns that can be used for a certain collaboration, create additional configured tables with the same underlying AWS Glue table.
- Specify the functions in the analysis rule that are necessary for analysis in the collaboration. This can help mitigate risk from rare function errors that can present information on an individual data point. To customize the functions that can be used for a certain collaboration, create additional configured tables with the same underlying AWS Glue table.
- Add aggregation constraints on any columns whose values at a row-level are sensitive. This includes columns in your configured table that also exist in other collaboration members’ tables and analysis rules as an aggregation constraint. This also includes columns in your configured table that aren't queryable, that is, columns that are in your configured table but are not in the analysis rule. Aggregation constraints can help mitigate risk from correlating query results with data outside the collaboration.
- Create test collaborations and analysis rules to test restrictions created with specified analysis rules.
- Review collaborator configured tables and members’ analysis rules on configured tables to check that they match what was agreed upon for the collaboration. This can help mitigate risk from other members engineering their own data to run queries that weren't agreed upon.
- Review the example query provided (console only) that is enabled on your configured table after you set up the analysis rule.

> **Note**
>
> In addition to the provided example query, other queries are possible based on the analysis rule and other collaboration member tables and analysis rules.

- You can add or update an analysis rule for a configured table in a collaboration. When you do, review all the collaborations where the configured table is associated and its resulting impact. This helps make sure that no collaborations use obsolete analysis rules.
- Review the queries run in the collaboration to check that the queries match the use cases or queries that were agreed upon for the collaboration. (The queries are available in the query logs when the **Query logging** feature is turned on.) This can help mitigate risk from members running analysis that was not agreed upon and potential attacks such as side channel attacks.
- Review the configured table columns used in collaboration members' analysis rules and in queries to check that they match what was agreed upon in the collaboration. (The queries are available in the query logs when that feature is turned on.) This can help mitigate risk from other members engineering their own data to do queries that weren't agreed upon.

Identity and Access Management for AWS Clean Rooms

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Clean Rooms resources. IAM is an AWS service that you can use with no additional charge.

Topics

- IAM behaviors for AWS Clean Rooms ML

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in AWS Clean Rooms.

Service user – If you use the AWS Clean Rooms service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS Clean Rooms features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in AWS Clean Rooms, see Troubleshooting AWS Clean Rooms identity and access.

Service administrator – If you're in charge of AWS Clean Rooms resources at your company, you probably have full access to AWS Clean Rooms. It's your job to determine which AWS Clean Rooms features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with AWS Clean Rooms, see How AWS Clean Rooms works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS Clean Rooms. To view example AWS Clean Rooms identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Clean Rooms.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users or your company's single sign-on authentication are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see AWS account root user credentials and IAM identities in the *AWS General Reference*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [Cross account resource access in IAM](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).
  - **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Clean Rooms works with IAM

Before you use IAM to manage access to AWS Clean Rooms, learn what IAM features are available to use with AWS Clean Rooms.

IAM features you can use with AWS Clean Rooms

IAM feature	AWS Clean Rooms support
Identity-based policies	Yes
Resource-based policies	Partial
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Partial
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	No

To get a high-level view of how AWS Clean Rooms and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for AWS Clean Rooms

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for AWS Clean Rooms

To view examples of AWS Clean Rooms identity-based policies, see Identity-based policy examples for AWS Clean Rooms.

Resource-based policies within AWS Clean Rooms

Supports resource-based policies	Partial

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see [Cross account resource access in IAM]() in the *IAM User Guide*.

The AWS Clean Rooms service supports only one type of resource-based policy called a *configured lookalike model managed resource policy*, which is attached to a configured lookalike model. This policy defines which principals can perform actions on the configured lookalike model.

To learn how to attach a resource-based policy to a configured lookalike model, see **IAM behaviors for AWS Clean Rooms ML**.

Policy actions for AWS Clean Rooms

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Clean Rooms actions, see Actions defined by AWS Clean Rooms in the *Service Authorization Reference*.

Policy actions in AWS Clean Rooms use the following prefix before the action.

```
cleanrooms
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
```

```
      "cleanrooms:action1",
      "cleanrooms:action2"
         ]
```

To view examples of AWS Clean Rooms identity-based policies, see Identity-based policy examples for AWS Clean Rooms.

Policy resources for AWS Clean Rooms

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of AWS Clean Rooms resource types and their ARNs, see Resources defined by AWS Clean Rooms in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Clean Rooms.

To view examples of AWS Clean Rooms identity-based policies, see Identity-based policy examples for AWS Clean Rooms.

Policy condition keys for AWS Clean Rooms

Supports service-specific policy condition keys	Partial

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To learn how AWS Clean Rooms ML uses policy condition keys, see **IAM behaviors for AWS Clean Rooms ML**.

ACLs in AWS Clean Rooms

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS Clean Rooms

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS Clean Rooms

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Forward access sessions for AWS Clean Rooms

Supports forward access sessions (FAS)	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS Clean Rooms

Supports service roles	Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break AWS Clean Rooms functionality. Edit service roles only when AWS Clean Rooms provides guidance to do so.

Service-linked roles for AWS Clean Rooms

Supports service-linked roles	No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see [AWS services that work with IAM](). Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Clean Rooms

By default, users and roles don't have permission to create or modify AWS Clean Rooms resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see [Creating IAM policies]() in the *IAM User Guide*.

For details about actions and resource types defined by AWS Clean Rooms, including the format of the ARNs for each of the resource types, see [Actions, resources, and condition keys for AWS Clean Rooms]() in the *Service Authorization Reference*.

Topics

- [Policy best practices]()
- [Using the AWS Clean Rooms console]()
- [Allow users to view their own permissions]()

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Clean Rooms resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see [AWS managed policies]() or [AWS managed policies for job functions]() in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on

specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see [Policies and permissions in IAM] in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see [IAM JSON policy elements: Condition] in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see [IAM Access Analyzer policy validation] in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see [Configuring MFA-protected API access] in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM] in the *IAM User Guide*.

Using the AWS Clean Rooms console

To access the AWS Clean Rooms console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Clean Rooms resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the AWS Clean Rooms console, also attach the AWS Clean Rooms `FullAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see [Adding permissions to a user] in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policies for AWS Clean Rooms

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: `AWSCleanRoomsReadOnlyAccess`

You can attach `AWSCleanRoomsReadOnlyAccess` to your IAM principals.

This policy grants read-only permissions to resources and metadata in an `AWSCleanRoomsReadOnlyAccess` collaboration.

Permissions details

This policy includes the following permissions:

- `CleanRoomsRead` – Allows principals read-only access to the service.
- `ConsoleDisplayTables` – Allows principals read-only access to the AWS Glue metadata needed to show data about the underlying AWS Glue tables on the console.
- `ConsoleLogSummaryQueryLogs` – Allows principals to see the query logs.
- `ConsoleLogSummaryObtainLogs` – Allows principals to retrieve the log results.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
```

```
    "Sid": "CleanRoomsRead",
    "Effect": "Allow",
    "Action": [
     "cleanrooms:BatchGet*",
     "cleanrooms:Get*",
     "cleanrooms:List*"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleDisplayTables",
    "Effect": "Allow",
    "Action": [
     "glue:GetDatabase",
     "glue:GetDatabases",
     "glue:GetTable",
     "glue:GetTables",
     "glue:GetPartition",
     "glue:GetPartitions",
     "glue:GetSchema",
     "glue:GetSchemaVersion",
     "glue:BatchGetPartition"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleLogSummaryQueryLogs",
    "Effect": "Allow",
    "Action": [
     "logs:StartQuery"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*"
   },
   {
    "Sid": "ConsoleLogSummaryObtainLogs",
    "Effect": "Allow",
    "Action": [
     "logs:GetQueryResults"
    ],
    "Resource": "*"
   }
  ]
 }
```

AWS managed policy: `AWSCleanRoomsFullAccess`

You can attach `AWSCleanRoomsFullAccess` to your IAM principals.

This policy grants administrative permissions that allow full access (read, write, and update) to resources and metadata in an AWS Clean Rooms collaboration. This policy includes access to perform queries.

Permissions details

This policy includes the following permissions:

- `CleanRoomsAccess` – Grants full access to all actions on all resources for AWS Clean Rooms.
- `PassServiceRole` – Grants access to pass a service role to only the service (`PassedToService` condition) that has "cleanrooms" in its name.
- `ListRolesToPickServiceRole` – Allows principals to list all their roles in order to choose a service role when using AWS Clean Rooms.
- `GetRoleAndListRolePoliciesToInspectServiceRole` – Allows principals to see the service role and corresponding policy in IAM.
- `ListPoliciesToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `GetPolicyToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `ConsoleDisplayTables` – Allows principals read-only access to the AWS Glue metadata needed to show data about the underlying AWS Glue tables on the console.
- `ConsolePickQueryResultsBucketListAll` – Allows principals to choose an Amazon S3 bucket from a list of all available S3 buckets into which their query results are written.
- `SetQueryResultsBucket` – Allows principals to choose an S3 bucket into which their query results are written.
- `ConsoleDisplayQueryResults` – Allows principals to show the query results to the customer, read from the S3 bucket.
- `WriteQueryResults` – Allows principals to write the query results into a customer-owned S3 bucket.
- `EstablishLogDeliveries` – Allows principals to deliver query logs to a customer's Amazon CloudWatch Logs log group.

- `SetupLogGroupsDescribe` – Allows principals to use the Amazon CloudWatch Logs log group creation process.
- `SetupLogGroupsCreate` – Allows principals to create an Amazon CloudWatch Logs log group.
- `SetupLogGroupsResourcePolicy` – Allows principals to set up a resource policy on the Amazon CloudWatch Logs log group.
- `ConsoleLogSummaryQueryLogs` – Allows principals to see the query logs.
- `ConsoleLogSummaryObtainLogs` – Allows principals to retrieve the log results.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CleanRoomsAccess",
   "Effect": "Allow",
   "Action": [
    "cleanrooms:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "PassServiceRole",
   "Effect": "Allow",
   "Action": [
    "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::*:role/service-role/*cleanrooms*",
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "cleanrooms.amazonaws.com"
    }
   }
  },
  {
   "Sid": "ListRolesToPickServiceRole",
   "Effect": "Allow",
   "Action": [
    "iam:ListRoles"
   ],
   "Resource": "*"
  },
```

```
    {
     "Sid": "GetRoleAndListRolePoliciesToInspectServiceRole",
     "Effect": "Allow",
     "Action": [
      "iam:GetRole",
      "iam:ListRolePolicies",
      "iam:ListAttachedRolePolicies"
     ],
     "Resource": "arn:aws:iam::*:role/service-role/*cleanrooms*"
    },
    {
     "Sid": "ListPoliciesToInspectServiceRolePolicy",
     "Effect": "Allow",
     "Action": [
      "iam:ListPolicies"
     ],
     "Resource": "*"
    },
    {
     "Sid": "GetPolicyToInspectServiceRolePolicy",
     "Effect": "Allow",
     "Action": [
      "iam:GetPolicy",
      "iam:GetPolicyVersion"
     ],
     "Resource": "arn:aws:iam::*:policy/*cleanrooms*"
    },
    {
     "Sid": "ConsoleDisplayTables",
     "Effect": "Allow",
     "Action": [
      "glue:GetDatabase",
      "glue:GetDatabases",
      "glue:GetTable",
      "glue:GetTables",
      "glue:GetPartition",
      "glue:GetPartitions",
      "glue:GetSchema",
      "glue:GetSchemaVersion",
      "glue:BatchGetPartition"
     ],
     "Resource": "*"
    },
    {
```

```
   "Sid": "ConsolePickQueryResultsBucketListAll",
   "Effect": "Allow",
   "Action": [
    "s3:ListAllMyBuckets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "SetQueryResultsBucket",
   "Effect": "Allow",
   "Action": [
    "s3:GetBucketLocation",
    "s3:ListBucketVersions"
   ],
   "Resource": "arn:aws:s3:::cleanrooms-queryresults*"
  },
  {
   "Sid": "WriteQueryResults",
   "Effect": "Allow",
   "Action": [
    "s3:ListBucket",
    "s3:PutObject"
   ],
   "Resource": "arn:aws:s3:::cleanrooms-queryresults*",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": "cleanrooms.amazonaws.com"
    }
   }
  },
  {
   "Sid": "ConsoleDisplayQueryResults",
   "Effect": "Allow",
   "Action": [
    "s3:GetObject"
   ],
   "Resource": "arn:aws:s3:::cleanrooms-queryresults*"
  },
  {
   "Sid": "EstablishLogDeliveries",
   "Effect": "Allow",
   "Action": [
    "logs:CreateLogDelivery",
    "logs:GetLogDelivery",
```

```
      "logs:UpdateLogDelivery",
      "logs:DeleteLogDelivery",
      "logs:ListLogDeliveries"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
    }
  },
  {
    "Sid": "SetupLogGroupsDescribe",
    "Effect": "Allow",
    "Action": [
      "logs:DescribeLogGroups"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
    }
  },
  {
    "Sid": "SetupLogGroupsCreate",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
    }
  },
  {
    "Sid": "SetupLogGroupsResourcePolicy",
    "Effect": "Allow",
    "Action": [
        "logs:DescribeResourcePolicies",
      "logs:PutResourcePolicy"
    ],
```

```
    "Resource": "*",
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cleanrooms.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleLogSummaryQueryLogs",
    "Effect": "Allow",
    "Action": [
     "logs:StartQuery"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*"
   },
   {
    "Sid": "ConsoleLogSummaryObtainLogs",
    "Effect": "Allow",
    "Action": [
     "logs:GetQueryResults"
    ],
    "Resource": "*"
   }
  ]
 }
```

AWS managed policy: `AWSCleanRoomsFullAccessNoQuerying`

You can attach `AWSCleanRoomsFullAccessNoQuerying` to your IAM principals.

This policy grants administrative permissions that allow full access (read, write, and update) to resources and metadata in an AWS Clean Rooms collaboration. This policy excludes access to perform queries.

Permissions details

This policy includes the following permissions:

- `CleanRoomsAccess` – Grants full access to all actions on all resources for AWS Clean Rooms, except for querying in collaborations.
- `CleanRoomsNoQuerying` – Explicitly denies `StartProtectedQuery` and `UpdateProtectedQuery` to prevent querying.

- `PassServiceRole` – Grants access to pass a service role to only the service (`PassedToService` condition) that has "cleanrooms" in its name.
- `ListRolesToPickServiceRole` – Allows principals to list all their roles in order to choose a service role when using AWS Clean Rooms.
- `GetRoleAndListRolePoliciesToInspectServiceRole` – Allows principals to see the service role and corresponding policy in IAM.
- `ListPoliciesToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `GetPolicyToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `ConsoleDisplayTables` – Allows principals read-only access to the AWS Glue metadata needed to show data about the underlying AWS Glue tables on the console.
- `EstablishLogDeliveries` – Allows principals to deliver query logs to a customer's Amazon CloudWatch Logs log group.
- `SetupLogGroupsDescribe` – Allows principals to use the Amazon CloudWatch Logs log group creation process.
- `SetupLogGroupsCreate` – Allows principals to create an Amazon CloudWatch Logs log group.
- `SetupLogGroupsResourcePolicy` – Allows principals to set up a resource policy on the Amazon CloudWatch Logs log group.
- `ConsoleLogSummaryQueryLogs` – Allows principals to see the query logs.
- `ConsoleLogSummaryObtainLogs` – Allows principals to retrieve the log results.
- `cleanrooms` – Manage collaborations, analysis templates, configured tables, memberships, and associated resources within the AWS Clean Rooms service. Perform various operations such as creating, updating, deleting, listing, and retrieving information about these resources.
- `iam` – Pass service roles with names containing "`cleanrooms`" to the AWS Clean Rooms service. List roles, policies, and inspect service roles and policies related to the AWS Clean Rooms service.
- `glue` – Retrieve information about databases, tables, partitions, and schemas from AWS Glue. This is required for the AWS Clean Rooms service to display and interact with the underlying data sources.
- `logs` – Manage log deliveries, log groups, and resource policies for CloudWatch Logs. Query and retrieve logs related to the AWS Clean Rooms service. These permissions are necessary for monitoring, auditing, and troubleshooting purposes within the service.

The policy also explicitly denies the actions `cleanrooms:StartProtectedQuery` and `cleanrooms:UpdateProtectedQuery` to prevent users from directly executing or updating protected queries, which should be done through the AWS Clean Rooms controlled mechanisms.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CleanRoomsAccess",
   "Effect": "Allow",
   "Action": [
    "cleanrooms:BatchGetCollaborationAnalysisTemplate",
    "cleanrooms:BatchGetSchema",
    "cleanrooms:BatchGetSchemaAnalysisRule",
    "cleanrooms:CreateAnalysisTemplate",
    "cleanrooms:CreateCollaboration",
    "cleanrooms:CreateConfiguredTable",
    "cleanrooms:CreateConfiguredTableAnalysisRule",
    "cleanrooms:CreateConfiguredTableAssociation",
    "cleanrooms:CreateMembership",
    "cleanrooms:DeleteAnalysisTemplate",
    "cleanrooms:DeleteCollaboration",
    "cleanrooms:DeleteConfiguredTable",
    "cleanrooms:DeleteConfiguredTableAnalysisRule",
    "cleanrooms:DeleteConfiguredTableAssociation",
    "cleanrooms:DeleteMember",
    "cleanrooms:DeleteMembership",
    "cleanrooms:GetAnalysisTemplate",
    "cleanrooms:GetCollaboration",
    "cleanrooms:GetCollaborationAnalysisTemplate",
    "cleanrooms:GetConfiguredTable",
    "cleanrooms:GetConfiguredTableAnalysisRule",
    "cleanrooms:GetConfiguredTableAssociation",
    "cleanrooms:GetMembership",
    "cleanrooms:GetProtectedQuery",
    "cleanrooms:GetSchema",
    "cleanrooms:GetSchemaAnalysisRule",
    "cleanrooms:ListAnalysisTemplates",
    "cleanrooms:ListCollaborationAnalysisTemplates",
    "cleanrooms:ListCollaborations",
    "cleanrooms:ListConfiguredTableAssociations",
    "cleanrooms:ListConfiguredTables",
    "cleanrooms:ListMembers",
```

```
    "cleanrooms:ListMemberships",
    "cleanrooms:ListProtectedQueries",
    "cleanrooms:ListSchemas",
    "cleanrooms:UpdateAnalysisTemplate",
    "cleanrooms:UpdateCollaboration",
    "cleanrooms:UpdateConfiguredTable",
    "cleanrooms:UpdateConfiguredTableAnalysisRule",
    "cleanrooms:UpdateConfiguredTableAssociation",
    "cleanrooms:UpdateMembership",
    "cleanrooms:ListTagsForResource",
    "cleanrooms:UntagResource",
    "cleanrooms:TagResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "CleanRoomsNoQuerying",
   "Effect": "Deny",
   "Action": [
    "cleanrooms:StartProtectedQuery",
    "cleanrooms:UpdateProtectedQuery"
   ],
   "Resource": "*"
  },
  {
   "Sid": "PassServiceRole",
   "Effect": "Allow",
   "Action": [
    "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::*:role/service-role/*cleanrooms*",
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "cleanrooms.amazonaws.com"
    }
   }
  },
  {
   "Sid": "ListRolesToPickServiceRole",
   "Effect": "Allow",
   "Action": [
    "iam:ListRoles"
   ],
   "Resource": "*"
```

```
    },
    {
     "Sid": "GetRoleAndListRolePoliciesToInspectServiceRole",
     "Effect": "Allow",
     "Action": [
      "iam:GetRole",
      "iam:ListRolePolicies",
      "iam:ListAttachedRolePolicies"
     ],
     "Resource": "arn:aws:iam::*:role/service-role/*cleanrooms*"
    },
    {
     "Sid": "ListPoliciesToInspectServiceRolePolicy",
     "Effect": "Allow",
     "Action": [
      "iam:ListPolicies"
     ],
     "Resource": "*"
    },
    {
     "Sid": "GetPolicyToInspectServiceRolePolicy",
     "Effect": "Allow",
     "Action": [
      "iam:GetPolicy",
      "iam:GetPolicyVersion"
     ],
     "Resource": "arn:aws:iam::*:policy/*cleanrooms*"
    },
    {
     "Sid": "ConsoleDisplayTables",
     "Effect": "Allow",
     "Action": [
      "glue:GetDatabase",
      "glue:GetDatabases",
      "glue:GetTable",
      "glue:GetTables",
      "glue:GetPartition",
      "glue:GetPartitions",
      "glue:GetSchema",
      "glue:GetSchemaVersion",
      "glue:BatchGetPartition"
     ],
     "Resource": "*"
    },
```

```
    {
     "Sid": "EstablishLogDeliveries",
     "Effect": "Allow",
     "Action": [
      "logs:CreateLogDelivery",
      "logs:GetLogDelivery",
      "logs:UpdateLogDelivery",
      "logs:DeleteLogDelivery",
      "logs:ListLogDeliveries"
     ],
     "Resource": "*",
     "Condition": {
      "ForAnyValue:StringEquals": {
       "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
     }
    },
    {
     "Sid": "SetupLogGroupsDescribe",
     "Effect": "Allow",
     "Action": [
      "logs:DescribeLogGroups"
     ],
     "Resource": "*",
     "Condition": {
      "ForAnyValue:StringEquals": {
       "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
     }
    },
    {
     "Sid": "SetupLogGroupsCreate",
     "Effect": "Allow",
     "Action": [
      "logs:CreateLogGroup"
     ],
     "Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*",
     "Condition": {
      "ForAnyValue:StringEquals": {
       "aws:CalledVia": "cleanrooms.amazonaws.com"
      }
     }
    },
    {
```

```
    "Sid": "SetupLogGroupsResourcePolicy",
    "Effect": "Allow",
    "Action": [
      "logs:DescribeResourcePolicies",
     "logs:PutResourcePolicy"
    ],
    "Resource": "*",
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cleanrooms.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleLogSummaryQueryLogs",
    "Effect": "Allow",
    "Action": [
     "logs:StartQuery"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*"
   },
   {
    "Sid": "ConsoleLogSummaryObtainLogs",
    "Effect": "Allow",
    "Action": [
     "logs:GetQueryResults"
    ],
    "Resource": "*"
   }
  ]
}
```

AWS managed policy: `AWSCleanRoomsMLReadOnlyAccess`

You can attach `AWSCleanRoomsMLReadOnlyAccess` to your IAM principals.

This policy grants read-only permissions to resources and metadata in an `AWSCleanRoomsMLReadOnlyAccess` collaboration.

This policy includes the following permissions:

- `CleanRoomsConsoleNavigation` – Grants access to view the screens of the AWS Clean Rooms console.

- `CleanRoomsMLRead` – Allows principals read-only access to the Clean Rooms ML service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CleanRoomsConsoleNavigation",
            "Effect": "Allow",
            "Action": [
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredAudienceModelAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:ListAnalysisTemplates",
                "cleanrooms:ListCollaborationAnalysisTemplates",
                "cleanrooms:ListCollaborationConfiguredAudienceModelAssociations",
                "cleanrooms:ListCollaborations",
                "cleanrooms:ListConfiguredTableAssociations",
                "cleanrooms:ListConfiguredTables",
                "cleanrooms:ListMembers",
                "cleanrooms:ListMemberships",
                "cleanrooms:ListProtectedQueries",
                "cleanrooms:ListSchemas",
                "cleanrooms:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CleanRoomsMLRead",
            "Effect": "Allow",
            "Action": [
                "cleanrooms-ml:Get*",
                "cleanrooms-ml:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: `AWSCleanRoomsMLFullAccess`

You can attach `AWSCleanRoomsMLFullAcces` to your IAM principals. This policy grants administrative permissions that allow full access (read, write, and update) to resources and metadata needed by Clean Rooms ML.

Permissions details

This policy includes the following permissions:

- `CleanRoomsMLFullAccess` – Grants access to all Clean Rooms ML actions.
- `PassServiceRole` – Grants access to pass a service role to only the service (`PassedToService` condition) that has "cleanrooms-ml" in its name.
- `CleanRoomsConsoleNavigation` – Grants access to view the screens of the AWS Clean Rooms console.
- `CollaborationMembershipCheck` – When you start an audience generation (lookalike segment) job within a collaboration, the Clean Rooms ML service calls `ListMembers` to check that that the collaboration is valid, the caller is an active member, and the configured audience model owner is an active member. This permission is always required; the console navigation SID is only required for console users.
- `AssociateModels` – Allows principals to associate a Clean Rooms ML model with your collaboration.
- `TagAssociations` – Allows principals to add tags to the association between a lookalike model and a collaboration.
- `ListRolesToPickServiceRole` – Allows principals to list all their roles in order to choose a service role when using AWS Clean Rooms.
- `GetRoleAndListRolePoliciesToInspectServiceRole` – Allows principals to see the service role and corresponding policy in IAM.
- `ListPoliciesToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `GetPolicyToInspectServiceRolePolicy` – Allows principals to see the service role and corresponding policy in IAM.
- `ConsoleDisplayTables` – Allows principals read-only access to the AWS Glue metadata needed to show data about the underlying AWS Glue tables on the console.
- `ConsolePickOutputBucket` – Allows principals to select Amazon S3 buckets for configured audience model outputs.

- `ConsolePickS3Location` – Allows principals to select the location within a bucket for configured audience model outputs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CleanRoomsMLFullAccess",
            "Effect": "Allow",
            "Action": [
                "cleanrooms-ml:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/cleanrooms-ml*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "cleanrooms-ml.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CleanRoomsConsoleNavigation",
            "Effect": "Allow",
            "Action": [
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredAudienceModelAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:ListAnalysisTemplates",
                "cleanrooms:ListCollaborationAnalysisTemplates",
                "cleanrooms:ListCollaborationConfiguredAudienceModelAssociations",
                "cleanrooms:ListCollaborations",
                "cleanrooms:ListConfiguredTableAssociations",
                "cleanrooms:ListConfiguredTables",
```

```
                "cleanrooms:ListMembers",
                "cleanrooms:ListMemberships",
                "cleanrooms:ListProtectedQueries",
                "cleanrooms:ListSchemas",
                "cleanrooms:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CollaborationMembershipCheck",
            "Effect": "Allow",
            "Action": [
                "cleanrooms:ListMembers"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["cleanrooms-ml.amazonaws.com"]
                }
            }
        },
        {
            "Sid": "AssociateModels",
            "Effect": "Allow",
            "Action": [
                "cleanrooms:CreateConfiguredAudienceModelAssociation"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagAssociations",
            "Effect": "Allow",
            "Action": [
                "cleanrooms:TagResource"
            ],
            "Resource": "arn:aws:cleanrooms:*:*:membership/*/
configuredaudiencemodelassociation/*"
        },
        {
            "Sid": "ListRolesToPickServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "GetRoleAndListRolePoliciesToInspectServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListRolePolicies",
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/cleanrooms-ml*",
                "arn:aws:iam::*:role/role/cleanrooms-ml*"
            ]
        },
        {
            "Sid": "ListPoliciesToInspectServiceRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:ListPolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "GetPolicyToInspectServiceRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicy",
                "iam:GetPolicyVersion"
            ],
            "Resource": "arn:aws:iam::*:policy/*cleanroomsml*"
        },
        {
            "Sid": "ConsoleDisplayTables",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:GetSchema",
                "glue:GetSchemaVersion",
```

```
                "glue:BatchGetPartition"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ConsolePickOutputBucket",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ConsolePickS3Location",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*cleanrooms-ml*"
        }
    ]
}
```

AWS Clean Rooms updates to AWS managed policies

View details about updates to AWS managed policies for AWS Clean Rooms since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS Clean Rooms Document history page.

Change	Description	Date
AWSCleanRoomsFullAccessNoQuerying – Update to existing policy	Added cleanrooms:BatchGetSchemaAnalysisRule to CleanRoomsAccess.	May 13, 2024
AWSCleanRoomsFullAccess – Update to existing policy	Updated the Statement ID in AWSCleanRoomsFullAccess from ConsolePickQueryResultsBucket to SetQueryResultsBucket in this policy to better represent the permissions	March 21, 2024

Change	Description	Date
	since the permissions are needed for setting the query results bucket both with and without the console.	
[AWSCleanRoomsMLReadOnlyAccess](#) – New policy [AWSCleanRoomsMLFullAccess](#) – New policy	Added AWSCleanRoomsMLReadOnlyAccess and AWSCleanRoomsMLFullAccess to support AWS Clean Rooms ML.	November 29, 2023
[AWSCleanRoomsFullAccessNoQuerying](#) – Update to existing policy	Added cleanrooms:CreateAnalysisTemplate, cleanrooms:GetAnalysisTemplate, cleanrooms:UpdateAnalysisTemplate, cleanrooms:DeleteAnalysisTemplate, cleanrooms:ListAnalysisTemplates, cleanrooms:GetCollaborationAnalysisTemplate, cleanrooms:BatchGetCollaborationAnalysisTemplate, and cleanrooms:ListCollaborationAnalysisTemplates to CleanRoomsAccess to enable the new analysis templates feature.	July 31, 2023
[AWSCleanRoomsFullAccessNoQuerying](#) – Update to existing policy	Added cleanrooms:ListTagsForResource, cleanrooms:UntagResource, and cleanrooms:TagResource to CleanRoomsAccess to enable resource tagging.	March 21, 2023
AWS Clean Rooms started tracking changes	AWS Clean Rooms started tracking changes for its AWS managed policies.	January 12, 2023

Troubleshooting AWS Clean Rooms identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Clean Rooms and IAM.

Topics

I am not authorized to perform an action in AWS Clean Rooms

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but does not have the fictional `cleanrooms:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cleanrooms:GetWidget on resource: my-example-widget
```

In this case, Mateo's policy must be updated to allow him to access the *`my-example-widget`* resource using the `cleanrooms:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS Clean Rooms.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS Clean Rooms. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS Clean Rooms resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role.

To learn more, consult the following:

- To learn whether AWS Clean Rooms supports these features, see How AWS Clean Rooms works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` global condition context keys in resource policies to limit the permissions that AWS Clean Rooms gives another service to the resource. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. In AWS Clean Rooms, you also have to compare against the `sts:ExternalId` condition key.

The value of `aws:SourceArn` must be set to the ARN of the membership of the assumed role.

The following example shows how you can use the `aws:SourceArn` global condition context key in AWS Clean Rooms to prevent the confused deputy problem.

> **Note**
>
> The example policy applies to the trust policy of the service role that AWS Clean Rooms uses to access customer data.
> The value of *`membershipID`* is your AWS Clean Rooms membership ID in the collaboration.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIfExternalIdMatches",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "sts:ExternalId": "arn:aws:*:aws-region:*:dbuser:*/membershipID*"
                }
            }
        },
        {
            "Sid": "AllowIfSourceArnMatches",
            "Effect": "Allow",
            "Principal": {
                "Service": "cleanrooms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ForAnyValue:ArnEquals": {
```

```
                     "aws:SourceArn": "arn:aws:cleanrooms:aws-
region:123456789012:membership/membershipID"
                 }
             }
         }
     ]
}
```

IAM behaviors for AWS Clean Rooms ML

Cross-account jobs

Clean Rooms ML allows certain resources created by one AWS account to be securely accessed in their account by another AWS account. When a client in AWS account A calls `StartAudienceGenerationJob` on a `ConfiguredAudienceModel` resource owned by AWS account B, Clean Rooms ML creates two ARNs for the job. One ARN in AWS account A and another in AWS account B. The ARNs are identical except for their AWS account.

Clean Rooms ML creates two ARNs for the job to ensure that both accounts can apply their own IAM policies to the jobs. For example, both accounts can use tag-based access control and apply policies from their AWS organization. The job processes data from both accounts, so both accounts can delete the job and its associated data. Neither account can block the other account from deleting the job.

There is only one job execution and both accounts can see the job when they call `ListAudienceGenerationJobs`. Both accounts can call the `Get`, `Delete`, and `Export` APIs on the job using the ARN with their own AWS account ID.

Neither AWS account can access the job when using an ARN with the other AWS account ID.

The name of the job must be unique within an AWS account. The name in AWS account B is *`$accountA-$name`*. The name chosen by AWS account A is prefixed with AWS account A when the job is viewed in AWS account B.

In order for a cross-account `StartAudienceGenerationJob` to succeed, AWS account B must allow that action on both the new job in AWS account B and the `ConfiguredAudienceModel` in AWS account B using a resource policy similar to the following example:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Clean-Rooms-<CAMA ID>",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "accountA"
                ]
            },
            "Action": [
                "cleanrooms-ml:StartAudienceGenerationJob"
            ],
            "Resource": [
                "arn:aws:cleanrooms-ml:us-west-1:AccountB:configured-audience-
model/id",
                "arn:aws:cleanrooms-ml:us-west-1:AccountB:audience-generation-job/*"
            ],
            // optional - always set by AWS Clean Rooms
            "Condition":{"StringEquals":{"cleanrooms-ml:CollaborationId":"UUID"}}
        }
    ]
}
```

If you use the [AWS Clean Rooms ML API](https://docs.aws.amazon.com/cleanrooms-ml/latest/APIReference/Welcome.html) to create a configured lookalike model with `manageResourcePolicies` set to true, AWS Clean Rooms creates this policy for you.

Additionally, the identity policy of the caller in AWS account A needs `StartAudienceGenerationJob` permission on `arn:aws:cleanrooms-ml:us-west-1:AccountA:audience-generation-job/*`. So there are three IAM Resources for Action `StartAudienceGenerationJob`: the AWS account A job, the AWS account B job, and the AWS account B `ConfiguredAudienceModel`.

Warning

The AWS account that started the job receives an AWS CloudTrail audit log event about the job. The AWS account that owns the `ConfiguredAudienceModel` does not receive a AWS CloudTrail audit log event.

Tagging jobs

When you set the `childResourceTagOnCreatePolicy=FROM_PARENT_RESOURCE` parameter of `CreateConfiguredAudienceModel`, all lookalike segment generation jobs within your account that are created from that configured lookalike model default to having the same tags as the configured lookalike model. The configured lookalike model is the parent and the lookalike segment generation job is the child.

If you are creating a job within your own account, the request tags of the job override the parent tags. Jobs created by other accounts never create tags in your account. If you set `childResourceTagOnCreatePolicy=FROM_PARENT_RESOURCE` and another account creates a job, there are two copies of the job. The copy in your account has the parent resource tags and the copy in the job submitter's account has tags from the request.

Validating collaborators

When granting permissions to other members of an AWS Clean Rooms collaboration, the resource policy should include the condition key `cleanrooms-ml:CollaborationId`. This enforces that the `collaborationId` parameter is included in the StartAudienceGenerationJob request. When the `collaborationId` parameter is included in the request, Clean Rooms ML validates that the collaboration exists, the job submitter is an active member of the collaboration, and the configured lookalike model owner is an active member of the collaboration.

When AWS Clean Rooms manages your configured lookalike model resource policy (the `manageResourcePolicies` parameter is TRUE in CreateConfiguredAudienceModelAssociation request), this condition key will be set in the resource policy. Therefore, you must specify the `collaborationId` in StartAudienceGenerationJob.

Cross-account access

Only `StartAudienceGenerationJob` can be called across accounts. All other Clean Rooms ML APIs can only be used with resources in your own account. This ensures that your training data, lookalike model configuration, and other information stays private.

Clean Rooms ML never reveals Amazon S3 or AWS Glue locations across accounts. The training data location, configured lookalike model output location, and lookalike segment generation job seed location are never visible across accounts. If you `Get` an audience generation job that another account submitted, the service does not show the seed location.

Compliance validation for AWS Clean Rooms

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

> **Note**
>
> Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your

compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Clean Rooms

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Clean Rooms

As a managed service, AWS Clean Rooms is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access AWS Clean Rooms through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Network security

When AWS Clean Rooms reads from your S3 bucket during query execution, the traffic between AWS Clean Rooms and Amazon S3 is securely routed through the AWS private network. In-flight traffic is signed using Amazon Signature Version 4 protocol (SIGv4) and encrypted using HTTPS. This traffic is authorized based on the IAM service role which you have set up for your configured table.

You can connect programmatically to AWS Clean Rooms through an endpoint. For a list of service endpoints, see AWS Clean Rooms endpoints and quotas in the *AWS General Reference*.

All service endpoints are HTTPS-only. You can use Amazon Virtual Private Cloud (VPC) endpoints in case you want to connect to AWS Clean Rooms from your VPC and do not want to have internet connectivity. For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

You can assign IAM policies to your IAM principals which make use of the aws:SourceVpce context keys to restrict your IAM principal to only be able to make calls to AWS Clean Rooms through a VPC endpoint and not over the internet.

Access AWS Clean Rooms or AWS Clean Rooms ML using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your virtual private cloud (VPC) and AWS Clean Rooms or AWS Clean Rooms ML. You can access AWS Clean Rooms or AWS Clean Rooms ML as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to access AWS Clean Rooms.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for AWS Clean Rooms.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Considerations for AWS Clean Rooms

Before you set up an interface endpoint for AWS Clean Rooms, review Considerations in the *AWS PrivateLink Guide*.

AWS Clean Rooms and AWS Clean Rooms ML support making calls to all of their API actions through the interface endpoint.

VPC endpoint policies are not supported for AWS Clean Rooms or AWS Clean Rooms ML. By default, full access to AWS Clean Rooms and AWS Clean Rooms ML is allowed through the interface endpoint. Alternatively, you can associate a security group with the endpoint network interfaces to control traffic to AWS Clean Rooms or AWS Clean Rooms ML through the interface endpoint.

Create an interface endpoint for AWS Clean Rooms

You can create an interface endpoint for AWS Clean Rooms or AWS Clean Rooms ML using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create an interface endpoint for AWS Clean Rooms using the following service name.

```
com.amazonaws.region.cleanrooms
```

Create an interface endpoint for AWS Clean Rooms ML using the following service name.

```
com.amazonaws.region.cleanrooms-ml
```

If you enable private DNS for the interface endpoint, you can make API requests to AWS Clean Rooms using its default Regional DNS name. For example, `cleanrooms-ml.us-east-1.amazonaws.com`.

Monitoring AWS Clean Rooms

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Clean Rooms and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Clean Rooms, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, and other sources. Amazon CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.

  Clean Rooms ML allows cross-account jobs for certain API actions. The AWS account that started the job receives the AWS CloudTrail audit log event for the job. For more information, see IAM behaviors for AWS Clean Rooms ML
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Logging AWS Clean Rooms API calls using AWS CloudTrail

AWS Clean Rooms is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Clean Rooms. CloudTrail captures all API calls for AWS Clean Rooms as events. The calls captured include calls from the AWS Clean Rooms console and code calls to the AWS Clean Rooms API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Clean Rooms. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Clean Rooms, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

AWS Clean Rooms information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Clean Rooms, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for AWS Clean Rooms, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions
- Receiving CloudTrail log files from multiple accounts

All AWS Clean Rooms actions are logged by CloudTrail and are documented in the *AWS Clean Rooms API Reference*.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding AWS Clean Rooms log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Example AWS Clean Rooms CloudTrail events

The following examples demonstrate CloudTrail events for:

Topics

StartProtectedQuery (successful)


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:sts::123456789012:assumed-role/query-runner/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:role/query-runner",
                "accountId": "123456789012",
                "userName": "query-runner"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-04-07T19:34:32Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-04-07T19:53:32Z",
    "eventSource": "cleanrooms.amazonaws.com",
    "eventName": "StartProtectedQuery",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "203.0.113.1",
```

```
    "userAgent": "aws-internal/3",
    "requestParameters": {
        "resultConfiguration": {
            "outputConfiguration": {
                "s3": {
                    "resultFormat": "CSV",
                    "bucket": "cleanrooms-queryresults-jdoe-test",
                    "keyPrefix": "test"
                }
            }
        },
        "sqlParameters": "***",
        "membershipIdentifier": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "type": "SQL"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
        "protectedQuery": {
            "createTime": 1680897212.279,
            "id": "f5988bf1-771a-4141-82a8-26fcc4e41c9f",
            "membershipArn": "arn:aws:cleanrooms:us-east-2:123456789012:membership/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "membershipId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "resultConfiguration": {
                "outputConfiguration": {
                    "s3": {
                        "bucket": "cleanrooms-queryresults-jdoe-test",
                        "keyPrefix": "test",
                        "resultFormat": "CSV"
                    }
                }
            },
            "sqlParameters": "***",
            "status": "SUBMITTED"
        }
    },
    "requestID": "7464211b-2277-4b55-9723-fb4f259aefd2",
    "eventID": "f7610f5e-74b9-420f-ae43-206571ebcbf7",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
```

```
}
```

StartProtectedQuery (failed)


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:sts::123456789012:assumed-role/query-runner/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:role/query-runner",
                "accountId": "123456789012",
                "userName": "query-runner"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-04-07T19:34:32Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-04-07T19:47:27Z",
    "eventSource": "cleanrooms.amazonaws.com",
    "eventName": "StartProtectedQuery",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "203.0.113.1",
    "userAgent": "aws-internal/3",
    "errorCode": "ValidationException",
    "requestParameters": {
        "resultConfiguration": {
            "outputConfiguration": {
                "s3": {
                    "resultFormat": "CSV",
                    "bucket": "cleanrooms-queryresults-jdoe-test",
                    "keyPrefix": "test"
                }
            }
```

```
        },
        "sqlParameters": "***",
        "membershipIdentifier": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "type": "SQL"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
        "message": "Column(s) [identifier] is not allowed in select"
    },
    "requestID": "e29f9f74-8299-4a83-9d18-5ddce7302f07",
    "eventID": "c8ee3498-8e4e-44b5-87e4-ab9477e56eb5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Creating AWS Clean Rooms resources with AWS CloudFormation

AWS Clean Rooms is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources. As a result of this integration, you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want, and AWS CloudFormation provisions and configures those resources for you. Examples of resources include collaborations, configured tables, configured table associations, and memberships.

When you use AWS CloudFormation, you can reuse your template to set up your AWS Clean Rooms resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and AWS Regions.

AWS Clean Rooms and AWS CloudFormation templates

To provision and configure resources for AWS Clean Rooms and related services, you must understand AWS CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

AWS Clean Rooms supports creating collaborations, configured tables, configured table associations, and memberships in AWS CloudFormation. For more information, including examples of JSON and YAML templates for collaborations, configured tables, configured table associations, and memberships, see the AWS Clean Rooms resource type reference in the *AWS CloudFormation User Guide*.

The following templates are available:

- *Analysis template*

  Specify an AWS Clean Rooms analysis template, including a name, description, format, source, parameters, and tags.

  For more information, see the following topics:

AWS::CleanRooms::AnalysisTemplate in the *AWS Clean Rooms User Guide*

CreateAnalysisTemplate in the *AWS Clean Rooms API Reference*

- *Collaboration*

  Specify an AWS Clean Rooms collaboration, including a name, description, type, parameters, and tags.

  For more information, see the following topics:

  AWS::CleanRooms::Collaboration in the *AWS CloudFormation User Guide*

  CreateCollaboration in the *AWS Clean Rooms API Reference*

- *Configured table*

  Specify a configured table in AWS Clean Rooms, including allowed columns, analysis method, description, name, table reference, privacy budget, and tags. Configured tables represent a reference to an existing table in the AWS Glue Data Catalog that has been configured for use in AWS Clean Rooms. A configured table contains an analysis rule that determines how the data can be used.

  For more information, see the following topics:

  AWS::CleanRooms::ConfiguredTable in the *AWS CloudFormation User Guide*

  CreateConfiguredTable in the *AWS Clean Rooms API Reference*

- *Configured table association*

  Specify a configured table association in AWS Clean Rooms, including ID, description, membership ID, name, role, Amazon Resource Name (ARN), and tags. A configured table association links a configured table with a collaboration.

  For more information, see the following topics:

  AWS::CleanRooms::ConfiguredTableAssociation in the *AWS CloudFormation User Guide*

  CreateConfiguredTableAssociation in the *AWS Clean Rooms API Reference*

- *Membership*

Specify membership for a specific collaboration identifier and join the collaboration in AWS Clean Rooms.

For more information, see the following topics:

AWS::CleanRooms::Membership in the *AWS CloudFormation User Guide*

CreateMembership in the *AWS Clean Rooms API Reference*

- *Privacy Budget Template*

  Specify an AWS Clean Rooms privacy budget template, including a privacy budget, noise added per query, and monthly privacy budget refresh.

  For more information, see the following topics:

  AWS::CleanRooms::PrivacyBudgetTemplate in the *AWS CloudFormation User Guide*

  CreatePrivacyBudgetTemplate in the *AWS Clean Rooms API Reference*

- *Create training dataset*

  Specify a training dataset for a Clean Rooms ML model from a AWS Glue table.

  For more information, see the following topics:

  AWS::CleanRoomsML::TrainingDataset in the *AWS CloudFormation User Guide*

  CreateTrainingDataset in the *Clean Rooms ML API Reference*

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

Quotas for AWS Clean Rooms

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is specific to an AWS Region. You can request increases for some quotas, and other quotas can't be increased.

To view the quotas for AWS Clean Rooms, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **AWS Clean Rooms**.

To request a quota increase, see Requesting a Quota Increase in the *Service Quotas User Guide*. If the quota isn't yet available in Service Quotas, use the Service limit increase form.

Your AWS account has the following quotas related to AWS Clean Rooms.

Resource	Default	Description
Members invited per collaboration	5	Maximum number of members invited per collaboration
Memberships per account	100	Maximum number of memberships for an account
Collaborations created per account	10	Maximum number of collaborations created per account
Configured tables per account	60	Maximum number of configured tables that can be created by an account
Table associations per membership	25	Maximum number of tables associated per active membership
Concurrent ongoing queries per membership	5	Maximum number of concurrent ongoing queries per membership

Resource	Default	Description
Columns per configured table allowlist	100	Maximum number of columns that can be allowlisted per configured table
Configured tables per protected query	15	Maximum number of configured tables in a protected query
Analysis templates per membership	25	Maximum number of analysis templates per membership
Configured lookalike model (audience model) associations per membership	5	Maximum number of configured lookalike model association per membership.

Resource parameter limits

Resource	Default	Description
Analysis rule size	100 KB	Maximum size of JSON for an analysis rule
Query text length	90 KB (8KB for differential privacy queries)	Maximum text length for a SQL query statement
Query run time	12 hours	Maximum duration a query is run before timeout
Query data file output size	6.2 GB	Maximum size of an output file from a protected query

Your AWS account has the following API transaction per second (TPS) per account per endpoint quotas.

API throttling quotas

Resource	Rate limit	Description
Rate of `BatchGetCollaborationAnalysisTemplate` requests	5 TPS	Maximum number of `BatchGetCollaborationAnalysisTemplate` API calls per second
Rate of `BatchGetSchema` requests	5 TPS	Maximum number of `BatchGetSchema` API calls per second
Rate of `CreateAnalysisTemplate` requests	5 TPS	Maximum number of `CreateAnalysisTemplate` API calls per second
Rate of `CreateCollaboration` requests	5 TPS	Maximum number of `CreateCollaboration` API calls per second
Rate of `CreateConfiguredAudienceModelAssociation` requests	5 TPS	Maximum number of `CreateConfiguredAudienceModelAssociation` calls per second
Rate of `CreateConfiguredTable` requests	5 TPS	Maximum number of `CreateConfiguredTable` calls per second
Rate of `CreateConfiguredTableAnalysisRule` requests	5 TPS	Maximum number of `CreateConfiguredTableAnalysisRule` calls per second
Rate of `CreateConfiguredTableAssociation` requests	5 TPS	Maximum number of `CreateConfiguredTableAssociation` calls per second

Resource	Rate limit	Description
Rate of `CreateMembership` requests	5 TPS	Maximum number of `CreateMembership` calls per second
Rate of `CreatePrivacyBudgetTempate` requests	5 TPS	Maximum number of `CreatePrivacyBudgetTemplate` calls per second
Rate of `DeleteAnalysisTemplate` requests	5 TPS	Maximum number of `DeleteAnalysisTemplate` calls per second
Rate of `DeleteCollaboration` requests	5 TPS	Maximum number of `DeleteCollaboration` calls per second
Rate of `DeleteConfiguredAudienceModelAssociation` requests	5 TPS	Maximum number of `DeleteConfiguredAudienceModelAssociation` calls per second
Rate of `DeleteConfiguredTable` requests	5 TPS	Maximum number of `DeleteConfiguredTable` calls per second
Rate of `DeleteConfiguredTableAnalysisRule` requests	5 TPS	Maximum number of `DeleteConfiguredTableAnalysisRule` calls per second
Rate of `DeleteConfiguredTableAssociation` requests	5 TPS	Maximum number of `DeleteConfiguredTableAssociation` calls per second

Resource	Rate limit	Description
Rate of `DeleteMember` requests	5 TPS	Maximum number of `DeleteMember` calls per second
Rate of `DeleteMembership` requests	5 TPS	Maximum number of `DeleteMembership` calls per second
Rate of `DeletePrivacyBudgetTemplate` requests	5 TPS	Maximum number of `DeletePrivacyBudgetTemplate` calls per second
Rate of `GetAnalysisTemplate` requests	5 TPS	Maximum number of `GetAnalysisTemplate` calls per second
Rate of `GetCollaboration` requests	5 TPS	Maximum number of `GetCollaboration` calls per second
Rate of `GetCollaborationConfiguredAudienceModelAssociation` requests	5 TPS	Maximum number of `GetCollaborationConfiguredAudienceModelAssociation` calls per second
Rate of `GetCollaborationPrivacyBudgetTemplate` requests	5 TPS	Maximum number of `GetCollaborationPrivacyBudgetTemplate` calls per second
Rate of `GetConfiguredAudienceModelAssociation` requests	5 TPS	Maximum number of `GetConfiguredAudienceModelAssociation` calls per second

Resource	Rate limit	Description
Rate of `GetConfiguredTable` requests	5 TPS	Maximum number of `GetConfiguredTable` calls per second
Rate of `GetConfiguredTableAnalysisRule` requests	5 TPS	Maximum number of `GetConfiguredTableAnalysisRule` calls per second
Rate of `GetConfiguredTableAssociation` requests	20 TPS	Maximum number of `GetConfiguredTableAssociation` calls per second
Rate of `GetMembership` requests	5 TPS	Maximum number of `GetMembership` calls per second
Rate of `GetPrivacyBudgetTemplate` requests	5 TPS	Maximum number of `GetPrivacyBudgetTemplate` calls per second
Rate of `GetProtectedQuery` requests	20 TPS	Maximum number of `GetProtectedQuery` calls per second
Rate of `GetSchema` requests	5 TPS	Maximum number of `GetSchema` calls per second
Rate of `GetSchemaAnalysisRule` requests	5 TPS	Maximum number of `GetSchemaAnalysisRule` calls per second
Rate of `ListAnalysisTemplates` requests	5 TPS	Maximum number of `ListAnalysisTemplates` calls per second

Resource	Rate limit	Description
Rate of `ListCollaborationConfiguredAudienceModelAssociations` requests	5 TPS	Maximum number of `ListCollaborationConfiguredAudienceModelAssociations` calls per second
Rate of `ListCollaborationPrivacyBudgets` requests	5 TPS	Maximum number of `ListCollaborationPrivacyBudgets` calls per second
Rate of `ListCollaborationPrivacyBudgetTemplates` requests	5 TPS	Maximum number of `ListCollaborationPrivacyBudgetTemplates` calls per second
Rate of `ListCollaborations` requests	5 TPS	Maximum number of `ListCollaborations` calls per second
Rate of `ListConfiguredAudienceModelAssociations` requests	5 TPS	Maximum number of `ListConfiguredAudienceModelAssociations` calls per second
Rate of `ListConfiguredTableAssociations` requests	5 TPS	Maximum number of `ListConfiguredTableAssociations` calls per second
Rate of `ListConfiguredTables` requests	5 TPS	Maximum number of `ListConfiguredTables` calls per second

Resource	Rate limit	Description
Rate of `ListMembers` requests	5 TPS	Maximum number of `ListMembers` calls per second
Rate of `ListMemberships` requests	5 TPS	Maximum number of `ListMemberships` calls per second
Rate of `ListPrivacyBudgets` requests	5 TPS	Maximum number of `ListPrivacyBudgets` calls per second
Rate of `ListPrivacyBudgetTemplates` requests	5 TPS	Maximum number of `ListPrivacyBudgetTemplates` calls per second
Rate of `ListProtectedQueries` requests	5 TPS	Maximum number of `ListProtectedQueries` calls per second
Rate of `ListSchemas` requests	5 TPS	Maximum number of `ListSchemas` calls per second
Rate of `StartProtectedQuery` requests	5 TPS	Maximum number of `StartProtectedQuery` calls per second
Rate of `UpdateAnalysisTemplate` requests	5 TPS	Maximum number of `UpdateAnalysisTemplate` calls per second
Rate of `UpdateCollaboration` requests	5 TPS	Maximum number of `UpdateCollaboration` calls per second

Resource	Rate limit	Description
Rate of `UpdateConfiguredAudienceModelAssociation` requests	5 TPS	Maximum number of `UpdateConfiguredAudienceModelAssociation` calls per second
Rate of `UpdateConfiguredTable` requests	5 TPS	Maximum number of `UpdateConfiguredTable` calls per second
Rate of `UpdateConfiguredTableAnalysisRule` requests	5 TPS	Maximum number of `UpdateConfiguredTableAnalysisRule` calls per second
Rate of `UpdateConfiguredTableAssociation` requests	5 TPS	Maximum number of `UpdateConfiguredTableAssociation` calls per second
Rate of `UpdatePrivacyBudgetTemplate` requests	5 TPS	Maximum number of `UpdatePrivacyBudgetTemplate` calls per second

AWS Clean Rooms ML API throttling quotas

Resource	Rate limit	Description
Rate of `CreateAudienceModel` requests	1 TPS rate, 3 TPS burst	Maximum number of `CreateAudienceModel` API calls per second
Rate of `CreateConfiguredAudienceModel` requests	10 TPS	Maximum number of `CreateConfiguredAu`

Resource	Rate limit	Description
		dienceModel API calls per second
Rate of CreateTrainingDataset requests	10 TPS	Maximum number of CreateTrainingDataset API calls per second
Rate of DeleteAudienceGenerationJob requests	2 TPS rate, 10 TPS burst	Maximum number of DeleteAudienceGenerationJob API calls per second
Rate of DeleteAudienceModel requests	2 TPS rate, 10 TPS burst	Maximum number of DeleteAudienceModel API calls per second
Rate of DeleteConfiguredAudienceModel requests	10 TPS	Maximum number of DeleteConfiguredAudienceModel API calls per second
Rate of DeleteConfiguredAudienceModelPolicy requests	25 TPS	Maximum number of DeleteConfiguredAudienceModelPolicy API calls per second
Rate of DeleteTrainingDataset requests	10 TPS	Maximum number of DeleteTrainingDataset API calls per second
Rate of GetAudienceGenerationJob requests	50 TPS	Maximum number of GetAudienceGenerationJob API calls per second
Rate of GetAudienceModel requests	50 TPS	Maximum number of GetAudienceModel API calls per second

Resource	Rate limit	Description
Rate of `GetConfiguredAudienceModel` requests	50 TPS	Maximum number of `GetConfiguredAudienceModel` API calls per second
Rate of `GetConfiguredAudienceModelPolicy` requests	50 TPS	Maximum number of `GetConfiguredAudienceModelPolicy` API calls per second
Rate of `GetTrainingDataset` requests	50 TPS	Maximum number of `GetTrainingDataset` API calls per second
Rate of `ListAudienceExportJobs` requests	50 TPS	Maximum number of `ListAudienceExportJobs` API calls per second
Rate of `ListAudienceGenerationJobs` requests	50 TPS	Maximum number of `ListAudienceGenerationJobs` API calls per second
Rate of `ListAudienceModels` requests	50 TPS	Maximum number of `ListAudienceModels` API calls per second
Rate of `ListConfiguredAudienceModels` requests	50 TPS	Maximum number of `ListConfiguredAudienceModels` API calls per second
Rate of `ListTagsForResource` requests	50 TPS	Maximum number of `ListTagsForResource` API calls per second

Resource	Rate limit	Description
Rate of `ListTrainingDatasets` requests	50 TPS	Maximum number of `ListTrainingDatasets` API calls per second
Rate of `PutConfiguredAudienceModelPolicy` requests	25 TPS	Maximum number of `PutConfiguredAudienceModelPolicy` API calls per second
Rate of `StartAudienceExportJob` requests	1 TPS rate, 3 TPS burst	Maximum number of `StartAudienceExportJob` API calls per second
Rate of `StartAudienceGenerationJob` requests	1 TPS rate, 5 TPS burst	Maximum number of `StartAudienceGenerationJob` API calls per second
Rate of `TagResource` requests	10 TPS	Maximum number of `TagResource` API calls per second
Rate of `UntagResource` requests	50 TPS	Maximum number of `UntagResource` API calls per second
Rate of `UpdateConfiguredAudienceModel` requests	10 TPS	Maximum number of `UpdateConfiguredAudienceModel` API calls per second

Name	Default	Adjustable	Description
Active audience export jobs per audience generation job	Each supported Region: 25	No	The maximum number of active audience export jobs for an audience generation job
Pending/In-progress audience export jobs per customer	Each supported Region: 20	No	The maximum number of pending/in-progress audience export jobs per customer
Pending/In-progress audience generation jobs per customer	Each supported Region: 10	Yes	The maximum number of pending/in-progress audience generation jobs per customer
Pending/In-progress audience models per customer	Each supported Region: 2	Yes	The maximum number of pending/in-progress audience model training jobs per customer

Clean Rooms ML quotas

Resource	Default	Description
Datasets	*per job*	
Maximum number of interactions	20 billion	Maximum number of interactions allowed in training data. Larger inputs are sampled down.
Minimum number of interactions	1 million	

Resource	Default	Description
Maximum number of distinct users for lookalike model training	1 million	If more are included, only the top 100 million are used, ranked by number of interacti ons.
Minimum number of distinct users for lookalike model training	100,000	
Minimum number of users for export lookalike segment (audience) job	10,000	
Maximum number of distinct items used for model training.	1 million	You can include up to 50 million items, but only the most popular 1 million are used.
Maximum number of feature columns in the training dataset.	10	
Minimum number of distinct items per user	2	AWS Clean Rooms ML requires that each row or user has two or more items, including repeated items.
Maximum size of the seed audience	500,000	
Minimum size of the seed audience	500	The training data provider can set this value to as low as 25.
APIs	*per customer*	
Total number of active training datasets	500	

Resource	Default	Description
Total number of active lookalike models (audience models)	500	
Total number of active configured lookalike models (audience models)	10,000	
Total number of completed lookalike segment (audience) generation jobs	No limit	
Total number of completed export lookalike segment (audience) jobs	No limit	
Maximum duration of a lookalike model (audience model) generation job	1 day (24 hours)	
Maximum duration of a lookalike segment (audience) generation job	10 hours	After you provide a seed, Clean Rooms ML takes a maximum of 10 hours to generate a lookalike segment.
Minimum percentage for a segment (audience) size bin	1%	
Maximum percentage for a segment (audience) size bin	20%	
Minimum absolute size for a segment (audience) size bin	1% of the number of distinct users	
Maximum absolute size for a segment (audience) size bin	20% of the number of distinct users	

Document history for the AWS Clean Rooms User Guide

The following table describes the documentation releases for AWS Clean Rooms.

For notification about updates to this documentation, you can subscribe to the RSS feed. To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser you are using.

Change	Description	Date
Update to existing policy	The following new permission has been added to the `AWSCleanRoomsFullAccessNoQuerying` managed policy: `cleanrooms:BatchGetSchemaAnalysisRule` .	May 13, 2024
AWS Clean Rooms ML is now fully available	AWS Clean Rooms ML provides a privacy-enhancing method for two parties to identify similar users in their data without the need to share their data with each other.	April 3, 2024
Update to existing policy	The Statement ID in the `AWSCleanRoomsFullAccess` managed policy has been updated from ConsolePickQueryResultsBucket to SetQueryResultsBucket to better represent the permissions since the permissions.	March 21, 2024
New managed policies for AWS Clean Rooms ML	Two new managed policies have been added: `AWSCleanRoomsMLReadOnlyAcce`	November 29, 2023

	ss and AWSCleanR oomsMLFullAccess .	
AWS Clean Rooms ML (preview)	AWS Clean Rooms ML provides a privacy-enhancing method for two parties to identify similar users in their data without the need to share their data with each other.	November 29, 2023
AWS Clean Rooms Differential Privacy (preview)	Customers can now use AWS Clean Rooms Differential Privacy to help protect the privacy of their users.	November 29, 2023
Payment configuration	The collaboration creator can now configure either the member who can run queries or a different member in the collaboration to be billed for query compute costs.	November 14, 2023
Query run time - update	The maximum duration a query is run before timeout has been updated from 4 hours to 12 hours.	October 6, 2023

AWS CloudFormation resources - update	AWS Clean Rooms has added the following new resources : `AWS::CleanRooms::MembershipProtectedQueryOutputConfiguration` , `AWS::CleanRooms::MembershipProtectedQueryResultConfiguration` , and `AWS::CleanRooms::MembershipProtectedQueryS3OutputConfiguration` .	September 7, 2023
AWS CloudFormation resources - update	AWS Clean Rooms has added the following new resources : `AWS::CleanRooms::AnalysisTemplate` and `AWS::CleanRooms::ConfiguredTableAnalysisRuleCustom` .	August 31, 2023
Separate member abilities	The collaboration creator can now designate one member as the member who can query and another member as the member who can receive results. This gives the collaboration creator the ability to make sure that the member who can query doesn't have access to the query results.	August 30, 2023
AWS Clean Rooms Glossary	Documentation-only update to add a glossary of AWS Clean Rooms terms.	August 30, 2023

Support for Apache Iceberg tables (preview)	AWS Clean Rooms now supports Apache Iceberg tables (preview).	August 25, 2023
Quotas update	The Quotas section has been updated to reflect the new default quota for memberships per account.	August 9, 2023
Update to existing policy	The following new permissions have been added to the `AWSCleanRoomsFullAccessNoQuerying` managed policy: `cleanrooms:CreateAnalysisTemplate` , `cleanrooms:GetAnalysisTemplate` , `cleanrooms:UpdateAnalysisTemplate` , `cleanrooms:DeleteAnalysisTemplate` , `cleanrooms:ListAnalysisTemplates` , `cleanrooms:GetCollaborationAnalysisTemplate` , `cleanrooms:BatchGetCollaborationAnalysisTemplate` , and `cleanrooms:ListCollaborationAnalysisTemplates` .	July 31, 2023

Analysis templates and Custom analysis rule	AWS Clean Rooms now supports analysis templates and the **Custom** analysis rule. Analysis templates enable collaborators to build or import their own custom SQL query to use in the collaboration. With the **Custom** analysis rule, the table owner can approve custom SQL queries on their configured tables.	July 31, 2023
Analysis rules support the OR logical condition	AWS Clean Rooms analysis rules now support the OR logical condition in the JOIN clause.	June 29, 2023
CloudFormation integration	AWS Clean Rooms now integrates with AWS CloudFormation.	June 15, 2023
Analysis builder	Members who can query and receive results now have the ability to run queries on some tables without writing SQL code by using the **Analysis builder UI**.	June 15, 2023
SQL functions	Documentation-only update to clarify supported SQL functions.	May 5, 2023
Troubleshooting	Documentation-only update to add a Troubleshooting section for common issues.	April 27, 2023

Supported data types for AWS Clean Rooms	Documentation-only update to add a new section that lists supported AWS Glue Data Catalog data types.	April 26, 2023
Examples of AWS CloudTrail events	Documentation-only update to add examples of CloudTrail events for StartProtectedQuery (successful) and StartProtectedQuery (failed).	April 20, 2023
Update to existing policy	The following new permissions have been added to the `AWSCleanRoomsFullAccessNoQuerying` managed policy: `cleanrooms:ListTagsForResource`, `cleanrooms:UntagResource`, and `cleanrooms:TagResource`. For more information, see AWS managed policies.	March 21, 2023
General availability	AWS Clean Rooms is now generally available.	March 21, 2023
Preview release	Preview release of the AWS Clean Rooms User Guide	January 12, 2023

AWS Clean Rooms Glossary

Consult this glossary to become familiar with terminology that is used for AWS Clean Rooms.

Aggregation analysis rule

The query restriction that allows queries that aggregate analysis using COUNT, SUM, or AVG functions along optional dimensions. These queries won't reveal row-level information.

Supports use cases such as campaign planning, media reach, frequency, and conversion measurement.

Other types of analysis rules are custom and list.

Analysis rules

The query restrictions that authorize a specific type of query.

The analysis rule type determines what kind of analysis can be run on the configured table. Each type has a predefined query structure. You control how your table columns can be used in the structure through the query controls.

The types of analysis rules are aggregation, list, and custom.

Analysis template

A collaboration-specific, pre-approved query that can be reused.

Supports custom SQL queries supported in AWS Clean Rooms.

Can contain parameters wherever a literal value could typically appear in a SQL query. For more information about supported parameter types, see Data types in the *AWS Clean Rooms SQL Reference*.

Analysis templates only work with the custom analysis rule.

C3R encryption client

The Cryptographic Computing for Clean Rooms (C3R) encryption client.

Used to encrypt and decrypt data, C3R is a client-side encryption SDK with a command line interface.

Cleartext column

A column that is not cryptographically protected for either a JOIN or SELECT SQL construct.

Cleartext columns can be used in any part of the SQL query.

Collaboration

A secure logical boundary in AWS Clean Rooms in which members can perform SQL queries on configured tables.

Collaborations are created by the collaboration creator.

Only members who have been invited to the collaboration can join the collaboration.

A collaboration can have only one member who can query data, one member who can receive results, and one member paying for query compute costs.

All members can see the list of invited participants in the collaboration before they join the collaboration.

Collaboration creator

The member who creates a collaboration.

There is only one collaboration creator per collaboration.

Only the collaboration creator can remove members from the collaboration or delete the collaboration.

Configured table

Each configured table represents a reference to an existing table in the AWS Glue Data Catalog that has been configured for use in AWS Clean Rooms. A configured table contains an analysis rule that determines how the data can be used.

Currently, AWS Clean Rooms supports associating data stored in Amazon Simple Storage Service (Amazon S3) that is cataloged through AWS Glue.

For more information about AWS Glue, see the AWS Glue Developer Guide.

Configured tables can be associated to one or more collaborations.

> **Note**
> AWS Clean Rooms does not currently support Amazon S3 bucket locations that are registered with AWS Lake Formation.

Custom analysis rule

The query restriction that allows a specific set of pre-approved queries (analysis templates) or allows a specific set of accounts that can provide queries that use your data.

Supports use cases such as first-touch attribution, incremental analyses, and audience discovery analyses.

Supports differential privacy.

Decryption

The process of transforming encrypted data back to its original form. Decryption can only be performed if you have access to the secret key.

Differential privacy

A mathematically-rigorous technique that protects the collaboration data from the member who can receive results learning about a specific individual.

Encryption

The process of encoding data into a form that appears random using a secret value called a key. It's impossible to determine the original plaintext without access to the key.

Fingerprint column

A column that is cryptographically protected for a JOIN SQL construct.

List analysis rule

The query restriction that allows queries that output row-level attribute analysis of the overlap between this table and the tables of the member who can query.

Supports use cases such as enrichment and audience building or suppression.

Member

An AWS customer who is a participant in a collaboration.

A member is identified using their AWS account.

All members can contribute data.

Member who can query

The member who can query data in the collaboration.

There is only one member who can query per collaboration, and that member is immutable.

An administrative user can use AWS Identity and Access Management (IAM) permissions to control which of their IAM principals (such as users or roles) can query data in the collaboration. For more information, see Create a service role to read data.

Member who can receive results

The member who can receive query results. The member who can receive results specifies query results settings for the Amazon S3 destination and the query result format.

There is only one member who can receive results per collaboration, and that member is immutable.

Member paying for query compute costs

The member who is responsible for paying for query compute costs.

There is only one member who is responsible for paying for query compute costs per collaboration, and that member is immutable.

If the collaboration creator hasn't specified anyone as the member paying for query compute costs, then the member who can query is the default payer.

The member paying for query compute costs receives a bill for the queries that have been run in the collaboration.

Membership

A resource created when a member joins a collaboration.

All resources that the member associates to a collaboration are a part of the membership or are associated with the membership.

Only the member that owns the membership can add, remove, or edit resources in that membership.

Sealed column

A column that is cryptographically protected for a SELECT SQL construct.