AWS Clean Rooms: User Guide
Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon’s trademarks and trade dress may not be used in connection with any product or service that is not Amazon’s, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.
# Table of Contents

What is AWS Clean Rooms? .......................................................... 1
Are you a first-time AWS Clean Rooms user? .................................. 1
How AWS Clean Rooms works .................................................... 1
Related services ........................................................................... 3
Accessing AWS Clean Rooms ...................................................... 3
Pricing for AWS Clean Rooms ...................................................... 3
Billing for AWS Clean Rooms ...................................................... 4
Analysis rules ............................................................................... 5
Analysis rule types ........................................................................ 5
  Supported use cases .................................................................. 5
  Supported controls ................................................................. 6
Aggregation analysis rule ............................................................. 7
  Aggregation query structure and syntax .................................... 7
  Aggregation analysis rule - query controls .............................. 10
  Aggregation analysis rule - query results controls .................. 14
  Aggregation analysis rule structure ........................................ 14
  Aggregation analysis rule - example ...................................... 15
Troubleshooting aggregation analysis rule issues ........................ 18
List analysis rule .......................................................................... 18
  List query structure and syntax .............................................. 19
  List analysis rule - query controls ........................................ 20
  List analysis rule predefined structure ................................... 22
  List analysis rule - example .................................................. 22
Custom analysis rule ................................................................... 24
  Custom analysis rule predefined structure .............................. 24
  Custom analysis rule - example ............................................. 25
Cryptographic computing ............................................................ 27
  Considerations ....................................................................... 28
    Allowing mixed cleartext and encrypted data in your tables .... 28
    Allowing repeated values in fingerprint columns .................. 28
    Loosening restrictions on how fingerprint columns are named .. 29
    Determining how NULL values are represented ..................... 29
Supported file and data types ....................................................... 29
  CSV files ............................................................................. 30
  Parquet files ......................................................................... 32
  Encrypting non-string values .................................................. 32
Column names .............................................................................. 33
  Normalization of column header names .................................. 33
Column types ............................................................................... 33
  Fingerprint columns ............................................................ 34
  Sealed columns .................................................................... 34
  Cleartext columns ............................................................... 35
Parameters .................................................................................... 35
  Allow cleartext columns parameter ....................................... 35
  Allow duplicates parameter .................................................. 36
  Allow JOIN of columns with different names parameter .......... 36
  Preserve NULL values parameter .......................................... 37
Optional flags ............................................................................... 38
  --csvInputNULLValue flag .................................................. 38
  --csvOutputNULLValue flag ............................................... 39
  --enableStackTraces flag ..................................................... 39
  --dryRun flag ................................................................... 39
  --tempDir flag .................................................................. 40
Queries with C3R ........................................................................ 40
Configuring an analysis rule to a configured table .......................................................... 101
Configuring an aggregation analysis rule to a table (guided flow) .................................. 101
Configuring a list analysis rule to a table (guided flow) .................................................. 103
Configuring a custom analysis rule to a table (guided flow) ........................................ 104
Configuring analysis rule to a table (JSON editor) ....................................................... 105
Next steps ......................................................................................................................... 106
Associating a configured table to a collaboration ............................................................ 107
Associate a configured table from the configured table detail page ............................ 107
Associate a configured table from the collaboration detail page ............................... 109
Next steps ......................................................................................................................... 110
Reviewing an analysis template ....................................................................................... 111
Review an analysis template ........................................................................................... 111
Next steps ......................................................................................................................... 112
Querying data in a collaboration ...................................................................................... 112
Using the SQL code editor ............................................................................................. 112
Using the analysis builder .............................................................................................. 114
Use the analysis builder to query a single table (aggregation) ...................................... 115
Use the analysis builder to query two tables (aggregation or list) ............................... 116
Using an analysis template ............................................................................................. 117
Viewing recent queries .................................................................................................... 118
Viewing query details ....................................................................................................... 118
Receiving query results .................................................................................................... 120
Receive query results ...................................................................................................... 120
Edit default values for query results settings ................................................................. 121
Using query output in other AWS services ................................................................... 121
Decrypting data tables ..................................................................................................... 122
Managing AWS Clean Rooms .......................................................................................... 123
Managing collaborations ................................................................................................. 123
Editing collaborations ...................................................................................................... 123
Deleting collaborations .................................................................................................... 126
Viewing collaborations .................................................................................................... 126
Viewing tables and analysis rules .................................................................................. 126
Monitoring member status ............................................................................................. 127
Removing a member from a collaboration .................................................................... 127
Leaving a collaboration .................................................................................................... 127
Editing configured table associations ........................................................................... 128
Disassociating configured tables .................................................................................... 128
Managing configured tables .......................................................................................... 129
Editing configured table details ....................................................................................... 129
Editing configured table tags ........................................................................................ 129
Editing configured table analysis rule ........................................................................ 130
Deleting configured table analysis rule ........................................................................ 130
Troubleshooting ............................................................................................................... 131
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. ...................................................... 131
One of the underlying datasets has an unsupported file format ..................................... 131
Query results are not as expected when using Cryptographic Computing for Clean Rooms. ......................................................................................................................... 131
Security ............................................................................................................................. 133
Data protection ............................................................................................................... 133
Encryption at rest .............................................................................................................. 134
Encryption in transit ......................................................................................................... 134
Encrypting underlying data ............................................................................................... 134
Best practices .................................................................................................................. 134
Best practices with AWS Clean Rooms ......................................................................... 135
Best practices for using analysis rules in AWS Clean Rooms ....................................... 135
Identity and Access Management .................................................................................... 136
Audience .......................................................................................................................... 136
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticating with identities</td>
<td>137</td>
</tr>
<tr>
<td>Managing access using policies</td>
<td>139</td>
</tr>
<tr>
<td>How AWS Clean Rooms works with IAM</td>
<td>140</td>
</tr>
<tr>
<td>Identity-based policy examples</td>
<td>145</td>
</tr>
<tr>
<td>AWS managed policies</td>
<td>147</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>156</td>
</tr>
<tr>
<td>Cross-service confused deputy prevention</td>
<td>158</td>
</tr>
<tr>
<td>Compliance validation</td>
<td>159</td>
</tr>
<tr>
<td>Resilience</td>
<td>159</td>
</tr>
<tr>
<td>Infrastructure security</td>
<td>160</td>
</tr>
<tr>
<td>Network security</td>
<td>160</td>
</tr>
<tr>
<td>AWS PrivateLink considerations</td>
<td>161</td>
</tr>
<tr>
<td>Create an interface endpoint</td>
<td>161</td>
</tr>
<tr>
<td>Monitoring</td>
<td>162</td>
</tr>
<tr>
<td>CloudTrail logs</td>
<td>162</td>
</tr>
<tr>
<td>AWS Clean Rooms information in CloudTrail</td>
<td>162</td>
</tr>
<tr>
<td>Understanding AWS Clean Rooms log file entries</td>
<td>163</td>
</tr>
<tr>
<td>Example AWS Clean Rooms CloudTrail events</td>
<td>163</td>
</tr>
<tr>
<td>AWS CloudFormation resources</td>
<td>166</td>
</tr>
<tr>
<td>AWS Clean Rooms and AWS CloudFormation templates</td>
<td>166</td>
</tr>
<tr>
<td>Learn more about AWS CloudFormation</td>
<td>167</td>
</tr>
<tr>
<td>Quotas</td>
<td>168</td>
</tr>
<tr>
<td>Document history</td>
<td>172</td>
</tr>
<tr>
<td>Glossary</td>
<td>175</td>
</tr>
<tr>
<td>Aggregation analysis rule</td>
<td>175</td>
</tr>
<tr>
<td>Analysis rules</td>
<td>175</td>
</tr>
<tr>
<td>Analysis template</td>
<td>175</td>
</tr>
<tr>
<td>C3R encryption client</td>
<td>175</td>
</tr>
<tr>
<td>Cleartext column</td>
<td>175</td>
</tr>
<tr>
<td>Collaboration</td>
<td>176</td>
</tr>
<tr>
<td>Collaboration creator</td>
<td>176</td>
</tr>
<tr>
<td>Configured table</td>
<td>176</td>
</tr>
<tr>
<td>Custom analysis rule</td>
<td>176</td>
</tr>
<tr>
<td>Decryption</td>
<td>176</td>
</tr>
<tr>
<td>Encryption</td>
<td>176</td>
</tr>
<tr>
<td>Fingerprint column</td>
<td>177</td>
</tr>
<tr>
<td>List analysis rule</td>
<td>177</td>
</tr>
<tr>
<td>Member</td>
<td>177</td>
</tr>
<tr>
<td>Member who can query</td>
<td>177</td>
</tr>
<tr>
<td>Member who can receive results</td>
<td>177</td>
</tr>
<tr>
<td>Member paying for query compute costs</td>
<td>177</td>
</tr>
<tr>
<td>Membership</td>
<td>178</td>
</tr>
<tr>
<td>Sealed column</td>
<td>178</td>
</tr>
</tbody>
</table>
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 to create your own clean rooms in minutes, and start analyzing your collective datasets with just a few steps. You can choose the partners you want to collaborate with, 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 (p. 5) to restrict SQL queries and provide output constraints
- Cryptographic Computing for Clean Rooms (p. 27) to keep data encrypted, even as queries are processed, to comply with stringent data handling policies
- Query logs (p. 57) to review queries and help support audits

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 (p. 1)
- Accessing AWS Clean Rooms (p. 3)
- Setting up AWS Clean Rooms (p. 59)
- AWS Clean Rooms Glossary (p. 175)

How AWS Clean Rooms works

The following workflow assumes that:

- The collaboration member has already uploaded their data tables to Amazon S3 (p. 76) and created an AWS Glue table (p. 76).
- (Optional) For encrypted (p. 177) data tables only, the collaboration member has already prepared encrypted data tables (p. 81) using the C3R encryption client.

In summary, the workflow for AWS Clean Rooms is as follows:
1. The **collaboration creator** (p. 176) does the following tasks:
   - Creates a collaboration (p. 67).
   - Invites one or more members (p. 177) to the collaboration (p. 176).
   - Assigns abilities to members, such as the **member who can query** (p. 177) and the **member who can receive results** (p. 177).
   
   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 (p. 177).

2. The invited member **joins the collaboration by creating a membership resource** (p. 72).

   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** (p. 177) **configures an existing AWS Glue table for use in AWS Clean Rooms** (p. 97).

   (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 (p. 76).

   a. The member names the **configured table** (p. 176) and chooses which columns to use in the collaboration.
   b. The member **configures one of the following analysis rules to the configured table** (p. 101):
      - **Aggregation analysis rule** (p. 175) or list analysis rule (p. 177) – To control the type of analysis that can be run on the table.
      - **Custom analysis rule** (p. 176) – To allow a specific set of pre-approved queries or a specific set of accounts that can provide queries that use your data.

   **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** (p. 107) 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.

5. The member who can query **runs SQL queries on the configured tables** (p. 112).

   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 7).

6. The **member who can receive results** (p. 177) 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 (p. 177) 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 (p. 176) 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 (p. 75)
  - 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 (p. 75)
  - 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 (p. 166).

- **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 (p. 162).

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 (p. 177) and the member paying for query compute costs (p. 177) 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 won’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 will 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 (p. 67).

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 has not allowed the analysis template.

  Because Collaborator B has not 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 (p. 7), list (p. 18) and custom (p. 24). The following tables compare the analysis rule types. Each type has a separate section which describes specifying the analysis rule.

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

### Supported use cases

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

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

### Supported analyses

<table>
<thead>
<tr>
<th>Use case</th>
<th>Aggregation (p. 7)</th>
<th>List (p. 18)</th>
<th>Custom (p. 24)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supported analyses</strong></td>
<td>Queries that aggregate statistics using COUNT, SUM, and/or AVG functions along optional dimensions</td>
<td>Queries that output row-level lists of the overlap between multiple tables</td>
<td>Any custom analysis as long as the analysis template or the analysis creator have been reviewed and allowed</td>
</tr>
<tr>
<td><strong>Common use cases</strong></td>
<td>Segment analysis, measurement, attribution</td>
<td>Enrichment, segment building</td>
<td>First-touch attribution, incremental analyses, audience discovery</td>
</tr>
</tbody>
</table>
| **SQL constructs**     | • JOIN statements: INNER JOIN  
                          • Aggregate functions: COUNT/COUNT DISTINCT, SUM/SUM DISTINCT, and AVG  
                          • Scalar functions: Limited subset | • JOIN statements: INNER JOIN  
                          • Scalar functions: None | Majority of SQL functions and SQL constructs available with the SELECT command |
| **Subqueries and common table expressions (CTEs)** | No | No | Yes |
| **Analysis templates** | No | No | Yes |
### 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](p. 7).

The parameters of the analysis rule, defined in [Aggregation analysis rule - query controls](p. 10), 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 ] [, ...]
```
Aggregation query structure and syntax

```
-- 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.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
</table>
| select_aggregate_function_expression | A comma-separated list containing the following expressions:  
• select_aggregation_function_expression  
• select_aggregate_expression | SELECT SUM(PRICE), user_segment                                                               |
| 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. | AVG(PRICE)  
COUNT(DISTINCT user_id) |
| Note                           | There must be at least one select_aggregation_function_expression in the select_aggregate_expression. |
| select_grouping_column_expression | An expression that can contain any expression using the following:  
• Table column names  
• Supported scalar functions  
• String literals | TRUNC(timestampColumn)  
UPPER(campaignName) |
### Expression

<table>
<thead>
<tr>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical literals</td>
<td>FROM consumer_table INNER JOIN provider_table ON consumer_table.identifier1 = provider_table.identifier1 AND consumer_table.identifier2 = provider_table.identifier2</td>
</tr>
</tbody>
</table>

#### Note

select_aggregate_expression can alias columns with or without the AS parameter. For more information, see the AWS Clean Rooms SQL Reference.

### table_expression

A table, or join of tables, connecting join conditional expressions with join_condition.

**join_condition** returns a Boolean.

The table_expression supports:

- A specific JOIN type (INNER JOIN)
- The equality comparison condition within a join_condition (=)
- Logical operators (AND, OR).

### 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 (\(=\), \(\geq\), \(<\), \(\leq\), \(\neq\), \(!=\), \(\not\in\), \(\in\), \(\text{LIKE}\), \(\text{IS NULL}\), \(\text{IS NOT NULL}\)).

Supported logical operators are (AND, OR).

The where_expression is optional.
Aggregation analysis rule - query controls

<table>
<thead>
<tr>
<th>Expression</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_by_expression</td>
<td>A comma-separated list of expressions that match the requirements for the select_grouping_column_expression.</td>
<td>GROUP BY TRUNC(timestampColumn), UPPER(campaignName), segment.</td>
</tr>
<tr>
<td>having_expression</td>
<td>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.</td>
<td>HAVING SUM(SALES) &gt; 500</td>
</tr>
<tr>
<td>order_by_expression</td>
<td>A comma-separated list of expressions that is compatible with the same requirements defined in select_aggregate_expression defined earlier.</td>
<td>ORDER BY SUM(SALES), UPPER(campaignName)</td>
</tr>
</tbody>
</table>

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.
Aggregation analysis rule - query controls

Topics
- Aggregation controls (p. 11)
- Join controls (p. 11)
- Dimension controls (p. 13)
- Scalar functions (p. 13)

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.

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregateColumns</td>
<td>Columns of configured table columns you allow for use within aggregation functions.</td>
<td>aggregateColumns can be used inside an aggregation 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).</td>
</tr>
<tr>
<td>function</td>
<td>The COUNT, SUM, and AVG functions you allow for use on top of the aggregateColumns.</td>
<td>function can be applied to an aggregateColumns that is associated to it.</td>
</tr>
</tbody>
</table>

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.

```sql
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.

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

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
</table>
| 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 an aggregateColumn (see Aggregation controls (p. 11)).
|               |                                                                           | The same column can't be used both as joinColumn and dimensionColumns (see later).                                                  |
|               |                                                                           | Unless it has also been categorized as an aggregateColumn, 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.

```sql
ON
my_table.identifier = 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.

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimensionColumns</td>
<td>The columns (if any) that you allow the member who can query to use in SELECT, WHERE, GROUP BY, and ORDER BY.</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Scalar functions

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

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>scalarFunctions</td>
<td>The scalar functions that can be used on dimensionColumns in the query.</td>
<td>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.</td>
</tr>
</tbody>
</table>

The following scalar functions are supported:

- Math functions – ABS, CEILING, LOG, LN, ROUND, SQRT, FLOOR
- Data type formatting functions – CAST
- String functions – LOWER, UPPER, RTRIM
  - For RTRIM, custom character sets to trim aren't allowed.
- Conditional expressions – COALESCE
- Other functions – TRUNC

For more details, see the [AWS Clean Rooms SQL Reference](https://aws.amazon.com/blogs/aws/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.

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>columnName</code></td>
<td>The aggregateColumn that is used in the condition that each output row must meet.</td>
<td>Can be any column in the configured table.</td>
</tr>
<tr>
<td><code>minimum</code></td>
<td>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.</td>
<td>The minimum must be at least value of 2.</td>
</tr>
</tbody>
</table>

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.

```json
{
    "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.

```json
{
    "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",
            "constraint": 15
        }
    ]
}
```
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.

```json
{
  "aggregateColumns": [
    {
      "columnNames": [
        "identifier"
      ],
      "function": "COUNT_DISTINCT"
    },
    {
      "columnNames": [
        "returns"
      ],
      "function": "AVG"
    },
    {
      "columnNames": [
        "returns"
      ],
      "function": "SUM"
    }
  ],
  "joinColumns": [
    "hashedemail"
  ],
  "joinRequired": [
    "QUERY_RUNNER"
  ],
  "dimensionColumns": [
    "state",
    "popularpurchases",
```
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(companyAhashedemail))
FROM sales companyA
INNER JOIN returns companyB
ON companyA.identifier = companyB.identifier
WHERE companyA.purchasedate > '2022-01-01'
AND companyA.purchasedate < '2023-01-01'
AND TRUNC(companyB.returndate) > companyA.purchasedate

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 (p. 18)

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 (p. 15).

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 (p. 22).

The parameters of the list analysis rule, defined in List analysis rule - query controls (p. 20), 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 (p. 7).

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

Topics

• List query structure and syntax (p. 19)
• List analysis rule - query controls (p. 20)
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.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>select_list_expression</code></td>
<td>A comma-separated list containing at least one table column name.</td>
<td>SELECT DISTINCT segment</td>
</tr>
<tr>
<td></td>
<td>A DISTINCT parameter is required.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The <code>select_list_expression</code> can alias columns with or without the <code>AS</code> parameter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>It also supports the <code>TOP</code> parameter. For more information, see the <code>AWS Clean Rooms SQL Reference</code>.</td>
<td></td>
</tr>
<tr>
<td><code>table_expression</code></td>
<td>A table, or join of tables, with <code>join_condition</code> to connect it to <code>join_condition</code>.</td>
<td>FROM consumer_table INNER JOIN provider_table ON consumer_table.identifier1 = provider_table.identifier1 AND consumer_table.identifier2 = provider_table.identifier2</td>
</tr>
<tr>
<td></td>
<td><code>join_condition</code> returns a Boolean.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The <code>table_expression</code> supports:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A specific JOIN type (INNER JOIN)</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>Definition</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
| **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 |
| **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 analysis 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 (p. 21)
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.

```sql
ON
my_table.identifier = 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.

```sql
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.

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

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>joinColumns</td>
<td>The columns that you want to allow the member who can query to use in the INNER JOIN statement.</td>
<td>The same column can't be categorized as both a joinColumn and listColumn (see List controls (p. 22)).</td>
</tr>
</tbody>
</table>
**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.

```json
{
  "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.

```json
{
    "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.

```json
{
    "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.

```sql
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 which 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 (p. 10) and list (p. 19) 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 (p. 7) and list (p. 19) queries.

The custom analysis rule supports more advanced use cases than what 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.

To configure the custom analysis rule, data owners can choose to allow specific custom queries, stored in analysis templates (p. 99), 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 (p. 99). 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 (p. 24)
- Custom analysis rule - example (p. 25)

Custom analysis rule predefined structure

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

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

You can either:

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

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

- Add member AWS account IDs to the allowedAnalysisProviders control. In this case, you add ANY_QUERY to the allowedAnalyses control.
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.
7. Company B adds 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.

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

```

```

```sql
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 labeleddatahashedemail = providerhashedemail
WHERE MONTH(labeleddata.purchasedate) > :campaignmonth
AND testgroup = :group
"
}

10. Company A adds their account (for example, 657845239416) 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": [
    "657845239416"
  ]
}

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 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-1:657835239466:membership/41327cc4-bbf0-43f1-b70c-a160dddeba8/analyisistemplate/1ff1bf9d-781c-418d-a6ac-2b80c89d6292"
  ]
}

14. Company A runs the analysis template and uses the parameter value 05-01-2023.
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 (p. 5). 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 (p. 177) 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 (p. 34). 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 (p. 28)
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 (p. 28)
- Allowing repeated values in fingerprint columns (p. 28)
- Loosening restrictions on how fingerprint columns are named (p. 29)
- Determining how NULL values are represented (p. 29)

For more information about how to set parameters for these scenarios, see Cryptographic computing parameters (p. 35).

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 (p. 35).

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.
Loosening restrictions on how fingerprint columns are named

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 (p. 36).

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 (p. 36).

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 (p. 37).

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 (p. 30)
• Parquet files (p. 32)
• Encrypting non-string values (p. 32)

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.

<table>
<thead>
<tr>
<th>&lt;csv-input-null&gt;</th>
<th>&lt;csv-output-null&gt;</th>
<th>Input entry</th>
<th>Output entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None</td>
<td>,AnyProduct</td>
<td>,AnyProduct</td>
</tr>
</tbody>
</table>
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 (p. 90).

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:

<table>
<thead>
<tr>
<th>Equivalence class</th>
<th>Supported AWS Clean Rooms data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOLEAN</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>INTEGRAL</td>
<td>BIGINT, INT, SMALLINT</td>
</tr>
<tr>
<td>STRING</td>
<td>CHAR, STRING, VARCHAR</td>
</tr>
</tbody>
</table>

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.
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.

Padding data for a fingerprint column before encryption

When you specify that a column should be a fingerprint 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 is not 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.

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.

Cleartext columns

Cleartext columns are columns that are not 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 (p. 67). 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 (p. 35)
• Allow duplicates parameter (p. 36)
• Allow JOIN of columns with different names parameter (p. 36)
• Preserve NULL values parameter (p. 37)

Allow cleartext columns parameter

In the console, you can set the Allow cleartext columns parameter when creating a collaboration (p. 67) 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.

<table>
<thead>
<tr>
<th>Parameter value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Cleartext columns aren't allowed in the encrypted table. All data is cryptographically protected.</td>
</tr>
<tr>
<td>Yes</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 (p. 67) 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** (p. 36), and **Preserve NULL values** (p. 37) parameters have separate but related effects.

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

<table>
<thead>
<tr>
<th>Parameter value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Repeated values are not allowed in a fingerprint column. All values in a single fingerprint column must be unique.</td>
</tr>
<tr>
<td>Yes</td>
<td>Repeated values are allowed in a fingerprint column. 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.</td>
</tr>
</tbody>
</table>

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 (p. 67) to specify if JOIN statements between columns with different names are supported.

For more information, see [Normalization of column header names](#) (p. 33)

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

<table>
<thead>
<tr>
<th>Parameter value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Joins of fingerprint columns with different names are not supported. JOIN statements only provide accurate results on columns that have the same name.</td>
</tr>
</tbody>
</table>
Preserve NULL values parameter

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 (p. 82). 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 (p. 67) to indicate that there is no value present for that column.

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

<table>
<thead>
<tr>
<th>Parameter value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>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.</td>
</tr>
</tbody>
</table>
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** (p. 38)
- **--csvOutputNULLValue flag** (p. 39)
- **--enableStackTraces flag** (p. 39)
- **--dryRun flag** (p. 39)
- **--tempDir flag** (p. 40)

**--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.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional. Users can specify custom encodings for NULL entries in the input data.</td>
<td>User-specified encoding of NULL values in the input CSV file</td>
</tr>
</tbody>
</table>

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 (p. 88) using the C3R encryption client.

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

<table>
<thead>
<tr>
<th>Usage</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional. Users can specify custom encodings in the generated output file for NULL entries.</td>
<td>User-specified encoding of NULL values in the output CSV file</td>
</tr>
</tbody>
</table>

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 (p. 88) 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.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional. Used to provide additional contextual information for error reporting when the C3R encryption client encounters an error.</td>
<td>None</td>
</tr>
</tbody>
</table>

**--dryRun flag**

The encrypt (p. 88) and decrypt (p. 122) 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.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional. Causes the C3R encryption client to parse parameters and check files, but performs no encryption or decryption.</td>
<td>None</td>
</tr>
</tbody>
</table>

**`--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 (p. 88) 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.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users can specify the location where temporary files can be created while processing the input.</td>
<td>Defaults to the system temporary directory.</td>
</tr>
</tbody>
</table>

**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 (p. 40)
- Mapping one source column to multiple target columns (p. 41)
- Using the same data for both JOIN and SELECT queries (p. 41)

**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 (p. 41).

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 (p. 27).

Topics
- Performance implications for column types (p. 41)
- Troubleshooting unanticipated increases in ciphertext size (p. 55)

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 (p. 35)
- Fingerprint columns (p. 42)
- Sealed columns (p. 45)
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 (p. 42)
- Collaboration settings for fingerprint columns (p. 42)
- Example data for a fingerprint column (p. 43)
- Troubleshooting fingerprint columns (p. 44)

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 $BASE\_OVERHEAD = C3R\_DESIGNATION + (MAC * 1.33)$

![Diagram showing base overhead calculation](image)

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**: wJalrXUtntFEMI/K7MDfNg/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**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>TRUE</td>
</tr>
<tr>
<td>Output</td>
<td>null</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>FALSE</td>
</tr>
<tr>
<td>Output</td>
<td>01:hmac:3lkFjthvV3IUu6mMvFc1a+XAHgwg/E4mOq4p3Yg25kk=</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>52</td>
</tr>
</tbody>
</table>

**Example 3**

<table>
<thead>
<tr>
<th>Input</th>
<th>empty string</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:hmac:oKTgi3Gba +eUb3JteSz2EMgUXkF1WgM77UP0Ydw5kPQ=</td>
</tr>
</tbody>
</table>
Performance implications for column types

<table>
<thead>
<tr>
<th>Deterministic</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>52</td>
</tr>
</tbody>
</table>

**Example 4**

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyz</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:hmac:kU/IqwG7FMzzshr0B9scomE0UJUEE7j9keTctp1Gww=</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>26</td>
</tr>
<tr>
<td>Output bytes</td>
<td>52</td>
</tr>
</tbody>
</table>

**Example 5**

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:hmac:ks3htnQbw2vdhCRFF6JNzW5LMndJaHG57uvE26mBt</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>62</td>
</tr>
<tr>
<td>Output bytes</td>
<td>52</td>
</tr>
</tbody>
</table>

**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 section called "Parameters" (p. 35) 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 (p. 45)
- Collaboration settings for sealed columns (p. 45)
- Schema settings sealed columns: padding types (p. 46)
- Example data for a sealed column (p. 46)
- Troubleshooting sealed columns (p. 54)

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 prepended 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 (p. 46)
- Pad type of fixed (p. 46)
- Pad type of max (p. 46)

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).

$$\text{PADDING\_OVERHEAD} = (\text{PAD\_LENGTH} - \text{AVG\_CLEARTEXT\_LENGTH}) \times 1.33 \times \text{ROW\_COUNT}$$

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**: wJalrXUtFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

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

**Topics**
- **Pad type of none** (p. 47)
- **Pad type of fixed (Example 1)** (p. 48)
- **Pad type of fixed (Example 2)** (p. 50)
- **Pad type of max (Example 1)** (p. 51)
- **Pad type of max (Example 2)** (p. 53)

**Pad type of none**

**Example 1**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>TRUE</td>
</tr>
<tr>
<td>Output</td>
<td>null</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>FALSE</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMEYyMzQ1NjcyOTM</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>91</td>
</tr>
</tbody>
</table>

**Example 3**

<table>
<thead>
<tr>
<th>Input</th>
<th>empty string</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMEYyMzQ1NjcyOTM</td>
</tr>
</tbody>
</table>
AWS Clean Rooms User Guide
Performance implications for column types

<table>
<thead>
<tr>
<th>Deterministic</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>91</td>
</tr>
</tbody>
</table>

**Example 4**

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyz</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmg0MIDEyMzQ1Njc4OTBTcJc=</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>26</td>
</tr>
<tr>
<td>Output bytes</td>
<td>127</td>
</tr>
</tbody>
</table>

**Example 5**

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmg0MIDEyMzQ1Njc4OTBTcJc=</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>62</td>
</tr>
<tr>
<td>Output bytes</td>
<td>175</td>
</tr>
</tbody>
</table>

**Pad type of fixed (Example 1)**

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

**Example 1**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>TRUE</td>
</tr>
<tr>
<td>Output</td>
<td>null</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
</table>
### Performance implications for column types

<table>
<thead>
<tr>
<th>preserveNulls</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDExMzQ1Njc4OTT+Mx9jy48Fcg1yOPvBqR570qy1V3UKyYTLEZb/hCz7oaineVsirc0NpATs0zn6nkor4L/aSuA=</td>
</tr>
</tbody>
</table>

Deterministic No
Input bytes 0
Output bytes 175

---

**Example 3**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty string</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDExMzQ1Njc4OTT+Mx9jy48Fcg1yOPvBqR570qy1V3UKyYTLEZb/hCz7oaineVsirc0NpATs0zn6nkor4L/aSuA=</td>
</tr>
</tbody>
</table>

Deterministic No
Input bytes 0
Output bytes 175

---

**Example 4**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcdefghijklmnopqrstuvwxyz</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDExMzQ1Njc4OTT+Mx9jy48Fcg1yOPvBqR570qy1V3UKyYTLEZb/hCz7oaineVsirc0NpATs0zn6nkor4L/aSuA=</td>
</tr>
</tbody>
</table>

Deterministic No
Input bytes 26
Output bytes 175

---

**Example 5**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDExMzQ1Njc4OTT+Mx9jy48Fcg1yOPvBqR570qy1V3UKyYTLEZb/hCz7oaineVsirc0NpATs0zn6nkor4L/aSuA=</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

### Example 2

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Ø1:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OT +Mx9jy48Fcg1y0PvBqRS7oqy1V3UKFyTLEZb/hCz7oaIneVsr cnkB0xbLWD7zNdAqQGR0ix0SE5dW0I0vpNoGc +8o4WtG/ClipNcjDXvXVtK4vfCojhCA6uwimwv/xAySX +xcntotL703aB7Bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>

### Example 3

<table>
<thead>
<tr>
<th>Input</th>
<th>empty string</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Ø1:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmNlMDEyMzQ1Njc4OT +Mx9jy48Fcg1y0PvBqRS7oqy1V3UKFyTLEZb/hCz7oaIneVsr cnkB0xbLWD7zNdAqQGR0ix0SE5dW0I0vpNoGc +8o4WtG/ClipNcjDXvXVtK4vfCojhCA6uwimwv841VaT9Yd +6oQx65/+gdVT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>
### Example 4

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th>abcdefghijklmnopqrstuvwxyz</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>preserveNulls</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnIMDEyMzQ1Njc4OTTHCz7oaIevsrcnkB0xbLWD7zNdAqQGR0rXoSESdW0I0vpNoGc8O4WtG/ClipNcjDXvXVtK4vfCohcCA6uwImwtX5Hn1WyfO6ks3QMaRDGSf</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Input bytes</strong></td>
<td>26</td>
</tr>
<tr>
<td><strong>Output bytes</strong></td>
<td>307</td>
</tr>
</tbody>
</table>

### Example 5

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>preserveNulls</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnIMDEyMzQ1Njc4OTTHQQ3cXb/pbvPcnkB0xbLWD7zNdAqQGR0rXoSESdW0I0vpNoGcBfBv4cJbG8O4WtG/ClipNcjDXvXVtK4vfCohcCA6uwImwjkJXQZ0gPdeFX9Yr/8ali</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Input bytes</strong></td>
<td>62</td>
</tr>
<tr>
<td><strong>Output bytes</strong></td>
<td>307</td>
</tr>
</tbody>
</table>

**Pad type of max (Example 1)**

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

### Example 1

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>preserveNulls</strong></td>
<td>TRUE</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>null</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Input Bytes</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Output Bytes</strong></td>
<td>0</td>
</tr>
</tbody>
</table>

### Example 2

<table>
<thead>
<tr>
<th><strong>Input</strong></th>
<th>null</th>
</tr>
</thead>
</table>
### Performance implications for column types

<table>
<thead>
<tr>
<th>preserveNulls</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDEyMzQ1Njc4OTT +Mx9jy48Fcgiy0PvBqR52Z0qy1V3UKfYtLZzb/hCz7ao1eVsrcoNpATs0GzbnLkor4L+/aSuA=</td>
</tr>
</tbody>
</table>

**Deterministic**: No
**Input bytes**: 0
**Output bytes**: 175

---

#### Example 3

<table>
<thead>
<tr>
<th>Input</th>
<th>preserveNulls</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty string</td>
<td>-</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDEyMzQ1Njc4OTT +Mx9jy48Fcgiy0PvBqR52Z0qy1V3UKfYtLZzb/hCz7ao1eVsrcoNpATs0GzbnLkor4L+/aSuA=</td>
</tr>
</tbody>
</table>

**Deterministic**: No
**Input bytes**: 0
**Output bytes**: 175

---

#### Example 4

<table>
<thead>
<tr>
<th>Input</th>
<th>preserveNulls</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcdefghijklmnopqrstuvwxyz</td>
<td>-</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDEyMzQ1Njc4OTT +hCz7ao1eVsrctBAcO+Mb9tuU2KIH31AWg=</td>
</tr>
</tbody>
</table>

**Deterministic**: No
**Input bytes**: 26
**Output bytes**: 175

---

#### Example 5

<table>
<thead>
<tr>
<th>Input</th>
<th>preserveNulls</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</td>
<td>-</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbmnLMDEyMzQ1Njc4OTT +QQQ3Xb/pbvPcnohrHIGS5x4ua1/jfcVjc=</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>TRUE</td>
</tr>
<tr>
<td>Output</td>
<td>null</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2**

<table>
<thead>
<tr>
<th>Input</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>FALSE</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uYTUwMTIzNDU2Nzg5MG5vbnmMDEyMzQ1Nj40T0+Mx9ry48Fcg1yOPvBqRZ7oqy1V3UKfYTEZb/hCz7oaIneVsrncnB0xbLWD7zNdAqQGR0iXo5E5dW0I0vNoGc+8o4WtG/ClipNcDXvVXvK4vfCohCA6uwImwv/xAySX+xcntotL703aBTBb</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>

**Example 3**

<table>
<thead>
<tr>
<th>Input</th>
<th>empty string</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uYTUwMTIzNDU2Nzg5MG5vbnmMDEyMzQ1Nj40T0+Mx9ry48Fcg1yOPvBqRZ7oqy1V3UKfYTEZb/hCz7oaIneVsrncnB0xbLWD7zNdAqQGR0iXo5E5dW0I0vNoGc+8o4WtG/ClipNcDXvVXvK4vfCohCA6uwImwv84lVat9Yd+6oq65/+gdVT</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>0</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>
### Example 4

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyz</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbN1MDEyMzQ1Njc40T1c7oa1eVsncn80XbL5Wd7zNdAqQGR0KxOsesdW0I0vpNoGc +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6uwimwtX5Hnl +Wyf0Gks3QMARSGf</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>26</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>

### Example 5

<table>
<thead>
<tr>
<th>Input</th>
<th>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789</th>
</tr>
</thead>
<tbody>
<tr>
<td>preserveNulls</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>01:enc:bm9uY2UwMTIzNDU2Nzg5MG5vbN1MDEyMzQ1Njc40T1c7oa1eVsncn80XbL5Wd7zNdAqQGR0KxOsesdW0I0vpNoGc +8o4WtG/ClipNcjDXvXVtK4vfCohcCA6uwimwtX5Hnl +Wyf0Gks3QMARSGf</td>
</tr>
<tr>
<td>Deterministic</td>
<td>No</td>
</tr>
<tr>
<td>Input bytes</td>
<td>62</td>
</tr>
<tr>
<td>Output bytes</td>
<td>307</td>
</tr>
</tbody>
</table>

### 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 name (747 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 (p. 27) 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? (p. 55)
- Is the size increase coming from a sealed column? (p. 56)

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 (p. 42) 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 (p. 27).

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 (p. 45) 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 and how enabling it impacts the privacy assurances of your data, see Cryptographic computing (p. 27).
Query logging

Query logging is a feature in AWS Clean Rooms. When you create a collaboration (p. 67) 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 (p. 72).

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](#) (p. 67).

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.

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

Topics
• Sign up for AWS (p. 59)
• Create an administrator user (p. 59)
• Create an IAM role for a collaboration member (p. 60)
• Create a service role to read data (p. 61)
• Create a service role to write results (p. 63)

Sign up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account
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.

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.

<table>
<thead>
<tr>
<th>Choose one way to manage your administra</th>
<th>To</th>
<th>By</th>
<th>You can also</th>
</tr>
</thead>
<tbody>
<tr>
<td>In IAM Identity Center (Recommended)</td>
<td>Use short-term credentials to access AWS.</td>
<td>Following the instructions in Getting started in the AWS IAM Identity Center User Guide.</td>
<td>Configure programmatic access by Configuring the AWS CLI to use AWS IAM Identity Center in the AWS Command Line Interface User Guide.</td>
</tr>
</tbody>
</table>
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

2. Under Access management, choose Roles. Roles enables you to create short-term credentials, which is recommended for increased security. You can also choose Users to create long-term credentials.
3. Choose Create role.
4. In the Create role wizard, for Trusted entity type, choose AWS account.
5. Leave the option This account selected, and then choose Next.
6. For Add permissions, choose Create policy.
   a. In the Policy editor, select the JSON tab, 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:

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Then use ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>View the resources and</td>
<td>AWS managed policy: AWSCleanRoomsReadOnlyAccess (p. 147)</td>
</tr>
<tr>
<td>metadata</td>
<td></td>
</tr>
<tr>
<td>Query</td>
<td>AWS managed policy: AWSCleanRoomsFullAccess (p. 148)</td>
</tr>
<tr>
<td>Receive results</td>
<td>AWS managed policy: AWSCleanRoomsFullAccessNoQuerying (p. 152)</td>
</tr>
<tr>
<td>Query and receive results</td>
<td>AWS managed policy: AWSCleanRoomsFullAccess (p. 148)</td>
</tr>
</tbody>
</table>

b. Choose Next: Tags, add tags (optional), and then choose Next: Review.
c. For Review policy, enter a Name and Description, and review the Summary.
d. Choose Create policy.
   You have created a policy for a collaboration member.
e. Go back to your original tab and under Add permissions, enter the name of the policy you just created. (You might need to reload the page.)
f. Select the check box next to the name of the policy you created, and then choose Next.
7. For Name, review, and create, enter the Role name and Description.
   a. Review Select trusted entities, enter the AWS account for the person or persons who will assume the role (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.

Create a service role to read data

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

To create a service role to read data
1. Sign in to the IAM console (https://console.aws.amazon.com/iam/) with your administrator account.
2. Under Access management, choose Roles.
   Roles enables you to create short-term credentials, which is recommended for increased security. You can also choose Users to create long-term credentials.
3. Choose Create role.
4. In the Create role wizard, for Trusted entity type, choose Custom trust policy.
5. Copy and paste the following custom trust policy into the JSON editor.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "RoleTrustPolicyForCleanRoomsService",
         "Effect": "Allow",
         "Principal": {
            "Service": "cleanrooms.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
      }
   ]
}
```
6. Choose Next.
7. For Add permissions, choose Create policy.
   A new tab appears.
   a. 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. Your AWS Glue resources and underlying Amazon S3 resources must be in the same AWS Region as the AWS Clean Rooms collaboration.
   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](p. 158).
Create a service role to read data

```json
{
    "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: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"
                    ]
                }
            }
        }
    ]
}
```
Create a service role to write results

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

AWS Clean Rooms uses a service role to write results of the queried data in the collaboration to the specified Amazon S3 bucket. You can create this role using the console if you have the necessary IAM permissions. If you do not have CreateRole permissions, ask your administrator to create the service role.

To create a service role to write results

1. Sign in to the IAM console (https://console.aws.amazon.com/iam/) with your administrator account.
2. Under Access management, choose Roles.
   Roles enables you to create short-term credentials, which is recommended for increased security.
   You can also choose Users to create long-term credentials.
3. Choose Create role.
4. In the Create role wizard, for Trusted entity type, choose Custom trust policy.
5. Copy and paste the following custom trust policy into the JSON editor.

Example custom trust policy

In the following example, 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-EXAMPLE11111` – 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.

```json
{
    "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-EXAMPLE11111"
                ]
            }
        }
    ]
}
```

6. Choose **Next**.

7. For **Add permissions**, choose **Create policy**.

A new tab appears.

a. In the **Policy editor**, choose the **JSON** button, 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. Your AWS Glue resources and underlying Amazon S3 resources must be in the same **AWS Region** as the AWS Clean Rooms collaboration.
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 (p. 158).

Example service role policy

In the following example, replace each placeholder with your own information:

- **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.
- **111122223333** – 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.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket_name"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket_name/optional_key_prefix/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

b. Choose **Next**.

c. For **Review and create**,  
   i. For **Policy details**, enter a **Name** and **Description**.  
   ii. Review the **Permissions defined in this policy**.

d. Add tags (optional).

e. Choose **Create policy**.

You have created a policy for AWS Clean Rooms.
8. Go back to your original tab and under Add permissions, enter the name of the policy you just created. (You might need to reload the page.)
9. Select the check box next to the name of the policy you created, and then choose Next.
10. 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.
11. 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 (p. 177) and another member who can receive results (p. 177).

In most cases, the member who can query is also the member paying for query compute costs (p. 177). 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 (p. 67)
- Next steps (p. 71)

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](#) 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,

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query the data in the collaboration and receive the results</td>
<td>1. Choose yourself as the member who can <strong>Run queries</strong>.</td>
</tr>
<tr>
<td></td>
<td>2. Leave the default setting of the member who can <strong>Receive results</strong> is the <strong>Same as who runs queries</strong>.</td>
</tr>
<tr>
<td>Query the data in the collaboration and assign a different member to receive results</td>
<td>1. Choose yourself as the member who can <strong>Run queries</strong>.</td>
</tr>
<tr>
<td></td>
<td>2. Select the member who can <strong>Receive results</strong> from the dropdown list.</td>
</tr>
<tr>
<td>Receive the results of the query in the collaboration and assign a different member to query the data</td>
<td>1. Select the member who can <strong>Run queries</strong> from the dropdown list.</td>
</tr>
<tr>
<td></td>
<td>2. Choose yourself as member who can <strong>Receive results</strong> from the dropdown list.</td>
</tr>
<tr>
<td>Create and manage the collaboration, assign a different member to query the data, and assign a different member to receive results</td>
<td>1. Select the member who can <strong>Run queries</strong> from the dropdown list.</td>
</tr>
<tr>
<td></td>
<td>2. Select the member who can <strong>Receive results</strong> from the dropdown list.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign the member who can <strong>Run queries</strong> to be the member who pays for the query compute costs</td>
<td>Leave the default setting of the member who will <strong>Pay for queries</strong> is the <strong>Same as who runs queries</strong>.</td>
</tr>
<tr>
<td>Assign a different member to pay for the query compute costs</td>
<td>Select the member who will <strong>Pay for queries</strong> from the dropdown list.</td>
</tr>
</tbody>
</table>

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* (p. 35).

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* (p. 81).

**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:

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yes, join by creating membership now</strong></td>
<td>Both the collaboration and your membership are created.</td>
</tr>
<tr>
<td></td>
<td>Your status in the collaboration is active.</td>
</tr>
<tr>
<td><strong>No, I will create a membership later</strong></td>
<td>Only the collaboration is created.</td>
</tr>
<tr>
<td></td>
<td>Your status in the collaboration is inactive.</td>
</tr>
</tbody>
</table>
b. If you are the member who can **Receive results**, under **Query results settings defaults**, choose one option:

<table>
<thead>
<tr>
<th>If you ...</th>
<th>Then ...</th>
</tr>
</thead>
</table>
| 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. |

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

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
</table>
| **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 (p. 177)** but not the **member who can query (p. 177)**, 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 (p. 127). 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:

<table>
<thead>
<tr>
<th>If you have chosen to...</th>
<th>Then choose...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a membership with the collaboration</td>
<td>Create collaboration and membership</td>
</tr>
<tr>
<td>(Yes, join by creating membership now)</td>
<td></td>
</tr>
<tr>
<td>Create the collaboration, and not to create a membership at</td>
<td>Create collaboration</td>
</tr>
<tr>
<td>this time (No, I will create a membership later)</td>
<td></td>
</tr>
</tbody>
</table>

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 (p. 75). (Optional if you want to query your own data.)
- Associate the configured table to your collaboration (p. 107). (Optional if you want to query your own data.)
- Configure an analysis rule for the configured table (p. 101). (Optional if you want to query your own data.)
- Create a membership and join a collaboration (p. 72).
- Manage your collaboration (p. 123).
Creating a membership and joining a collaboration in AWS Clean Rooms

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 (p. 177) data, member who can receive results (p. 177) of a query, or both. You can also join a collaboration as a member paying for query compute costs (p. 177). 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 (p. 72)
- Next steps (p. 74)

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 into 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:

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn on</td>
<td>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.</td>
</tr>
</tbody>
</table>
Create a membership and join a collaboration

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Turn off</strong></td>
<td>The query logs relevant to you aren’t stored in your Amazon CloudWatch Logs account.</td>
</tr>
</tbody>
</table>

**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**.

      For more information, see Create a service role to write results (p. 63).

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 (p. 3).

If you are the **member paying for query compute costs** (p. 177) but not the **member who can query** (p. 177), 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 (p. 127). 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.
Next steps

For information about how to leave a collaboration, see Leaving a collaboration (p. 127).

You are now ready to:

- Prepare your data table to be queried in AWS Clean Rooms (p. 75). (Optional if you want to query your own data.)
- Associate the configured table to your collaboration (p. 107).
- Configure an analysis rule for the configured table (p. 101).
Preparing data tables for queries in AWS Clean Rooms

As a member in the collaboration, you must prepare your data tables so that the member who can query can query them in AWS Clean Rooms. Preparing data tables can take place before or after you have joined a collaboration.

If you're the collaboration creator and you don't want to query 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 (p. 97).

Preparing your data tables involves the following steps:

• Step 1: Complete the prerequisites (p. 75)
• Step 2: (Optional) Prepare your data for cryptographic computing (p. 75)
• Step 3: Upload your data table to Amazon S3 (p. 76)
• Step 4: Create an AWS Glue table (p. 76)
• Next steps (p. 98)

For more information about the data formats that you can use for queries, see Data formats for AWS Clean Rooms (p. 77).

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 (p. 77).
• Your datasets must be cataloged in AWS Glue and use the supported data types for AWS Clean Rooms (p. 78).
• All of your datasets 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 (p. 67).
• 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 (p. 81).

### 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 (p. 81).

**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 about supported AWS Glue Data Catalog data types, see Supported data types (p. 78).

**Note**
AWS Clean Rooms does not 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. Sign in to the AWS Management Console and open the AWS Glue console at https://console.aws.amazon.com/glue/.
2. From the navigation bar, select Crawlers.
3. Select your S3 bucket from the list, and then choose Add crawler.
4. On the Add crawler page, enter a Crawler name and then choose Next.
5. Continue through the Add crawler page, specifying the details.
6. On the Choose an IAM role page, choose Choose an existing IAM role and then choose Next.
   
   You can also choose Create an IAM role or have your administrator create the IAM role if needed.
7. For Create a schedule for this crawler, keep the Frequency default (Run on demand) and then choose Next.
8. For **Configure the crawler's output**, enter the AWS Glue database and then choose **Next**.
9. Review all of the details, and then choose **Finish**.
10. On the **Crawlers** page, select the check box next to your S3 bucket and then choose **Run crawler**.
11. After the crawler is finished running, on the AWS Glue navigation bar, choose **Databases**, and then choose your database name.
12. On the **Database** page, choose **Tables in {your database name}**.
   a. View the tables in the AWS Glue database.
   b. To view a table's schema, select a specific table.
13. Make a note of the AWS Glue database name and AWS Glue table name.

**Next steps**

Now that you have prepared your datasets, you are ready to:

- Create a configured table (p. 97).

**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** (p. 79)
- 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.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>File extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>GZIP</td>
<td>.gz</td>
</tr>
<tr>
<td>Bzip2</td>
<td>.bz2</td>
</tr>
<tr>
<td>Snappy</td>
<td>.snappy</td>
</tr>
</tbody>
</table>

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.


Using Apache Iceberg tables in AWS Clean Rooms (preview)

This is prerelease documentation for support of Apache Iceberg tables, which is in preview release. The documentation and the feature are both subject to change. We recommend that you use this feature only in test environments, and not in production environments. For preview terms and conditions, see Betas and Previews in the *AWS Service Terms*.

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 (p. 27).

Preparing encrypted data tables with C3R involves the following steps:

Steps
- Step 1: Complete the prerequisites (p. 81)
- Step 2: Download the C3R encryption client (p. 82)
- (Optional) Step 3: View available commands in the C3R encryption client (p. 82)
- Step 4: Generate an encryption schema for a tabular file. (p. 82)
- Step 5: Create a shared secret key (p. 87)
- Step 6: Store the shared secret key in an environment variable (p. 87)
- Step 7: Encrypt data (p. 88)
- Step 8: Verify data encryption (p. 89)
- (Optional) Create a schema (advanced users) (p. 90)

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 (p. 87).
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](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](https://github.com/aws/c3r)) 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](https://github.com/aws/c3r).

(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)](https://github.com/aws/c3r) (p. 90).
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 (p. 35).

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

<table>
<thead>
<tr>
<th>Decision</th>
<th>Number of target columns from source column &lt;'name-of-column'&gt;?</th>
<th>Target column type: [c] cleartext, [f] fingerprint, or [s] sealed?</th>
<th>Target column headername &lt;default 'name-of-column'&gt;</th>
<th>Add suffix &lt;suffix&gt; to header to indicate how it was encrypted, [y] yes or [n] no &lt;default 'yes'&gt;</th>
<th>&lt;='name-of-column_sealed'&gt; padding type: [n] one, [f] fixed, or [m] max &lt;default 'max'&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leave the column unencrypted.</td>
<td>1</td>
<td>c</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Encrypt the column as a fingerprint column.</td>
<td>1</td>
<td>f</td>
<td>Choose default or enter a new header name.</td>
<td>Enter y to choose default (_fingerprint) or enter n.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Encrypt the column as a sealed column.</td>
<td>1</td>
<td>s</td>
<td>Choose default or enter a new header name.</td>
<td>Enter y to choose default (_sealed) or enter n.</td>
<td>Choose padding type. For more information,</td>
</tr>
</tbody>
</table>
### 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)* (p. 90).
- 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](p. 88).

---

**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) (p. 90).

   • 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.
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 (p. 87)
- Example: Key generation on Windows using PowerShell (p. 87)

**Example: Key generation using OpenSSL**

For a common general purpose cryptography library, run the following command to create a shared secret key.

```bash
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 (p. 87).

**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.

```powershell
$bs = New-Object Byte[](32);
[Security.Cryptography.RandomNumberGenerator]::Create().GetBytes($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 (p. 87) 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 (p. 81).

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 (p. 87).
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 (p. 35) and Optional flags in Cryptographic Computing for Clean Rooms (p. 38).
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** (p. 76).
2. **Create an AWS Glue table** (p. 76).
3. **Create a configured table in AWS Clean Rooms** (p. 97).

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 issue 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 (p. 85) 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.

```json
{
}
```
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",  
        "type": "fingerprint"  
      },  
      {  
        "sourceHeader": "City",  
        "targetHeader": "City",  
        "type": "sealed"  
      }  
    ]  
}
```
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, 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 |

As a more complex example, the following is an example .csv file with headers.

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.

```json
{
  "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": "fixed",
        "length": 20
      }
    }
  ]
}
```
'"type": "none"
}
]
}
}

The following is the .csv file that results from the mapped schema.

givenname,surname,state_fingerprint,state
John,Doe,01:hmac:UK8s8Cn/WR23O/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:FQ3n3Ahv9BQQNWQ6cugeHHzHyZE1vaphHa2Uu4SRgS4tZ3qObjPA4TsHT+B0kMK8cnHWI13BeGG/SBqmnj7vKpi=
Paulo,Santos,01:enc:CHF4eItjTNqAooU9v4h90jc+txBnMidQ7jdjWuaDTTA=,01:enc:KZ5n5GtAAXcc065AKk4BBBO2duruN2ULC4YxMcB8N2ZKKJiksU1lWdFAdAvv41BQ1busSTU5d
Mateo,Jackson,01:hmac:iIRnjfNItzyusUJ1w35lgNzeY1RQ1b5f6PDH58Xrkb=,01:enc:mLkp5SHIO5gphdEsrzhEdiP/eneNnBO2gAbiygty0Ftn4Laly9n9xyj/XWx1mnn82fFe2T4kyDTDbkgG0vQE6uX4Ufk=
Diego,Ramirez,01:hmac:UK8s8Cn/WR23O/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:izmZTH98Zm+II6w1ujj51jP41rW/AA1tBLMchvNyFrgnWP623VFQ6aUnhsb2MDqEw4G5Wg5rK2zepUxx5ukBFk=
Jorge,Souza,01:hmac:IK8s8Cn/WR23O/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:5aqwC1VRb5k5b0gnuR7q0zxVPcvEj
Terry,Whitlock,01:hmac:UK8s8Cn/WR23O/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:3rWEb0O0/xbQj6GuclL7ZtZ8sP4+Syr1y3kudFAxubMQ2uRdU/q7rbyy3jx2S8M2u3s13j/1Dg7y7cM=
Jane,Doe,01:hmac:UK8s8Cn/WR23O/To2dTxWD73aDEe2ZUXeSHy3Tv+1Mk=,01:enc:9RWv46YLveykeNZ/G6Nd1lYFg+AVd0nu05hHAYTqKPLHnyX+0/jbzd/g9ZT86CG99aB5bV4ooJIXHGBVMXcjrQ=

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.

```json
{
  "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.

```json
{
  "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",
```
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,John'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.

```json
{
    "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:iIRnjfNzryusJ1lw35lgNzeY1RQ1bSfq6PDHW8Xrkb=,01:enc:ENS6QD3cMV19vQEGfe9MNwFR0UUpQ8m/Y5SA89dJ3wkJpT5rGPe8e36h6k1wDo8lFzGvU0=
Jorge,Souza,01:hmac:3Bx3dxiFFyZ8HBBbYNNqehB5vqhM0d7s2ZiKuE7qIy08=,01:enc:Lko0zirz2+XtIIIMRJAsGMDwudgingl7ystc+1FPi+iUf1BNeZDJjteF+1Z+zb2FQwWjyS2Rr7HqvAb2WIk1omnk=
Pualo,Santos,01:hmac:CHF4eIrtTNgAooU9v4h90jc+txBnMidQTdjjWuaD774=,01:enc:MyQKyWw39kvKixLQWtv3F+yrbRzoxjuU/Y1B6g5KFQOn9pK+M77g+ZNq2EPcPz41ht1u0t/wbTagzDCLXQF=
Jane,Doee,01:hmac:UKB88Cn1WR2J0/To2dTwD73aDDee2ZXEXeS1h3T+1Mk=,01:enc:Pd8sbITBF08/ttUB4svVsgoyKdfnDvgkxvzecI0Yxq54lsL5wcyj03/B50C3cpkkkn56dovCwzgmmPNwmmCmtb4=
Terry,Whitlock01:hmac:UKB88Cn1WR2J0/To2dTwD73aDDee2ZXEXeS1h3T+1Mk=,01:enc:Qmtzu3BGAKHk2khRYTIEAaMopYedsSdF2e/
ADUib9Qkv2CxpKzWYTD3zttmKPMka19dHer5VnUHnpO30+j1AQB=
Diego,Ramirez,01:hmac:UKB88Cn1WR2J0/To2dTwD73aDDee2ZXEXeS1h3T+1Mk=,01:enc:ysdg+GHKdeZrS/gbeBl0oEPLH6g8MsWpx1dh3xjbf+gS5rMfMqUcJLnuuYBHhHAlcchM2Wv1fMhkBX3mvZNvkc=
John,Doee,01:hmac:UKB88Cn1WR2J0/To2dTwD73aDDee2ZXEXeS1h3T+1Mk=,01:enc:9uX0wZuO7kAPAw/HF6vujQwnkWqPSkT5SwgQI3Se5aXFquKWCK6yZNOX5Ea2N3bn03Uj1kh0agDw0iP9FRZ3J3A4=
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*.

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*.

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All columns</strong></td>
<td>All columns are allowed for use in AWS Clean Rooms (subject to analysis rules).</td>
</tr>
<tr>
<td><strong>Custom list</strong></td>
<td>Choose one or more columns that you want to allow from the <em>Specify allowed columns</em> dropdown list.</td>
</tr>
</tbody>
</table>

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 (p. 101)
- Associate the configured table to a collaboration (p. 107)
Creating an analysis template in AWS Clean Rooms

Analysis templates work with the Custom analysis rule in AWS Clean Rooms (p. 24). 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.

For information about how to create an analysis template using the AWS SDKs, see the AWS Clean Rooms API Reference.

Topics
- Create an analysis template (p. 99)
- Next steps (p. 100)

Create an analysis template

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.
Next steps

You are now ready to:

- Inform your collaboration member that they can Review an analysis template (p. 111). (Optional if you want to query your own data.)
- Configure an analysis rule to the configured table (p. 101)
Configuring an analysis rule to a configured table

The following sections describe how to configure an analysis rule to your configured table. Defining the analysis rules enables you to 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 (p. 175), list (p. 177), and custom (p. 176).

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 (p. 5).

For more information about the aggregation analysis rule, see Aggregation analysis rule (p. 7).

For more information about the list analysis rule, see List analysis rule (p. 18).

For more information about the custom analysis rule, see Custom analysis rule in AWS Clean Rooms (p. 24).

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) (p. 101)
• Configuring a list analysis rule to a table (guided flow) (p. 103)
• Configuring a custom analysis rule to a table (guided flow) (p. 104)
• Configuring analysis rule to a table (JSON editor) (p. 105)
• Next steps (p. 106)

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/or 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 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?

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>All currently supported by AWS Clean Rooms</td>
<td>You allow all scalar functions currently supported by AWS Clean Rooms.</td>
</tr>
<tr>
<td></td>
<td>• You can choose View list to see the entire list of Scalar functions supported in AWS Clean Rooms.</td>
</tr>
<tr>
<td>A custom list</td>
<td>You can customize which scalar functions to allow.</td>
</tr>
<tr>
<td></td>
<td>• Choose one or more options from the Specify allowed scalar functions dropdown.</td>
</tr>
<tr>
<td>None</td>
<td>You don’t want to allow any scalar functions.</td>
</tr>
</tbody>
</table>

For more information, see Scalar functions (p. 13).

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’ll 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.

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>You can include AND in the INNER JOIN match conditions to join one column to another column between tables.</td>
</tr>
<tr>
<td>OR</td>
<td>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.</td>
</tr>
</tbody>
</table>

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'll 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 (p. 99).

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: Specify query controls**,
   a. For **Control type**:

<table>
<thead>
<tr>
<th>If you want to …</th>
<th>Then choose …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review each new analysis template before it’s run on your configured table</td>
<td>Review each new analysis before it is allowed to be run on this table</td>
</tr>
<tr>
<td>Let any analysis template or direct query be performed on your configured table</td>
<td>Allow any queries created by specific collaborators to run without review on this table</td>
</tr>
</tbody>
</table>

b. For **Analysis templates allowed to be run**:

<table>
<thead>
<tr>
<th>If you've chosen …</th>
<th>Then …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review each new analysis before it is allowed to be run on this table</td>
<td>Under <strong>Analysis templates allowed to be run</strong>, choose Add analysis template, and then choose the appropriate Collaboration and the Analysis template from the dropdown lists.</td>
</tr>
<tr>
<td>Allow any queries created by specific collaborators to run without review on this table</td>
<td>Under <strong>AWS accounts allowed to create any query</strong>, choose Add AWS account, and then choose the appropriate AWS account ID.</td>
</tr>
</tbody>
</table>

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’ll 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** 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](p. 10).

For more information about list analysis rule parameters, see [List analysis rule - query controls](p. 20).

**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.

### Next steps

Now that you configured an analysis rule to your configured table, you are ready to:

- [Associate a configured table to a collaboration](p. 107)  
- [Query the data tables](p. 112)  (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**
- Associate a configured table from the configured table detail page (p. 107)
- Associate a configured table from the collaboration detail page (p. 109)
- Next steps (p. 110)

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 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:
### AWS Clean Rooms User Guide

#### Associate a configured table from the configured table detail page

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure a new table</td>
<td>Choose <strong>Configure table</strong> and follow the prompts on the <strong>Configure table</strong> page.</td>
</tr>
<tr>
<td>View the schema and analysis rule for the configured table</td>
<td>Turn on <strong>View schema and analysis rule</strong>.</td>
</tr>
</tbody>
</table>

8. Specify the **Service access** permissions by selecting either **Create and use a new service role** or **Use an existing service role**.

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
</table>
| **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. |
| **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 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 AWS IAM** external link.  
If there are no existing service roles, the option to **Use an existing service role** is unavailable.  
3. By default, AWS Clean Rooms does not attempt to update the existing role policy to add necessary permissions.  
(Optional) Select the **Add a pre-configured policy with necessary permissions 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](p. 147).
- 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.
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:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose an existing configured table</td>
<td>Choose the Configured table name that you want to associate with the collaboration from the dropdown list.</td>
</tr>
<tr>
<td>Configure a new table</td>
<td>Choose Configure table and follow the prompts on the Configure table page.</td>
</tr>
<tr>
<td>View the schema and analysis rule for the configured table</td>
<td>Turn on View schema and analysis rule.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>If you choose...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create and use a new service role</td>
<td>• AWS Clean Rooms creates a service role with the required policy for this table.</td>
</tr>
<tr>
<td></td>
<td>• The default Service role name is cleanrooms-&lt;timestamp&gt;.</td>
</tr>
<tr>
<td></td>
<td>• You must have permissions to create roles and attach policies.</td>
</tr>
<tr>
<td>Use an existing service role</td>
<td>1. Choose an Existing service role name from the dropdown list.</td>
</tr>
</tbody>
</table>
### Next steps

**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](p. 147).
- 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** (p. 123), if you're the collaboration creator
- **Query the data tables** (p. 112) (as a member who can query)
Reviewing an analysis template in AWS Clean Rooms

After a collaboration member has created an analysis template, you can review and approve it. After the analysis template is reviewed and approved, it can be used in a query in AWS Clean Rooms.

Topics
- Review an analysis template (p. 111)
- Next steps (p. 100)

Review an analysis template

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.

<table>
<thead>
<tr>
<th>If you ...</th>
<th>Then choose ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approve the analysis template</td>
<td>Allow template on table. Confirm your approval by choosing Allow.</td>
</tr>
<tr>
<td>Don't approve the analysis template</td>
<td>Disallow</td>
</tr>
</tbody>
</table>

Next steps

You are now ready to:
- Query the data tables (p. 112) (as a member who can query).
Querying data in a collaboration

As the member who can query (p. 177), 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 (p. 175).

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.

Note
When using Cryptographic Computing for Clean Rooms (p. 27), 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
- Using the SQL code editor (p. 112)
- Using the analysis builder (p. 114)
- Using an analysis template (p. 117)
- Viewing recent queries (p. 118)
- Viewing query details (p. 118)

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 (p. 57).

Note
If you run a query on encrypted (p. 177) data tables, the results from the encrypted columns are encrypted.

For information about receiving query results, see Receiving query results (p. 120).

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 (p. 114).

**Important**
If you start writing a SQL query in the code editor and then turn on the Analysis builder UI, your query won’t be 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 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 will only display if the member who can receive results and the member who is responsible to pay for query compute costs has 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 (p. 107).
   - The tables don't have an analysis rule configured (p. 101).

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.

<table>
<thead>
<tr>
<th>(Optional) If you want to use an example query</th>
<th>(Optional) If you want to insert column names or functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select the three vertical dots next to the table.</td>
<td>1. Select the three vertical dots next to a column.</td>
</tr>
<tr>
<td>2. Under <strong>Insert in editor</strong>, choose <strong>Example query</strong>.</td>
<td>2. Under <strong>Insert in editor</strong>, choose <strong>Column name</strong>.</td>
</tr>
<tr>
<td><strong>Note</strong> Inserting an Example query appends the query already in the editor.</td>
<td>3. To manually insert a function that is permitted on a column, select <strong>Insert in editor</strong>, and then select the name of the permitted function (such as INNER JOIN, SUM, SUM DISTINCT, or COUNT).</td>
</tr>
<tr>
<td>The query example appears. All of the tables listed under <strong>Tables</strong> are included in the query.</td>
<td>4. Press <strong>Ctrl + Space</strong> to view the table schemas in the code editor.</td>
</tr>
<tr>
<td>3. Edit the placeholder values in the query.</td>
<td><strong>Note</strong> Members who can query can view and use the partition columns in</td>
</tr>
</tbody>
</table>
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 (p. 7) with no JOIN required
- Two tables (one from each member) that both use the aggregation analysis rule (p. 175)
- Two tables (one from each member) that both use the list analysis rule (p. 177)
- 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 (p. 112).

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 won’t be 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) (p. 115)
- Use the analysis builder to query two tables (aggregation or list) (p. 116)
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 (p. 7) 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 (p. 107).
• The table doesn't have an analysis rule configured (p. 101).

5. Under the Analysis section, turn on Analysis builder UI.

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.


Note
You won’t be able to 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 (p. 131).

**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 (p. 7)
- Two tables (one from each member) that both use the list analysis rule (p. 18)
- 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 (p. 107).
   - The tables don’t have an analysis rule configured (p. 101).
5. Under the Analysis section, turn on Analysis builder UI.

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.

<table>
<thead>
<tr>
<th>If the two tables use the aggregation analysis rule</th>
<th>If the two tables use the list analysis rule</th>
</tr>
</thead>
</table>
| 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. | 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. |
If the two tables use the aggregation analysis rule

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>When using the analysis builder, you can match only on a single pair of columns.</td>
</tr>
</tbody>
</table>

3. (Optional) For **Add segments – optional**, choose one or more parameters.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add segments – optional</strong> is only displayed if dimensions are specified for the table.</td>
</tr>
</tbody>
</table>

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**.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
</table>
| **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.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
</table>
| **LIMIT** is not supported for list queries.  
| Only the AND operator is supported in filters. |

If the two tables use the list analysis rule

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>When using the analysis builder, you can match only on a single pair of columns.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
</table>
| **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.

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.

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**.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>You won’t be able to run the query if the member who can receive results hasn’t configured the query results settings</td>
</tr>
</tbody>
</table>

9. Continue to adjust parameters and run your query again, or choose the + button to start a new query in a new tab.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 (p. 131).</td>
</tr>
</tbody>
</table>

Using an analysis template
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 (p. 107).
   • The tables don’t have an analysis rule configured (p. 101).
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 are not supported.
7. Choose Run.
   Note
   You won’t be able to 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.

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 (p. 177), 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 will 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 (p. 177), 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

- Receive query results (p. 120)
- Edit default values for query results settings (p. 121)
- Using query output in other AWS services (p. 121)

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 (p. 57).

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:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then choose...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy the results.</td>
<td>Copy</td>
</tr>
<tr>
<td>Download the results.</td>
<td>Download</td>
</tr>
<tr>
<td>Note</td>
<td>By default, the downloaded file's name is the corresponding Query id that was displayed when the query was run in AWS Clean Rooms.</td>
</tr>
<tr>
<td>View the results in Amazon S3.</td>
<td>View in Amazon S3</td>
</tr>
<tr>
<td>The Amazon S3 console opens in a separate tab.</td>
<td></td>
</tr>
</tbody>
</table>
6. If you're using encrypted data, you can now decrypt (p. 176) the data tables.

   For more information, see Decrypting data tables with the C3R encryption client (p. 122).

## 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.
Decrypteding 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 (p. 112).

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 (p. 82).
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 (p. 87).
4. From the AWS Command Line Interface (AWS CLI), run the following command.
   
   ```java
   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 (p. 123)
• Managing configured tables in AWS Clean Rooms (p. 129)

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 (p. 123)
• Deleting collaborations (p. 126)
• Viewing collaborations (p. 126)
• Viewing tables and analysis rules (p. 126)
• Monitoring member status (p. 127)
• Removing a member from a collaboration (p. 127)
• Leaving a collaboration (p. 127)
• Editing configured table associations (p. 128)
• Disassociating configured tables (p. 128)

Editing collaborations

Topics

• Edit collaboration name and description (p. 123)
• Edit collaboration tags (p. 124)
• Edit membership tags (p. 124)
• Edit associated table tags (p. 125)
• Edit analysis template tags (p. 125)

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](https://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. 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://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:

<table>
<thead>
<tr>
<th>If you are...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>A member of the collaboration</td>
<td>Choose the <strong>Details</strong> tab.</td>
</tr>
<tr>
<td>The collaboration creator but not a member of the collaboration</td>
<td>Scroll down the page to the <strong>Tags</strong> section.</td>
</tr>
</tbody>
</table>

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://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 associated 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 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 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.
Deleting collaborations

As a collaboration creator, you can delete a collaboration that you created.

**Note**
When you delete a collaboration, you and all members won't be able to run queries, receive results, or contribute data. Each collaboration member will continue 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://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://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](https://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.
4. Choose the **Tables** tab.
5. Choose one of the following:
   - 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.

## 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://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 will also remove 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://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** ([p. 177](#)), no more queries will be 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** ([p. 177](#)) and you delete your membership, no more queries will be 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 that you are a member of.
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 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 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.

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 (p. 129)
- Editing configured table tags (p. 129)
- Editing configured table analysis rule (p. 130)
- Deleting configured table analysis rule (p. 130)

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.
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 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. (p. 131)
• One of the underlying datasets has an unsupported file format. (p. 131)
• Query results are not as expected when using Cryptographic Computing for Clean Rooms. (p. 131)

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 (p. 59).

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 (p. 77).

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:
Query results are not as expected when using Cryptographic Computing for Clean Rooms.

- 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 (p. 27) and the section called “Column types” (p. 33).
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](https://aws.amazon.com/compliance/).

- **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 (p. 133)
- Best practices for data collaborations in AWS Clean Rooms (p. 134)
- Identity and Access Management for AWS Clean Rooms (p. 136)
- Compliance validation for AWS Clean Rooms (p. 159)
- Resilience in AWS Clean Rooms (p. 159)
- Infrastructure security in AWS Clean Rooms (p. 160)
- Access AWS Clean Rooms using an interface endpoint (AWS PrivateLink) (p. 160)

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. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](https://aws.amazon.com/privacy/). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](https://aws.amazon.com/gdpr/) blog post on the [AWS Security Blog](https://aws.amazon.com/security/).

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.
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.

**Note**
You can use the encryption options in Amazon S3 to protect your data at rest. For more information, see [Specifying Amazon S3 encryption](https://docs.aws.amazon.com/AmazonS3/latest/userguide/specifying-encryption.html) 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.

Encrypting underlying data

For more information about how to encrypt your underlying data, see [Cryptographic Computing for Clean Rooms](https://docs.aws.amazon.com/AmazonS3/latest/userguide/cryptographic-computing.html) (p. 27).

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](https://aws.amazon.com/sharedResponsibility/). AWS Clean Rooms offers [analysis rules](https://docs.aws.amazon.com/clean-rooms/latest/userguide/clean-rooms-analysis-rules.html) (p. 5) 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.
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
- Audience (p. 136)
- Authenticating with identities (p. 137)
- Managing access using policies (p. 139)
- How AWS Clean Rooms works with IAM (p. 140)
- Identity-based policy examples for AWS Clean Rooms (p. 145)
- AWS managed policies for AWS Clean Rooms (p. 147)
- Troubleshooting AWS Clean Rooms identity and access (p. 156)
- Cross-service confused deputy prevention (p. 158)

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 (p. 156).

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 (p. 140).

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 (p. 145).

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 How IAM roles differ from resource-based policies 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.

- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for AWS Clean Rooms in the Service Authorization Reference.

- **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](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles实例.html) 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)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_instances.html) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_policies_overview.html) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_policies_create.html) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_policies_create-managed.html) 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

<table>
<thead>
<tr>
<th>IAM feature</th>
<th>AWS Clean Rooms support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity-based policies (p. 141)</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource-based policies (p. 141)</td>
<td>No</td>
</tr>
<tr>
<td>Policy actions (p. 142)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy resources (p. 142)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy condition keys (service-specific) (p. 143)</td>
<td>No</td>
</tr>
</tbody>
</table>
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

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 (p. 145).

Resource-based policies within AWS Clean Rooms

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 How IAM roles differ from resource-based policies in the IAM User Guide.

Policy actions for AWS Clean Rooms

<table>
<thead>
<tr>
<th>Supports policy actions</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 (p. 145).

Policy resources for AWS Clean Rooms

<table>
<thead>
<tr>
<th>Supports policy resources</th>
<th>Yes</th>
</tr>
</thead>
</table>

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": 

142
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 (p. 145).

**Policy condition keys for AWS Clean Rooms**

| Supports service-specific policy condition keys | No |

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 view examples of AWS Clean Rooms identity-based policies, see Identity-based policy examples for AWS Clean Rooms (p. 145).

**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) | No |

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

<table>
<thead>
<tr>
<th>Supports temporary credentials</th>
<th>Yes</th>
</tr>
</thead>
</table>

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.

### Cross-service principal permissions for AWS Clean Rooms

<table>
<thead>
<tr>
<th>Supports principal permissions</th>
<th>Yes</th>
</tr>
</thead>
</table>

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for AWS Clean Rooms in the Service Authorization Reference.

### Service roles for AWS Clean Rooms

<table>
<thead>
<tr>
<th>Supports service roles</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 (p. 145)
- Using the AWS Clean Rooms console (p. 146)
- Allow users to view their own permissions (p. 146)

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
Identity-based policy examples 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.

```json
{
   "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",
```

---

146
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. AWS Clean Rooms also attaches this policy to a service role that allows AWS Clean Rooms to perform actions on your behalf.

This policy grants read-only permissions that allow read-only access 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*"
```
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.
• ConsolePickQueryResultsBucket – 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.

```json
{
    "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": "*"
        }
    ]
}
```
{ "Sid": "ConsolePickQueryResultsBucketListAll", "Effect": "Allow", "Action": [ "s3:ListAllMyBuckets" ], "Resource": "*" },
{ "Sid": "WriteQueryResults", "Effect": "Allow", "Action": [ "s3:ListBucket", "s3:PutObject" ], "Resource": "arn:aws:s3:::cleanrooms-queryresults*", "Condition": { "ForAnyValue:StringEquals": { "aws:CalledVia": "cleanrooms.amazonaws.com" } } }
AWS managed policies

```
{
  "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"
    }
  }
}
```
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.

- **ConsolePickQueryResultsBucketListAll** – Allows principals to choose an S3 bucket from a list of all available S3 buckets into which their query results are written.

- **ConsolePickQueryResultsBucket** – 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.

- **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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CleanRoomsAccess",
      "Effect": "Allow",
      "Action": [
        "cleanrooms:BatchGetCollaborationAnalysisTemplate",
        "cleanrooms:BatchGetSchema",
        "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": "*"
    }
  ],
  "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:StartQueryLogSummary",
"logs:GetQueryLogSummary",
"logs:StopQueryLogSummary"
],
"Resource": "*",
"Condition": {
"ForAnyValue:StringEquals": {
 "aws:CalledVia": "cleanrooms.amazonaws.com"
}
}
}
AWS Clean Rooms User Guide
Troubleshooting

"Action": [
  "logs:StartQuery"
],
"Resource": "arn:aws:logs:*:*:log-group:/aws/cleanrooms*"
},
{
  "Sid": "ConsoleLogSummaryObtainLogs",
  "Effect": "Allow",
  "Action": [
    "logs:GetQueryResults"
  ],
  "Resource": "*"
}
]

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.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSCleanRoomsFullAccessNoQuerying</td>
<td>- Update to existing policy 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.</td>
<td>July 31, 2023</td>
</tr>
<tr>
<td>AWS Clean Rooms started tracking changes</td>
<td>AWS Clean Rooms started tracking changes for its AWS managed policies.</td>
<td>January 12, 2023</td>
</tr>
</tbody>
</table>

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 (p. 157)
• I am not authorized to perform iam:PassRole (p. 157)
• I want to allow people outside of my AWS account to access my AWS Clean Rooms resources (p. 157)

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 (p. 140).
• 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.
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.

```json
{
  "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": {
        "StringLike": {
          "sts:ExternalId": "arn:aws::*:aws-region::dbuser::*:membershipID*"
        }
      }
    }
  ]
}
```
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.
- **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.
- **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 programatically 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 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. You can access AWS Clean Rooms 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 supports making calls to all of its API actions through the interface endpoint.

VPC endpoint policies are not supported for AWS Clean Rooms. By default, full access to AWS Clean Rooms 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 through the interface endpoint.

Create an interface endpoint for AWS Clean Rooms

You can create an interface endpoint for AWS Clean Rooms 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

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.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.

- **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)](p. 163)
- [StartProtectedQuery (failed)](p. 164)

**StartProtectedQuery (successful)**

```json
{
  "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",
}
```

163
Example AWS Clean Rooms CloudTrail events

```
"sourceIPAddress": "203.0.113.1",
"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" } }
```

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" } }
```

164
Example AWS Clean Rooms CloudTrail events

```
{
  "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-EXAMPLE111111",
    "type": "SQL"
  },
  "responseElements": {
    "Access-Control-Expose-Headers": "x-amzn-RequestId,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, 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

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 cannot 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 is not yet available in Service Quotas, use the Service limit increase form.

Your AWS account has the following quotas related to AWS Clean Rooms.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Members invited per collaboration</td>
<td>5</td>
<td>Maximum number of members invited per collaboration</td>
</tr>
<tr>
<td>Memberships per account</td>
<td>100</td>
<td>Maximum number of memberships for an account</td>
</tr>
<tr>
<td>Collaborations created per account</td>
<td>10</td>
<td>Maximum number of collaborations created per account</td>
</tr>
<tr>
<td>Configured tables per account</td>
<td>60</td>
<td>Maximum number of configured tables that can be created by an account</td>
</tr>
<tr>
<td>Table associations per membership</td>
<td>25</td>
<td>Maximum number of tables associated per active membership</td>
</tr>
<tr>
<td>Concurrent ongoing queries per membership</td>
<td>5</td>
<td>Maximum number of concurrent ongoing queries per membership</td>
</tr>
<tr>
<td>Columns per configured table allowlist</td>
<td>100</td>
<td>Maximum number of columns that can be allowlisted per configured table</td>
</tr>
<tr>
<td>Configured tables per protected query</td>
<td>15</td>
<td>Maximum number of configured tables in a protected query</td>
</tr>
<tr>
<td>Analysis templates per membership</td>
<td>25</td>
<td>Maximum number of analysis templates per membership</td>
</tr>
</tbody>
</table>

Resource parameter limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis rule size</td>
<td>100 KB</td>
<td>Maximum size of JSON for an analysis rule</td>
</tr>
<tr>
<td>Query text length</td>
<td>16 KB</td>
<td>Maximum text length for a SQL query statement</td>
</tr>
</tbody>
</table>
### Query run time

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query run time</td>
<td>12 hours</td>
<td>Maximum duration a query is run before timeout</td>
</tr>
</tbody>
</table>

### API throttling quotas

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API TPS per endpoint per account</td>
<td>5 TPS</td>
<td>Maximum requests per second per API endpoint per account</td>
</tr>
<tr>
<td>Rate of BatchGetSchema requests</td>
<td>5 TPS</td>
<td>Maximum number of BatchGetSchema API calls per second</td>
</tr>
<tr>
<td>Rate of CreateCollaboration requests</td>
<td>5 TPS</td>
<td>Maximum number of CreateCollaboration API calls per second</td>
</tr>
<tr>
<td>Rate of CreateConfiguredTable requests</td>
<td>5 TPS</td>
<td>Maximum number of CreateConfiguredTable calls per second</td>
</tr>
<tr>
<td>Rate of CreateConfiguredTableAnalysisRule requests</td>
<td>5 TPS</td>
<td>Maximum number of CreateConfiguredTableAnalysisRule calls per second</td>
</tr>
<tr>
<td>Rate of CreateConfiguredTableAssociation requests</td>
<td>5 TPS</td>
<td>Maximum number of CreateConfiguredTableAssociation calls per second</td>
</tr>
<tr>
<td>Rate of CreateMembership requests</td>
<td>5 TPS</td>
<td>Maximum number of CreateMembership calls per second</td>
</tr>
<tr>
<td>Rate of DeleteCollaboration requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteCollaboration calls per second</td>
</tr>
<tr>
<td>Rate of DeleteConfiguredTable requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteConfiguredTable calls per second</td>
</tr>
<tr>
<td>Rate of DeleteConfiguredTableAnalysisRule requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteConfiguredTableAnalysisRule calls per second</td>
</tr>
<tr>
<td>Rate of DeleteConfiguredTableAssociation requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteConfiguredTableAssociation calls per second</td>
</tr>
<tr>
<td>Rate of DeleteMember requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteMember calls per second</td>
</tr>
<tr>
<td>Rate of DeleteMembership requests</td>
<td>5 TPS</td>
<td>Maximum number of DeleteMembership calls per second</td>
</tr>
<tr>
<td>Resource</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Rate of GetCollaboration requests</td>
<td>5 TPS</td>
<td>Maximum number of GetCollaboration calls per second</td>
</tr>
<tr>
<td>Rate of GetConfiguredTable requests</td>
<td>5 TPS</td>
<td>Maximum number of GetConfiguredTable calls per second</td>
</tr>
<tr>
<td>Rate of GetConfiguredTableAnalysisRule requests</td>
<td>5 TPS</td>
<td>Maximum number of GetConfiguredTableAnalysisRule calls per second</td>
</tr>
<tr>
<td>Rate of GetConfiguredTableAssociation requests</td>
<td>20 TPS</td>
<td>Maximum number of GetConfiguredTableAssociation calls per second</td>
</tr>
<tr>
<td>Rate of GetMembership requests</td>
<td>5 TPS</td>
<td>Maximum number of GetMembership calls per second</td>
</tr>
<tr>
<td>Rate of GetProtectedQuery requests</td>
<td>20 TPS</td>
<td>Maximum number of GetProtectedQuery calls per second</td>
</tr>
<tr>
<td>Rate of GetSchema requests</td>
<td>5 TPS</td>
<td>Maximum number of GetSchema calls per second</td>
</tr>
<tr>
<td>Rate of GetSchemaAnalysisRule requests</td>
<td>5 TPS</td>
<td>Maximum number of GetSchemaAnalysisRule calls per second</td>
</tr>
<tr>
<td>Rate of ListCollaborations requests</td>
<td>5 TPS</td>
<td>Maximum number of ListCollaborations calls per second</td>
</tr>
<tr>
<td>Rate of ListConfiguredTableAssociations requests</td>
<td>5 TPS</td>
<td>Maximum number of ListConfiguredTableAssociations calls per second</td>
</tr>
<tr>
<td>Rate of ListConfiguredTables requests</td>
<td>5 TPS</td>
<td>Maximum number of ListConfiguredTables calls per second</td>
</tr>
<tr>
<td>Rate of ListMembers requests</td>
<td>5 TPS</td>
<td>Maximum number of ListMembers calls per second</td>
</tr>
<tr>
<td>Rate of ListMemberships requests</td>
<td>5 TPS</td>
<td>Maximum number of ListMemberships calls per second</td>
</tr>
<tr>
<td>Rate of ListProtectedQueries requests</td>
<td>5 TPS</td>
<td>Maximum number of ListProtectedQueries calls per second</td>
</tr>
<tr>
<td>Rate of ListSchemas requests</td>
<td>5 TPS</td>
<td>Maximum number of ListSchemas calls per second</td>
</tr>
<tr>
<td>Resource</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Rate of StartProtectedQuery requests</td>
<td>5 TPS</td>
<td>Maximum number of StartProtectedQuery calls per second</td>
</tr>
<tr>
<td>Rate of UpdateCollaboration requests</td>
<td>5 TPS</td>
<td>Maximum number of UpdateCollaboration calls per second</td>
</tr>
<tr>
<td>Rate of UpdateConfiguredTable requests</td>
<td>5 TPS</td>
<td>Maximum number of UpdateConfiguredTable calls per second</td>
</tr>
<tr>
<td>Rate of UpdateConfiguredTableAnalysisRule requests</td>
<td>5 TPS</td>
<td>Maximum number of UpdateConfiguredTableAnalysisRule calls per second</td>
</tr>
<tr>
<td>Rate of UpdateConfiguredTableAssociation requests</td>
<td>5 TPS</td>
<td>Maximum number of UpdateConfiguredTableAssociation calls per second</td>
</tr>
</tbody>
</table>
## 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.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Payment configuration</strong></td>
<td>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.</td>
<td>November 14, 2023</td>
</tr>
<tr>
<td><strong>Query run time - update</strong></td>
<td>The maximum duration a query is run before timeout has been updated from 4 hours to 12 hours.</td>
<td>October 6, 2023</td>
</tr>
<tr>
<td><strong>AWS CloudFormation resources - update</strong></td>
<td>AWS Clean Rooms has added the following new resources: AWS::CleanRooms::Membership ProtectedQueryOutputConfiguration, AWS::CleanRooms::Membership ProtectedQueryResultConfiguration, and AWS::CleanRooms::Membership ProtectedQueryS3OutputConfiguration.</td>
<td>September 7, 2023</td>
</tr>
<tr>
<td><strong>AWS CloudFormation resources - update</strong></td>
<td>AWS Clean Rooms has added the following new resources: AWS::CleanRooms::AnalysisTemplate and AWS::CleanRooms::ConfiguredTable AnalysisRuleCustom.</td>
<td>August 31, 2023</td>
</tr>
<tr>
<td><strong>Separate member abilities</strong></td>
<td>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.</td>
<td>August 30, 2023</td>
</tr>
<tr>
<td><strong>AWS Clean Rooms Glossary</strong></td>
<td>Documentation-only update to add a glossary of AWS Clean Rooms terms.</td>
<td>August 30, 2023</td>
</tr>
<tr>
<td>Topic</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td><strong>Support for Apache Iceberg tables (preview)</strong></td>
<td>AWS Clean Rooms now supports Apache Iceberg tables (preview).</td>
<td>August 25, 2023</td>
</tr>
<tr>
<td><strong>Quotas update</strong></td>
<td>The <a href="#">Quotas section</a> has been updated to reflect the new default quota for memberships per account.</td>
<td>August 9, 2023</td>
</tr>
<tr>
<td><strong>Update to existing policy</strong></td>
<td>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.</td>
<td>July 31, 2023</td>
</tr>
<tr>
<td><strong>Analysis templates and Custom analysis rule</strong></td>
<td>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.</td>
<td>July 31, 2023</td>
</tr>
<tr>
<td><strong>Analysis rules support the OR logical condition</strong></td>
<td>AWS Clean Rooms analysis rules now support the OR logical condition in the JOIN clause.</td>
<td>June 29, 2023</td>
</tr>
<tr>
<td><strong>CloudFormation integration</strong></td>
<td>AWS Clean Rooms now integrates with AWS CloudFormation.</td>
<td>June 15, 2023</td>
</tr>
<tr>
<td><strong>Analysis builder</strong></td>
<td>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.</td>
<td>June 15, 2023</td>
</tr>
<tr>
<td><strong>SQL functions</strong></td>
<td>Documentation-only update to clarify supported SQL functions.</td>
<td>May 5, 2023</td>
</tr>
<tr>
<td><strong>Troubleshooting</strong></td>
<td>Documentation-only update to add a Troubleshooting section for common issues.</td>
<td>April 27, 2023</td>
</tr>
<tr>
<td>Supported data types for AWS Clean Rooms</td>
<td>Documentation-only update to add a new section that lists supported AWS Glue Data Catalog data types.</td>
<td>April 26, 2023</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Examples of AWS CloudTrail events</td>
<td>Documentation-only update to add examples of CloudTrail events for StartProtectedQuery (successful) and StartProtectedQuery (failed).</td>
<td>April 20, 2023</td>
</tr>
<tr>
<td>Update to existing policy (p. 172)</td>
<td>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.</td>
<td>March 21, 2023</td>
</tr>
<tr>
<td>General availability (p. 172)</td>
<td>AWS Clean Rooms is now generally available.</td>
<td>March 21, 2023</td>
</tr>
<tr>
<td>Preview release (p. 172)</td>
<td>Preview release of the AWS Clean Rooms User Guide</td>
<td>January 12, 2023</td>
</tr>
</tbody>
</table>
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 (p. 176).

Only members who have been invited to the collaboration can join the collaboration.

A collaboration can have only one member who can query (p. 177) data, one member who can receive results (p. 177), and one member paying for query compute costs (p. 177).

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 (p. 175)) 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.

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.
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 (p. 176).
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 (p. 176).
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 (p. 61).

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 (p. 177) 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 (p. 177) joins a collaboration (p. 176).

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.