User Guide

AWS Data Exports

AWS Data Exports: User Guide

Table of Contents

What is AWS Data Exports?

AWS Data Exports enables you to create billing and cost management data exports and carbon emissions data exports using basic SQL, and visualize data by integrating with Amazon QuickSight.

You can create exports using the AWS Billing and Cost Management console, AWS CLI, or AWS SDK. In the console, you can use custom column selections. In the AWS CLI or AWS SDK, you can write SQL queries, select columns, filter rows, and rename columns. This allows you to select only the data you want to process, remove any sensitive cost information, and control the output schema of your exports.

There are five export types:

- **Standard data export**, with four different tables to choose from:
  - Cost and Usage Report 2.0 (CUR 2.0)

    > **Note**
    >
    > Cost and Usage Report 2.0 (CUR 2.0) is the new and recommended way to receive your detailed AWS cost and usage data. CUR 2.0 has several improvements over the previous Cost and Usage Reports (CUR). For more information, see Migrating from CUR to CUR 2.0 in Data Exports.

  - Cost optimization recommendations (from Cost Optimization Hub)
  - FOCUS 1.2 with AWS columns
  - FOCUS 1.0 with AWS columns
  - Carbon emissions
- **Cost and usage dashboard**: An export and integration to Amazon QuickSight to deploy a pre-built cost and usage dashboard.
- **Legacy data export**: An export of the Legacy Cost and Usage Reports (CUR). However, Legacy CUR is accessed with a different set of actions (see CUR actions reference) compared to the Data Exports actions (see Data Exports actions reference).

Data Exports includes the following benefits:

- Create recurring exports with the most granular data available, and store them in Amazon S3.

- Customize your data exports with SQL queries using column selections and row filters.
- Create exports with consistent schemas including only the columns you want.
- Remove sensitive cost data or charges associated with certain linked AWS account IDs.
- Reduce the size of your exports by selecting only the columns or rows that you need.
- Automate exporting of cost data and carbon footprint data to support downstream analysis.

To get started with Data Exports

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.

From the **Data Exports** page, you can create new exports, manage existing exports, and create an export that integrates with Amazon QuickSight and deploys a pre-built cost and usage dashboard.

You can also access the Customer Carbon Footprint Tool and the AWS Usage Report from the **Data Exports** page.

Migrating from CUR to Data Exports CUR 2.0

AWS Data Exports allows you to create exports of Cost and Usage Report 2.0 (CUR 2.0). The CUR 2.0 table provides the same information as Cost and Usage Reports (CUR) along with some improvements. Data Exports enables you to create a CUR 2.0 export that is backwards compatible with the data pipelines you've been using to process CUR.

CUR 2.0 provides the following improvements over CUR:

- **Consistent schema:** CUR 2.0 contains a fixed set of columns, whereas the columns included for CUR can vary monthly depending on your usage of AWS services, cost categories, and resource tags.
- **Nested data:** CUR 2.0 reduces data sparsity by collapsing certain columns from CUR into individual columns with key-value pairs of the collapsed columns. Optionally, you can query the nested keys in Data Exports as separate columns to match the original CUR schema and data.
- **Additional columns:** CUR 2.0 contains two additional columns: **bill_payer_account_name** and **line_item_usage_account_name**.

The following table outlines the differences between CUR 2.0 and legacy CUR in more detail:

	CUR 2.0	Legacy CUR
Data schema	Fixed schema. For the complete column list, see Cost and Usage Report (CUR) 2.0.	Dynamic schema based on AWS usage and activity. For the partial column list, see Data dictionary.
Exclusive columns	`bill_payer_account_name` `line_item_usage_account_name`	None
Export customization	Enables basic SQL for column selections, row filtering, and column aliasing (renaming).	Not supported. You must manually set up Athena/Qu

	CUR 2.0	Legacy CUR
	For details about the supported SQL syntax, see Data query.	ickSight to create the view you require.

Nested columns with key-value pairs	`resource_tags`
	`cost_category`
	`product`
	`discount`
		No nested columns.
		The four nested columns in CUR 2.0 are split into separate columns in legacy CUR (for example, `resource_tags_user_creator`).

File delivery destination	S3 bucket	S3 bucket
File output formats	GZIP, Parquet	ZIP, GZIP, Parquet
Integration with other AWS services	Amazon QuickSight	Amazon Athena, Amazon Redshift, Amazon QuickSight
Amazon CloudFormation support	Yes For details, see AWS Data Exports resource type reference in the *AWS CloudFormation User Guide*.	Yes For details, see AWS Cost and Usage Report resource type reference in the *AWS CloudFormation User Guide*.

	CUR 2.0	Legacy CUR
Tag and cost category data	Tag and cost category names are normalized to remove special characters and spaces. In the event that there are conflicting tags or cost categories after normalization, only one value is kept. For more information, see Column names.	The behavior is different between legacy CUR Parquet and CSV file formats. **Legacy CUR Parquet:** Tag and cost category names are normalized to remove special characters and spaces. In the event that there are conflicting tags or cost categories after normalization, only one value is kept. For more information, see Column names. **Legacy CUR CSV:** Tag and cost category names are not changed.

For more detailed information about the schema of CUR 2.0, see the Data Exports table dictionary.

You can migrate to CUR 2.0 in Data Exports in two ways:

- Method one: Create an export with an SQL query using the CUR schema
- Method two: Create an export of CUR 2.0 with its new schema

Method one: Create an export with an SQL query using the CUR schema

You can create an export with an SQL query. The export schema matches what you receive today in CUR. You do this using the AWS API or SDK.

1. Determine (a) the list of columns and (b) the CUR content settings (**Include resource IDs**, **Split cost allocation data**, and **Time granularity**) needed in order to match your CUR today.

   1. You can determine the list of columns either by viewing the schema of one of your CUR files or going to the manifest file and extracting the list of columns from there.
   2. You can determine the CUR content settings by going to Data Exports in the console and choosing your CUR export to view its details.

2. Write an SQL query that selects the columns you identified from the CUR 2.0 table named `COST_AND_USAGE_REPORT`.

   1. All column names in the CUR 2.0 table are in snake case (for example, `line_item_usage_amount`). For your SQL statement, you might need to convert the previous column names to snake case.
   2. For your SQL statement, you need to convert all `resource_tag` and `cost_category` columns, and certain `product` and `discount` columns, to have the dot operator in order to select the nested columns in CUR 2.0. For example, to select the `product_from_location` column in CUR 2.0, write an SQL statement selecting `product.from_location`.

      Example: `SELECT product.from_location FROM COST_AND_USAGE_REPORT`

      This selects the `from_location` column of the `product` map column.
   3. By default, the column selected with a dot operator is named by the attribute (for example, `from_location`). To match your existing CUR, you'll need to declare an alias for the column in order to have the same as before.

      Example: `SELECT product.from_location AS product_from_location FROM COST_AND_USAGE_REPORT`

      For more details on nested columns, see the Data Exports table dictionary.

3. Write the CUR content settings, identified in step 1, into the table configuration format for the `CreateExport` API. You need to provide these table configurations with your data query in the next step.

4. In the AWS SDK/CLI for Data Exports, use the `CreateExport` API to input your SQL query and table configurations into the data-query field.

   1. Specify delivery preferences, such as the target Amazon S3 bucket and the overwrite preference. We recommend choosing the same delivery preferences you had before. For more information on the required fields, see AWS Data Exports in the *AWS Billing and Cost Management API Reference*.

   2. Update the permissions of the target Amazon S3 bucket to allow Data Exports to write to the bucket. For more information, see Setting up an Amazon S3 bucket for data exports.

5. Direct your data ingestion pipeline to read data from the directory in the Amazon S3 bucket where your CUR 2.0 is being delivered.

Method two: Create an export of CUR 2.0 with its new schema

You can create an export of CUR 2.0 with its new schema of nested columns and additional columns. However, you'll need to adjust your current data pipeline to process these new columns. You do this using the console, the AWS API, or SDK.

1. Determine the CUR content settings (**Include resource IDs**, **Split cost allocation data**, and **Time granularity**) needed in order to match your CUR today.

   - You can determine the CUR content settings by going to Data Exports in the console and choosing your CUR export to view its details.

2. Using either the Data Exports console page (**Option A**) or the AWS SDK/CLI (**Option B**), create an export of CUR 2.0 that selects all columns from the "Cost and usage report" table.
3. **(Option A)** To create the export in the console:

   1. In the navigation pane, choose **Data Exports**.
   2. On the **Data Exports** page, choose **Create**.
   3. Choose **Standard data export**.

      For the **Cost and Usage Report (CUR 2.0)** table, all columns are selected by default.
   4. Specify the CUR content settings that you identified in step 1.
   5. Under **Data table delivery options**, choose your options.
   6. Choose **Create**.

4. **(Option B)** To create the export using the AWS API/SDK, first write a query that selects all columns in the `COST_AND_USAGE_REPORT` table.

   1. Use the `GetTable` API to determine the complete list of columns and receive the full schema.
   2. Write the CUR content settings, identified in step 1, into the table configuration format for the `CreateExport` API.

   3. Use the `CreateExport` API to input your SQL query and table configurations into the `data-query` field.
   4. Specify delivery preferences, such as the target Amazon S3 bucket and the overwrite preference. We recommend choosing the same delivery preferences you had before. For more information on the required fields, see AWS Data Exports in the *AWS Billing and Cost Management API Reference*.
   5. Update the permissions of the target Amazon S3 bucket to allow Data Exports to write to the bucket. For more information, see Setting up an Amazon S3 bucket for data exports.

5. Direct your data ingestion pipeline to read data from the directory in the Amazon S3 bucket where your CUR 2.0 is being delivered.

   You also need to update your data ingestion pipeline and your business intelligence tools to process the following new columns with nested key-values: `product`, `resource_tags`, `cost_category`, and `discounts`.

Creating data exports

You can use the **Data Exports** page in the Billing and Cost Management console to create data exports of three different types: standard exports, cost and usage dashboard exports, and legacy exports.

There are the following limits on the number of exports you can create per table:

- **Cost and Usage Report 2.0 (CUR 2.0)**: 5 exports
- **Cost optimization recommendations**: 2 exports
- **FOCUS 1.0 with AWS columns**: 2 exports
- **Cost and usage dashboard**: 2 exports
- **Carbon emissions**: 2 exports

For more information, see Quotas and restrictions.

Set up an export in minutes by either creating an export in the console and selecting the table you want to export, or creating an export in the AWS SDK/CLI and defining an SQL query of column selections and row filters from the data table you want.

When creating an export in the console, you can create an Amazon S3 bucket for your data export storage. When creating an export in the AWS SDK/CLI, you need to create an Amazon S3 bucket with the correct bucket policy in advance. For more information, see Setting up an Amazon S3 bucket for data exports.

Once you create a new data export, Data Exports starts to export the data to the Amazon S3 bucket.

> **Note**
>
> It can take up to 24 hours for AWS to start delivering exports to your Amazon S3 bucket. Once delivery starts, AWS refreshes the billing and cost management export output at least once a day and the carbon emissions export output at least once a month in your S3 bucket. The actual refresh rate may be different due to various factors.

Topics

- Setting up an Amazon S3 bucket for data exports
- Creating a standard export
- Creating a cost and usage dashboard
- Creating a Legacy CUR export
- Creating exports with billing views
- Data query–SQL query and table configurations
- Configuring Cost and Usage Reports 2.0 using AWS Billing Conductor

Setting up an Amazon S3 bucket for data exports

You must have an Amazon S3 bucket in your AWS account to receive and store your data exports. When creating an export in the console, you can select an existing S3 bucket that you own, or you can create a new bucket. In either case, you need to review and confirm the application of the following default S3 bucket policy. Editing this policy in the Amazon S3 console or changing the S3 bucket owner after you've created an export prevents Data Exports from delivering your exports. Storing the exports data in your S3 bucket is billed at standard Amazon S3 rates. For more information, see Quotas and restrictions.

> **Note**
>
> The account that creates the export must also own the S3 bucket that AWS sends the exports to. You cannot configure an export to deliver to an S3 bucket owned by another account.

The following policy is applied to every S3 bucket when creating a data export:

This S3 bucket policy ensures that Data Exports can only deliver exports to the S3 bucket on behalf of the account that created the export. It also allows Data Exports to verify that the S3 bucket is still owned by the account that created the export.

- To deliver exports to your S3 bucket, AWS needs write permissions for that S3 bucket. To do this, the S3 bucket policy grants the Data Exports service (`bcm-data-exports.amazonaws.com`) permission to deliver (`s3:PutObject`) reports to the S3 bucket you own (`arn:aws:s3:::<EXAMPLE-BUCKET>/*`).

- Every time Data Exports makes the request to write to the S3 bucket, it must provide the account ID of the account that created the export. The condition keys `aws:SourceArn` and `aws:SourceAccount` enforce this.
- This S3 bucket policy does not give AWS permissions to read or delete any objects in your S3 bucket, including the Cost and Usage Reports after they've been delivered.

For an Amazon S3 bucket that has access control list (ACL) enabled, Data Exports applies a `BucketOwnerFullControl` ACL to the reports when delivering them. By default, Amazon S3 objects, such as these reports, can only be read by the user or service principal who wrote them. To provide you or the S3 bucket owner with permission to read the reports, AWS needs to apply the `BucketOwnerFullControl` ACL. The ACL grants the S3 bucket owner `Permission.FullControl` for these reports. However, it's recommended to disable ACL and use an S3 bucket policy to control access.

> **Note**
>
> For newly-created S3 buckets, ACLs are disabled by default. For more information, see Controlling ownership of objects and disabling ACLs for your bucket.

If you see an **Invalid bucket** error in the **Data Exports** console page, verify that the policy and S3 bucket ownership haven't changed since report setup.

Creating a standard export

You can create a standard data export that you can analyze using other processing tools (Amazon Athena, for example).

To create a standard data export

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. Choose **Create export**.
4. On the **Create export** page, under **Export type**, choose **Standard data export**.
5. For **Export name**, enter a name for your export.

Export names can have up to 128 characters and must be unique. Valid characters are a-z, A-Z, 0-9, - (hyphen), and _ (underscore).

6. Under **Data table configurations**, you can specify the table and columns to be contained within your export. First, select the table you want to export.

> **Note**
>
> Exporting the Cost optimization recommendations table requires a service-linked role. For more information, see Service-linked roles for Data Exports.
> Exporting the Carbon emissions table requires the IAM permission `sustainability:GetCarbonFootprintSummary` to access the Customer Carbon Footprint Tool and data.

With the exception of FOCUS 1.0 with AWS columns and Carbon emissions, there are different table configurations to add data to your export.

1. For **CUR 2.0**:

    a. Select **Include resource IDs** to include the IDs of each individual resource in the export.

    > **Note**
    >
    > Including resource IDs creates individual line items for each of your resources. This might increase the size of your export significantly, based on your AWS usage.
    > Selecting resource ID will add a Tag column containing data about users, accounts, cost categories, and resources when you create a new report. You can deselect the columns to avoid redundant information.

    b. Select **Split cost allocation data** to include detailed cost and usage for shared resources (Amazon ECS and Amazon EKS).

> **Note**
>
> Including split cost allocation data creates individual line items for each of your resources (that is, ECS tasks and Kubernetes pods). This might increase the size of your Cost and Usage Report significantly, based on your AWS usage.

c. Select **Include Capacity reservation data** to include the Capacity reservation columns and row-level granularity in the export.

> **Note**
>
> Including Capacity reservation data creates 3 new columns and can split the instance line items, based on your AWS usage.

d. Select **Enable manual discount format** to convert your discounts so that they appear in the Cost and Usage Report in the manual discount format instead of the standard automated format.

> **Note**
>
> This option only appears if you are on the discount automation program.

e. For **Time granularity**, choose between hourly, daily, or monthly to have the line items in the export aggregated by that time granularity.

2. For **FOCUS 1.0 with AWS columns**, there are no table configurations.
3. For **Carbon emissions**, there are no table configurations.
4. For **Cost optimization recommendations**:

   a. Select **Include all recommendations** to remove the lowest savings value recommendation of recommendations that are incompatible with one another.

   b. Add **Recommendation filters** if you want certain types of recommendations to be filtered out before incompatible recommendations are removed.

> **Note**
>
> If you specified these settings in the Cost Optimization Hub console, they will be carried over to Data Exports when you choose **Create an export** in Cost Optimization Hub.

7. For **Column selection**, select the columns you want to include in your export. If unsure, select all columns by selecting the first check box at the top of the table. Selecting more columns may increase the file size of your export.
8. Under **Data table delivery options**, for **Data export refresh cadence**: .

   - For billing and cost management data exports, the only option available is **Daily - export is refreshed up to one time per day**.
   - For carbon emissions data exports, the only option available is **Monthly - export is refreshed once per month**. Each update provides the carbon emissions data from the previous month (for example, a February update contains January data).
9. For **Compression type and file format**, choose between the following for your export:

   - Parquet – Parquet
   - gzip – text/csv
10. For **File versioning**, choose between the following which determines whether your export is overwritten with each update:

    - **Overwrite existing data export file**: Each export refresh overwrites the previous delivery within the data partition (for example, billing periods). Overwriting exports can save on Amazon S3 storage costs.

      > **Note**
      >
      > Overwrite is not supported for exports of cost optimization recommendations.
    - **Create new data export file**: Each export refresh is written to a separate directory, even for deliveries of the same partition (for example, billing period). Creating new export versions allows you to track the changes in cost and usage data over time.
11. Under **Data export storage settings**, for **S3 bucket** name, choose **Configure**.
12. In the **Configure S3 bucket** dialog box, do one of the following:

- Select existing bucket.
- Choose **Create a bucket**, enter an **S3 bucket name**, and then choose the **Region** where you want to create a new bucket.

13. Review the **Bucket policy**. If you're selecting an existing bucket, you need to acknowledge that Data Exports will overwrite you existing S3 bucket policy. The new policy will allow both CUR and Data Exports to deliver exports.
14. For **S3 path prefix**, enter a name for the directory that will be created in your S3 bucket to store all the export data.
15. Under **Tags**, you can choose to add up to 50 tags in order to search and filter your resources or track your AWS costs.

    > **Note**
    >
    > Adding tags is optional.

16. Choose **Create** to complete the creation of your export.

Creating a cost and usage dashboard

You can visualize your billing and cost management data by deploying a pre-built Cost and Usage Dashboard powered by Amazon QuickSight.

To create a cost and usage dashboard

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. On the **Data Exports** page, choose either **Create** or the **Cost and usage dashboard** tile.
4. On the **Create** page, under **Export type**, choose **Cost and usage dashboard powered by QuickSight**.
5. For **Export name**, enter a name for your dashboard.

   Export names can have up to 128 characters and must be unique. Valid characters are a-z, A-Z, 0-9, - (hyphen), and _ (underscore).

6. For **QuickSight dashboard settings** your QuickSight account details such as **account name**, **account ID**, **account edition**, and **authentication method** are automatically populated.

   1. If the QuickSight account details don't populate automatically, choose **Create account** to sign up if you're new to QuickSight, or log in to your QuickSight account if you're an existing QuickSight customer.
   2. Once you successfully create or log in to your QuickSight account, you'll see a success message. Close the window and return to **Data Exports**.
   3. Under **QuickSight dashboard settings**, choose **Refresh**.

   > **Note**
   >
   > This feature requires Enterprise Edition.

7. For **QuickSight namespace**, enter your namespace.
8. For **QuickSight username**, enter the details for the user who has permissions to access the QuickSight dashboard.
9. For **QuickSight region**, choose the AWS Region where you want to create the QuickSight dashboard.
10. The **Data table content settings** and **Data table delivery options** are preset and can't be edited.
11. Under **Data export storage settings**, for **S3 bucket** name, choose **Configure**.
12. In the **Configure S3 bucket** dialog box, do one of the following:
    - Select existing bucket.
    - Choose **Create a bucket**, enter an **S3 bucket name**, and then choose the **Region** where you want to create a new bucket.
13. Review the **Bucket policy**, and then choose **Create bucket**.
14. For **S3 path prefix**, enter the S3 path prefix that you want prepended to the name of your export.
15. Under **Service access**, choose a method to authorize QuickSight:
    - Create a new service role (default)
    - Use an existing service role

16. Under **Tags**, you can choose to add up to 50 tags in order to search and filter your resources or track your AWS costs.

    > **Note**
    >
    > Adding tags is optional.

17. Choose **Create**.

You can always return to the **Data Exports** page of the AWS Billing and Cost Management console to see when your Cost and Usage Dashboard was last updated.

Creating a Legacy CUR export

You can create a data export of your legacy Cost and Usage Report (CUR). This workflow uses the legacy `cur` APIs and doesn't allow you to use SQL to define your export contents. CUR 2.0 with its additional columns and SQL access is only available as a standard data export.

To create a legacy data export

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. Choose **Create**.
4. On the **Create** page, under **Export type**, choose **Legacy CUR export**.
5. For **Export name**, enter a name for your export.
6. Under **Export content**, select the data to include in your CUR export.

   - For **Additional export content**, select **Include resource IDs** to include the IDs of each individual resource in the export.

     > **Note**
     >
     > Including resource IDs creates individual line items for each of your resources. This might increase the size of your export significantly, based on your AWS usage.

- Select **Split cost allocation data** to include detailed cost and usage for shared resources (Amazon ECS and Amazon EKS).

  > **Note**
  >
  > Including split cost allocation data creates individual line items for each of your resources (that is, ECS tasks and Kubernetes pods). This might increase the size of your Cost and Usage Report significantly, based on your AWS usage.

- Select **Enable manual discount format** to convert your discounts so that they appear in the Cost and Usage Report in the manual discount format instead of the standard automated format.

  > **Note**
  >
  > This is only available for customers on Discount Automation.

7. Under **Data table delivery options**, for **Time granularity**, choose one of the following:

   - **Hourly** if you want the line items in the export to be aggregated by the hour.
   - **Daily** if you want the line items in the export to be aggregated by the day.
   - **Monthly** if you want the line items in the export to be aggregated by month.

8. For **Report versioning**, choose between the following:

   - **Create new report version**: Each report refresh will be written to a separate directory, even for deliveries of the same billing period. Choose this to improve the ability to audit your exports over time.
   - **Overwrite existing report**: Each report refresh will overwrite the previous delivery within the same billing period. Deliveries of new billing periods be delivered as new files and directories. Choose this to save on Amazon S3 storage costs.

9. For **Report data integration**, choose whether you want to enable your Cost and Usage Reports to integrate with Amazon Athena, Amazon Redshift, or Amazon QuickSight. The report is compressed in the following formats:

   - **Amazon Athena**: Selects the delivery options optimal for Amazon Athena which are Parquet file format and overwrite existing report. Also delivers a script that can be used to set up the integration.

- **Amazon Redshift**: Selects the delivery option optimal for Amazon Redshift which is gzip/csv file format. Also delivers a script that can be used to set up the integration.
- **Amazon QuickSight**: Selects the delivery option optimal for Amazon QuickSight which is gzip/csv file format.

10. For **Compression type and file format**, choose between the following:
    - Parquet – Parquet
    - gzip – text/csv
    - zip – text/csv
11. Under **Data export storage settings**, for **S3 bucket** name, choose **Configure**.
12. In the **Configure S3 bucket** dialog box, do one of the following:
    - Select existing bucket.
    - Choose **Create a bucket**, enter an **S3 bucket name**, and then choose the **Region** where you want to create a new bucket.
13. Review the **Bucket policy**, and then choose **Create bucket**.
14. For **S3 path prefix**, enter the S3 path prefix that you want prepended to the name of your export.
15. Under **Tags**, you can choose to add up to 50 tags in order to search and filter your resources or track your AWS costs.

    > **Note**
    >
    > Adding tags is optional.
16. Choose **Create report**.

Creating exports with billing views

When you sign in as a bill transfer account using billing transfer, or as a management account using AWS Billing Conductor, you can create an export based on your AWS managed billing views (billing groups and billing transfer views).

⚠ **Important**

- Custom billing views aren't supported.
- You can create billing view-based reports only from the Data Exports page. The legacy Cost and Usage Reports page doesn't support creating reports based on billing views.

You can create reports based on billing views whether billing view mode is enabled or disabled, because reports are resources of your account.

To create a report based on billing views

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. Choose **Create report**.
4. Choose the billing view type (managed views only).
5. Choose the specific view for your report.
6. Complete the remaining steps to create your report.

ⓘ **Note**

When creating a report based on a billing transfer showback/chargeback view or billing group view, you must disable the Split Cost Allocation Data functionality.

For more information about Data Exports for billing transfer use cases, see billing transfer best practices.

Data query–SQL query and table configurations

Data Exports enables you to write SQL queries (column selections, row filtering, column aliasing) that are executed against the tables provided–CUR 2.0, for example. Each table might also have table configurations that alter the data contained within the table. For example, with CUR 2.0, you can specify a configuration to choose a time granularity of hourly, daily, or monthly, or a configuration to add cost and usage data at resource-level granularity.

For an export data query to be fully defined, you must specify the following two attributes:

- **SQL query:** The SQL statement is executed against a table and determines what data is returned by the export.
- **Table configurations:** The table configuration settings change what data is contained within the table before the SQL query is executed against it.

In the **Data Exports** console page, you can use the workflow that builds the SQL statement and table configurations based on your selections. In the Data Exports SDK/CLI, you can write your own SQL statement and table configurations.

Data Exports SQL statements (`QueryStatement`) use the following syntax:

```
SELECT <column_name_a>, <column_name_b>.<attribute_name> AS <new_name>, ...
FROM <TABLE_NAME>
[ WHERE <column_name> OPERATOR <value> AND|OR ... ]
[ LIMIT number ]
```

Data Exports table configurations (`TableConfigurations`) use the following syntax:

```
{"<TABLE_NAME>":
    {"<CONFIGURATION_NAME_A>": "<value>",
     "<CONFIGURATION_NAME_B>": "<value>",
     ...}
            }
```

SQL query

The SQL query is executed against a table and determines what data is returned in an export. The SQL statement can be altered after an export has been created, but the table selected can't be changed.

SQL statements (in the QueryStatement field) can have a maximum of 36,000 characters.

The possible keywords in a Data Exports SQL query are as follows.

Note

The keywords are not case-sensitive. The column names and table names are case-sensitive.

SELECT

Required.

Specifies which columns are to be selected from the table. There can only be one SELECT statement per query.

Use the dot operator `.` to specify selecting an attribute of a MAP or STRUCT column as a separate column. The name of the resulting column in the SQL output is the attribute name by default.

For example, you can select attributes from the product MAP column.

```
SELECT product.from_location FROM COST_AND_USAGE_REPORT
```

This selects the `from_location` attribute from the `product` column and creates a new column with the attribute's data. By default, in the output, this column's name will be `from_location`. However, it can be renamed with AS.

For more information on the MAP and STRUCT columns available in each table, and the attributes these columns have, see the Data Exports table dictionary.

AS

Optional.

Enables renaming of the column being selected. The new column name can't have spaces or characters other than alphanumeric characters (a-z, A-Z, and 0-9) and underscores (_). You can't use quotes when defining the column alias in order to use other characters.

Aliasing can be useful when selecting an attribute of a MAP or STRUCT column to rename the resulting column to match the schema of the CUR. For example, to match how the CUR shows the `product_from_location` column, write the following query in Data Exports with the CUR 2.0 table.

```
SELECT product.from_location AS product_from_location FROM
COST_AND_USAGE_REPORT
```

This creates an export with a column named `product_from_location`.

FROM

Required.

Specifies the table to be queried. There can only be one FROM statement per query.

WHERE

Optional.

Filters the rows to only those that match your specified clause.

The WHERE clause supports the following operators:

- **=** Value must match the string or number.
- **!= and <>** Value must not match the specified string or number.
- **<, <=, >,** and **>=** Value must be less than, less than or equal to, greater than, or greater than or equal to the number.
- **AND** Both conditions that are specified must be true to match. You can use multiple **AND** keywords to specify two or more conditions.
- **OR** Either conditions that are specified must be true to match. You can use multiple **OR** keywords to specify two or more conditions.
- **NOT** The condition specified must not be true to match.
- **IN** Any of the values specified within the parentheses after the keyword must be true to match.
- Parentheses can be used to construct multi-conditional WHERE clauses

> **Note**
>
> When expressing strings as the value following an operator, use single quotes `'` instead of double quotes. You don't need to escape the single quotes. For example you can write the following WHERE statement:
>
> `WHERE line_item_type = 'Discount' OR line_item_type = 'Usage'`

LIMIT

Optional.

Limits the number of rows returned by the query to the value that you specify.

Table configurations

Table configurations are user-controlled properties that a user can set to change the data or schema of a table before it's queried in Data Exports. The table configurations are saved as a JSON statement and are either specified through user input in the AWS SDK/CLI or user selections in the console.

For example, CUR 2.0 has table configurations to change data granularity (hourly, daily, monthly), whether resource-level granular data is included, and whether split cost allocation data is included. Not all tables have configurations. For more information on the configurations available for each table, see the Data Exports table dictionary.

Each table configuration parameter has a default value that is assumed if a table configuration is not specified by the user. Table configurations can't be changed after an export is created.

Configuring Cost and Usage Reports 2.0 using AWS Billing Conductor

With AWS Billing Conductor, you can create pro forma AWS Cost and Usage Report (AWS CUR) 2.0 for each billing group. These pro forma reports use the same file format, granularity, and columns as the standard AWS CUR 2.0, providing the most comprehensive cost and usage data available for a given time period.

For more information about AWS Billing Conductor, see the AWS Billing Conductor User Guide.

Topics

- Comparing standard and AWS Billing Conductor Cost and Usage Reports
- Creating pro forma Cost and Usage Reports for a billing group

Comparing standard and AWS Billing Conductor Cost and Usage Reports

There are a few differences between the standard Cost and Usage Reports and pro forma AWS CUR created using the AWS Billing Conductor configuration.

Account coverage

- Standard AWS CUR – Includes cost and usage data for all accounts in your consolidated billing family
- Pro forma AWS CUR – Includes only accounts that belong to the specific billing group at the time of report generation

Invoice handling

- Standard AWS CUR – Populates the invoice column after AWS generates an invoice
- Pro forma AWS CUR – Does not populate the invoice column because AWS does not generate or issue invoices based on pro forma billing data

Creating pro forma Cost and Usage Reports for a billing group

Use the following steps to generate a pro forma AWS CUR for a billing group.

To create pro forma Cost and Usage Reports for a billing group

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. Choose **Create**.
4. In the **Export details** section, choose **Standard data export**.
5. For **Export name**, enter a name for your export.
6. Under **Data table content settings**, choose **CUR 2.0**.
7. Under **Data table configurations**, choose **Include resource IDs** to include the IDs of each individual resources in the report.

   Split cost allocation data is disabled when pro forma data export is enabled.

8. Choose **Next**.
9. For **S3 bucket**, choose **Configure**.
10. In the **Configure S3 Bucket** dialog box, do one of the following:

    - Choose an existing bucket from the drop down list and choose **Next**.
    - Enter a bucket name and the AWS Region where you want to create a new bucket and choose **Next**.
11. Review the **Bucket policy**, select **I have confirmed that this policy is correct**, and choose **Save**.
12. For **S3 path prefix**, enter the S3 path prefix that you want prepended to the name of your export.
13. For **Time granularity**, choose one of the following:

    - **Hourly** if you want the line items in the report to be aggregated by the hour.
    - **Daily** if you want the line items in the report to be aggregated by the day.
    - **Monthly** if you want the line items in the report to be aggregated by the month.
14. For **Report versioning**, choose whether you want each version of the report to overwrite the previous version of the report, or to be delivered in addition to the previous versions.

    Overwriting reports can save on Amazon S3 storage costs. Delivering new report versions can improve auditability of billing data over time.
15. Choose **Next**.
16. After you have reviewed the settings for your report, choose **Review and Complete**.

Viewing and managing data exports

To view details about your exports, use the **Data Exports** page in the AWS Billing and Cost Management console. To view your export files, use the S3 console link for your Amazon S3 bucket on the **Data Exports** page. To view your export dashboards, use the QuickSight link on the **Data Exports** page, or go directly to the QuickSight console and find your dashboard.

To view your export details, files, and dashboards

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. In the **Exports and dashboards** list, find the name of the export that you want to view.
4. To view the export details, choose the link in the **Export name** column to view the summary page describing the export settings.
5. To view the export files, choose the link in the **S3 bucket** column to be brought to the S3 console for your bucket.
6. To view the QuickSight dashboard, choose the **Cost and usage dashboard** link in the **Export type** column.

The following is an overview of the columns in the **Exports and Dashboards** list:

- **Export name**: The name you chose when creating the export.
- **Status**: The health of your export. It can have two values:
  - **Healthy**: This status indicates that the most recent export delivery was successful.

> **Note**
>
> Your Cost and Usage Dashboard could be missing the data for the current month because it can take up to 24 hours for all your data to be populated in your dashboard. If the export status says “Healthy”, allow 24 hours for your dashboard to update with the current month's data.
> When you use billing transfer as a bill transfer account, or AWS Billing Conductor as a management account, you can see all billing view-based exports with billing

view mode disabled. When you enable billing view mode, you can see only exports associated with the selected billing view.

  - **Unhealthy**: This status indicates that the most recent export delivery failed.
- **Export type**: The type of export created. Data Exports has three types of exports:
  - **Standard data export**: A customized export of a table that delivers to Amazon S3 on a recurring basis.
  - **Cost and usage dashboard**: An export and integration to Amazon QuickSight that deploys a pre-built cost and usage dashboard. This becomes a link to a dashboard.
  - **Legacy CUR export**: An export of the Legacy Cost and Usage Report (CUR).
- **Data table**: The table that your export is querying.
- **Date created**: The time and date when your export was created.
- **Date last refreshed**: The time and date when your export was last refreshed.
- **S3 bucket**: The S3 bucket to which your export is being delivered to.

Understanding export delivery

In the following sections, you'll find information about your export delivery.

- **Export S3 parent directory structure:** How export data is structured in the S3 directory to which your export is delivered to.
- **Export refreshing:** How often your export updates in your S3 directory.
- **Export overwriting and create new:** How your export delivery changes with overwriting and creates new delivery preferences.
- **Export data file names and chunks:** How the export files (gzip/csv or Parquet) are named.

Export S3 parent directory structure

Each export delivers the data from the query to S3 (as one or more gzip/csv or Parquet files) and a `Manifest.json` metadata file containing information about the export definition at the time the export was executed.

Data

The data resulting from the export query is stored in the following S3 file path:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/
```

The partition corresponds to the table that is being queried. For CUR 2.0, the partition corresponds to the “billing period” of a given CUR 2.0 export.

`prefix`: The S3 file prefix that you assign to the export.

`export-name`: The name that you assign to the export.

`partition`: The partition describes how a single table is partitioned into separate tables for delivery. For CUR 2.0, the partition corresponds to the “billing period” in the format `BILLING_PERIOD=YYYY-MM`. For example, the partition for November 2023 is 2023-11.

The following is an example of an S3 file path:

```
s3://my-data-export-s3-bucket/my-cur-files/business_group_a_cur/data/
BILLING_PERIOD=2023-11
```

Metadata

The `Manifest.json` metadata file for the query is stored in the following S3 file path:

```
s3://<bucket-name>/<prefix>/<export-name>/metadata/<partition>/<export-
name>-Manifest.json
```

The `Manifest.json` file is updated each time the export is refreshed. A new `Manifest.json` file is created for each new partition created by the export. For CUR 2.0, this means a new `Manifest.json` file is generated when a new billing period begins.

Manifest files contain the following information:

- All of the columns that are included in the export.
- A list of the export files and their file path. We recommend identifying which files to ingest by programmatically reading this list.
- The time period covered by the export.

The `Manifest.json` is only delivered once all of the export data files have been delivered to S3.

Export refreshing

Data Exports refreshes your exports each time the source data is updated. For CUR 2.0, this occurs at least once a day. The current billing period (partition) is refreshed until the billing period ends, at which point deliveries of the next billing period begin. Deliveries of the next billing period only contain charges and billing data for that billing period. After the billing period ends, AWS may update the export delivery for the previous billing period within the first two weeks after it ended.

Export overwriting and create new

When you create an export, you can choose to either create new export files or overwrite the existing export files with each refresh.

Create new

Creating new export files uses more S3 storage because all export refreshes are kept. Overwriting the previous export files uses less S3 storage because only the latest version of each billing period refresh is kept.

When in “create new” mode, the export files are delivered to the following S3 path:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/<timestamp>-<execution-id>
```

The `timestamp` is the date and time when the export was executed. The `execution-id` is the unique ID assigned to the execution.

For "create new", two `Manifest.json` files are delivered with each export execution. One is stored in the `metadata/<partition>/<timestamp>-<execution-id>` directory, and the other is overwritten in the `metadata/<partition>` directory. The manifest in the `metadata/<partition>` directory always represents the most recent refresh and its data is used to identify the location of the most recently refreshed export files.

Overwrite

Overwriting only applies for refreshes of the same partition (that is, billing period). Once a new billing period begins, the export creates a new S3 directory with a name based on the latest partition or billing period, and begins delivering the new export partition there. The export of the previous partition is not overwritten unless the data for that specific partition is updated.

When in “overwrite” mode, the export files are delivered to the following S3 path:

`s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/`

The export files in this file directory are overwritten with each delivery of the same partition (that is, billing period).

Export files are delivered as multiple “chunks” (separate gzip/csv or Parquet files) when the export becomes sufficiently big. If the export ever decreases in size during the month (due to a changed query or correction to data), fewer chunks may be needed to deliver the export refresh. In this case, Data Exports overwrites any extra chunks from the last refresh with empty data.

For overwriting, one `Manifest.json` file is delivered with each export execution. It is stored in the `metadata/<partition>` directory and is overwritten with each refresh.

Export data file names and chunks

Exports either deliver the results of one execution as one file (gzip/csv or Parquet) or in multiple “chunks” (separate gzip/csv or Parquet files) when the export becomes sufficiently big.

Exports are named as follows for the gzip/csv file format:

`<export-name>-<chunk-number>.csv.gz`

Exports are named as follows for the Parquet format:

`<export-name>-<chunk-number>.snappy.parquet`

Chunk numbers always have five digits. Chunk numbers are enumerated starting at `00001`.

Summary

Export data file names with directory for create new

Parquet:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/<timestamp>-<execution-id>/<export-name>-<chunk-number>.snappy.parquet
```

gzip/csv:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/<timestamp>-<execution-id>/<export-name>-<chunk-number>.csv.gz
```

Export data file names with directory for overwrite

Parquet:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/<export-
name>-<chunk-number>.snappy.parquet
```

gzip/csv:

```
s3://<bucket-name>/<prefix>/<export-name>/data/<partition>/<export-
name>-<chunk-number>.csv.gz
```

Manifest file names with directory for create new

The “create new” mode delivers `Manifest.json` to two locations.

The first location is in a folder representing a specific execution of an export (named by `timestamp` and `execution-id`). This Manifest corresponds to that specific execution. The file path is as follows:

```
s3://<bucket-name>/<prefix>/<export-name>/metadata/<partition>/
<timestamp>-<execution-id>
```

The second location is in a partition folder containing all executions. This Manifest is the same file from the most recent execution of the export. You can read this Manifest to identify the exact file paths of all recent export files. The file path is as follows:

```
s3://<bucket-name>/<prefix>/<export-name>/metadata/<partition>/
Manifest.json
```

Manifest file names with directory for overwrite

The “overwrite” mode delivers `Manifest.json` to one location.

```
s3://<bucket-name>/<prefix>/<export-name>/metadata/<partition>
```

The Manifest in this directory is overwritten with each refresh of a given partition (that is, billing period).

Editing export details

You can use the **Data Exports** page in the AWS Billing and Cost Management console to edit your export details.

To edit your export details

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. From your list of exports, choose the name of the export that you want to edit.
4. On the **Export details** page, choose **Edit**.
5. In **Edit** mode, you can update the column selection for your export and the export overwrite preference.

   You can't update the data table, the additional export content (table configurations), or the compression type and file format for an existing export. To change these details, you need to create a new export.

Editing export tags

You can use the **Data Exports** page in the AWS Billing and Cost Management console to edit your export tags.

To edit your export tags

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. From your list of exports, choose the name of the export that you want to edit.
4. In the **Tags** section at the bottom of the page, choose **Manage tags**.
5. On the **Manage tags** page, do the following:
   - To add a new tag, choose **Add new tag**. You can add up to 50 tags to search and filter your resources or track your AWS costs.
   - To delete an existing tag, choose **Remove** beside the entry you want to remove.
6. Choose **Save**.

Deleting exports

You can use the **Data Exports** page in the AWS Billing and Cost Management console to delete your exports.

To delete an export

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Data Exports**.
3. From your list of exports, choose the name of the export that you want to delete.
4. On the **Export details** page, choose **Delete**.
5. Choose **Delete** once more to confirm that you want to delete the export.

> **Note**
>
> This procedure deletes your export from Data Exports. However, it doesn't delete the objects stored in your Amazon S3 bucket.
> For a Cost and Usage Dashboard, the above procedure deletes the Cost and Usage Dashboard from Data Exports. However, it doesn't delete the objects stored in your S3 bucket, QuickSight dashboard, and additional QuickSight resources. To delete your Cost and Usage Dashboard from QuickSight, see Deleting an Amazon QuickSight dashboard. When you delete an Amazon QuickSight dashboard, the dashboard is permanently removed from your account and all folders the dashboard was a part of. You'll no longer be able to access the deleted dashboard. You can only delete dashboards that you own or co-own.

Using Data Exports with AWS Organizations

Data Exports can work with AWS Organizations so that management accounts can generate exports with data for all accounts in your organization. Member accounts can also create data exports, but these exports only contain the billing and cost management data for that specific member account. The settings that control whether the management account receives data for all member accounts varies across the Data Exports tables. Refer to the following sections for information about how it is determined whether to include member account data for each table.

- Cost and usage report 2.0 (CUR 2.0)
- Cost optimization recommendations (from Cost Optimization Hub)
- FOCUS 1.0 with AWS columns
- Cost and usage dashboard
- Carbon emissions

The IAM policies that allow or restrict the ability to create an export are the same for both management and member accounts.

If you are an administrator of an AWS Organizations management account and you don’t want member accounts to create an export, you can apply a service control policy (SCP) that prevents member accounts from creating exports. While the SCP prevents member accounts from creating new exports, it doesn’t delete previously created exports.

> **Note**
>
> SCPs apply only to member accounts. To prevent a management account from creating an export, modify the IAM policies attached to the user roles in the management account.

Data Exports table dictionary

Data Exports provides several tables that you can query when creating an export.

The following topics describe the schema for each available table with definitions for each column present.

Topics

Cost and Usage Report (CUR) 2.0

The CUR 2.0 table provides the same information as Cost and Usage Reports (CUR) with a few improvements.

Cost and Usage Reports 2.0 provides the following improvements over Cost and Usage Reports:

- **Consistent schema**: CUR 2.0 contains a fixed set of columns, whereas the columns included for CUR can vary monthly depending on your usage of AWS services, cost categories, and resource tags.
- **Nested data**: CUR 2.0 reduces data sparsity by collapsing certain columns from CUR into individual columns with key-value pairs of the collapsed columns. The nested keys can optionally be queried in Data Exports as separate columns to match the original CUR schema and data.
- **Additional columns**: CUR 2.0 contains additional columns: **bill_payer_account_name**, **line_item_usage_account_name**, **capacity_reservation_capacity_reservation_arn**, **capacity_reservation_capacity_reservation_status** and **capacity_reservation_capacity_reservation_type**.

The SQL table name for CUR 2.0 is `COST_AND_USAGE_REPORT`.

Table configurations

Table configurations are user-controlled properties that a user can set to change the data or schema of a table before it's queried in Data Exports. The table configurations are saved as a JSON statement and are either specified through user input in the AWS SDK/CLI or user selections in the console.

CUR 2.0 has the following table configurations:

Configuration name	Description	Valid values
TIME_GRANULARITY	This configuration changes the cost and usage line items in the CUR 2.0 table to have different time granularities. For example, selecting "HOURLY" will make all line items represent a single hour of usage.	HOURLY, DAILY, MONTHLY
INCLUDE_RESOURCES	This configuration changes the cost and usage line items in the CUR 2.0 table to have resource-level granularity and adds the "line_item_resource_id" column to the table schema. Enabling this configuration causes the CUR 2.0 table to have a line item for each resource that incurred usage for a given service, instead of showing combined total usage for that service.	TRUE, FALSE

Configuration name	Description	Valid values
	Enabling this configuration can greatly increase the number of rows, and also the file size of your export.	
INCLUDE_SPLIT_COST _ALLOCATION_DATA	This configuration adds split cost allocation data and columns (split_line_item_*) to the CUR 2.0 table. This data indicates how the usage of certain AWS resources can be allocated to different business units or teams. Enabling this configuration can add additional rows and columns which show how an EC2 instance can be allocated to different containers running in that instance. For more information, see Understanding split cost allocation data.	TRUE, FALSE

Configuration name	Description	Valid values
INCLUDE_CAPACITY_RESERVATION_DATA	**Note:** This configuration only adds data in the new columns starting November 1, 2025. Enabling this configuration changes the cost and usage line items in the CUR 2.0 table to have resource-level granularity when an instance usage is split across multiple capacity reservations or used partially in a capacity reservationin an hour. This also adds 3 new columns to the table schema, which show how an EC2 instance is launched in a capacity reservation.	TRUE, FALSE

Configuration name	Description	Valid values
INCLUDE_MANUAL_DISCOUNT_COMPATIBILITY	**Note:** This configuration only applies to AWS customers who have onboarded to the Discount Automation program where discounts are computed automatically. This configuration changes the discounts in the CUR 2.0 table to appear as when they were added "manually" to the CUR, usually as separate line items, and removes two columns from the schema ("discount" and "total_discount").	TRUE, FALSE

AWS Organizations support

The CUR 2.0 table inherits the settings you made in the consolidated billing feature in AWS Organizations. When consolidated billing is enabled, there are different behaviors for management and member accounts. If you're using a management account, your CUR 2.0 table includes cost and usage data for the management account and all member accounts in your organization. If you're using a member account, your CUR 2.0 table only includes cost and usage data for that member account.

After joining an organization, a member account can only export data for the time that the account has been a member of the organization. For example, let's say that a member account leaves organization A and joins organization B on the 15th of the month. Then, the member account creates an export. Because the member account created an export after joining organization B, the member account's export of CUR 2.0 for the month only includes cost and usage data for the time that the account has been a member of organization B.

After a member account joins a new organization, the member account's cost and usage data is recorded in the new organization’s exports. This is the same outcome for a management account that converts to a member account and joins a new organization.

When a member account leaves an organization or converts to a standalone account, the member account can still access previous exports as long as they have permissions to the Amazon S3 bucket where the previous exports are stored.

For more information, see Consolidated billing for AWS Organizations in the *AWS Billing User Guide*.

CUR 2.0 column groups

There are 125 possible columns in the CUR 2.0 table, grouped as follows:

- **Bill:** Data about your bill for the billing period.
- **Cost category:** Data about cost categories that apply to the line item.
- **Capacity reservation:** Data about capacity reservation that applies to the line item.
- **Discount:** Data about any discounts you are receiving.
- **Identity:** Data to identify a line item.
- **Line item:** Data about cost, usage, type of usage, pricing rates, product name, and more.
- **Pricing:** Data about the pricing for a line item.
- **Product:** Data about the product that is being charged in the line item.
- **Reservation:** Data about a reservation that applies to the line item.
- **Resource tags:** Data about resource tags that apply to the line item.
- **Savings plan:** Data about savings plans that apply to the line item.
- **Split line item:** Data about split cost allocation for another line item.
- **Capacity Reservation:** Data about capacity reservation that applies to the line item.
- **Tags:** Data about user, account, cost category and resource tags.

Bill columns

Bill columns contain data about your bill for the billing period.

Column name	Description	Data type
bill_bill_type	The type of bill that this report covers. There are three bill types: - **Anniversary:** Line items for services that you used during the month. - **Purchase:** Line items for upfront service fees. - **Refund:** Line items for refunds.	string
bill_billing_entity	Helps you identify whether your invoices or transactions are for AWS Marketplace or for purchases of other AWS services.	string
bill_billing_period_end_date	The end date of the billing period that is covered by this report, in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ` .	timestamp
bill_billing_period_start_date	The start date of the billing period that is covered by this report, in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ` .	timestamp
bill_invoice_id	The ID associated with a specific line item. Until	string

Column name	Description	Data type
	the report is final, the `InvoiceId` is blank.	
bill_invoicing_entity	The AWS entity that issues the invoice.	string
bill_payer_account_id	The account ID of the paying account. For an organizat ion in AWS Organizations, this is the account ID of the management account.	string
bill_payer_account_name	The account name of the paying account. For an organization in AWS Organizations, this is the name of the management account.	string

Cost category columns

Cost category columns contain data about cost categories that apply to the line item. Note that you don’t need to select this column if you selected the Tags column because resource tags are also included under the Tags column.

Column name	Description	Data type
cost_category	A map column containing key-value pairs of the cost categories and their values for a given line item. These keys and values are populated based on the categorization	map <string, string>

Column name	Description	Data type
	rules you create in the cost categories feature. A cost category key only appears in the map column if it has a value that applies to the specific line item. The keys of this column can be queried as individual columns using the dot operator. For more information, see Data query.	

Discount columns

Discount columns contain data about any discounts you are receiving.

Column name	Description	Data type
discount	**Table configuration:** Removed by: INCLUDE MANUAL DISCOUNT COMPATIBILITY A "struct" column containing key-value pairs of any specific discounts that apply to this line item. The keys correspond to a discount type and the values correspond to either the discount value or other information. The values in this column are either data	map <string, double>

Column name	Description	Data type
	type "numeric" or "string" depending on the specific key. The keys of this column can be queried as individua l columns by using the dot operator. For more informati on, see Data query. This column is not available when "Manual discount compatibility" is enabled. When it's enabled, discounts are populated as separate line items and not in this column.	

Column name	Description	Data type
discount_bundled_discount	The bundled discount applied to the line item. A bundled discount is a usage-based discount that provides free or discounted usage of a service or feature based on the usage of another service or feature. As of August 2025, bundled discounts are applied using an "Owner-first approach" where discounts are first applied to the account that generates the source usage. Within the source account, discounts are applied based on the following sequence: • Highest per-unit price • Earlier usage timestamps • Larger usage amounts when other factors are equal Any remaining discounts are distributed across other accounts in the Consolidated Billing Family (CBF) based on the following sequence: • Highest per-unit price	double

Column name	Description	Data type
	- Ascending order of account Id - Earlier usage timestamps - Larger usage amounts Examples of bundled discounts include: - If you use AWS Shield Advanced, then you don't have to pay for AWS WAF separately. AWS WAF usage is bundled with AWS Shield Advanced. For more information about AWS Shield Advanced, see Amazon CloudFront pricing. - If you create a NAT gateway with AWS Network Firewall, then the standard NAT gateway processing and per-hour usage charges are waived on a one-to-one basis with the firewall's processing per GB and usage hours. For more information, see AWS Network Firewall pricing. - With Amazon Interactive Video Service (IVS) Chat,	

Column name	Description	Data type
	for every hour of video input sent, you get 2,700 sent messages and 270,000 delivered messages at no additional cost. For more information, see Amazon Interactive Video Service pricing.	
discount_total_discount	**Table configuration:** Removed by: `INCLUDE MANUAL DISCOUNT COMPATIBILITY` The sum of all the discount columns for the correspon ding line item. This column is not available when "Manual discount compatibility" is enabled. When it's enabled, discounts are populated as separate line items and not in this column.	double

Identity columns

Identity columns contain data to identify a line item.

Column name	Description	Data type
identity_line_item_id	This field is generated for each line item and is unique in a given partition. This	string

Column name	Description	Data type
	does not guarantee that the field will be unique across an entire delivery (that is, all partitions in an update) of the AWS CUR. The line item ID isn't consistent between different Cost and Usage Reports and can't be used to identify the same line item across different reports.	
identity_time_interval	The time interval that this line item applies to, in the following format: `YYYY-MM-DDTHH:mm:ssZ/YYYY-MM-DDTHH:mm:ssZ` . The time interval is in UTC and can either be daily or hourly, depending on the report granularity.	string

Line item columns

Line item columns contain data about cost, usage, type of usage, pricing rates, product name, and more.

Column name	Description	Data type
line_item_usage_account_name	The name of the account that used this line item. For organizations, this can be either the management account or a member account. You can	string

Column name	Description	Data type
	use this field to track costs or usage by account.	
line_item_availability_zone	The Availability Zone that hosts this line item. For example, `us-east-1a` or `us-east-1b` .	string
line_item_blended_cost	The `BlendedRate` multiplied by the `UsageAmount` . **BlendedCost** is blank for line items that have a **LineItemType** of **Discount**. Discounts are calculated using only the unblended cost of a member account, aggregated by member account and SKU. As a result, **BlendedCost** is not available for discounts.	double

Column name	Description	Data type
line_item_blended_rate	The `BlendedRate` is the average cost incurred for each SKU across an organization. For example, the Amazon S3 blended rates are the total cost of storage divided by the amount of data stored per month. For accounts with RIs, the blended rates are calculated as the average costs of the RIs and the On-Demand Instances. Blended rates are calculated at the management account level, and used to allocate costs to each member account. For more information, see Blended Rates and Costs in the *AWS Billing User Guide*.	string
line_item_currency_code	The currency that this line item is shown in. All AWS customers are billed in US dollars by default. To change your billing currency, see Changing which currency you use to pay your bill in the *AWS Billing User Guide*.	string

Column name	Description	Data type
line_item_legal_entity	The Seller of Record of a specific product or service. In most cases, the invoicing entity and legal entity are the same. The values might differ for third-party AWS Marketplace transactions. Possible values include: • **Amazon Web Services, Inc.**: The entity that sells AWS services. • **Amazon Web Services India Private Limited**: The local Indian entity that acts as a reseller for AWS services in India.	string

Column name	Description	Data type
line_item_line_item_description	The description of the line item type. For example, the description of a usage line item summarizes the type of usage incurred during a specific time period. For size-flexible RIs, the description corresponds to the RI the benefit was applied to. For example, if a line item corresponds to a `t2.micro` and a `t2.small` RI was applied to the usage, the **line_item_line_item_description** displays `t2.small`. The description for a usage line item with an RI discount contains the pricing plan covered by the line item.	string

Column name	Description	Data type
line_item_line_item_type	The type of charge covered by this line item. The possible types are as follows: • **BundledDiscount**: A usage-based discount that provides free or discounted usage of a service or feature based on the usage of another service or feature. • **Credit**: Any credits that AWS applied to your bill. See the **Description** column for details. AWS might update reports after they've been finalized, if AWS applies a credit to your account for the month after finalizing your bill. • **Discount**: Any discounts that AWS applied to your usage. This specific line item name may vary and require parsing based on the discount. For more information, refer to the **lineItem/LineItemDescription** column. •	string

Column name	Description	Data type
	DiscountedUsage: The rate of any instances for which you had Reserved Instance (RI) benefits. • **Fee**: Any upfront annual fee that you paid for subscriptions. For example, the upfront fee that you paid for an **All Upfront RI** or a **Partial Upfront RI**. • **Refund**: The negative charges that AWS refunded money for. Review the **Description** column for details. AWS might update reports after they've been finalized, if AWS applies a refund to your account for the month after finalizing your bill. • **RIFee**: The monthly recurring fee for subscriptions. For example, the recurring fee for **Partial Upfront RIs**, **No Upfront RIs**, and **All Upfronts** that you pay every month. Although the RIFee might be $0 for all upfront reservations, this line is still populated for those r	

Column name	Description	Data type
	eservation types to provide other columns such as **reservation/AmortizedUpfrontFeeForBillingPeriod** and **reservation/ReservationARN**. • **Tax**: Any taxes that AWS applied to your bill. For example, VAT or US sales tax. • **Usage**: Any usage that is charged at On-Demand Instance rates. • **SavingsPlanUpfrontFee**: Any one-time upfront fee from your purchase of an All Upfront or Partial Upfront Savings Plan. • **SavingsPlanRecurringFee**: Any recurring hourly charges that correspond with your No Upfront or Partial Upfront Savings Plan. The Savings Plan recurring fee is initially added to your bill on the day that you purchase a No Upfront or Partial Upfront Savings Plan. After the initial purchase, AWS adds the recurring fee to the	

Column name	Description	Data type
	first day of each billing period thereafter. - **SavingsPlanCoveredUsage**: Any On-Demand cost that is covered by your Savings Plan. Savings Plan covered usage line items are offset by the corresponding Savings Plan negation items. - **SavingsPlanNegation –** Any offset cost through your Savings Plan benefit that's associated with the corresponding Savings Plan covered usage item.	
line_item_net_unblended_cost	The actual after-discount cost that you're paying for the line item. This column is only included in your report when your account has a discount in the applicable billing period.	double
line_item_net_unblended_rate	The actual after-discount rate that you're paying for the line item. This column is only included in your report when your account has a discount in the applicable billing period.	string

Column name	Description	Data type
line_item_normalization_factor	As long as the instance has shared tenancy, AWS can apply all Regional Linux or Unix Amazon EC2 and Amazon RDS RI discounts to all instance sizes in an instance family and AWS Region. This also applies to RI discounts for member accounts in an organization. All new and existing Amazon EC2 and Amazon RDS size-flexible RIs are sized according to a normalization factor, based on the instance size.	double
line_item_normalized_usage_amount	The amount of usage that you incurred, in normalized units, for size-flexible RIs. The **NormalizedUsageAmount** is equal to **UsageAmount** multiplied by **NormalizationFactor**.	double
line_item_operation	The specific AWS operation covered by this line item. This describes the specific usage of the line item. For example, a value of `RunInstances` indicates the operation of an Amazon EC2 instance.	string

Column name	Description	Data type
line_item_product_code	The code of the product measured. For example, Amazon EC2 is the product code for Amazon Elastic Compute Cloud.	string
line_item_resource_id	**Table configuration:** Added by: INCLUDE RESOURCES (Optional) If you chose to include individual resource IDs in your report, this column contains the ID of the resource that you provisioned. For example, an Amazon S3 storage bucket, an Amazon EC2 compute instance, or an Amazon RDS database can each have a resource ID. This field is blank for usage types that aren't associate d with an instantiated host, such as data transfers and API requests, and line item types such as discounts, credits, and taxes.	string
line_item_tax_type	The type of tax that AWS applied to this line item.	string
line_item_unblended_cost	The `UnblendedCost` is the `UnblendedRate` multiplied by the `UsageAmount` .	double

Column name	Description	Data type
line_item_unblended_rate	In consolidated billing for accounts using AWS Organizations, the unblended rate is the rate associated with an individual account's service usage. For Amazon EC2 and Amazon RDS line items that have an RI discount applied to them, the `UnblendedRate` is `0`. Line items with an RI discount have a `LineItemType` of `DiscountedUsage` .	string
line_item_usage_account_id	The account ID of the account that used this line item. For organizations, this can be either the management account or a member account. You can use this field to track costs or usage by account.	string

Column name	Description	Data type
line_item_usage_amount	The amount of usage that you incurred during the specified time period. For size-flexible Reserved Instances, use the **reservation/TotalReservedUnits** column instead. Certain subscription charges will have a **UsageAmount** of `0`.	double
line_item_usage_end_date	The end date and time for the corresponding line item in UTC, exclusive. The format is `YYYY-MM-DDTHH:mm:ssZ` .	timestamp
line_item_usage_start_date	The start date and time for the line item in UTC, inclusive . The format is `YYYY-MM-DDTHH:mm:ssZ` .	timestamp
line_item_usage_type	The usage details of the line item. For example, `USW2-BoxUsage:m2.2xlarge` describes an M2 High Memory Double Extra Large instance in the US West (Oregon) Region.	string

Column name	Description	Data type
line_item_user_identifier	The Identity Access Management (IAM) Identity Center identifier of a workforce user. The monthly flat-rate subscription and on-demand charges are calculate d for the user identified by this identifier.	string

Pricing columns

Pricing columns contain data about the pricing for a line item.

Column name	Description	Data type
pricing_currency	The currency that the pricing data is shown in.	string
pricing_lease_contract_length	The length of time that your RI is reserved for.	string
pricing_offering_class	The offering class of the Reserved Instance.	string
pricing_public_on_demand_cost	The total cost for the line item based on public On-Demand Instance rates. If you have SKUs with multiple On-Demand public costs, the equivalent cost for the highest tier is displayed. For	double

Column name	Description	Data type
	example, services offering free-tiers or tiered pricing.	
pricing_public_on_demand_rate	The public On-Demand Instance rate in this billing period for the specific line item of usage. If you have SKUs with multiple On-Demand public rates, the equivalent rate for the highest tier is displayed. For example, services offering free-tiers or tiered pricing.	string
pricing_purchase_option	How you chose to pay for this line item. Valid values are `All Upfront`, `Partial Upfront`, and `No Upfront`.	string
pricing_rate_code	A unique code for a product/offer/pricing-tier combination. The product and term combinations can have multiple price dimensions, such as a free tier, low-use tier, and high-use tier.	string
pricing_rate_id	The ID of the rate for a line item.	string
pricing_term	Whether your AWS usage is Reserved or On-Demand.	string

Column name	Description	Data type
pricing_unit	The pricing unit AWS used to calculate your usage cost. For example, the pricing unit for Amazon EC2 instance usage is in hours.	string

Product columns

Product columns contain data about the product that is being charged in the line item.

Column name	Description	Data type
product	A map column containing key-value pairs of multiple product attributes and their values for a given line item. A product attribute only appears in the map column if it has a value that applies to the specific line item. **Note** Any product column that appeared in legacy CUR, but is not part of the CUR 2.0 static schema, appears in this map column.	map <string, string>

Column name	Description	Data type
	The keys of this column can be queried as individual columns by using the dot operator. For more information, see Data query.	
product_comment	A comment regarding the product.	string
product_fee_code	The code that refers to the fee.	string
product_fee_description	The description for the product fee.	string
product_from_location	Describes the location where the usage originated from.	string
product_from_location_type	Describes the location type where the usage originated from.	string
product_from_region_code	Describes the source Region code for the AWS service.	string
product_instanceSKU	The SKU of the product instance	string

Column name	Description	Data type
product_instance_family	Describes your Amazon EC2 instance family. Amazon EC2 provides you with a large number of options across 10 different instance types, each with one or more size options, organized into distinct instance families optimized for different types of applications.	string
product_instance_type	Describes the instance type, size, and family, which define the CPU, networking, and storage capacity of your instance.	string
product_location	Describes the Region that your Amazon S3 bucket resides in.	string
product_location_type	Describes the endpoint of your task.	string
product_operation	Describes the specific AWS operation that this line item covers.	string
product_pricing_unit	The smallest billing unit for an AWS service. For example, 0.01c per API call.	string

Column name	Description	Data type
product_product_family	The category for the type of product.	string
product_region_code	A Region is a physical location around the world where data centers are clustered. AWS calls each group of logical data centers an Availability Zone (AZ). Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographical area. The Region code attribute has the same name as an AWS Region, and specifies where the AWS service is available.	string
product_sku	A unique code for a product. The SKU is created by combining the `ProductCode` , `UsageType` , and `Operation` . For size-flexible RIs, the SKU uses the instance that was used. For example, if you used a `t2.micro` instance and AWS applied a `t2.small` RI discount to the usage, the line item SKU is created with the `t2.micro`.	string

Column name	Description	Data type
product_servicecode	This identifies the specific AWS service to the customer as a unique short abbreviation.	string
product_to _location_type	Describes the destination location of the service usage.	string
product_to_location	Describes the location usage destination.	string
product_to_region_code	Describes the source Region code for the AWS service.	string
product_usagetype	Describes the usage details of the line item.	string

Reservation columns

Reservation columns contain data about a reservation that applies to the line item.

Column name	Description	Data type
reservation_amortized_upfront_cost_for_usage	The initial upfront payment for all upfront RIs and partial upfront RIs amortized for usage time. The value is equal to: `RIAmortizedUpfrontFeeForBillingPeriod` * `The normalized usage amount for Discounte`	double

Column name	Description	Data type
	`dUsage line items / The normalized usage amount for the RIFee`. Because there are no upfront payments for no upfront RIs, the value for a no upfront RI is `0`. We don't provide this value for Dedicated Host reservations at this time. The change will be made in a future update.	
reservation_amortized_upfront_fee_for_billing_period	Describes how much of the upfront fee for this reservation is costing you for the billing period. The initial upfront payment for all upfront RIs and partial upfront RIs, amortized over this month. Because there are no upfront fees for no upfront RIs, the value for no upfront RIs is `0`. We don't provide this value for Dedicated Host reservations at this time. The change will be made in a future update.	double
reservation_availability_zone	The Availability Zone of the resource that is associated with this line item.	string

Column name	Description	Data type
reservation_effective_cost	The sum of both the upfront and hourly rate of your RI, averaged into an effective hourly rate. `Effective Cost` is calculated by taking the `amortized UpfrontCostForUsage` and adding it to the `recurringFeeForUsage` .	double
reservation_end_time	The end date of the associated RI lease term.	string
reservation_modification_status	Shows whether the RI lease was modified or if it is unaltered. **Original:** The purchased RI was never modified. **System:** The purchased RI was modified using the console or API. **Manual:** The purchased RI was modified using AWS Support assistance. **ManualWithData:** The purchased RI was modified using AWS Support assistance, and AWS calculated estimates for the RI.	string

Column name	Description	Data type
reservation_net_amortized_upfront_cost_for_usage	The initial upfront payment for All Upfront RIs and Partial Upfront RIs amortized for usage time, if applicable. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_net_amortized_upfront_fee_for_billing_period	The cost of the reservation's upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_net_effective_cost	The sum of both the upfront fee and the hourly rate of your RI, averaged into an effective hourly rate. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_net_recurring_fee_for_usage	The after-discount cost of the recurring usage fee. This column is included in your report only when your account has a discount in the applicable billing period.	double

Column name	Description	Data type
reservation_net_unused_amortized_upfront_fee_for_billing_period	The net unused amortized upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_net_unused_recurring_fee	The recurring fees associated with unused reservation hours for Partial Upfront and No Upfront RIs after discounts. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_net_upfront_value	The upfront value of the RI with discounts applied. This column is included in your report only when your account has a discount in the applicable billing period.	double
reservation_normalized_units_per_reservation	The number of normalized units for each instance of a reservation subscription.	string

Column name	Description	Data type
reservation_number_of_reservations	The number of reservations that are covered by this subscription. For example, one RI subscription might have four associated RI reservations.	string
reservation_recurring_fee_for_usage	The recurring fee amortized for usage time, for partial upfront RIs and no upfront RIs. The value is equal to: `The unblended cost of the RIFee` * `The sum of the normalized usage amount of Usage line items` / `The normalized usage amount of the RIFee for size flexible Reserved Instances`. Because all upfront RIs don't have recurring fee payments greater than 0, the value for all upfront RIs is `0`.	double

Column name	Description	Data type
reservation_reservation_a_r_n	The Amazon Resource Name (ARN) of the RI that this line item benefited from. This is also called the "RI Lease ID". This is a unique identifier of this particular AWS Reserved Instance. The value string also contains the AWS service name and the Region where the RI was purchased.	string
reservation_start_time	The start date of the term of the associated Reserved Instance.	string
reservation_subscription_id	A unique identifier that maps a line item with the associate d offer. We recommend you use the RI ARN as your identifier of an AWS Reserved Instance, but both can be used.	string
reservation_total_reserved_normalized_units	The total number of reserved normalized units for all instances for a reservation subscription. AWS computes total normalized units by multiplying the `reservation/NormalizedUnitsPerReservation` with `reservation/NumberOfReservations`.	string

Column name	Description	Data type
reservation_total_reserved_units	`TotalReservedUnits` populates for both Fee and RIFee line items with distinct values.	string
reservation_units_per_reservation	`UnitsPerReservation` populates for both Fee and RIFee line items with distinct values.	string
reservation_unused_amortized_upfront_fee_for_billing_period	The amortized-upfront-fee-for-billing-period-column amortized portion of the initial upfront fee for all upfront RIs and partial upfront RIs. Because there are no upfront payments for no upfront RIs, the value for no upfront RIs is `0`. We don't provide this value for Dedicated Host reservations at this time. The change will be made in a future update.	double
reservation_unused_normalized_unit_quantity	The number of unused normalized units for a size-flexible Regional RI that you didn't use during this billing period.	double
reservation_unused_quantity	The number of RI hours that you didn't use during this billing period.	double

Column name	Description	Data type
reservation_unused_recurring_fee	The recurring fees associated with your unused reservation hours for partial upfront and no upfront RIs. Because all upfront RIs don't have recurring fees greater than 0, the value for All Upfront RIs is 0.	double
reservation_upfront_value	The upfront price paid for your AWS Reserved Instance. For no upfront RIs, this value is 0.	double

Resource tags columns

Resource tags columns contain data about resource tags that apply to the line item. Note that you don’t need to select this column if you selected the Tags column because resource tags are also included under the Tags column.

Column name	Description	Data type
resource_tags	A map column containing key-value pairs of resource tags and their values for a given line item. The values in this column are all of data type "string". Resource tag keys only appear in this column if they've been enabled as cost allocation tags in the Billing	map <string, string>

Column name	Description	Data type
	console. After being enabled, a particular key only appears in the map column if it has a value that applies to the specific line item. The keys of this column can be queried as individual columns using the dot operator. For more information, see Data query.	

Savings plan columns

Saving Plan columns contain data about savings plans that apply to the line item.

Column name	Description	Data type
savings_plan_amortized_upfront_commitment_for_billing_period	The amount of upfront fee a Savings Plan subscription is costing you for the billing period. The initial upfront payment for **All Upfront Savings Plan** and **Partial Upfront Savings Plan** amortized over the current month. For **No Upfront Savings Plan**, the value is `0`.	double
savings_plan_end_time	The expiration date for the Savings Plan agreement.	string
savings_plan_instance_type_family		string

Column name	Description	Data type
	The instance family that is associated with the specified usage.	
savings_plan_net_amortized_upfront_commitment_for_billing_period	The cost of a Savings Plan subscription upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.	double
savings_plan_net_recurring_commitment_for_billing_period	The net unblended cost of the Savings Plan fee. This column is included in your report only when your account has a discount in the applicable billing period.	double
savings_plan_net_savings_plan_effective_cost	The effective cost for Savings Plans, which is your usage divided by the fees. This column is included in your report only when your account has a discount in the applicable billing period.	double
savings_plan_offering_type	Describes the type of Savings Plan purchased.	string
savings_plan_payment_option	The payment options available for your Savings Plan.	string

Column name	Description	Data type
savings_plan_purchase_term	Describes the duration, or term, of the Savings Plan.	string
savings_plan_recurring_commitment_for_billing_period	The monthly recurring fee for your Savings Plan subscriptions. For example, the recurring monthly fee for a **Partial Upfront Savings Plan** or **No Upfront Savings Plan**.	double
savings_plan_region	The AWS Region (geographic area) that hosts your AWS services. You can use this field to analyze spend across a particular AWS Region.	string
savings_plan_savings_plan_a_r_n	The unique Savings Plan identifier.	string
savings_plan_savings_plan_effective_cost	The proportion of the Savings Plan monthly commitment amount (upfront and recurring) that is allocated to each usage line.	double
savings_plan_savings_plan_rate	The Savings Plan rate for the usage.	double
savings_plan_start_time	The start date of the Savings Plan agreement.	string

Column name	Description	Data type
savings_plan_total _commitment_to_date	The total amortized upfront commitment and recurring commitment to date, for that hour.	double
savings_plan_used_ commitment	The total dollar amount of the Savings Plan commitment used. (SavingsPlanRate multiplied by usage)	double

Split line item columns

Columns under the **split_line_item** header are fields that appear in Data Exports if you've opted in to the split cost allocation data feature. For more information, see Understanding split cost allocation data. The feature is limited to Amazon ECS (including Fargate), AWS Batch, and Amazon EKS only.

Column name	Description	Data type
split_line_item_actual_usage	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The usage for vCPU, memory, or accelerator resources (based on lineItem/UsageType) you incurred for the specified time period for the Amazon ECS task or Kubernetes pod.	double
split_line_item_net_split_cost		double

Column name	Description	Data type
	Table configuration: Added by: INCLUDE SPLIT COST ALLOCATION DATA The effective cost for Amazon ECS tasks or Kubernetes pods after all discounts have been applied. This column is included in your report only when your account has a discount in the applicable billing period.	
split_line_item_net_unused_cost	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The effective unused cost for Amazon ECS tasks or Kubernetes pods after all discounts have been applied. This column is included in your report only when your account has a discount in the applicable billing period.	double

Column name	Description	Data type
split_line_item_parent_resource_id	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The resource ID of the parent EC2 instance associated with the Amazon ECS task or Amazon EKS pod (referenced in the lineItem/ResourceId column). The parent resource ID implies that the ECS task or Kubernetes pod workload for the specified time period ran on the parent EC2 instance. This applies only for Amazon ECS tasks or Kubernetes pods with EC2 launch type.	string
split_line_item_public_on_demand_split_cost	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod based on public On-Demand Instance rates (referenced in the pricing/publicOnDemandRate column).	double

Column name	Description	Data type
split_line_item_public_on_demand_unused_cost	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The unused cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod based on public On-Demand Instance rates. Unused costs are costs associated with resources (CPU or memory) on the EC2 instance (referenced in the splitLineItem/ParentResourceIdcolumn) that were not utilized for the specified time period.	double
split_line_item_reserved_usage	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The usage for vCPU, memory, or accelerator resources (based on lineItem/UsageType) that you configured for the specified time period for the Amazon ECS task or Kubernetes pod.	double

Column name	Description	Data type
split_line_item_split_cost	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The cost for vCPU or memory (based on lineItem/ UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod. This includes amortized costs if the EC2 instance (referenced in the splitLineItem/ parentResourceId column) has upfront or partial upfront charges for reservations or Savings Plans.	double
split_line_item_split_usage	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The usage for vCPU or memory (based on lineItem/ UsageType) allocated for the specified time period to the Amazon ECS task or Kubernetes pod. This is defined as the maximum usage of splitLineItem/ReservedUsage or splitLineItem/ActualUsage.	double

Column name	Description	Data type
split_line_item_split_usage _ratio	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The ratio of vCPU, memory, or accelerator resources (based on lineItem/UsageType) allocated to the Amazon ECS task or Kubernetes pod compared to the overall CPU, memory, or accelerator resources available on the EC2 instance (referenced in the splitLineItem/ParentResourceId column).	double

Column name	Description	Data type
split_line_item_unused_cost	**Table configuration:** Added by: INCLUDE SPLIT COST ALLOCATION DATA The unused cost for vCPU, memory, or accelerator resources (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod. Unused costs are costs associated with resources (CPU, memory, or accelerator resources) on the EC2 instance (referenced in the splitLineItem/ParentResourceId column) that were not utilized for the specified time period. This includes amortized costs if the EC2 instance (splitLineItem/parentResourceId) has upfront or partial upfront charges for reservations or Savings Plans.	double

Tags Column

Tags column contains data about user, account, cost category and resource tags that apply to the line item. If you select this column, you need not select Resource tags and Cost category columns in your CUR 2.0.

Column name	Description	Data type
tags		map <string, string>

Column name	Description	Data type
	A map column containing key-value pairs of all tags and their values for a given line item. The values in this column are all of data type "string". Tag keys only appear in this column if they've been enabled as cost allocation tags in the Billing console. After being enabled, a particular key only appears in the map column if it has a value that applies to the specific line item. The keys of this column can be queried as individual columns using the dot operator. For more information, see Data query.	

Understanding Tag Prefixes and Overlapping Tag Keys

When using cost allocation tags alongside other AWS tagging mechanisms, you may encounter situations where the same tag key (such as "department" or "aws:createdBy") appears across different tagging contexts. AWS automatically prefixes these tags to prevent conflicts and ensure accurate cost allocation.

Tag Prefix Types

AWS uses the following prefixes to distinguish between different tag sources:

1. resourceTags/#- Tags applied directly to AWS resources.
2. userAttribute/#- User attributes imported from IAM Identity Center.

3. accountTag/#- Tags applied at the AWS account level.
4. costCategory/#- Tags derived from AWS Cost Categories.

Example: Overlapping Tag Keys

Consider a scenario where multiple tagging mechanisms use the same tag keys. Here's how AWS handles them:

```
{
  "resourceTags/department": "teamA",
  "resourceTags/appName": "app1",
  "userAttribute/Department": "teamB",
  "accountTag/department": "teamC",
  "accountTag/appName": "app3",
  "costCategory/department": "teamD"
}
```

In this example:

- The resource is tagged with department "teamA" at the resource level
- The user who accessed the resource belongs to the "teamB" department in IAM Identity Center
- The AWS account has an account-level tag indicating the "teamC" department
- A cost category rule has assigned this cost to the "teamD" department

Each tag is preserved with its unique prefix, allowing you to analyze costs from multiple perspectives simultaneously. This enables you to:

- Track which resources belong to which teams (`resourceTags/department`)
- Understand which users from which departments are consuming resources (`userAttribute/Department`)
- Allocate costs based on account ownership (`accountTag/department`)
- Apply custom business logic through cost categories (`costCategory/department`)

Capacity reservation columns

Capacity reservation columns contain data about capacity reservations that apply to the line item.

Column name	Description	Data Type	Nullability	Properties
capacity_reservation_capacity_reservation_arn	**Table configuration:** Added by: INCLUDE CAPACITY RESERVATION DATA The capacity reservation ARN represents the unique identifier of the capacity reservation	String	Nullable	This field is not null when a charge is related to a capacity reservation This field is not null when a charge represents the unused portion of a capacity reservation This field is null when a charge is not related to a capacity reservation
capacity_reservation_capacity_reservation_status	**Table configuration:** Added by: INCLUDE CAPACITY RESERVATION DATA Indicates whether the line item represents either the consumption of the capacity reservation	String	Nullable	This field is null when capacity_reservation_capacity_reservation_arn is null This field is not null when capacity_reservation_capacity_reservation_arn is

Column name	Description	Data Type	Nullability	Properties
	identified in the capacity _reservation_capacity_reservation_arn column or when the capacity reservation is unused or when the capacity reservation is reserved.			not null and line_item _line_item_type is **Usage** or **SavingsPlannedCoveredUsage** or **DiscountedUsage** This field contains one of the allowed values – Reserved, Used or Unused

Column name	Description	Data Type	Nullability	Properties
capacity_reservation_capacity_reservation_type	**Table configuration:** Added by: INCLUDE CAPACITY RESERVATION DATA The capacity reservation type field represents the type of capacity reservation purchased. Currently, there are 2 types ODCR and EC2 Capacity Blocks for ML	String	Nullable	This field is null when capacity_reservation_capacity_reservation_arn is null This field is not null when capacity_reservation_capacity_reservation_arn is not null and line_item_line_item_type is **Usage** or **SavingsPlannedCoveredUsage** or **DiscountedUsage** This field contains one of the allowed values – ODCR or EC2 Capacity Blocks for ML

Cost optimization recommendations (from Cost Optimization Hub)

The cost optimization recommendations table contains your cost optimization recommendations from Cost Optimization Hub. Cost Optimization Hub recommendations are consolidated from AWS Compute Optimizer and consist of over 15 types of optimizations, such as resource rightsizing, idle resource deletion, Savings Plans, and Reserved Instances. For more detailed information, see Cost Optimization Hub in the *AWS Cost Management User Guide*.

The SQL table name for cost optimization recommendations is `COST_OPTIMIZATION_RECOMMENDATIONS`.

Table configurations

Table configurations are user-controlled properties that a user can set to change the data or schema of a table before it's queried in Data Exports. The table configurations are saved as a JSON statement and are either specified through user input in the AWS SDK/CLI or user selections in the console.

Cost optimization recommendations has the following table configurations:

Configuration name	Description	Valid values
INCLUDE_ALL_RECOMMENDATIONS	When set to "FALSE", only the highest savings value recommendation is kept in the table from any set of recommendations that are incompatible with one another. For example, only "Terminate instance" is kept from a recommendation to terminate an instance and a recommendation to rightsize the same instance.	TRUE, FALSE

Configuration name	Description	Valid values
	When set to "TRUE", all recommendations are kept in the table. This is also known as **Group related recommendations** in the Cost Optimization Hub console. For more information, see Grouping related recommendations in the *AWS Cost Management User Guide*.	

Configuration name	Description	Valid values
FILTER	This allows you to filter recommendations based on different recommendation attributes. Filters are applied to the table before the savings deduplication algorithm is applied. You can filter using the same parameters as in the Cost Optimization Hub console. For more information, see [Prioritizing your cost optimization opportunities]() in the *AWS Cost Management User Guide*. Filter statements are provided for this configuration using the same JSON structure that is used in the `filter` parameter in the Cost Optimization Hub `list-recommendations` API . It must be provided as a JSON string. For details, see the [`list-recommendations` structure]().	Any JSON string that is valid for the Cost Optimization Hub `list-recommendations` API.

Service-linked role

A service-linked role for Data Exports is required to create an export of the cost optimization recommendations table. For information on how to create the service-linked role, see [Service-linked roles for Data Exports]() in the *AWS Cost Management User Guide*.

AWS Organizations support

Cost Optimization Hub integrates with AWS Organizations to control whether a management account can see member account recommendations in Cost Optimization Hub. For more information, see Getting started with Cost Optimization Hub in the *AWS Cost Management User Guide*.

The cost optimization recommendations table for a given AWS account inherits the same AWS Organizations settings you configured for Cost Optimization Hub. This means the content of the cost optimization recommendations table matches the recommendations that appear in Cost Optimization Hub for each AWS account.

Cost optimization recommendations columns

Column name	Description	Data type	Null value allowed
account_id	The account ID that the recommendation is for.	string	No
account_name	The account name that the recommend ation is for.	string	No
action_type	The type of action you can take by adopting the recommendation.	string	No
currency_code	The currency code used for the recommendation.	string	No
current_resource_d etails	The details for the resource in JSON string format.	string	Yes
current_resource_s ummary	A description of the current resource.	string	Yes

Column name	Description	Data type	Null value allowed
current_resource_type	The type of resource.	string	Yes
estimated_monthly_cost_after_discount	The estimated monthly cost of the current resource after discounts. For Reserved Instances and Savings Plans, it refers to the cost for eligible usage.	double	Yes
estimated_monthly_cost_before_discount	The estimated monthly cost of the current resource before discounts. For Reserved Instances and Savings Plans, it refers to the cost for eligible usage.	double	No
estimated_monthly_savings_after_discount	The estimated monthly savings amount for the recommendation after discounts.	double	Yes
estimated_monthly_savings_before_discount	The estimated monthly savings amount for the recommendation before discounts.	double	No

Column name	Description	Data type	Null value allowed
estimated_savings_percentage_after_discount	The estimated savings percentage after discounts relative to the total cost over the cost calculation lookback period.	double	Yes
estimated_savings_percentage_before_discount	The estimated savings percentage before discounts relative to the total cost over the cost calculation lookback period.	double	No
implementation_effort	The effort required to implement the recommendation.	string	No
last_refresh_timestamp	The time when the recommendation was last generated.	timestamp	No
recommendation_ID	The ID for the recommendation.	string	No
recommendation_lookback_period_in_days	The lookback period that's used to generate the recommendation.	integer	No
recommendation_source	The source of the recommendation.	string	No

Column name	Description	Data type	Null value allowed
recommended_resource_details	The details about the recommended resource in JSON string format.	string	Yes
recommended_resource_summary	A description of the recommended resource.	string	Yes
recommended_resource_type	The resource type of the recommendation.	string	Yes
region	The AWS Region of the resource.	string	Yes
resource_arn	The Amazon Resource Name (ARN) of the resource.	string	Yes
restart_needed	Whether or not implementing the recommendation requires a restart.	boolean	No
rollback_possible	Whether or not implementing the recommendation can be rolled back.	boolean	No
tags	A list of tags associated with the resource for which the recommendation exists.	map	Yes

FOCUS 1.2 with AWS columns

The FOCUS 1.2 with AWS columns table contains your cost and usage data formatted with FinOps Open Cost and Usage Specification (FOCUS) 1.0, along with three additional columns from AWS that contain proprietary billing data. These columns are **x_Discounts**, **x_Operation**, and **x_ServiceCode**. For more information about the FOCUS open-source specification, refer to the FOCUS website.

The SQL table name for FOCUS 1.2 with AWS columns is `FOCUS_1_2_AWS`

Table configurations

Table configurations are user-controlled properties that a user can set to change the data or schema of a table before it's queried in Data Exports. The table configurations are saved as a JSON statement and are either specified through user input in the AWS SDK/CLI or user selections in the console.

FOCUS 1.2 has the following table configurations:

Configuration name	Description	Valid values
TIME_GRANULARITY	This configuration changes the cost and usage line items in the FOCUS 1.2 table to have different time granularities. For example, selecting "HOURLY" will make all line items represent a single hour of usage.	HOURLY, DAILY, MONTHLY

AWS Organizations support

The FOCUS 1.2 with AWS columns table inherits the settings you made in the consolidated billing feature in AWS Organizations. When consolidated billing is enabled, there are different behaviors for management and member accounts. If you're using a management account, your FOCUS 1.2 with AWS columns table includes cost and usage data for the management account and all member accounts in your organization. If you're using a member account, your FOCUS 1.2 with AWS columns table only includes cost and usage data for that member account.

After joining an organization, a member account can only export data for the time that the account has been a member of the organization. For example, let's say that a member account leaves organization A and joins organization B on the 15th of the month. Then, the member account creates an export. Because the member account created an export after joining organization B, the member account's export of FOCUS 1.2 with AWS columns for the month only includes cost and usage data for the time that the account has been a member of organization B.

FOCUS 1.2 with AWS columns

Column name	Description	Data type	Null value allowed
AvailabilityZone	A provider-assigned identifier for a physically separated and isolated area within a Region that provides high availability and fault tolerance.	string	Yes
BilledCost	A charge serving as the basis for invoicing, inclusive of all reduced rates and discounts while excluding the amortization of upfront charges (one-time or recurring).	double	No
BillingAccountId	The identifier assigned to a billing	string	No

Column name	Description	Data type	Null value allowed
	account by the provider.		
BillingAccountName	The display name assigned to a billing account.	string	Yes
BillingAccountType	A provider-assigned name to identify the type of billing account.	string	No
BillingCurrency	Represents the currency that a charge was billed in.	string	No
BillingPeriodEnd	The exclusive end bound of a billing period.	timestamp_milliseconds	No
BillingPeriodStart	The inclusive start bound of a billing period.	timestamp_milliseconds	No
CapacityReservationId	The identifier assigned to a capacity reservation by the provider.	string	Yes
CapacityReservationStatus	Indicates whether the charge represents either the consumption of a capacity reservation or when a capacity reservation is unused.	string	Yes

Column name	Description	Data type	Null value allowed
ChargeCategory	Represents the highest-level classification of a charge based on the nature of how it is billed.	string	No
ChargeClass	Indicates whether the row represents a correction to a previously invoiced billing period.	string	Yes
ChargeDescription	Self-contained summary of the charge's purpose and price.	string	Yes
ChargeFrequency	Indicates how often a charge will occur.	string	No
ChargePeriodEnd	The exclusive end bound of a charge period.	timestamp_milliseconds	No
ChargePeriodStart	The inclusive start bound of a charge period.	timestamp_milliseconds	No
CommitmentDiscountCategory	Indicates whether the commitment discount identified in the CommitmentDiscountId column is based on usage quantity or cost (aka "spend").	string	Yes

Column name	Description	Data type	Null value allowed
CommitmentDiscount Id	The identifier assigned to a commitment discount by the provider.	string	Yes
CommitmentDiscount Name	The display name assigned to a commitment discount.	string	Yes
CommitmentDiscount Quantity	The amount of a commitment discount purchased or accounted for in commitment discount related rows that is denominat ed in Commitment Discount Units.	double	Yes
CommitmentDiscount Status	Indicates whether the charge corresponds with the consumpti on of a commitmen t discount or the unused portion of the committed amount.	string	Yes
CommitmentDiscount Type	A provider-assigned identifier for the type of commitmen t discount applied to the row.	string	Yes

Column name	Description	Data type	Null value allowed
CommitmentDiscountUnit	The provider-specified measurement unit indicating how a provider measures the Commitment Discount Quantity of a commitment discount.	string	Yes
ConsumedQuantity	The volume of a metered SKU associated with a resource or service used, based on the Consumed Unit.	double	Yes
ConsumedUnit	Provider-specified measurement unit indicating how a provider measures usage of a metered SKU associated with a resource or service.	string	Yes
ContractedCost	Cost calculated by multiplying contracted unit price and the corresponding Pricing Quantity.	double	No

Column name	Description	Data type	Null value allowed
ContractedUnitPrice	The agreed-upon unit price for a single Pricing Unit of the associated SKU, inclusive of negotiated discounts, if present, while excluding negotiated commitment discounts or any other discounts.	double	Yes
EffectiveCost	The amortized cost of the charge after applying all reduced rates, discounts, and the applicable portion of relevant, prepaid purchases (one-time or recurring) that covered this charge.	double	No
InvoiceId	The provider-assigned identifier for an invoice encapsulating some or all charges in the corresponding billing period for a given billing account.	string	Yes

Column name	Description	Data type	Null value allowed
InvoiceIssuerName	The name of the entity responsible for invoicing for the resources or services consumed.	string	No
ListCost	Cost calculated by multiplying List Unit Price and the corresponding Pricing Quantity.	double	No
ListUnitPrice	The suggested provider-published unit price for a single Pricing Unit of the associated SKU, exclusive of any discounts.	double	Yes
PricingCategory	Describes the pricing model used for a charge at the time of use or purchase.	string	Yes
PricingCurrency	The national or virtual currency denomination that a resource or service was priced in.	string	Yes

Column name	Description	Data type	Null value allowed
PricingCurrencyContractedUnitPrice	The agreed-upon unit price for a single Pricing Unit of the associated SKU, inclusive of negotiated discounts, if present, while excluding negotiated commitment discounts or any other discounts, and expressed in Pricing Currency.	double	Yes
PricingCurrencyEffectiveCost	The cost of the charge after applying all reduced rates, discounts, and the applicable portion of relevant, prepaid purchases (one-time or recurring) that covered this charge, as denominated in Pricing Currency.	double	Yes
PricingCurrencyListUnitPrice	The suggested provider-published unit price for a single Pricing Unit of the associated SKU, exclusive of any discounts and expressed in Pricing Currency.	double	Yes

Column name	Description	Data type	Null value allowed
PricingQuantity	The volume of a given SKU associated with a resource or service used or purchased, based on the Pricing Unit.	double	Yes
PricingUnit	Provider-specified measurement unit for determining unit prices, indicating how the provider rates measured usage and purchase quantities after applying pricing rules like block pricing.	string	Yes
ProviderName	The name of the entity that made the resources or services available for purchase.	string	No
PublisherName	The name of the entity that produced the resources or services that were purchased.	string	No

Column name	Description	Data type	Null value allowed
RegionId	Provider-assigned identifier for an isolated geographic area where a resource is provisioned or a service is provided.	string	Yes
RegionName	The name of an isolated geographic area where a resource is provisioned or a service is provided.	string	Yes
ResourceId	Identifier assigned to a resource by the provider.	string	Yes
ResourceName	Display name assigned to a resource.	string	Yes
ResourceType	The kind of resource the charge applies to.	string	Yes
ServiceCategory	Highest-level classification of a service based on the core function of the service.	string	No

Column name	Description	Data type	Null value allowed
ServiceName	An offering that can be purchased from a provider (e.g., cloud virtual machine, SaaS database, professional services from a systems integrator).	string	No
ServiceSubcategory	Secondary classification of the Service Category for a service based on its core function.	string	No
SkuId	Provider-specified unique identifier that represents a specific SKU (e.g., a quantifiable good or service offering).	string	Yes
SkuMeter	Describes the functionality being metered or measured by a particular SKU in a charge.	string	Yes
SkuPriceId	A provider-specified unique identifier that represents a specific SKU Price associated with a resource or service used or purchased.	string	Yes

Column name	Description	Data type	Null value allowed
SkuPriceDetails	A set of properties of a SKU Price ID which are meaningful and common to all instances of that SKU Price ID.	map <string, string>	Yes
SubAccountId	An ID assigned to a grouping of resources or services, often used to manage access and/or cost.	string	Yes
SubAccountName	A name assigned to a grouping of resources or services, often used to manage access and/or cost.	string	Yes
SubAccountType	A provider-assigned name to identify the type of sub account.	string	Yes
Tags	The set of tags assigned to tag sources that account for potential provider-defined or user-defined tag evaluations.	map <string, string>	Yes
x_Discounts	A map column containing key-value pairs of any specific discounts that apply to this line item.	map <string, double>	Yes

Column name	Description	Data type	Null value allowed
x_Operation	The specific AWS operation covered by this line item. This describes the specific usage of the line item. For example, a value of RunInstances indicates the operation of an Amazon EC2 instance.	string	Yes
x_ServiceCode	The code of the product measured. For example, Amazon EC2 is the product code for Amazon Elastic Compute Cloud.	string	Yes

FOCUS 1.2 with AWS columns conformance gaps

The following table provides all of the conformance gaps that might exist in an export of the FOCUS 1.2 with AWS columns table. A particular conformance gap will not apply to your export if you're not receiving cost and usage data for the related scenario.

Conformance gap type	Affected column	FOCUS 1.2 requirement	Conformance gap description
Missing data	ContractedUnitPrice	ContractedUnitPrice **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	ContractedUnitPrice might be null for certain product offerings.

Conformance gap type	Affected column	FOCUS 1.2 requirement	Conformance gap description
	InvoiceIssuerName	InvoiceIssuerName **must not** be null.	InvoiceIssuerName might be null for certain charges.
	ListUnitPrice	ListUnitPrice **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	ListUnitPrice might be null for certain product offerings.
	PricingUnit	PricingUnit **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	PricingUnit might be null for certain product offerings.
	PublisherName	PublisherName **must not** be null.	PublisherName may be null for certain charges.
	SkuId	SkuId **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	SkuId might be null for certain product offerings.
	SkuPriceId	SkuPriceId **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	SkuPriceId might be null for certain line items when it should not be.

Conformance gap type	Affected column	FOCUS 1.2 requirement	Conformance gap description
	CapacityReservationStatus	CapacityReservationStatus **must not** be null when CapacityReservationId is not null and ChargeCategory is Usage	CapacityReservationStatus might be null for a capacity reservation in a reserved state, and it is neither "used" nor "unused."
Incorrect data	ConsumedQuantity	ConsumedQuantity is a required column for showing the amount of usage you actually used.	ConsumedQuantity will contain the amount of usage you were charged for. This means that ConsumedQuantity could be incorrect in situations where a minimum charge quantity applied for a particular service. For example, there is a 10MB minimum for an Athena query and a 10 minute minimum Glue crawler run. For these services, ConsumedQuantity will show the value that includes the minimum charged quantity.

Conformance gap type	Affected column	FOCUS 1.2 requirement	Conformance gap description
	SkuMeter	SkuMeter **must** be null when SkuId is null	SkuMeter might be null for certain product offerings.
	ConsumedUnit	ConsumedUnit nullability **must** match ConsumedQuantity	ConsumedUnit might be null for certain product offerings with ConsumedQuantity.

Migrating from FOCUS 1.0 to FOCUS 1.2

AWS Data Exports allows you to create exports of FOCUS 1.2 with AWS columns, which provides the same standardized cost and usage information as FOCUS 1.0 along with several enhancements for invoice reconciliation, capacity reservation tracking, and SaaS integration. However, FOCUS 1.2 introduces breaking changes that affect existing row counts and column values. Review these changes carefully before migrating.

FOCUS 1.2 provides the following improvements over FOCUS 1.0:

- Invoice reconciliation: FOCUS 1.2 includes an InvoiceID column that enables streamlined financial close processes and invoice reconciliation.
- Capacity reservation tracking: FOCUS 1.2 contains CapacityReservationId and CapacityReservationStatus columns that help you identify and track unused capacity reservations.
- Virtual currency support: FOCUS 1.2 includes new pricing currency columns (PricingCurrency, PricingCurrencyEffectiveCost, PricingCurrencyListUnitPrice, and PricingCurrencyContractedUnitPrice) that enable you to join with SaaS providers cost and usage data in FOCUS 1.2 format.

The following table outlines the differences between FOCUS 1.2 and FOCUS 1.0 in more detail:

Feature	FOCUS 1.2	FOCUS 1.0
Data schema	FOCUS 1.2 specification with 57 FOCUS columns + 3 AWS columns (60 total). For the complete column list, see *FOCUS 1.2 with AWS columns*.	FOCUS 1.0 specification with 43 FOCUS columns + 5 AWS columns (48 total). For the complete column list, see *FOCUS 1.0 with AWS columns*.
New columns	InvoiceIssuerId, CapacityReservationId, CapacityReservationStatus, CommitmentDiscountQuantity, CommitmentDiscountUnit, ServiceSubcategory, SkuMeter, SkuPriceDetails, PricingCurrency, PricingCurrencyEffectiveCost, PricingCurrencyListUnitPrice, PricingCurrencyContractedUnitPrice, BillingAccountType, SubAccountType	
Removed columns	x_UsageType (This is now the SkuMeter column) x_CostCategories (This is now included in the Tags column with the "aws:tags:CostCategory/" prefix)	
Row count	On Demand Capacity Reservation (ODCR) and EC2 Capacity Blocks for ML line items are now split into separate "Used" and "Unused" status. This increases the row count.	Does not provide capacity reservation status breakdown details.

Feature	FOCUS 1.2	FOCUS 1.0
PricingCategory column values	Commitment discount purchases, such as Reserved Instance and Savings Plan upfront fees, are now categorized as "Standard."	Commitment discount purchases categorized as "Committed".
ConsumedQuantity column nullability	Null when the Commitmen tDiscountStatus is "Unused".	Value is "0" when the CommitmentDiscountStatus is "Unused".
ConsumedUnit column nullability	Tied to ConsumedQuantity nullability (required to be null when ConsumedQuantity is null).	Independent nullability rules.
Tag column requirement	Include user-defined and provider-defined resource tags, and user-defined cost category tags.	Include only user-defined and provider-defined resource tags.
File delivery destination	S3 bucket	S3 bucket
File output formats	GZIP, Parquet	GZIP, Parquet
SQL table name	FOCUS_1_2_AWS	FOCUS_1_0_AWS
Manifest FocusVersion	"1.2"	"1.0"
Table configurations	Allows TIME_GRANULARITY to be set to HOURLY, DAILY, or MONTHLY	Only exports in HOURLY.

For more detailed information about the schema of FOCUS 1.2, see FOCUS 1.2 with AWS columns.

FOCUS 1.0 with AWS columns

The FOCUS 1.0 with AWS columns table contains your cost and usage data formatted with FinOps Open Cost and Usage Specification (FOCUS) 1.0, along with five additional columns from AWS that contain proprietary billing data. These columns are **x_CostCategories**, **x_Discounts**, **x_Operation**, **x_ServiceCode**, and **x_UsageType**. For more information about the FOCUS open-source specification, refer to the FOCUS website.

The SQL table name for FOCUS 1.0 with AWS columns is `FOCUS_1_0_AWS`

Table configurations

There are no table configurations for the FOCUS 1.0 with AWS columns table.

AWS Organizations support

The FOCUS 1.0 with AWS columns table inherits the settings you made in the consolidated billing feature in AWS Organizations. When consolidated billing is enabled, there are different behaviors for management and member accounts. If you’re using a management account, your FOCUS 1.0 with AWS columns table includes cost and usage data for the management account and all member accounts in your organization. If you’re using a member account, your FOCUS 1.0 with AWS columns table only includes cost and usage data for that member account.

After joining an organization, a member account can only export data for the time that the account has been a member of the organization. For example, let's say that a member account leaves organization A and joins organization B on the 15th of the month. Then, the member account creates an export. Because the member account created an export after joining organization B, the member account’s export of FOCUS 1.0 with AWS columns for the month only includes cost and usage data for the time that the account has been a member of organization B.

FOCUS 1.0 with AWS columns

Column name	Description	Data type	Null value allowed
AvailabilityZone	A provider-assigned identifier for a physically separated and isolated area	string	Yes

Column name	Description	Data type	Null value allowed
	within a Region that provides high availability and fault tolerance.		
BilledCost	A charge that is the basis for invoicing , inclusive of all reduced rates and discounts while excluding the amortization of relevant purchases paid to cover future eligible charges.	double	No
BillingAccountId	A provider-assigned identifier for a billing account.	string	No
BillingAccountName	A provider-assigned name for a billing account.	string	Yes
BillingCurrency	An identifier that represents the currency that a charge for resources or services was billed in.	string	No
BillingPeriodEnd	The end date and time of the billing period.	timestamp_millisec onds	No

Column name	Description	Data type	Null value allowed
BillingPeriodStart	The start date and time of the billing period.	timestamp_milliseconds	No
ChargeCategory	An indicator of whether the row represents an upfront or recurring fee, cost of usage that already occurred, an after-the-fact adjustment (for example, credits), or taxes.	string	No
ChargeClass	An indicator of whether the row represents a regular charge, or a correction to one or more previous charges.	string	Yes
ChargeDescription	A high-level context of a row without requiring additional discovery.	string	Yes
ChargeFrequency	An indicator of how often a charge will occur.	string	No
ChargePeriodEnd	The end date and time of the charge period.	timestamp_milliseconds	No

Column name	Description	Data type	Null value allowed
ChargePeriodStart	The start date and time of the charge period.	timestamp_milliseconds	No
CommitmentDiscountCategory	An indicator of whether the commitment-based discount identified in the CommitmentDiscountId column is based on usage quantity or cost (that is, spend).	string	Yes
CommitmentDiscountId	A provider-assigned identifier for a commitment-based discount.	string	Yes
CommitmentDiscountName	The display name assigned to a commitment-based discount.	string	Yes
CommitmentDiscountStatus	An indicator of whether the charge corresponds to a used or unused commitment discount.	string	Yes
CommitmentDiscountType	A provider-assigned name to identify the type of commitment-based discount applied to the row.	string	Yes

Column name	Description	Data type	Null value allowed
ConsumedQuantity	The volume of a given resource or service used or purchased based on the ConsumedUnit.	double	Yes
ConsumedUnit	A provider-assigned measurement unit indicating how a provider measures usage of a given SKU associated with a resource or service.	string	Yes
ContractedCost	The cost calculated by multiplying ContractedUnitPrice and the corresponding PricingQuantity.	double	No
ContractedUnitPrice	The agreed-upon unit price for a single PricingUnit of the associated SKU, inclusive of any negotiated discounts while excluding negotiated commitment-based discounts or any other discounts.	double	Yes

Column name	Description	Data type	Null value allowed
EffectiveCost	A cost that includes all reduced rates and discounts, augmented with the amortization of relevant purchases (one-time or recurring) paid to cover future eligible charges.	double	No
InvoiceIssuerName	An entity responsible for invoicing the sources or services consumed. It is commonly used for cost analysis and reporting scenarios.	string	No
ListCost	The cost calculated by multiplying ListUnitPrice and the corresponding PricingQuantity.	double	No
ListUnitPrice	The suggested unit price, published by the provider, for a single PricingUnit of the associated SKU, excluding any discounts.	double	Yes

Column name	Description	Data type	Null value allowed
PricingCategory	The pricing model used for a charge at the time of use or purchase.	string	Yes
PricingQuantity	The volume of a given SKU associated with a resource or service used or purchased, based on the PricingUnit.	double	Yes
PricingUnit	A provider-assigned measurement unit for determining unit prices, indicating how the provider rates measured usage and purchase quantities after applying pricing rules such as block pricing.	string	Yes
ProviderName	The entity that made the resources or services available for purchase.	string	No
PublisherName	The entity that produced the resources or services that were purchased.	string	No

Column name	Description	Data type	Null value allowed
RegionId	A provider-assigned identifier for an isolated geographic area where a *resource* is provisioned or a *service* is provided.	string	Yes
RegionName	The name of an isolated geographic area where a *resource* is provisioned or a *service* is provided.	string	Yes
ResourceId	A provider-assigned identifier for a resource.	string	Yes
ResourceName	A display name assigned to a resource.	string	Yes
ResourceType	The type of resource the charge applies to.	string	Yes
ServiceCategory	The highest-level classification of a service based on the core function of the service.	string	No
ServiceName	A display name for the offering that was purchased.	string	No

Column name	Description	Data type	Null value allowed
SkuId	A unique identifier that defines a provider-supported construct for organizing properties that are common across one or more SKU prices.	string	Yes
SkuPriceId	A unique identifier that defines the unit price used to calculate the charge.	string	Yes
SubAccountId	An ID assigned to a grouping of resources or services, often used to manage access and/or cost.	string	Yes
SubAccountName	A name assigned to a grouping of resources or services, often used to manage access and/or cost.	string	Yes
Tags	The set of tags assigned to tag sources that also account for potential provider-defined or user-defined tag evaluations.	map <string, string>	Yes

Column name	Description	Data type	Null value allowed
x_CostCategories	A map column containing key-value pairs of the cost categories and their values for a given line item.	map <string, string>	Yes
x_Discounts	A map column containing key-value pairs of any specific discounts that apply to this line item.	map <string, double>	Yes
x_Operation	The specific AWS operation covered by this line item. This describes the specific usage of the line item.	string	Yes
x_ServiceCode	The code of the service used in this line item.	string	Yes
x_UsageType	The usage details of the line item.	string	Yes

FOCUS 1.0 with AWS columns conformance gaps

The following table provides all of the conformance gaps that might exist in an export of the FOCUS 1.0 with AWS columns table. A particular conformance gap will not apply to your export if you're not receiving cost and usage data for the related scenario.

Conformance gap type	Affected column	FOCUS 1.0 requirement	Conformance gap description
Missing data	ContractedUnitPrice	ContractedUnitPrice **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	ContractedUnitPrice might be null for certain product offerings.
	InvoiceIssuerName	InvoiceIssuerName **must not** be null.	InvoiceIssuerName might be null for certain charges.
	ListUnitPrice	ListUnitPrice **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	ListUnitPrice might be null for certain product offerings.
	PricingUnit	PricingUnit **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	PricingUnit might be null for certain product offerings.
	PublisherName	PublisherName **must not** be null.	PublisherName may be null for certain charges.
	SkuId	SkuId **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	SkuId might be null for certain product offerings.

Conformance gap type	Affected column	FOCUS 1.0 requirement	Conformance gap description
	SkuPriceId	SkuPriceId **must not** be null for Usage and Purchase ChargeCategory line items that are not corrections.	SkuPriceId might be null for certain line items when it should not be.
Incorrect data	ConsumedQuantity	ConsumedQuantity is a required column for showing the amount of usage you actually used.	ConsumedQuantity will contain the amount of usage you were charged for. This means that ConsumedQuantity could be incorrect in situations where a minimum charge quantity applied for a particular service. For example, there is a 10MB minimum for an Athena query and a 10 minute minimum Glue crawler run. For these services, ConsumedQuantity will show the value that includes the minimum charged quantity.

Cost and usage dashboard

The SQL table name for cost and usage dashboard is COST_AND_USAGE_DASHBOARD.

Table configurations

Table configurations are user-controlled properties that a user can set to change the data or schema of a table before it's queried in Data Exports.

The Cost and Usage Dashboard does not have any modifiable table configurations.

AWS Organizations support

The cost and usage dashboard table is generated from CUR 2.0 data, which means that it inherits the same AWS Organizations settings that apply to CUR 2.0 and the same behavior applies. To understand how AWS Organizations applies to the cost and usage dashboard, refer to AWS Organizations support in the CUR 2.0 section.

Cost and usage dashboard columns

Column name	Description
amortized_cost	The effective cost of the upfront and monthly reservation fees spread across the billing period. This is the sum of costs based on the type of line item. The cost is determined as follows: • If the line item type is 'SavingsPlanCoveredUsage', the cost is the effective cost of the savings plan. • If the line item type is 'SavingsPlanRecurringFee', the cost is the total commitment to date of the savings plan minus the used commitment.. •

Column name	Description
	If the line item type is 'SavingsPlanNegation' or 'SavingsPlanUpfrontFee', the cost is 0. • If the line item type is 'DiscountedUsage', the cost is the effective cost of the reservation. • If the line item type is 'RIFee', the cost is the sum of the unused amortized upfront fee for the billing period and the unused recurring fee of the reservation. • If the line item type is 'Fee' and there is a reservation ARN, the cost is 0. • For all other line item types, the cost is the unblended cost of the line item.
availability_zone	The Availability Zone that hosts this line item. For example, us-east-1a or us-east-1b.
billing_entity	Helps you identify whether your invoices or transactions are for AWS Marketplace or for purchases of other AWS services. Possible values include: **AWS:** Identifies a transaction for AWS services other than in AWS Marketplace. **AWS Marketplace:** Identifies a purchase in AWS Marketplace.

Column name	Description
billing_period	The start date of the billing period that is covered by dashboard, in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ` . **Example:** 2023-10-01T00:00:00.000Z
charge_category	The category of the charge covered by this line item. The following are the possible categories: Running_usage: When the charge_type is one of the following: 'DiscountedUsage', SavingsPlanCoveredUsage', or 'Usage'. Non_Usage: For everything else under charge_type.
charge_type	The type of charge covered by this line item. See lineItem/LineItemType here for all the possible values.
current_generation	Helps you identify whether your Amazon RDS instance is of current generation or not.
database_engine	Describes the database engine of your database. **Examples:** PostgreSQL, Oracle.

Column name	Description
instance_type	Describes the instance type, size, and family, which define the CPU, networking, and storage capacity of your instance. **Examples:** t2.small, m4.xlarge, t2.micro, m4.large, t2.large **Services:** - Amazon EC2 - Amazon RDS - OpenSearch Service - Amazon ElastiCache - Amazon EMR - For the full service list, download: Column_Attribute_Service.zip
instance_type_family	The instance family that is associated with the given usage. **Examples:** t2, m4, m3 **Services:** - Amazon DocumentDB - Amazon RDS
invoice_id	The ID associated with a specific line item. The invoice_id remains blank until the export is final.

Column name	Description
item_description	The description of the line item type. For example, the description of a usage line item summarizes what type of usage you incurred during a specific time period. For size-flexible RIs, the description corresponds to the RI the benefit was applied to. For example, if a line item corresponds to a t2.micro and a t2.small RI was applied to the usage, the lineItem/LineItemDescription displays t2.small. The description for a usage line item with an RI discount contains the pricing plan covered by the line item.
legal_entity	The Seller of Record of a specific product or service. In most cases, the invoicing entity and legal entity are the same. The values might differ for third-party AWS Marketplace transactions. Possible values include: **Amazon Web Services, Inc. :** The entity that sells AWS services. **Amazon Web Services India Private Limited :** The local Indian entity that acts as a reseller for AWS services in India.
linked_account_id	The account ID of the account that used this line item. For organizations, this can be either the management account or a member account. You can use this field to track costs or usage by account.

Column name	Description
linked_account_name	The name of the account that used this line item. For organizations, this can be either the management account or a member account. You can use this field to track costs or usage by account.
operation	The specific AWS operation covered by this line item. This describes the specific usage of the line item. For example, a value of RunInstances indicates the operation of an Amazon EC2 instance.
payer_account_id	The account ID of the paying account. For an organization in AWS Organizations, this is the account ID of the management account.
payer_account_name	The account name of the paying account. For an organization in AWS Organizations, this is the name of the management account.

Column name	Description
platform	Describes the operating system of your Amazon EC2 instance. **Examples:** Amazon Linux, Ubuntu, Windows Server, Oracle Linux, FreeBSD **Services:** • Amazon AppStream • Amazon EC2 • Amazon GameLift • Amazon Lightsail • Amazon WorkSpaces • Amazon CodeBuild
pricing_unit	The pricing unit that AWS used for calculating your usage cost. For example, the pricing unit for Amazon EC2 instance usage is in hours.

Column name	Description
processor	Describes the processor on your Amazon EC2 instance. **Examples:** High Frequency Intel Xeon E7-8880 v3 (Haswell), Intel Xeon E5-2670, AMD EPYC 7571 **Services:** • Amazon DocumentDB • Amazon EC2 • Amazon Neptune • Amazon RDS • AWS Database Migration Service
processor_features	Describes the processor features of your ins tances. **Examples:** Intel AVX, Intel AVX2, Intel AVX512, Intel Turbo **Services:** • AWS Database Migration Service • Amazon DocumentDB • Amazon EC2 • Amazon Neptune • Amazon RDS

Column name	Description
product_code	The code of the product measured. For example, AmazonEC2 is the product code for Amazon Elastic Compute Cloud.
product_family	The category for the type of product. **Examples:**Alarm, AWS Budgets, Stopped Instance, Storage Snapshot, Compute
product_from_location	Describes the location where the usage originated from. **Sample values:** External, US East (N. Virginia) , Global **Services:** • Amazon CloudFront • AWS DataTransfer

Column name	Description
product_group	A construct of several products that are similar by definition, or grouped together. For example, the Amazon EC2 team can categoriz e their products into shared instances, dedicated host, and dedicated usage. **Services:** - AWS Certificate Manager - AWS CodeCommit - AWS Glue - AWS IoT Analytics - AWS Lambda
product_name	The full name of the AWS service. **Examples:** AWS Backup, AWS Config, Amazon Registrar, Amazon Elastic File System, Amazon Elastic Compute Cloud
product_to_location	Describes the location usage destination. **Sample values:** External, US East (N. Virginia) **Services:** - Amazon CloudFront - AWS DataTransfer

Column name	Description
public_cost	The total cost for the line item based on public On-Demand Instance rates. If you have SKUs with multiple On-Demand public costs, the equivalent cost for the highest tier is displayed. For example, services offering free-tiers or tiered pricing.
purchase_option	The way in which you acquired and paid for the AWS resource in this line item. The purchase_option column contains 'SavingsPlan', 'Reserved', or 'Spot' for savings plans, reserved instances, and spot instances respectively. The purchase_option column has 'OnDemand' for the remaining records.
region	The geographical area that hosts your AWS services. Use this field to analyze spend across a particular Region. **Examples:** eu-west-3, us-west-1, us-east-1, ap-northeast-2, sa-east-1 **Services:** • Amazon EC2 • AWS Certificate Manager • Amazon S3 • Amazon RDS • Amazon DynamoDB • For the full service list, download: Column_Attribute_Service.zip

Column name	Description
ri_sp_arn	The unique Savings Plan or Reserved Instance identifier. It typically follows the format arn: aws: savingsplans: <region>:<account-id>: savingsplan/<savings-plan-id> or arn: aws: ec2:<region>:<account>reserved-instances/ <reserved-instance-id>.
ri_sp_trueup	This is the sum of adjustments based on the type of line item. True-up represents the difference between total upfront fees incur red in the period using an unblended cost, and the smaller portion of upfront fees applicabl e to the period using an amortized cost. The adjustment is determined as follows: • If the line item type is ‘SavingsPlanRecurr ingFee’, the adjustment is the negative of the amortized upfront commitment for the billing period of the Savings Plan. • If the line item type is ‘RIFee’, the adjustmen t is the negative of the amortized upfront fee for the billing period of the reservation. • For all other line item types, the adjustment is 0.
ri_sp_upfront_fee	The upfront fee refers to the initial payment you make when you choose certain types of Reserved Instances or Savings Plans.
service	The name of the AWS service. **Examples:** AmazonVPC, AmazonRDS, AmazonRoute53 etc.

Column name	Description
tenancy	The type of tenancy allowed on the Amazon EC2 instance. **Examples:** Dedicated, Reserved, Shared, NA, Host **Services:** - Amazon EC2 - Amazon ECS
unblended_cost	The UnblendedCost is the UnblendedRate multiplied by the UsageAmount.
usage_date	The start date and time for the line item in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ` . **Example:** 2023-10-01T00:00:00.000Z
usage_quantity	The amount of usage that you incurred during the specified time period. For size-flexible Reserved Instances, use the reservation/TotalReservedUnits column instead. **Note** Certain subscription charges will have a UsageAmount of 0.

Column name	Description
usage_type	The usage details of the line item. For example, USW2-BoxUsage:m2.2xlarge describes an M2 High Memory Double Extra Large instance in the US West (Oregon) Region.

Carbon emissions

The carbon emissions table contains your estimated carbon emissions. It provides detailed account-level and regional granularity of your carbon emissions data. You can configure automated monthly deliveries to Amazon S3 in either CSV or Parquet format, making it simple to integrate with your existing business intelligence tools and reporting systems. For more detailed information, see Viewing your carbon footprint in the *AWS Billing User Guide*.

The SQL table name for carbon emissions is `CARBON_EMISSIONS`.

Historical data

You will receive data going back up to January 2022 within 24 hours of setup, enabling you to perform baseline analysis and trend reporting without manual data gathering.

Table configurations

There are no table configurations for the Carbon emissions table.

Permissions

To access data in the Customer Carbon Footprint Tool or the Carbon emissions table, you need the IAM permission `sustainability:GetCarbonFootprintSummary`.

Model versions

The methodology for calculating your carbon emissions will evolve over time to better reflect your usage and align with carbon accounting best practices. Exports are partitioned in hierarchical order by “model_version=Y/” and “usage_period=YYYY-MM/”. The “model_version” partition that an export is stored under will correspond to the model version used to generate that export, while the “usage_period” partition corresponds to the dates the carbon emissions were generated. This structure enables you to differentiate between data with the old and new models by viewing the partition names.

AWS Organizations support

The Carbon emissions table inherits the settings you made in the consolidated billing feature in AWS Organizations. When consolidated billing is enabled, there are different behaviors for management and member accounts. If you're using a management account, your Carbon emissions table includes estimated carbon emissions data for the management account and all member accounts in your organization. If you're using a member account, your Carbon emissions table only includes estimated carbon emissions data for that member account.

After a member account joins a new organization, or a management account converts to a member account and joins a new organization, the account's carbon emissions data is recorded in the new organization's exports. Each management account contains member accounts' data for the time period it was linked to said management account. For example, a member account leaves organization A and joins organization B on the 15th of the month. Then, the member account creates an export. Because the member account created an export after joining organization B, the member account's export of the Carbon emissions table for the month includes estimated carbon emissions data for the time that the account has been a member of organization B.

When a member account leaves an organization or converts to a standalone account, the member account can still access previous exports if it has permissions to the Amazon S3 bucket where those exports are stored. Carbon emissions associated with terminated or suspended accounts will appear in the management account data exports for the periods when these accounts were active.

For more information, see Consolidated billing for AWS Organizations in the *AWS Billing User Guide*.

Carbon emissions columns

Column name	Description	Data type	Null value allowed
last_refresh_timestamp	The time when the carbon footprint value was last generated, in UTC. The format is `YYYY-`	timestamp	No

Column name	Description	Data type	Null value allowed
	`MM-DD HH:mm:ssZ` .		
location	Describes the human-readable location where the usage originated from, corresponding to the region_code. For example, US East (N. Virginia) . Emissions from Global Services, such as Amazon CloudFront, are reported under `Global`.	string	Yes
model_version	The version of the methodology the carbon footprint data was calculated with. For example, v2.0.0.	string	No
payer_account_id	The account ID of the paying account. For an organization in AWS Organizations, this is the account ID of the management account.	string	No

Column name	Description	Data type	Null value allowed
product_code	The code of the product measured. For example, Amazon EC2 is the product code for Amazon Elastic Compute Cloud.	string	Yes
region_code	A Region is a physical location around the world where data centers are clustered. AWS calls each group of logical data centers an Availability Zone (AZ). Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographical area. The Region code attribute has the same name as an AWS Region, and specifies where the AWS service is available. For example, us-east-1.	string	Yes

Column name	Description	Data type	Null value allowed
total_lbm_emissions_unit	The unit of measure used for location-based method (LBM) emissions. The unit we currently support is metric tons of carbon dioxide-equivalent (MTCO2e).	string	Yes
total_lbm_emissions_value	The total estimated location-based method (LBM) emissions value associated with the account.	double	Yes
total_mbm_emissions_unit	The unit of measure used for market-based method (MBM) emissions. The unit we currently support is metric tons of carbon dioxide-equivalent (MTCO2e).	string	Yes
total_mbm_emissions_value	The total estimated market-based method (MBM) emissions value associated with the account.	double	Yes

Column name	Description	Data type	Null value allowed
total_scope_1_emissions_value	The Scope 1 emissions value associated with the account.	double	No
total_scope_1_emissions_unit	The unit of measure used for Scope 1 emissions.	string	No
total_scope_2_lbm_emissions_value	The Scope 2 location-based method (LBM) emissions value associated with the account.	double	No
total_scope_2_lbm_emissions_unit	The unit of measure used for Scope 2 location-based method (LBM) emissions.	string	No
total_scope_2_mbm_emissions_value	The Scope 2 market-based method (MBM) emissions value associated with the account.	double	No
total_scope_2_mbm_emissions_unit	The unit of measure used for Scope 2 market-based method (MBM) emissions.	string	No

Column name	Description	Data type	Null value allowed
total_scope_3_lbm_emissions_value	The Scope 3 location-based method (LBM) emissions value associated with the account.	double	No
total_scope_3_lbm_emissions_unit	The unit of measure used for Scope 3 location-based method (LBM) emissions.	string	No
total_scope_3_mbm_emissions_value	The Scope 3 market-based method (MBM) emissions value associated with the account.	double	No
total_scope_3_mbm_emissions_unit	The unit of measure used for Scope 3 market-based method (MBM) emissions.	string	No
usage_account_id	The account ID of the account associated with the carbon footprint value. For organizations, this can be either the management account or a member account.	string	No

Column name	Description	Data type	Null value allowed
usage_period_end	The end date of the period that is covered by this report, in UTC. The format is `YYYY-MM-DD HH:mm:ssZ` .	timestamp	No
usage_period_start	The start date of the period that is covered by this report, in UTC. The format is `YYYY-MM-DD HH:mm:ssZ` .	timestamp	No

Processing data exports

In the following sections, you'll find information about processing your data exports.

Configuring Amazon Athena

Unlike Cost and Usage Reports (CUR), Data Exports doesn't offer an SQL file for setting up Athena to query your exports. You'll need to either use a CloudFormation template for Data Exports (see option 1) or manually configure Athena (see option 2).

(Option 1) Use a CloudFormation template: To locate the CloudFormation template and instructions for setting up Athena with Data Exports, refer to Data Exports in the Cloud Intelligence Dashboards Framework.

(Option 2) Use an AWS Glue crawler to build your table and partitions for Athena: When creating CUR or carbon emissions data exports for Athena, we suggest using the Apache Parquet file format; it offers better compression and column-oriented storage which contributes to smaller and less expensive Athena queries. The overwrite delivery preference is required so that each monthly partition always contains only one copy of each file and no duplicate line items appear when you execute queries with Amazon Athena.

We also recommend using AWS Glue with a Glue crawler to load your data into Athena.

To build a table and partitions for Athena using an AWS Glue crawler

1. Create an export of CUR 2.0 or Carbon emissions with the following data export delivery options:
   - Compression type and file format: Parquet - Parquet
   - File versioning: Overwrite existing data export file
2. In Athena, use the notebook editor with Trino SQL and choose **Create** to create a table with "AWS Glue crawler". Using the Glue crawler workflow, point the Glue crawler to run on the s3://<bucket-name>/<prefix>/<export-name>/data folder to automatically load all of the delivered partitions for the specified export to Athena.
3. After the Glue crawler is complete, you can use Athena to write queries on the table created by the Glue crawler.

Configuring Amazon Redshift

Amazon Redshift is a cloud data warehouse that can be accessed either in a provisioned capacity or serverless model. Amazon Redshift offers fast query performance for processing your data from Data Exports.

Currently, Data Exports doesn't provide the SQL file for setting up Redshift to query your exports like Cost and Usage Reports (CUR) does. However, you can still manually set up Redshift to query your exports. We recommend that you use the gzip/csv compression and file format for Redshift.

For information on setting up Redshift, see the *Amazon Redshift Getting Started Guide*.

Recommended SQL queries for processing CUR 2.0

After loading your CUR 2.0 export data into a data analytics tool such as Amazon Athena or Amazon Redshift, you can process it in order to gain cost and usage insights. AWS Well-Architected Labs provides a CUR query library that you can use to process CUR. For more information, see AWS CUR Query Library.

Note the following two pieces of information about SQL queries:

- The Well-Architected Labs SQL queries won't work in the data export query field, because Data Exports doesn't support aggregations and some of the other SQL syntax used in these queries.
- The Well-Architected Labs SQL queries will only work if you haven’t renamed your columns from the default names. Depending on the query, you may need to query some of the product columns as separate columns using the dot operator. For more information, see Data query–SQL query and table configurations.

Recommended SQL queries for processing carbon emissions data exports

To get the total carbon emissions per payer_account_id:

```
SELECT payer_account_id, SUM(total_mbm_emissions_value) AS total_emissions
FROM "ccft-data-exports"."ccft-data-exports-data" -- change to your table name
GROUP BY payer_account_id
ORDER BY total_emissions DESC;
```

To get the total carbon emissions per payer_account_id and per product_code:

```
SELECT payer_account_id, product_code, SUM(total_mbm_emissions_value) AS
 total_emissions
FROM "ccft-data-exports"."ccft-data-exports-data"-- change to your table name
GROUP BY payer_account_id, product_code
ORDER BY total_emissions DESC;
```

To get the total carbon emissions per payer_account_id and per region_code:

```
SELECT payer_account_id, region_code, SUM(total_mbm_emissions_value) AS total_emissions
FROM "ccft-data-exports"."ccft-data-exports-data" -- change to your table name
GROUP BY payer_account_id, region_code
ORDER BY total_emissions DESC;
```

Understanding the Cost and Usage Dashboard

The Cost and Usage Dashboard is an easy to deploy, secure, and pre-built dashboard powered by Amazon QuickSight, and inspired from the Cloud Intelligence Dashboards (CID) open source project. The Cost and Usage Dashboard includes a subset of the summary visuals without the resource-level views from the CUDOS dashboard, which is one of the six Cloud Intelligence Dashboards. The Cost and Usage Dashboard brings the benefits of the CUDOS solution into an AWS Billing and Cost Management console feature that is easy to set up, and frees you from maintaining the underlying infrastructure, such as Amazon Athena views or AWS Glue crawlers. You can deploy the Cost and Usage Dashboard from the **Data Exports** page in the AWS Billing and Cost Management console within minutes. CID involves AWS CloudFormation template-based deployment. For information on setting up the complete CID solution, see AWS well architected labs.

The following table summarizes the differences between a Cost and Usage Dashboard and Cloud Intelligence Dashboards (CID):

Feature	Cost and Usage Dashboard	Cloud Intelligence Dashboards
Deployment	Seamless deployment from AWS console	CloudFormation, Command Line, or Terraform
Deployment options for AWS Organizations	In management account only	In management account or delegated linked account
Multiple AWS Organizations aggregation	No	Yes
High-level cost and usage insights	Yes	Yes
Resource-level details	No	Yes
Reserved Instances and Savings Plans insights	No	Yes

Feature	Cost and Usage Dashboard	Cloud Intelligence Dashboards
Supported data sources	Cost and Usage Summary (dashboard view)	Cost and Usage Report (CUR), Compute Optimizer, Trusted Advisor, Cost Anomaly Detection

Understanding the Cost and Usage Report (CUR)

You can use CUR and CUR 2.0 to get the most detailed information about your AWS cost and usage. For more information on use cases such as tracking your Savings Plans and Reserved Instance (RI) utilization, charges, and split cost allocations, see Use cases.

Understanding the carbon emissions data export

You can use the carbon emissions data export to get carbon emissions estimates from the Customer Carbon Footprint Tool for all usage accounts linked to your payer account. For more information about how these carbon emissions are calculated and categorized, see Viewing your carbon footprint in the *AWS Billing User Guide*.

Security and permissions

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 Cost Management, see AWS Services in Scope by Compliance Program.

Security in the cloud: Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations. This documentation helps you understand how to apply the shared responsibility model when using Billing and Cost Management. The following topics show you how to configure Billing and Cost Management to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Billing and Cost Management resources.

Topics

Identity and access management for Data Exports

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 Billing resources. IAM is an AWS service that you can use with no additional charge.

To use Data Exports, an IAM user needs to be given access to actions in the `bcm-data-exports` namespace in IAM. See the following table for the available actions.

Data Exports action	Description	Access level	Resource types	Condition keys
CreateExport	Allows user to create an Export and specifies query, delivery configurations, schedule configurations, and content configurations.	Write	export table	aws:RequestTag/${TagKey} aws:TagKeys
UpdateExport	Allows user to update an existing Export.	Write	export table	aws:ResourceTag/${TagKey}
DeleteExport	Allows user to delete an existing Export.	Write	export	aws:ResourceTag/${TagKey}
GetExport	Allows user to view an existing Export.	Read	export	aws:ResourceTag/${TagKey}
ListExports	Allows user to list all existing Exports.	Read		
GetExecution	Allows user to see details of the given Execution, including metadata and schema of the exported data.	Read	export	aws:ResourceTag/${TagKey}

Data Exports action	Description	Access level	Resource types	Condition keys
ListExecutions	Allows user to list all Executions of the provided Export identifier.	Read	export	aws:ResourceTag/${TagKey}
GetTable	Allows user to get the schema of the given table.	Read	table	
ListTables	Allows user to list all available tables.	Read		

TagResource	Allows user to tag an existing Export.	Write	export	aws:ResourceTag/${TagKey}
				aws:RequestTag/${TagKey}
				aws:TagKeys

UntagResource	Allows user to untag an existing Export.	Write	export	aws:ResourceTag/${TagKey} aws:TagKeys
ListTagsForResource	Allows user to list tags associated with an existing Export.	Read	export	aws:ResourceTag/${TagKey}

For more information about how to use these context keys, see Controlling access to AWS resources using tags in the *IAM User Guide*.

The following table describes the resource types that are available in Data Exports.

Resource type	Description	ARN
export	An export is the resource created by the CreateExport API. An export generates a billing and cost management query output on a recurring basis.	arn:${Partition}:bcm-data-exports:${Region}:${Account}:export/${exportName}-{UUID}
table	A table is data in a row-column format that a user queries with an export. Tables are created and managed by AWS for customers. Tables cannot be deleted by customers.	arn:${Partition}:bcm-data-exports:${Region}:${Account}:table/${TableName}

To create exports of the COST_AND_USAGE_REPORT or COST_AND_USAGE_DASHBOARD table resources in Data Exports, IAM users must also have permissions for the respective `cur` action in IAM. This means that if an IAM user is blocked from using `cur` actions for any reason, such as lacking an explicit allow on `cur` or a service control policy (SCP) providing an explicit deny on `cur`, that IAM user will be blocked from creating or updating exports of these two tables.

The following table shows which `cur` action is required for which `bcm-data-exports` actions in Data Exports for these two tables.

Data Exports action	Table resources	Additional required actions in IAM
bcm-data-exports:CreateExport	COST_AND_USAGE_REPORT COST_AND_USAGE_DASHBOARD	cur:PutReportDefinition

Sample policy

Allow IAM user to have full access to CUR 2.0 exports in Data Exports.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewDataExportsTablesAndExports",
            "Effect": "Allow",
            "Action": [
                "bcm-data-exports:ListTables",
                "bcm-data-exports:ListExports",
                "bcm-data-exports:GetExport"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateCurExports",
            "Effect": "Allow",
            "Action": "bcm-data-exports:*",
            "Resource": [
                "arn:aws:bcm-data-exports:*:*:table/COST_AND_USAGE_REPORT",
                "arn:aws:bcm-data-exports:*:*:export/*"
                ]
        },
        {
            "Sid": "CurDataAccess",
            "Effect": "Allow",
            "Action":  "cur:PutReportDefinition",
            "Resource": "*"
         }
    ]
}
```

For more information on access control and IAM permissions to use Data Exports in Billing and Cost Management, see Overview of managing access permissions.

Create a pro forma AWS CUR 2.0

To create a pro forma CUR 2.0, you will need to include the following IAM policy:

Allow IAM user to have full access to CUR 2.0 and Billing Group Billing View.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateCur20AnyBillingView",
            "Effect": "Allow",
            "Action": "bcm-data-exports:CreateExport",
            "Resource": [
                "arn:aws:bcm-data-exports:*:*:table/COST_AND_USAGE_REPORT",
                "arn:aws:bcm-data-exports:*:*:export/*",
                "arn:aws:billing::*:billingview/*"
            ]
        },{
            "Sid": "CurDataAccess",
            "Effect": "Allow",
            "Action": "cur:PutReportDefinition",
            "Resource": "*"
        }
    ]
}
```

If you want an IAM role to have access to a specific billing group, you can add the Billing View ARN the role is allowed to access.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateSpecificBillingViewCur20",
            "Effect": "Allow",
            "Action": "bcm-data-exports:CreateExport",
            "Resource": [
                "arn:aws:bcm-data-exports:*:*:table/COST_AND_USAGE_REPORT",
                "arn:aws:bcm-data-exports:*:*:export/*",
                "arn:aws:billing::444455556666:billingview/billing-
group-111122223333"
        ]
        },{
            "Sid": "CurDataAccess",
            "Effect": "Allow",
            "Action": "cur:PutReportDefinition",
            "Resource": "*"
        }
    ]
}
```

Data protection in Data Exports

Learn how the AWS shared responsibility model applies to data protection in Data Exports.

S3 security best practices

Data Exports delivers your billing and cost management data to an Amazon S3 bucket. There are a number of steps you can take to make sure your S3 bucket is secure. For more information, see Security best practices for Amazon S3 in the *Amazon S3 User Guide*.

Data encryption in S3

By default, your data exports are encrypted using server-side encryption with Amazon S3 managed keys (SSE-S3). If you want to use Amazon Key Management Service (KMS) encryption (SSE-KMS) to encrypt your exports, you need to trigger encryption with KMS after the export has been delivered. For more information, see Setting default server-side encryption behavior for Amazon S3 buckets in the *Amazon S3 User Guide*.

Quotas and restrictions

The following table describes the current quotas and restrictions within Data Exports.

Quota type	Quota value
Number of exports of CUR 2.0 (`COST_AND_USAGE_REPORT`)	5
Number of exports of Cost Optimization Recommendations table (`COST_OPTIMIZATION_RECOMMENDATIONS`)	2
Number of exports of FOCUS 1.2 table (`FOCUS_1_2_AWS`)	2
Number of exports of FOCUS 1.0 table (`FOCUS_1_0_AWS`)	2
Number of exports of Cost and Usage Dashboard table (`COST_AND_USAGE_DASHBOARD`)	2
Number of exports of Carbon emissions table (`CARBON_EMISSIONS`)	2
Number of exports of legacy for billing transfer accounts	1000

Troubleshooting

As you use Data Exports or Cost and Usage Reports (CUR), you might encounter certain issues or challenges that require troubleshooting. Use this troubleshooting section for recommendations that help you to resolve common problems quickly and efficiently.

Topics

General troubleshooting

Topics

Why is my export unhealthy?

An “unhealthy” export is one that encountered an error when it last tried to deliver a refresh to your Amazon S3 bucket. You may see one of the following error messages when hovering your cursor over the “unhealthy” message or by calling the `GetExport` API.

- **Data Exports issues**
  - **Insufficient permission:** This means Data Exports was unable to deliver the export files to your S3 bucket. This can be fixed by updating your S3 bucket policy with the permission listed in Setting up an Amazon S3 bucket for data exports.

  - **Bill owner changed:** This error can occur when your AWS account moves to a new organization or leaves an organization in AWS Organizations. It can also happen when you're in an organization and your management account changes whether you belong to a billing group in AWS Billing Conductor. The best way to solve this problem is to create a new CUR and delete your old CUR. If you believe your account should not have changed organizations or billing groups, contact your account admin.
  - **Internal failure:** This error is due to an issue with the Data Exports internal infrastructure. Review the AWS Service Health Dashboard for updates on any service-wide issues that may be affecting Data Exports, or contact AWS Support for more information or help.
- **QuickSight integration issues**
  - **Insufficient SPICE capacity:** This error means that QuickSight does not have enough processing capacity provisioned to ingest your cost and usage data. For information on how to increase your SPICE capacity, see Managing SPICE memory capacity.
  - **Insufficient permission to access the manifest file:** The service role you assigned to QuickSight to access your S3 bucket is no longer working. Review your service policy to make sure it's giving read permissions to the S3 bucket storing your cost and usage data.
  - **Access denied when trying to access manifest file:** Your IAM role does not have access to the S3 bucket storing your export files to check whether a QuickSight dashboard exists for this export. The dashboard may or may not be working. You need `s3:GetObject` permissions on the S3 bucket storing the export data in order to be able to check for a QuickSight dashboard.
  - **QuickSight CreateBundle failed:** This error means your dashboard failed to be created in QuickSight. This may have happened due to a delay in IAM role propagation if you created a new service role, or if you selected an existing service role that did not have the right permissions. Use the retry action if you created a new service role. If you selected an existing service role, you should delete your export and create a new one with a new service role.
  - **Dashboard does not exist:** This error means your dashboard was deleted in QuickSight. You should delete your existing cost and usage dashboard export in Data Exports and recreate it.
  - **QuickSight account does not exist:** This error means your QuickSight account was deleted. You will need to recreate your QuickSight account to use a dashboard again. After recreating your QuickSight account, you should delete your existing cost and usage dashboard export in Data Exports and recreate it.

Why is my SQL statement not being accepted by Data Exports?

Data Exports supports a limited set of SQL syntax that is mainly focused on column selections and row filters. Make sure your SQL statement is using only the relevant keywords and operators. For full details, see Data query.

Why can't I locate a predefined SQL script for configuring Athena within Data Exports?

Unlike Cost and Usage Reports (CUR), Data Exports doesn't offer an SQL file for setting up Athena to query your exports. You'll need to either use a CloudFormation template for Data Exports or manually configure Athena. For more information, see Configuring Amazon Athena.

Why is one of my export partitions empty?

If an export is larger than most applications can handle, AWS splits the report into multiple files. If an export update is smaller than the previous export and you're using “overwrite” mode, AWS overwrites the unneeded partitions with empty data. The export manifest only lists the partitions that have data. Review the report’s manifest file to find any empty files that you don’t need to ingest.

Why are there no report files in the Amazon S3 bucket?

Confirm that the Amazon S3 bucket policy grants the **billingreports.amazonaws.com** service permission to put files in the bucket. For more information on the required bucket policy, see Setting up an Amazon S3 bucket for data exports or Setting up an Amazon S3 bucket for Cost and Usage Reports.

Troubleshooting CUR 2.0

Topics

- When attempting to create a data export with the same CSV format as the legacy CUR columns, I get an "Invalid QueryStatement" error. How can I resolve this?
- After migrating to Data Exports CUR 2.0, can I have a legacy CUR export and a CUR 2.0 export at the same time?
- When attempting to create an export of CUR 2.0, I get the error "This account is unable to create an export against this table". Why can't I create a CUR 2.0 export?

Why are some columns that were available in CUR not appearing in CUR 2.0?

In CUR 2.0, four column types became nested into four individual columns. The resulting nested columns are: `product`, `discount`, `resource_tag`, and `cost_category`.

In legacy CUR, there could be hundreds of columns with names that started with these strings. The variations depended on customer usage of AWS services or agreements with AWS. This schema design resulted in hundreds of columns that were often sparsely filled. The variability of the columns could also cause problems with SQL queries due to a charging schema.

As a result, the columns that could vary across different AWS columns were nested together into these four columns. Certain product columns that are commonly used were not nested.

You can recreate the schema of the CUR in your CUR 2.0 export by using the dot operator in SQL. To learn how to do this, see Migrating from CUR to Data Exports CUR 2.0.

What will happen to legacy Cost and Usage Reports; will it be deprecated?

We currently have no plans to deprecate legacy CUR. However, as CUR 2.0 in Data Exports offers several improvements such as a consistent schema, nested data, and additional columns (`bill_payer_account_name` and `line_item_usage_account_name`), we recommend migrating to CUR 2.0.

While there is no target date, we are planning to eventually deprecate **Cost and Usage Reports** under **Legacy Pages** in the console. However, all of the same functionality to create, update, and delete legacy CUR is available through the **Data Exports** console page.

Note

Detailed Billing Reports (DBR), another legacy billing feature, may be deprecated at a later date. The feature has been unavailable for new customers since July 8, 2019.

Does creating an export of CUR 2.0 affect my legacy CUR?

CUR and CUR 2.0 are two distinct reports. When creating CUR 2.0, there is no impact on your existing CUR settings. You can choose between legacy CUR and CUR 2.0 based on your preferences.

Why can’t I create an export of CUR 2.0 even though I have IAM permissions to use Data Exports and the CUR table?

Make sure you also have IAM permissions for `cur:PutReportDefinition`.

When attempting to create a data export with the same CSV format as the legacy CUR columns, I get an "Invalid QueryStatement" error. How can I resolve this?

Currently, you can't rename your columns to have special characters such as "/" to match the legacy CUR column names in CSV format. For information about the supported character types, see SQL query.

After migrating to Data Exports CUR 2.0, can I have a legacy CUR export and a CUR 2.0 export at the same time?

Yes, you can have up to 10 legacy CUR exports and 5 CUR 2.0 exports at the same time.

When attempting to create an export of CUR 2.0, I get the error "This account is unable to create an export against this table". Why can't I create a CUR 2.0 export?

Unlike legacy CUR, CUR 2.0 does not currently support creating an export of CUR 2.0 with pro forma billing data. If you are part of a billing group in AWS Billing Conductor, you are only allowed to receive pro forma billing data. As a result, you receive this error message when trying to create an export of CUR 2.0. You can still create a legacy CUR export.

Troubleshooting the cost and usage dashboard

Topics

Why did my cost and usage dashboard export fail right after I created it?

Your cost and usage dashboard export may have failed due to a delay in IAM role propagation. If you created a new service role for this export, Amazon QuickSight may not have had permission to access your S3 bucket and create your dashboard. When you see the error “Insufficient permission to access the manifest file” in the Export status, choose Export, and then choose Retry in the table action menu.

If you didn’t create a new service role for your cost and usage dashboard export, you may have specified an incorrect service role for QuickSight to use. In this case, you should delete your export and recreate it, while also creating a new service role in the cost and usage dashboard console workflow.

Why can't I access the dashboard?

You may not be able to access the cost and usage dashboard in Amazon QuickSight if you don’t have permission to view it. To troubleshoot, open your export by choosing the export name. Check the QuickSight created by field to see who created the dashboard. Ask the user to give you permission to view the dashboard.

Why am I being taken to the console admin page to unsubscribe the QuickSight account when I try to view the dashboard?

You may encounter this error if you're using the “Active directory” authentication method. Choose the cost and usage dashboard export name to view the details of your export. Choose **QUICKSIGHT SIGN IN** to sign in to your QuickSight account. You'll be able to see the dashboard if you have permission to view it.

Why don't I see any data in the cost and usage dashboard that I just created?

Your cost and usage dashboard could be missing the data for the current month because it can take up to 24 hours for all your data to be populated in your dashboard. Check the status of your cost and usage dashboard export. If the export status says “Healthy”, allow 24 hours for your dashboard to update with the current month's data. If you don’t see the current month's data in your dashboard after 24 hours, contact AWS Support. You can check the creation time of your cost and usage dashboard in the Exports and Dashboards table on the Data Exports console page.

Why can't I see historical data in the cost and usage dashboard?

Your cost and usage dashboard might be missing the six months of historical data for any of the following reasons:

- **No historical data exists:** If you have an account without six months of historical spending due to being a new account or recently changing membership in AWS Organizations, no historical data can populate the dashboard.
- **Historical backfill is still in progress:** Historical data backfill by Data Exports can take up to 24 hours to complete. You can use the SDK/CLI to check if any backfill executions failed with the `ListExecutions` API for this export, or if they are still in progress. Wait a little longer or use `ListExecutions` to ensure the backfill is not in progress.
- **Historical backfill failed:** Historical data backfill may have failed to complete due to an internal error. You can come to this conclusion if it's been more than 24 hours and the backfill is not complete, or you can use the `ListExecutions` API in the SDK/CLI and look for any failed executions for this export. If you believe the backfill has failed, try remaking the cost and usage dashboard in the console. If it fails a second time, we recommend reaching out to AWS Support.

Why did my QuickSight dashboard link disappear from the Data Exports console page?

The Data Exports console page reads from a file in your S3 bucket in order to identify the QuickSight dashboard that the export is linked to. If this file is altered or deleted, the console doesn't know that a dashboard exists for this export. While your dashboard still exists in QuickSight, you'll need to repair this file for the link to reappear.

How can I configure Amazon QuickSight to visualize resource tags in CUR 2.0?

The cost and usage dashboard feature does not support visualizing resource tags. However, you can still receive your resource tag data in the CUR 2.0 export. If you want an AWS supported QuickSight dashboard for visualizing your cost and usage with tags, refer to the CUDOS Dashboard from AWS Well-Architected Labs. It currently only uses data from legacy CUR, but will support CUR 2.0 in the future.

Troubleshooting Cost and Usage Reports

Topics

- How are All Upfront Reserved Instances amortized in my report?

Why does my Cost and Usage Report data not match the data shown in other Billing and Cost Management features?

Other Billing and Cost Management features (Cost Explorer, Detailed Billing Reports, Billing and Cost Management console) might present your costs differently for the following reasons:

- The billing features round cost data in different ways.
- The billing features might have different data refresh settings. For example, you can choose whether or not your Cost and Usage Report automatically refreshes a previously closed bill with any refunds, credits, or Support fees applied after the bill is finalized. Cost Explorer automatically reflects the same items. In this scenario, if you don’t activate the automatic refresh on your Cost and Usage Report, then the Cost and Usage Report data won’t match the Cost Explorer data.
- The billing features can group charges differently. For example, the **Bills** page in the Billing and Cost Management console shows data transfer charges as a separate **Data Transfer** grouping within your **AWS Service Charges**. Meanwhile, Cost and Usage Reports and Cost Explorer show data transfer charges as a usage type for each service.

If after reviewing these reasons you still believe you're seeing discrepancies between your Cost and Usage Report and other Billing and Cost Management features, open a support case to request a review of your cost data. In your support case, make sure to provide the report name and the billing period that you would like reviewed. For more information on opening a case, see Getting help with your exports and reports.

How can I backfill data after changing my report settings?

Open a support case to request a backfill of your cost data. In your support case, make sure to provide the report name and the billing period that you want backfilled. For more information on opening a case, see Getting help with your exports and reports.

Note that you can't get a backfill of cost data for the following scenarios:

- You can't get a backfill for cost data from before the date that you created the account.
- If you use AWS Organizations and the structure of your organization changed, such as which account is designated the management account, then you can't get a backfill of data with the previous organization structure.

- If you use AWS Organizations and you change organizations, then you can't get a backfill of data from prior to joining your current organization.

Why is my report file folder in Amazon S3 stored in an unnamed folder?

Any **/** character in the **Report path prefix** of your report generates an unnamed folder in your Amazon S3 bucket. To remove the unnamed folder in your next report update, edit your report settings and remove the **/** character from the **Report path prefix**. For instructions, see Editing your Cost and Usage Reports configuration.

Why can’t I select the option to include resource IDs on my report?

When you create your report, you can select the option to **Include resource ID**. If you create your report with **Report versioning** set to **Overwrite existing report**, then you can’t modify your **Include resource ID** selection after you create your report. To include resource IDs, you must create a new report and select the **Include resource ID** option.

Why don't my Cost and Usage Report queries for Amazon Athena work on Amazon Redshift, or my Amazon Redshift queries on Amazon Athena?

Amazon Athena and Amazon Redshift databases format Cost and Usage Report columns differently. Amazon Athena adds an underscore between words in the column name (line_item_normalized_usage_amount). Amazon Redshift adds an underscore between the column type and the attribute (lineitem_normalizedusageamount). Make sure to modify your queries to match the column name format in Amazon Athena or Amazon Redshift.

Why have the columns included in my report changed from a previous month

The columns that AWS includes in your report depend on your AWS usage. Every report includes columns with the **identity/**, **bill/**, and **lineItem/** prefixes:

- identity/LineItemId
- identity/TimeInterval
- bill/InvoiceId

- bill/BillingEntity
- bill/BillType
- bill/PayerAccountId
- bill/BillingPeriodStartDate
- bill/BillingPeriodEndDate
- lineItem/UsageAccountId
- lineItem/LineItemType
- lineItem/UsageStartDate
- lineItem/UsageEndDate
- lineItem/ProductCode
- lineItem/UsageType
- lineItem/Operation
- lineItem/AvailabilityZone
- lineItem/ResourceId
- lineItem/UsageAmount
- lineItem/NormalizationFactor
- lineItem/NormalizedUsageAmount
- lineItem/CurrencyCode
- lineItem/UnblendedRate
- lineItem/UnblendedCost
- lineItem/BlendedRate
- lineItem/BlendedCost
- lineItem/LineItemDescription
- lineItem/TaxType
- lineItem/LegalEntity

All other columns are included only if your monthly AWS usage generates data to populate those columns.

For example, your report includes **savingsPlan/** columns only if you used Savings Plans during that month.

Why don't my queries or tables work after the columns in my report changed?

The columns that AWS includes in your report depend on your AWS usage for the month. Because the columns included in your report can change, it’s a best practice to reference column names instead of column numbers in any custom queries or tables based on your report.

How do I query my report?

For detailed information about querying your Cost and Usage Report, see CUR Query Library Help in the AWS Well-Architected Labs website.

Where can I find the billing data for my Amazon EC2 Dedicated Host?

In the **ResourceID** column, look for the Dedicated Host ID rather than the instance ID. Because Dedicated Hosts are metered by Dedicated Host running hours, your report shows Dedicated Host usage by metered hours associated with the host ID.

How do I interpret the billing data for my Amazon EC2 Elastic IP addresses?

Amazon EC2 Elastic IP addresses are metered in aggregate. This means that each line item in your report doesn’t correspond with an individual Elastic IP address. Each line item represents the total number of chargeable hours. You can have one Elastic IP address assigned to a running instance at no charge. You’re charged per hour on a pro-rata basis for each additional Elastic IP address that you assign to the instance. Additionally, AWS charges an hourly fee for unassigned Elastic IP addresses.

How do unblended and blended rates or costs differ in consolidated billing?

With consolidated billing for AWS Organizations, unblended and blended rates or costs can help you understand how much an account's usage would cost for a standalone account versus a linked account in an organization. Some services offer pricing tiers that can lower unit costs as usage increases. Because AWS aggregates all usage for a service in an organization, individual accounts might access lower-priced tiers sooner when their usage is aggregated in an organization's monthly usage.

Unblended rates are the rates associated with an individual account's service usage. For a line item, the unblended cost is usage multiplied by the unblended rate. The unblended cost would be the cost of the account's usage if it were a standalone account. Blended rates are the rates associated with total usage in an organization averaged across accounts. For a line item, the blended cost is usage multiplied by the blended rate. The blended cost is the cost attributed to the account's usage as a linked account in an organization.

For more information and examples of calculating unblended and blended costs, see Understanding Consolidated Bills in the *AWS Billing User Guide*

Why do some line items in my report have a blended rate or blended cost of 0?

Amazon EC2 line items with a Reserved Instance discount have an blended rate of zero. For these line items, the **LineItemType** is **Discounted Usage**.

The blended cost is the usage multiplied by the blended rate. If the value for blended rate or usage is zero, then the blended cost is also zero.

How are All Upfront Reserved Instances amortized in my report?

Because All Upfront Reserved Instances are paid in full upfront, the amortized costs are reflected in your report as the upfront payment divided over the associated time period (one year or three years).

reservation/AmortizedUpfrontCostForUsage and **reservation/EffectiveCost** are the same rate for All Upfront Reserved Instances. This is because both columns are an equal division of the upfront payment for the Reserved Instance over the total hours of its term.

It's expected that your report has **RIFee** line items populated for All Upfront Reserved Instances, even though the **RIFee** is $0.00. These line items represent the recurring hourly costs for the month, and they have additional usage data in other columns. All Reserved Instances generate **RIFee** line items.

Troubleshooting carbon emissions data exports

Topics

- Why can't I create an export of the Carbon emissions table even though I have IAM permissions to use Data Exports and the CUR 2.0 table?

- Why can't I see carbon emissions data for some member accounts in my organization?
- Why is one of the files in my S3 bucket empty?
- Why does my S3 export show zero carbon emissions for some Regions and services when there is usage data?
- Is historical data backfill available in Data Exports for carbon emissions?
- How can I backfill data after changing my report settings or when a new methodology is released?
- Why can't I see historical data in my S3 bucket?
- Why don't I see the newly released columns in my export?

Why can't I create an export of the Carbon emissions table even though I have IAM permissions to use Data Exports and the CUR 2.0 table?

To access data in the Customer Carbon Footprint Tool or the Carbon emissions table, you need the IAM permission `sustainability:GetCarbonFootprintSummary`.

Why can't I see carbon emissions data for some member accounts in my organization?

If you're using a management (payer) account, you should automatically see carbon emissions data for your management account and all member (usage) accounts in the Carbon emissions table. No extra configuration is required.

However, there is a data lag of up to 21 days for carbon emissions data. For new member accounts, data won't appear in the management account’s carbon emissions data export until the export period that includes when the member account joined the organization. For example, if you linked a new member account in January, its data first appears in the February export.

Similarly, when a member account leaves the organization, its data continues to appear until the export period when it was removed.

Why is one of the files in my S3 bucket empty?

If your account doesn't have carbon emissions data for a given month, you'll receive a file in your S3 bucket for the given carbon model version and usage period, but the file will be empty.

Why does my S3 export show zero carbon emissions for some Regions and services when there is usage data?

If your total carbon emissions show as zero, it means they are lower than 0.0000005 MTCO2e, which is our display threshold.

Is historical data backfill available in Data Exports for carbon emissions?

Yes, upon creating an export you will receive data going back up to January 2022 with the first delivery and one month of data each month thereafter. If your account was created after January 2022, you'll receive carbon emissions estimates from your account creation date onward.

If you have an existing data export, you can request a backfill. See how in the question below.

How can I backfill data after changing my report settings or when a new methodology is released?

Open a support case to request a backfill of your carbon data. In your support case, make sure to provide the report name and the start date for your backfill. For more information on opening a case, see Getting help with your exports and reports.

Note that you can't get a backfill of carbon data for the following scenarios:

- You can't get a backfill for carbon data from before the date that you created the account.
- If you use AWS Organizations and the structure of your organization changed, such as which account is designated the management account, then you can't get a backfill of data with the previous organization structure.
- If you use AWS Organizations and you change organizations, then you can't get a backfill of data from prior to joining your current organization

Why can't I see historical data in my S3 bucket?

Your S3 bucket might be missing historical data for any of the following reasons:

- **No historical data exists:** If you have an account without historical carbon emissions estimates due to being a new account or recently changing membership in AWS Organizations, no historical data can be populated in your S3 bucket. If your account has been created after January 2022, you'll receive the carbon emissions estimates for the entire duration of your account being active.

- **Historical backfill is still in progress:** Historical data backfill by Data Exports can take up to 24 hours to complete. You can use the SDK/CLI to check if any backfill executions failed with the `ListExecutions` API for this export, or if they are still in progress. Wait a little longer or use `ListExecutions` to ensure the backfill is not in progress.
- **Historical backfill failed:** Historical data backfill may have failed to complete due to an internal error. You can come to this conclusion if it's been more than 24 hours and the backfill is not complete, or you can use the `ListExecutions` API in the SDK/CLI and look for any failed executions for this export. If you believe the backfill has failed, try creating a new export. If it fails a second time, we recommend reaching out to AWS Support.

Why don't I see the newly released columns in my export?

Existing exports continue with their original configuration and monthly updates until updated. To add new columns to an existing export, you must update your export configuration for future monthly exports (previously exported data remains unchanged). To backfill data with the new columns, you need to create a new export. This provides up to 38 months of historical data plus monthly updates.

Legacy Cost and Usage Reports

This section provides information on the Legacy Cost and Usage Reports feature.

Data Exports is the new and recommended way to receive your detailed cost and usage data from AWS. For more information, see Migrating from CUR to CUR 2.0 in Data Exports.

What are AWS Cost and Usage Reports?

AWS Cost and Usage Reports (AWS CUR) contains the most comprehensive set of cost and usage data available. You can use Cost and Usage Reports to publish your AWS billing reports to an Amazon Simple Storage Service (Amazon S3) bucket that you own. You can receive reports that break down your costs by the hour, day, or month, by product or product resource, or by tags that you define yourself. AWS updates the report in your bucket once a day in comma-separated value (CSV) format. You can view the reports using spreadsheet software such as Microsoft Excel or Apache OpenOffice Calc, or access them from an application using the Amazon S3 API.

AWS Cost and Usage Reports tracks your AWS usage and provides estimated charges associated with your account. Each report contains line items for each unique combination of AWS products, usage type, and operation that you use in your AWS account. You can customize the AWS Cost and Usage Reports to aggregate the information either by the hour, day, or month.

AWS Cost and Usage Reports can do the following:

- Deliver report files to your Amazon S3 bucket
- Update the report up to three times a day
- Create, retrieve, and delete your reports using the AWS CUR API Reference

How Cost and Usage Reports work

After you create a Cost and Usage Report, AWS sends your report to the Amazon S3 bucket that you specify. AWS updates your report at least once a day until your charges are finalized.

Your report files consist of a .csv file or a collection of .csv files and a manifest file. You can choose to configure your report data for integration with Amazon Athena, Amazon Redshift, or Quick.

Report timeline

After you create your report, it can take up to 24 hours for AWS to deliver the first report to your Amazon S3 bucket.

After delivery starts, AWS updates the report files at least once a day. Each report update in a given month is cumulative, so each version of the report includes all of the billing data for the month to date. The report updates that you receive throughout the month are estimates. The charges are subject to change as you continue to use your AWS services.

> **Note**
>
> Different AWS services provide your usage-based billing information at different times, so you might notice updates to a certain hour or day come in at different times.

AWS builds on previous reports until the end of the billing period. AWS finalizes your report's usage charges after issuing an invoice at the end of the month. After the end of the report billing period, AWS generates a new report for the next month with none of the information from the previous report.

After your report is finalized, AWS might update the report if AWS applies refunds, credits, or AWS Support fees to your usage for the month. Because Developer, Business, and Enterprise Support are calculated based on final usage charges, those are reflected on the sixth or seventh of the month for the prior month's Cost and Usage Report. AWS applies credits or refunds based on the terms of your agreement or contract with AWS.

Report files

Your report is a .csv file or a collection of .csv files stored in an Amazon S3 bucket. The number of files that your report generates depends on your selection for report versioning and the size of your report.

When you create a report, you can choose to create new report versions or overwrite the existing report version with every update. If you choose to create new report versions, then your report generates more files with every update.

The size of an individual report can grow to more than a gigabyte and might exceed the capacity of desktop spreadsheet applications to display every line. If a report is larger than most applications can handle (around 1 million rows), then AWS splits the report into multiple files that are stored in the same folder in the Amazon S3 bucket.

AWS also generates refunds into separate files. AWS issues refunds after the closing of a monthly bill.

For more information on report files, file-naming conventions, and versioning, see Understanding your report versions.

Report columns

Each report includes several columns with details about your AWS costs and usage. The columns that AWS includes in your report depend on your usage during the month.

Every report includes columns with the **identity/**, **bill/**, and **lineItem/** prefixes. All other columns are included only if your monthly AWS usage generates data to populate those columns.

For example, your report includes **savingsPlan/** columns only if you used Savings Plans during that month.

To learn more about the columns in your report, see the Data dictionary.

Using your report

You can download your report from the Amazon S3 console, query the report using Amazon Athena, or upload the report into Amazon Redshift or Quick.

- For more information about creating an Amazon S3 bucket and using Athena to query your data, see Querying Cost and Usage Reports using Amazon Athena.
- For more information about uploading to Amazon Redshift, see Loading report data to Amazon Redshift.
- For more information about uploading to Quick, see Loading report data to Amazon Quick.

Creating Cost and Usage Reports

Note

Data Exports enables you to create exports of the Cost and Usage Report (CUR) 2.0. This is the new and recommended way to receive your detailed cost and usage data from AWS. For more information, see Migrating from CUR to CUR 2.0 in Data Exports.

When you use billing transfer, you can create AWS Cost and Usage Report reports for your billing transfer views only from the Data Export page, not from the legacy AWS Cost and Usage Report page.

You can use the **Cost and Usage Reports** page in the Billing and Cost Management console to create Cost and Usage Reports. In the following section, you'll find information on how to get started with Cost and Usage Reports.

Topics

Setting up an Amazon S3 bucket for Cost and Usage Reports

To receive billing reports, you must have an Amazon S3 bucket in your AWS account to receive and store your reports. When creating a Cost and Usage Report in the billing console, you can select an existing Amazon S3 bucket that you own or create a new bucket. In either case, you'll be asked to review and confirm the application of the following default bucket policy. Editing this policy in the Amazon S3 console or changing the bucket owner after you've created a Cost and Usage Report will prevent AWS from being able to deliver your reports. Storing the billing reports data in your Amazon S3 bucket is billed at standard Amazon S3 rates. For more information, see Quotas and restrictions.

The following policy is applied to every bucket when creating a Cost and Usage Report:

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "billingreports.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketAcl",
                "s3:GetBucketPolicy"
            ],
            "Resource":"arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:cur:us-east-1:${AccountId}:definition/*",
                    "aws:SourceAccount": "${AccountId}"
                }
            }
        },
        {
            "Sid": "Stmt1335892526596",
            "Effect": "Allow",
            "Principal": {
                "Service": "billingreports.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:cur:us-east-1:${AccountId}:definition/*",
                    "aws:SourceAccount": "${AccountId}"
                }
            }
        }
    ]
}
```

This default policy helps ensure that the Cost and Usage Report data can be read by the bucket owner and confirms that the bucket is owned by the account that created the Cost and Usage Report. Specifically:

- Every time a Cost and Usage Report is delivered, AWS first confirms whether the bucket is still owned by the account which setup the report. If the bucket ownership has changed, the report will not be delivered. This helps to ensure the security of the account's billing data. This bucket policy allows AWS (`"Effect": "Allow"`) to check which account owns the bucket (`"Action": ["s3:GetBucketAcl", "s3:GetBucketPolicy"`).
- To deliver reports to your Amazon S3 bucket, AWS needs write permissions for that bucket. To do this, the bucket policy grants (`"Effect": "Allow"`) the AWS Cost and Usage Reports service (`"Service": "billingreports.amazonaws.com"`) permission to deliver (`"Action": "s3:PutObject"`) reports to the bucket you own (`"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"`).

  This bucket policy does not give AWS permissions to read or delete any objects in your bucket, including the Cost and Usage Reports after they've been delivered.

- For an Amazon S3 bucket that has ACL enabled, AWS further applies a `BucketOwnerFullControl` ACL to the reports when delivering them. By default, Amazon S3 objects, such as these reports, can only be read by the user or service principal who wrote them. To provide you or the bucket owner with permission to read the reports, AWS must apply the `BucketOwnerFullControl` ACL. The ACL grants the bucket owner `Permission.FullControl` for these reports. However, it's recommended to disable ACL and use an Amazon S3 bucket policy to control access. Note that Amazon S3 has changed the default settings and, for newly created buckets, ACLs are disabled by default. For more information, see Controlling ownership of objects and disabling ACLs for your bucket.

If you see an **Invalid bucket** error in your billing console for Cost and Usage Report, you should verify that this policy and bucket ownership haven't changed after report setup.

Creating reports

Note

Data Exports enables you to create exports of the Cost and Usage Report (CUR) 2.0. This is the new and recommended way to receive your detailed cost and usage data from AWS. For more information, see Migrating from CUR to CUR 2.0 in Data Exports.

You can use the **Cost and Usage Reports** page of the Billing and Cost Management console to create Cost and Usage Reports. You can create up to 10 reports for an individual AWS account.

Note

It can take up to 24 hours for AWS to start delivering reports to your Amazon S3 bucket. After delivery starts, AWS updates the AWS Cost and Usage Reports files at least once a day.

To create Cost and Usage Reports

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.

3. Choose **Create report**.

4. For **Report name**, enter a name for your report.

5. For **Report additional content**, select **Include resource IDs** to include the IDs of each individual resource in the report.

   Note

   Including resource IDs creates individual line items for each of your resources. This might increase the size of your Cost and Usage Reports files significantly, based on your AWS usage.

6. Select **Split cost allocation data** to include detailed cost and usage for shared resources (Amazon ECS and Amazon EKS).

   Note

   Including split cost allocation data creates individual line items for each of your resources (that is, ECS tasks and Kubernetes pods). This might increase the size of your Cost and Usage Reports files significantly, based on your AWS usage.

7. For **Data refresh settings**, select whether you want the AWS Cost and Usage Reports to refresh if AWS applies refunds, credits, or support fees to your account after finalizing your bill. When a report refreshes, a new report is uploaded to Amazon S3.

8. Choose **Next**.

9. For **S3 bucket**, choose **Configure**.

10. In the **Configure S3 bucket** dialog box, do one of the following:

    - Select an existing bucket.
    - Select **Create a bucket**, enter a bucket name, and then choose the Region where you want to create a new bucket.

11. Review the bucket policy, select **The following default policy will be applied to your bucket**, and then choose **Save**.

12. For **Report path prefix**, enter the report path prefix that you want prepended to the name of your report.

13. For **Time granularity**, choose one of the following:

   - **Hourly** if you want the line items in the report to be aggregated by the hour.
   - **Daily** if you want the line items in the report to be aggregated by the day.
   - **Monthly** if you want the line items in the report to be aggregated by month.

14. For **Report versioning**, choose whether you want each version of the report to overwrite the previous version of the report or to be delivered in addition to the previous versions.

    Overwriting reports can save on Amazon S3 storage costs. Delivering new report versions can improve auditability of billing data over time.

15. For **Report data integration**, select whether you want to enable your Cost and Usage Reports to integrate with Amazon Athena, Amazon Redshift, or Quick. The report is compressed in the following formats:

    - **Athena**: parquet format
    - **Amazon Redshift or Quick**: .gz compression

16. Choose **Next**.
17. After you have reviewed the settings for your report, choose **Review and Complete**.

You can always return to the **Cost and Usage Reports** page in the Billing and Cost Management console to see when your reports were last updated.

Viewing and managing reports

To view information about your Cost and Usage Report, use the Billing and Cost Management console. To view your report's files, you can use the Amazon S3 console.

Use the following procedures to find your report and report files.

To view your report details and files

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. From your list of reports, choose the name of the report that you want to view.
4. On the **Report Details** page, you can view the report's settings.
5. To view the report's files, note the **Report path prefix** on the **Report Details** page.

6. Choose the bucket name listed under **Amazon S3 bucket**. The link opens this bucket in the Amazon S3 console.
7. From the list of objects in the bucket, choose the folder named with the first part of the **Report path prefix** that you noted in step 5. For example, if your **Report path prefix** is **`example-report-prefix/example-report-name`**, then choose the folder named **example-report-prefix**.
8. From the list of objects in the folder, choose the folder named with the second part of the **Report path prefix** that you noted in step 5. For example, if your **Report path prefix** is **`example-report-prefix/example-report-name`**, then choose the folder named **example-report-name**. This folder contains your report files.

Viewing the latest report version

AWS updates your Cost and Usage Report at least once a day until your charges are finalized. When you create a report, you can choose to create new report versions or overwrite the existing report version with every update.

If you configured your report to create new report versions with every update, then use the **assemblyId** in the manifest file to find the latest report files.

To view your latest report files when you have multiple report versions

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. From your list of reports, choose the name of the report that you want to view.
4. On the **Report Details** page, note the **Report path prefix**.
5. Choose the bucket name listed under Amazon S3 bucket. The link opens this bucket in the Amazon S3 console.
6. From the list of objects in the bucket, choose the folder named with the first part of the **Report path prefix** that you noted in step 4. For example, if your **Report path prefix** is **`example-report-prefix/example-report-name`**, then choose the folder named **example-report-prefix**.
7. From the list of objects in the folder, choose the folder named with the second part of the **Report path prefix** that you noted in step 4. For example, if your **Report path prefix**

is **`example-report-prefix/example-report-name`**, then choose the folder named **example-report-name**.

8. Open the folder named with the latest billing period (in the YYYYMMDD-YYYYMMDD format).
9. Open the **`example-report-name`-Manifest.json** file.
10. At the top of the manifest file, note the **assemblyId**. The **assemblyId** value corresponds to the name of the folder with the latest report files.
11. Return to the Amazon S3 console page where you're viewing the folder named with the latest billing period.
12. Open the folder named with the **assemblyId** value that you noted in step 10. For example, if the **assemblyId** value is **`20210129T123456Z`**, then open the folder named **`20210129T123456Z/`**. This folder contains your latest report files.

Viewing your finalized report

After issuing your invoice at the end of the month, AWS finalizes your report's usage charges. To determine if a line item on your report is final, review the **bill/InvoiceId** column. If the line item is final, then the **bill/InvoiceId** column is populated with an AWS invoice ID. If the line item is not yet final, then the **bill/InvoiceId** column is blank.

To determine if your entire report is finalized, review the **bill/InvoiceId** column. If the report is final, then the **bill/InvoiceId** column is populated with invoice ID values. If the report is not yet final, then the **bill/InvoiceId** column is blank.

> **Note**
>
> After your report is finalized, AWS might update the report if AWS applies refunds, credits, or support fees to your usage for the month. Because Developer, Business, and Enterprise Support are calculated based on final usage charges, those are reflected on the sixth or seventh of the month for the prior month's report. AWS applies credits or refunds based on the terms of your agreement or contract with AWS.

Understanding your report versions

AWS updates your Cost and Usage Report at least once a day until your charges are finalized. When you create a report, you can choose to create new report versions or overwrite the existing report version with every update.

Your report files include a .csv file or a collection of .csv files and the manifest file. Your report can also include any additional files that support your data's integration with Amazon Athena, Amazon Redshift, or Quick.

The following sections describe the file organization and naming conventions based on the report versioning that you choose.

Cost and Usage Reports delivery timeline

During the report period, AWS delivers a new report and a new manifest file each time AWS updates the report. AWS builds on previous reports until the end of the billing period. After the end of the report billing period, AWS generates a new report with none of the information from the previous report.

Creating new Cost and Usage Report versions

When you choose to keep your previous Cost and Usage Reports, your AWS CUR uses the following Amazon S3 organization and naming conventions.

```
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<assemblyId>/<example-report-name>-<file-number>.csv.<zip|gz>
```

- `report-prefix` = The prefix that you assign to the report.
- `report-name` = The name that you assign to the report.
- `yyyymmdd-yyyymmdd` = The range of dates that the report covers. Reports are finalized at the end of the date range.
- `assemblyId` = An ID that AWS creates each time that the report is updated.
- `file-number` = If the update includes a large file, AWS might split it into multiple files. The `file-number` tracks the different files in an update.
- `csv` = The format of the report files.
- `zip` or `gz` = The type of compression applied to the report files.

For example, your report could be delivered as a collection of the following files.

```
<example-report-prefix>/<example-report-name>/20160101-20160131/<123456789>/<example-
report-name>-<1>.csv.<zip>
<example-report-prefix>/<example-report-name>/20160101-20160131/<123456789>/<example-
report-name>-<2>.csv.<zip>
<example-report-prefix>/<example-report-name>/20160101-20160131/<123456789>/<example-
report-name>-<3>.csv.<zip>
<example-report-prefix>/<example-report-name>/20160101-20160131/<123456789>/<example-
report-name>-Manifest.json
<example-report-prefix>/<example-report-name>/20160101-20160131/<example-report-name>-
Manifest.json
```

AWS delivers all reports in a report date range to the same `report-prefix/report-name/yyyymmdd-yyyymmdd` folder. AWS gives each report a unique ID and delivers it to the `assemblyId` subfolder in the date range folder. If the report is too large for a single file, the report is split into multiple files and delivered to the same `assemblyId` folder.

For more information on manifesting files when you keep a previous report, see Cost and Usage Reports manifest files

Overwriting previous Cost and Usage Reports

When you choose to overwrite your previous Cost and Usage Reports, your AWS CUR uses the following Amazon S3 organization and naming conventions.

```
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-
<file-number>.csv.<zip|gz>
```

- `report-prefix` = The prefix that you assign to the report.
- `report-name` = The name that you assign to the report.
- `yyyymmdd-yyyymmdd` = The range of dates that the report covers. AWS finalizes reports at the end of the date range.
-
- `file-number` = If the update includes a large file, AWS might split it into multiple files. The `file-number` tracks the different files in an update.
- `csv` = The format of the report files.
- `zip` or `gz` = The type of compression applied to the report files.

For example, your report could be delivered as a collection of the following files.

```
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-<1>.csv.<zip>
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-<2>.csv.<zip><example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-<3>.csv.<zip>
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-Manifest.json
```

Athena specifications

If you chose Athena support when you created your AWS CUR, the file naming conventions are the same as when you choose to overwrite your AWS CUR except for the format and compression. Athena AWS CUR files use `.parquet` instead. For example, your report could be delivered as a collection of the following files.

```
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>.parquet
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<cost_and_usage_data_status>
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-Manifest.json
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-create-table.sql
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/crawler-cfn.yml
```

CloudFormation specifications

In addition to the AWS CUR files, AWS also delivers an CloudFormation template that you can use to set up an CloudFormation stack that enables you to query Amazon S3 data using Athena. If you don't want to use the CloudFormation template, you can use the provided SQL to create your own Athena tables. For more information, see Querying Cost and Usage Reports using Amazon Athena.

Cost and Usage Reports manifest files

When AWS updates AWS CUR, AWS also creates and delivers manifest files that you can use for Amazon Athena, Amazon Redshift, or Quick.

Manifest files use the naming conventions, and lists the following:

- All of the detail columns that are included in the report to date
- A list of report files if the report was split into multiple files
- The time period covered by the report, and other information.

```
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<example-report-name>-
Manifest.json
<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<assemblyId>/<example-
report-name>-Manifest.json
<example-report-prefix>/<example-report-name>/<example-report-name>/year=2018/month=12/
<example-report-name>-Manifest.json
```

Creating new Cost and Usage Report versions

When you keep the previous Cost and Usage Reports, the manifest file is delivered to both the date range folder and the `assemblyId` folder. Each time AWS creates a new AWS CUR for a date range, it overwrites the manifest file stored in the date range folder with an updated manifest file. AWS delivers the same updated manifest file to the `assemblyId` folder along with the files for that update. Manifest files in the `assemblyId` folder aren't overwritten.

Overwriting the previous Cost and Usage Reports

When you overwrite the previous AWS CUR, the manifest file is delivered to the `month=mm` folder. The manifest file is overwritten along with the report files.

Amazon Redshift specifications

If you chose the option for Amazon Redshift support in your AWS CUR, AWS also creates and delivers a file with the SQL commands that you need to upload your report into Amazon Redshift. You can open the SQL file with a regular text editor. The SQL file uses the following naming convention.

```

<example-report-prefix>/<example-report-name>/yyyymmdd-yyyymmdd/<assemblyId>/<example-
report-name>-RedshiftCommands.sql
```

If you use the commands in the `RedshiftCommands` file, you don't need to open the `RedshiftManifest` file.

> **Important**
>
> The `manifest` file determines which report files the `copy` command in the `RedshiftCommands` file uploads. Deleting or removing the `manifest` file breaks the copy command in the `RedshiftCommands` file.

Amazon Athena specifications

If you chose the option for Amazon Athena support in your AWS CUR, AWS also creates and delivers multiple files to help set up all of the resources that you need. AWS delivers a CloudFormation template, a SQL file with the SQL to create your Athena table manually, and a file with the SQL to check your AWS CUR refresh status. These files use the following naming conventions.

```
<example-report-prefix>/<example-report-name>/<example-report-name>/yyyymmdd-yyyymmdd/
crawler-cfn.yml
<example-report-prefix>/<example-report-name>/<example-report-name>/yyyymmdd-yyyymmdd/
<example-report-name>-create-table.sql
<example-report-prefix>/<example-report-name>/<example-report-name>/yyyymmdd-yyyymmdd/
<cost_and_usage_data_status>
```

Editing your Cost and Usage Reports configuration

You can use the **Cost and Usage Reports** page in the Billing and Cost Management console to edit Cost and Usage Reports.

> **Note**
>
> Report names can't be edited. If you chose **Overwrite** for **Report versioning**, you're unable to edit the report name, whether the report includes resource IDs, time granularity, or the report versioning. If you delete a report set to **Overwrite** and create a new report with the same name, Amazon S3 bucket, and path prefix, your data could corrupt and become inaccurate.

To edit Cost and Usage Reports

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. Select the report that you want to edit and choose **Edit report**.
4. (Versioned reports only) For **Report additional content**, select **Include resource IDs** to include the IDs of each individual resource in the report.

   > **Note**
   >
   > Including resource IDs creates individual line items for each of your resources. This might increase the size of your Cost and Usage Reports files significantly, based on your AWS usage.

5. Select **Split cost allocation data** to include detailed cost and usage for shared resources (Amazon ECS and Amazon EKS).

   > **Note**
   >
   > Including split cost allocation data creates individual line items for each of your resources (that is, ECS tasks and Kubernetes pods). This might increase the size of your Cost and Usage Reports files significantly, based on your AWS usage.

6. For **Data refresh settings**, select whether you want the AWS Cost and Usage Reports to refresh if AWS applies refunds, credits, or support fees to your account after finalizing your bill. When a report refreshes, a new report is uploaded to Amazon S3.
7. Choose **Next**.
8. For **S3 bucket**, enter the name of the Amazon S3 bucket where you want the reports delivered.
9. Choose **Verify**.

   > **Note**
   >
   > The bucket must have appropriate permissions to be valid. For more information on adding permissions to the bucket, see Setting Bucket and Object Access Permissions in the *Amazon Simple Storage Service User Guide*.

10. For **Report path prefix**, enter the report path prefix that you want prepended to the name of your report.
11. (Versioned reports only) For **Time granularity**, choose one of the following:
    - **Hourly**: If you want the line items in the report to be aggregated by the hour.
    - **Daily**: If you want the line items in the report to be aggregated by the day.
    - **Monthly** if you want the line items in the report to be aggregated by month.
12. (Versioned reports only) For **Report versioning**, choose whether you want each version of the report to overwrite the previous version of the report, or to be delivered in addition to the previous versions.
13. For **Report data integration**, select whether you want to enable your AWS CUR to integrate with Amazon Athena, Amazon Redshift, or Quick. The report is compressed in the following formats:
    - **Athena**: Parquet format
    - **Amazon Redshift or Quick**: .gz compression
14. Choose **Save**.

Using Cost and Usage Reports for AWS Organizations

In AWS Organizations, both management accounts and member accounts can create Cost and Usage Reports. The IAM policies that allow or restrict the ability to create a report are the same for both types of accounts.

> **Note**
>
> The account that creates the Cost and Usage Report must also own the Amazon S3 bucket that AWS sends the reports to. You cannot configure a Cost and Usage Report to deliver to an Amazon S3 bucket owned by another account. For more information about Amazon S3 bucket setup requirements, see Setting up an Amazon S3 bucket for Cost and Usage Reports.

Managing Cost and Usage Reports as a member account

If you have permissions to create a Cost and Usage Report for a member account within an organization, you can create a report for only the member account's cost and usage data. The member account receives reports for its cost and usage during the time that the account has been a member of its current organization.

For example, say a member account leaves organization A and joins organization B on the 15th of the month. Then, the member account creates a report. Because the member account created a report after joining organization B, the member account's report for the month includes billing data for only the time that the account has been a member of organization B.

After a member account joins a new organization, the member account's cost and usage are recorded in the new organization's reports. This is the same outcome for a management account that converts to a member account and joins a new organization.

When a member account leaves an organization or converts into a standalone account, the member account can still access the previous reports as long as they have permissions to the Amazon S3 bucket where the previous reports are stored.

Managing Cost and Usage Reports as a management account

If you're an administrator of an AWS Organizations management account and you don't want member accounts to create a report, you can apply a service control policy (SCP) that prevents member accounts from creating reports. The SCP can prevent member accounts from creating new reports, but it doesn't delete reports created previously.

> ⓘ **Note**
>
> SCPs apply only to member accounts. To prevent a management account from creating a report, modify the IAM policies attached to the user roles in the management account.

For more information on consolidated billing, see Consolidated billing for AWS Organizations in the *AWS Billing User Guide*.

Querying Cost and Usage Reports using Amazon Athena

Amazon Athena is a serverless query service that you can use to analyze the data from your AWS Cost and Usage Reports (AWS CUR) in Amazon Simple Storage Service (Amazon S3) using standard

SQL. This helps you avoid having to create your own data warehouse solutions to query AWS CUR data.

We strongly recommend that you create both a new Amazon S3 bucket and a new AWS CUR report to use with Athena. AWS CUR supports only the Apache Parquet compression format for Athena and automatically overwrites previous reports that are stored in your S3 bucket.

This section outlines how to use Athena with Cost and Usage Reports. For a full description of the Athena service, see the Amazon Athena User Guide.

Topics

For a demonstration of querying reports using Athena, see the following video.

Analyze Cost and Usage Reports using Amazon Athena

Setting up Athena using CloudFormation templates

> **Important**
>
> CloudFormation doesn't support cross-Region resources. If you plan to use an CloudFormation template, you must create all resources in the same AWS Region. The Region must support the following services:
>
> - AWS Lambda
> - Amazon Simple Storage Service (Amazon S3)
> - AWS Glue
> - Amazon Athena

To streamline and automate integration of your Cost and Usage Reports with Athena, AWS provides an CloudFormation template with several key resources along with the reports that you set up for Athena integration. The CloudFormation template includes an AWS Glue crawler, an AWS Glue database, and an AWS Lambda event.

The Athena integration setup process using CloudFormation removes any Amazon S3 events that your bucket might already have. This can negatively affect any existing event-based processes that you have for an existing AWS CUR report. We strongly recommend that you create both a new Amazon S3 bucket and a new AWS CUR report to use with Athena.

Before you can use a CloudFormation template to automate Athena integration, make sure that you do the following:

- Create a new Amazon S3 bucket for your reports. For more information, see Creating a bucket in the *Amazon S3 User Guide*.
- Create a new report to use with Athena. During the setup process, for **Report data integration**, choose **Athena**.
- Wait for the first report to be delivered to your Amazon S3 bucket. It can take up to 24 hours for AWS to deliver your first report.

To use the Athena CloudFormation template

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. From the list of buckets, choose the bucket where you chose to receive your AWS CUR report.
3. Choose your report path prefix (`your-report-path-prefix/`). Then, choose your report name (`your-report-name/`).
4. Choose the `.yml` template file.
5. Choose **Object actions**, and then choose **Download as**.
6. Open the CloudFormation console at https://console.aws.amazon.com/cloudformation.
7. If you have never used CloudFormation before, choose **Create New Stack**. Otherwise, choose **Create Stack**.
8. Under **Prepare template**, choose **Template is ready**.
9. Under **Template source**, choose **Upload a template file**.
10. Choose **Choose file**.
11. Choose the downloaded `.yml` template, and then choose **Open**.
12. Choose **Next**.

13. For **Stack name**, enter a name for your template and choose **Next**.
14. Choose **Next**.
15. At the bottom of the page, select **I acknowledge that AWS CloudFormation might create IAM resources.**

    This template creates the following resources:

    - Three IAM roles
    - An AWS Glue database
    - An AWS Glue crawler
    - Two Lambda functions
    - An Amazon S3 notification
16. Choose **Create stack**.

To update the existing Athena CloudFormation template

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. From the list of buckets, choose the bucket where you chose to receive your AWS CUR report.
3. Choose your report path prefix (`*your-report-path-prefix/*`). Then, choose your report name (`*your-report-name/*`).
4. Choose the `.yml` template file.
5. Choose **Object actions**, and then choose **Download as**.
6. Open the CloudFormation console at https://console.aws.amazon.com/cloudformation.
7. Select the stack that was previously created, and then choose **Update**.
8. Under **Prepare template**, choose **Replace current template**.
9. Under **Template source**, choose **Upload a template file**.
10. Choose **Choose file**.
11. Choose the downloaded `.yml` template, and then choose **Open**.
12. Choose **Next**.
13. On the **Specify stack details** page, modify any details, and then choose **Next**.
14. Choose **Next**.
15. At the bottom of the page, select **I acknowledge that AWS CloudFormation might create IAM resources.**

16. Choose **Update stack**.

Setting up Athena manually

We strongly recommend that you use the AWS CloudFormation template to create your table instead of creating it yourself. The provided SQL query creates a table that covers only a single month of data, but the AWS CloudFormation template creates a table that can include multiple months and that updates automatically. For more information on how to set up the AWS CloudFormation template, see the section called "Setting up Athena with CloudFormation".

If you choose not to use the CloudFormation template to set up your Athena table, manually follow the steps below. You need to create a table before you can run SQL queries on your AWS CUR data. You will need to do this step at least once a month and the table only includes data from the current AWS CUR.

As part of the table creation process, AWS transforms the AWS CUR column names. For more information about the transformation process, see the section called "Column names".

- Creating an Athena table
- Creating a Cost and Usage Reports status table
- Uploading your report partitions

Creating an Athena table

AWS includes the SQL that you need to run to create this table in your AWS CUR bucket.

To create your Athena table

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. From the list of buckets, choose the bucket where you chose to receive your Cost and Usage Reports.
3. Navigate the path *your-report-prefix*-*your-report-name*-*path-to-report*.

   The exact path depends on whether your AWS CUR is set to overwrite previous versions. For more information, see Cost and Usage Reports delivery timeline.
4. Open the file *my-report-name*-create-table.sql.

5. Copy the SQL from the file, starting with `CREATE` and ending with `LOCATION 's3://`*`your-report-prefix`*`/`*`your-report-name`*`/`*`the-rest-of-the=path`*`'`. Take note of the first line, as you need the database name and table to create the Athena database.
6. Open the Athena console at https://console.aws.amazon.com/athena/.
7. In the **New query 1** query pane, paste the following SQL. For *`<database name>.<table name>`*, use the database and table name from the first line of the SQL that you copied.

   ```
   CREATE DATABASE <database name>
   ```

8. Choose **Run query**.
9. In the dropdown menu, choose the database that you just created.
10. In the **New query 1** query pane, paste the rest of the SQL from the SQL file.
11. Choose **Run query**.

After you create your table, you need to load your partitions before you can run a query. For more information, see [Uploading your report partitions]().

Creating a Cost and Usage Reports status table

AWS refreshes your AWS CUR multiple times a day. There isn't a way for Athena to tell when AWS is in the process of refreshing your report, which can lead to query results with a combination of old and new data. To mitigate this, create a table to track whether AWS is refreshing your Cost and Usage Reports and query that table to see if AWS is refreshing your data. You only need to create this table once. After that, AWS keeps the table up to date.

To create your refresh table

1. Open the Athena console at https://console.aws.amazon.com/athena/.
2. In the **New query 1** query pane, paste the following SQL.

   ```
   CREATE EXTERNAL TABLE IF NOT EXISTS cost_and_usage_data_status(
     status STRING)
   ROW FORMAT SERDE
     'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
   WITH SERDEPROPERTIES (
    'serialization.format' = '1'
   )
   LOCATION 's3://{S3_Bucket_Name}/{Report_Key}/cost_and_usage_data_status/'
   ```

3. Choose **Run query**.

To check whether AWS is refreshing your data, use the Athena console to run the following SQL query.

```
select status from cost_and_usage_data_status
```

Uploading your report partitions

To query your Cost and Usage Reports data, you need to upload the data into your Athena table. You must do this for each new AWS CUR report that AWS delivers to you.

To upload your latest partitions

1. Open the Athena console at https://console.aws.amazon.com/athena/.
2. Choose the vertical three dots next to your table name.
3. Choose **Load partitions**.

If you don't upload your partitions, Athena returns either no results or an error message that indicates missing data.

Running Amazon Athena queries

To run Athena queries on your data, first use the Athena console to check whether AWS is refreshing your data and then run your query on the Athena console. When you run your SQL, make sure that the correct database is selected from the dropdown list. You can use the following SQL to check the status.

```
select status from cost_and_usage_data_status
```

The two possible results are `READY` and `UPDATING`. If the status is `READY`, then you can query your Athena database. If the status is `UPDATING`, then Athena might return incomplete results.

After you've confirmed that AWS is refreshing your data, you can run your own queries. For example, the following query shows year-to-date costs by service for each month in the example database called `mycostandusage_parquet`. The following query shows 2018 year-to-date costs. Update the year to see current year-to-date costs.

```
SELECT line_item_product_code,
sum(line_item_blended_cost) AS cost, month
FROM mycostandusage_parquet
WHERE year='2018'
GROUP BY  line_item_product_code, month
HAVING sum(line_item_blended_cost) > 0
ORDER BY  line_item_product_code;
```

Column names

Athena column name restrictions are different from the Cost and Usage Reports column name restrictions. This means that when your AWS CUR data is uploaded into an Athena table, the column names change. AWS makes the following changes:

- An underscore is added in front of uppercase letters
- Uppercase letters are replaced with lowercase letters
- Any non-alphanumeric characters are replaced with an underscore
- Duplicate underscores are removed
- Any leading and trailing underscores are removed
- If the column name is longer than the allowed length of column names, underscores are removed from left to right

> **Note**
>
> After applying these rules, some of the resource tag columns will have duplicate names. AWS merges columns when there are more than one columns with the same name.

As examples, the column name `ExampleColumnName` becomes `example_column_name`, and the column name `Example Column Name` becomes `example_column_name`.

Loading report data to other resources

You can upload Cost and Usage Reports to Amazon Redshift and Amazon Quick to analyze your AWS cost and usage.

Topics

- Loading report data to Amazon Quick
- Loading report data to Amazon Redshift

Loading report data to Amazon Quick

You can upload your Cost and Usage Reports into Amazon Quick.

For more information about uploading to Quick, see Creating a Data Set Using Amazon S3 Files in the *Quick User Guide*.

Loading report data to Amazon Redshift

This section shows how you can upload AWS CUR to Amazon Redshift to analyze your AWS costs and usage.

> ⚠ **Important**
>
> Amazon Redshift columns aren't case sensitive and has stricter character limitations than user-defined tags. To prevent conflicts between Amazon Redshift and user-defined tags, AWS replaces your tags with the tags `userTag0`, `userTag1`, `userTag2`, etc. After you create an Amazon Redshift table and upload your report into it, you can create an Amazon Redshift table that maps the AWS-defined tags to your user-defined tags. The tag table allows you to look up your original tags.
> For example, if you have the tags `OWNER` and `Owner`, Amazon Redshift doesn't allow you to create a table with two columns named "owner". Instead, you create a report table with the columns `userTag0` and `userTag1` instead of `OWNER` and `Owner`, and then create a table with the columns `remappedUserTag` and `userTag`. The `remappedUserTag` column stores the AWS-defined tags `userTag0` and `userTag1`, and the `userTag` column stores your original tags, `OWNER` and `Owner`

AWS provides the commands to create your Amazon Redshift table, upload your report, create your tag table, and insert all of the tag rows into your tag table. The commands are provided to you in the `RedshiftCommands.sql` file that is stored alongside your manifest file in S3, and in the **Redshift file Helper file** in the Billing and Cost Management console. AWS also provides a RedshiftManifest file, which controls which report the commands in the RedshiftCommand file uploads. Deleting or removing the RedshiftManifest file breaks the copy command in the RedshiftCommands file.

To find the `RedshiftCommands.sql` file in the Billing and Cost Management console

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. Choose the report that you want to upload to Amazon Redshift.
4. Next to **You have enabled viewing reports in the following service(s):**, choose **Amazon Redshift**.
5. Copy the commands from the dialog box and paste them into your SQL client.

The following procedure assumes familiarity with databases and Amazon Redshift.

To upload an Cost and Usage Reports to Amazon Redshift

1. Create an Amazon Redshift cluster.

   For more information, see Creating a Cluster in the *Amazon Redshift Management Guide*.
2. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. Navigate to the Amazon S3 location where you store your AWS CUR.
4. Open the `RedshiftCommands.sql` file.

   The file contains customized commands to create an Amazon Redshift table, upload the AWS CUR from Amazon S3, and create a tag table that allows user-defined tags to be imported into Amazon Redshift.
5. In the `copy` command, replace *`<AWS_ROLE>`* with the ARN of an IAM role that has permissions to access the Amazon S3 bucket where you store your AWS CUR.
6. Replace *`<S3_BUCKET_REGION>`* with the Region your Amazon S3 bucket is in. For example, `us-east-1`.
7. Use a SQL client to connect to the cluster.

   For more information, see Accessing Amazon Redshift Clusters and Databases in the *Amazon Redshift Management Guide*.
8. Copy the SQL commands from the `RedshiftCommands.sql` file to your SQL client in the following order:

- create table - This command creates an Amazon Redshift table with a schema customized to match your report.
- copy - This command uses the provided IAM role to upload the AWS CUR files from S3 to Amazon Redshift.
- create tag table - This command creates a table that allows you to map AWS-defined tags to your user-defined tags.
- insert - These commands insert the user-defined tags into the tag table.

9. After you have copied all of the data from your AWS CUR into Amazon Redshift, you can query the data using SQL. For more information about querying data in Amazon Redshift, see Amazon Redshift SQL in the *Amazon Redshift Database Developer Guide*.

> **Note**
>
> The number of columns in Cost and Usage Reports can change from month to month, such as when a new cost allocation tag is created or a service adds a new product attribute. We recommend that you copy the data from your AWS CUR into a new table every month, and then copy the columns that interest you into a separate month-by-month table.

Configuring Cost and Usage Reports using Billing Conductor

You can create pro forma AWS Cost and Usage Reports (AWS CUR) for each billing group that you create in Billing Conductor. The pro forma AWS CUR has the same file format, granularity, and columns as the standard AWS CUR. Pro forma contains the most comprehensive set of cost and usage data available for a given period of time. For more information about Billing Conductor, see the *Billing Conductor User Guide*.

Topics

Understanding the differences between Billing Conductor AWS CUR and standard AWS CUR

There are a few differences between the standard Cost and Usage Reports and pro forma AWS CUR created using the Billing Conductor configuration.

- The standard AWS CUR computes the cost and usage for each account in your consolidated billing family. A pro forma AWS CUR per billing group only includes the accounts in the billing group at the time of computation.
- The standard AWS CUR populates the invoice column once and invoice is generated by AWS. A pro forma AWS CUR doesn't populate the invoice column. Currently, no invoice is generated, or issued by AWS based on pro forma billing data.

Create pro forma Cost and Usage Reports for a billing group

Use the following steps to generate a pro forma AWS CUR for a billing group.

> **Note**
>
> The legacy AWS Cost and Usage Report page supports reports only for billing group views. To create reports for billing transfer views, use the Data Export page.

To create pro forma Cost and Usage Reports for a billing group

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. On the top right of the **Report table**, choose **Settings**.
4. Turn on the **Pro forma** data view.
5. Choose **Enable**.
6. Choose **Create report**.
7. For **Report name**, enter a name for your report.
8. For **Data view**, choose **pro forma**.
9. Choose a billing group.

10. For **Additional report details**, choose **Include resource IDs** to include the IDs of each individual resources in the report.
11. For **Data refresh settings**, choose if you want Cost and Usage Reports to refresh if AWS applies refunds, credits, or support fees to your account after finalizing your bill. When a report refreshes, a new report is uploaded to Amazon S3.
12. Choose **Next**.
13. For **S3 bucket**, choose **Configure**.
14. In the **Configure S3 Bucket** dialog box, do one of the following:

    - Choose an existing bucket from the drop down list and choose **Next**.
    - Enter a bucket name and the AWS Region where you want to create a new bucket and choose **Next**.
15. Review the bucket policy, select **I have confirmed that this policy is correct**, and choose **Save**.
16. For **Report path prefix**, enter the report path prefix that you want prepended to the name of your report.

    This step is optional for Amazon Redshift or Quick, but required for Amazon Athena. If you don't specify a prefix, the default prefix is the name that you specified for the report in step 7, and the date range for the report in the following format: `/report-name/date-range/`
17. For **Time granularity**, choose one of the following:

    - **Hourly** if you want the line items in the report to be aggregated by the hour.
    - **Daily** if you want the line items in the report to be aggregated by the day.
18. For **Report versioning**, choose whether you want each version of the report to overwrite the previous version of the report, or to be delivered in addition to the previous versions.

    Overwriting reports can save on Amazon S3 storage costs. Delivering new report versions can improve auditability of billing data over time.
19. For **Report data integration**, choose whether you want to upload your Cost and Usage Reports to Amazon Athena, Amazon Redshift, or Quick. The report is compressed in the following formats:

    - **Athena**: parquet format
    - **Amazon Redshift or Quick**: .gz compression
20. Choose **Next**.

21. After you have reviewed the settings for your report, choose **Review and Complete**.

Data dictionary

Cost and Usage Reports contain details about your usage. The following sections list and describe a subset of columns that you see in your report, and the corresponding definitions.

To download the full list of the columns that can appear in AWS Cost and Usage Reports (AWS CUR) and the services that the columns apply to, download Column_Attribute_Service.zip. This comma-separated values (CSV) list includes **Identity**, **Bill**, **LineItem**, **Reservation**, **Pricing**, and **Product** columns.

Every Cost and Usage Report includes **Identity**, **Bill**, and **LineItem** columns. All other columns are included in your report only if your monthly AWS usage generates data to populate those columns.

Topics

Identity details

Columns under the **identity** header in AWS Cost and Usage Reports are static fields that appear in all Cost and Usage Reports.

You can use the identity line items in the report to find specific line items that have been split across multiple AWS CUR files. These include the following columns:

identity/LineItemId

- **Description:** This field is generated for each line item and is unique in a given partition. This does not guarantee that the field will be unique across an entire delivery (that is, all partitions in an update) of the AWS CUR. The line item ID isn't consistent between different Cost and Usage Reports and can't be used to identify the same line item across different reports.
- **Example:** A report created for November 29 can be large enough to require multiple files. The **LineItemId** is consistent between the November 29 AWS CUR files, but doesn't match the **LineItemId** for the same resource in the November 30 report.

identity/TimeInterval

- **Description:** The time interval that this line item applies to, in the following format: `YYYY-MM-DDTHH:mm:ssZ/YYYY-MM-DDTHH:mm:ssZ`. The time interval is in UTC and can be either daily or hourly, depending on the granularity of the report.
- **Example:** The **TimeInterval** `2017-11-01T00:00:00Z/2017-12-01T00:00:00Z` includes the entire month of November 2017.

Billing details

Columns under the **bill** header in AWS Cost and Usage Reports are static fields that appear in all Cost and Usage Reports. You can use the billing line items in the report to find details about the specific bill covered by the report, such as the charge type and the beginning and end of the billing period. This includes the following columns:

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

B

bill/BillingEntity

Helps you identify whether your invoices or transactions are for AWS Marketplace or for purchases of other AWS services. Possible values include:

- **AWS –** Identifies a transaction for AWS services other than in AWS Marketplace.
- **AWS Marketplace –** Identifies a purchase in AWS Marketplace.

bill/BillingPeriodEndDate

The end date of the billing period that is covered by this report, in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ`.

bill/BillingPeriodStartDate

The start date of the billing period that is covered by this report, in UTC. The format is `YYYY-MM-DDTHH:mm:ssZ`.

bill/BillType

The type of bill that this report covers. There are three bill types:

- **Anniversary –** Line items for services that you used during the month
- **Purchase –** Line items for upfront service fees
- **Refund –** Line items for refunds

I

bill/InvoiceId

The ID associated with a specific line item. Until the report is final, the **InvoiceId** is blank.

bill/InvoicingEntity

The AWS entity that issues the invoice. Possible values include:

- **Amazon Web Services, Inc. –** The entity that issues invoices to customer globally, where applicable.
- **Amazon Web Services India Private Limited –** The entity that issues invoices to customers based in India.
- **Amazon Web Services South Africa Proprietary Limited –** The entity that issues invoices to customers in South Africa.

P

bill/PayerAccountId

The account ID of the paying account. For an organization in AWS Organizations, this is the account ID of the management account.

Line item details

Columns under the **lineItem** header in AWS Cost and Usage Reports are static fields that appear in all Cost and Usage Reports. They cover all of the cost and usage information for your usage. This includes the following columns:

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

A

lineItem/AvailabilityZone

The Availability Zone that hosts this line item. For example, `us-east-1a` or `us-east-1b`.

B

lineItem/BlendedCost

The `BlendedRate` multiplied by the `UsageAmount`.

> **Note**
>
> **BlendedCost** is blank for line items that have a **LineItemType** of **Discount**. Discounts are calculated using only the unblended cost of a member account, aggregated by member account and SKU. As a result, **BlendedCost** is not available for discounts.

lineItem/BlendedRate

The `BlendedRate` is the average cost incurred for each SKU across an organization.

For example, the Amazon S3 blended rates are the total cost of storage divided by the amount of data stored per month. For accounts with RIs, the blended rates are calculated as the average costs of the RIs and the On-Demand Instances.

Blended rates are calculated at the management account level, and used to allocate costs to each member account. For more information, see Blended Rates and Costs in the *AWS Billing User Guide*.

C

lineItem/CurrencyCode

The currency that this line item is shown in. All AWS customers are billed in US dollars by default. To change your billing currency, see Changing which currency you use to pay your bill in the *AWS Billing User Guide*.

L

lineItem/LegalEntity

The Seller of Record of a specific product or service. In most cases, the invoicing entity and legal entity are the same. The values might differ for third-party AWS Marketplace transactions. Possible values include:

- **Amazon Web Services, Inc.** – The entity that sells AWS services.
- **Amazon Web Services India Private Limited** – The local Indian entity that acts as a reseller for AWS services in India.

lineItem/LineItemDescription

The description of the line item type. For example, the description of a usage line item summarizes what type of usage you incurred during a specific time period.

For size-flexible RIs, the description corresponds to the RI the benefit was applied to. For example, if a line item corresponds to a `t2.micro` and a `t2.small` RI was applied to the usage, the **lineItem/LineItemDescription** displays `t2.small`.

The description for a usage line item with an RI discount contains the pricing plan covered by the line item.

lineItem/LineItemType

The type of charge covered by this line item. Possible types are the following:

- **BundledDiscount** – A usage-based discount that provides free or discounted usage of a service or feature based on the usage of another service or feature.

- **Credit –** Any credits that AWS applied to your bill. See the **Description** column for details. AWS might update reports after they have been finalized if AWS applies a credit to your account for the month after finalizing your bill.
- **Discount –** Any discounts that AWS applied to your usage. This specific line item name may vary and require parsing based on the discount. For more information, refer to the **lineItem/LineItemDescription** column.
- **DiscountedUsage –** The rate for any instances for which you had Reserved Instance (RI) benefits.
- **Fee –** Any upfront annual fee that you paid for subscriptions. For example, the upfront fee that you paid for an **All Upfront RI** or a **Partial Upfront RI**.
- **Refund –** The negative charges that AWS refunded money for. Check the **Description** column for details. AWS might update reports after they have been finalized if AWS applies a refund to your account for the month after finalizing your bill.
- **RIFee –** The monthly recurring fee for subscriptions. For example, the recurring fee for **Partial Upfront RI**s, **No Upfront RI**s, and **All Upfront**s that you pay every month. Although the RIFee might be $0 for all upfront reservations, this line is still populated for those reservation types to provide other columns such as **reservation/AmortizedUpfrontFeeForBillingPeriod** and **reservation/ReservationARN**.
- **Tax –** Any taxes that AWS applied to your bill. For example, VAT or US sales tax.
- **Usage –** Any usage that is charged at On-Demand Instance rates.
- **SavingsPlanUpfrontFee –** Any one-time upfront fee from your purchase of an All Upfront or Partial Upfront Savings Plans.
- **SavingsPlanRecurringFee –** Any recurring hourly charges that correspond with your No Upfront or Partial Upfront Savings Plans. The Savings Plans recurring fee is initially added to your bill on the day that you purchase a No Upfront or Partial Upfront Savings Plans. After the initial purchase, AWS adds the recurring fee to the first day of each billing period thereafter.
- **SavingsPlanCoveredUsage –** Any on-demand cost that is covered by your Savings Plans. Savings Plans covered usage line items are offset by the corresponding Savings Plans negation items.
- **SavingsPlanNegation –** Any offset cost through your Savings Plans benefit that’s associated with the corresponding Savings Plans covered usage item.

For more information and examples of Savings Plans line items, see Understanding Savings Plans.

N

lineItem/NetUnblendedCost

The actual after-discount cost that you're paying for the line item. This column is included in your report only when your account has a discount in the applicable billing period.

lineItem/NetUnblendedRate

The actual after-discount rate that you're paying for the line item. This column is included in your report only when your account has a discount in the applicable billing period.

lineItem/NormalizationFactor

As long as the instance has shared tenancy, AWS can apply all Regional Linux or Unix Amazon EC2 and Amazon RDS RI discounts to all instance sizes in an instance family and AWS Region. This also applies to RI discounts for member accounts in an organization. All new and existing Amazon EC2 and Amazon RDS size-flexible RIs are sized according to a normalization factor, based on the instance size. The following table shows the normalization factor that AWS applies to each instance size.

Normalization factors for Amazon EC2 size-flexible RIs

Instance size	Normalization factor
`nano`	0.25
`micro`	0.5
`small`	1
`medium`	2
`large`	4
	8

Instance size	Normalization factor
xlarge	
2xlarge	16
4xlarge	32
8xlarge	64
10xlarge	80
16xlarge	128
32xlarge	256

lineItem/NormalizedUsageAmount

The amount of usage that you incurred, in normalized units, for size-flexible RIs. The **NormalizedUsageAmount** is equal to **UsageAmount** multiplied by **NormalizationFactor**.

O

lineItem/Operation

The specific AWS operation covered by this line item. This describes the specific usage of the line item. For example, a value of `RunInstances` indicates the operation of an Amazon EC2 instance.

P

lineItem/ProductCode

The code of the product measured. For example, Amazon EC2 is the product code for Amazon Elastic Compute Cloud.

R

lineItem/ResourceId

(Optional) If you chose to include individual resource IDs in your report, this column contains the ID of the resource that you provisioned. For example, an Amazon S3 storage bucket, an Amazon EC2 compute instance, or an Amazon RDS database can each have a resource ID. This field is blank for usage types that aren't associated with an instantiated host, such as data transfers and API requests, and line item types such as discounts, credits, and taxes. The following table shows a list of resource identifiers for common AWS services.

AWS resource identifiers

AWS service	Resource identifier
Amazon CloudFront	Distribution ID
Amazon CloudSearch	Search domain
Amazon DynamoDB	DynamoDB table
Amazon Elastic Compute Cloud - Amazon EBS	Amazon EBS volume
Amazon Elastic Compute Cloud	Instance ID
Amazon Elastic Compute Cloud - CloudWatch	CloudWatch charges for an instance ID
Amazon EMR	MapReduce cluster
Amazon ElastiCache	Cache cluster
Amazon OpenSearch Service	Search domain
Amazon Glacier	Vault

AWS service	Resource identifier
Amazon Relational Database Service	Database
Amazon Redshift	Amazon Redshift cluster
Amazon Simple Storage Service	Amazon S3 bucket
Amazon Virtual Private Cloud	VPN ID
AWS Lambda	Lambda function name

T

lineItem/TaxType

The type of tax that AWS applied to this line item.

U

lineItem/UnblendedCost

The `UnblendedCost` is the `UnblendedRate` multiplied by the `UsageAmount`.

lineItem/UnblendedRate

In consolidated billing for accounts using AWS Organizations, the unblended rate is the rate associated with an individual account's service usage.

For Amazon EC2 and Amazon RDS line items that have an RI discount applied to them, the `UnblendedRate` is zero. Line items with an RI discount have a `LineItemType` of `DiscountedUsage`.

lineItem/UsageAccountId

The account ID of the account that used this line item. For organizations, this can be either the management account or a member account. You can use this field to track costs or usage by account.

lineItem/UsageAmount

The amount of usage that you incurred during the specified time period. For size-flexible Reserved Instances, use the **reservation/TotalReservedUnits** column instead.

> **Note**
>
> Certain subscription charges will have a **UsageAmount** of `0`.

lineItem/UsageEndDate

The end date and time for the corresponding line item in UTC, exclusive. The format is `YYYY-MM-DDTHH:mm:ssZ`.

lineItem/UsageStartDate

The start date and time for the line item in UTC, inclusive. The format is `YYYY-MM-DDTHH:mm:ssZ`.

lineItem/UsageType

The usage details of the line item. For example, `USW2-BoxUsage:m2.2xlarge` describes an M2 High Memory Double Extra Large instance in the US West (Oregon) Region.

Reservation details

Columns under the **reservation** header in AWS Cost and Usage Reports provide details about reserved resources.

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

A

reservation/AmortizedUpfrontCostForUsage

- **Description:** The initial upfront payment for all upfront RIs and partial upfront RIs amortized for usage time. The value is equal to: `RIAmortizedUpfrontFeeForBillingPeriod` * The `normalized usage amount for DiscountedUsage line items` / `The normalized usage amount for the RIFee`. Because there are no upfront payments for no upfront RIs, the value for a no upfront RI is `0`. We do not provide this value for Dedicated Host reservations at this time. The change will be made in a future update.

- **Line items applicable:** DiscountedUsage
- **Sample values:** `0.05`, `0.17`, `0.15`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/AmortizedUpfrontFeeForBillingPeriod

- **Description:** Describes how much of the upfront fee for this reservation is costing you for the billing period. The initial upfront payment for all upfront RIs and partial upfront RIs, amortized over this month. Because there are no upfront fees for no upfront RIs, the value for no upfront RIs is `0`. We do not provide this value for Dedicated Host reservations at this time. The change will be made in a future update.
- **Line items applicable:** RIFee
- **Sample values:** `29.15`, `200.67`, `214.43`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/AvailabilityZone

- **Description:** The Availability Zone of the resource that is associated with this line item.
- **Line items applicable:** Fee, Refund, RIFee
- **Sample values:** `us-east-1`, `us-east-1b`, `eu-west-1b`, `ap-southeast-2a`
- **Services:**
  - Amazon EC2

E

reservation/EffectiveCost

- **Description:** The sum of both the upfront and hourly rate of your RI, averaged into an effective hourly rate. `EffectiveCost` is calculated by taking the `amortizedUpfrontCostForUsage` and adding it to the `recurringFeeForUsage`. For more information, see Amazon EC2 Reserved Instances Pricing.
- **Line items applicable:** DiscountedUsage
- **Sample values:** `0.23`, `0.68`, `0.10`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon ElastiCache
  - Amazon RDS

reservation/EndTime

- **Description:** The end date of the associated RI lease term.
- **Line items applicable:** RIFee
- **Sample values:** `2019-05-15T04:23:14.000Z`, `2020-02-08T17:32:15.000Z`, `2019-07-14T00:00:33.000Z`
- **Services:**
  - Amazon EC2

  - OpenSearch Service
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

M

reservation/ModificationStatus

- **Description:** Shows whether the RI lease was modified or if it is unaltered.
  - **Original:** The purchased RI was never modified.
  - **System:** The purchased RI was modified using the console or API.
  - **Manual:** The purchased RI was modified using AWS Support assistance.
  - **ManualWithData:** The purchased RI was modified using AWS Support assistance, and AWS calculated estimates for the RI.
- **Line items applicable:** RIFee
- **Sample values:** `Original`, `System`, `Manual`, `ManualWithData`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

N

reservation/NetAmortizedUpfrontCostForUsage

The initial upfront payment for All Upfront RIs and Partial Upfront RIs amortized for usage time, if applicable. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetAmortizedUpfrontFeeForBillingPeriod

The cost of the reservation's upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetEffectiveCost

The sum of both the upfront fee and the hourly rate of your RI, averaged into an effective hourly rate. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetRecurringFeeForUsage

The after-discount cost of the recurring usage fee. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetUnusedAmortizedUpfrontFeeForBillingPeriod

The net unused amortized upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetUnusedRecurringFee

The recurring fees associated with unused reservation hours for Partial Upfront and No Upfront RIs after discounts. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NetUpfrontValue

The upfront value of the RI with discounts applied. This column is included in your report only when your account has a discount in the applicable billing period.

reservation/NormalizedUnitsPerReservation

- **Description:** The number of normalized units for each instance of a reservation subscription.
- **Line items applicable:** RIFee
- **Sample values:** `1316`, `54.5`, `319`
- **Services:**
  - Amazon RDS

reservation/NumberOfReservations

- **Description:** The number of reservations that are covered by this subscription. For example, one RI subscription might have four associated RI reservations.
- **Line items applicable:** Fee, RIFee, Refund, Credit
- **Sample values:** `5`, `50`, `500`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

R

reservation/RecurringFeeForUsage

- **Description:** The recurring fee amortized for usage time, for partial upfront RIs and no upfront RIs. The value is equal to: `The unblended cost of the RIFee` * `The sum of the normalized usage amount of Usage line items` / `The normalized usage amount of the RIFee for size flexible Reserved Instances`. Because all upfront RIs don't have recurring fee payments greater than `0`, the value for all upfront RIs is `0`.
- **Line items applicable:** DiscountedUsage
- **Sample values:** `0.139`, `0.729`, `0.018`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon ElastiCache
  - Amazon RDS

reservation/ReservationARN

- **Description:** The Amazon Resource Name (ARN) of the RI that this line item benefited from. This is also called the "RI Lease ID". This is a unique identifier of this particular AWS Reserved Instance. The value string also contains the AWS service name and the Region where the RI was purchased.
- **Line items applicable:** Fee, RIFee, DiscountedUsage, Refund, Credit
- **Sample values:** `arn:aws:ec2:us-east-1:074108124787:reserved-instances/1d3fbc13-f181-4c40-9dd6-12b345678de9`, `arn:aws:ec2:us-east-1:499958231354:reserved-instances/be41234c3-b5c0-403e-a80c-1cfd12345678`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

S

reservation/StartTime

- **Description:** The start date of the term of the associated Reserved Instance.
- **Line items applicable:** RIFee
- **Sample values:** `2018-07-29T02:56:10.000Z`, `2017-08-21T15:58:47.000Z`, `2019-02-01T22:01:34.000Z`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/SubscriptionId

- **Description:** A unique identifier that maps a line item with the associated offer. We recommend you use the RI ARN as your identifier of an AWS Reserved Instance, but both can be used.
- **Line items applicable:** Fee, RIFee, DiscountedUsage, Refund, Credit, and Usage
- **Sample values:** `123456789`, `111122222`, `333344444`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

T

reservation/TotalReservedNormalizedUnits

- **Description:** The total number of reserved normalized units for all instances for a reservation subscription. AWS computes total normalized units by multiplying the `reservation/NormalizedUnitsPerReservation` with `reservation/NumberOfReservations`.
- **Line items applicable:** DiscountedUsage
- **Sample values:** `40320`, `3647.99`, `17928.77`
- **Services:**
  - Amazon EC2
  - Amazon RDS

reservation/TotalReservedUnits

- **Description:** `TotalReservedUnits` populates for both Fee and RIFee line items with distinct values.
  - Fee line items: The total number of units reserved, for the total quantity of leases purchased in your subscription for the entire term.

  This is calculated by multiplying the `NumberOfReservations` with `UnitsPerReservation`. For example, 5 RIs x 744 hours per month x 12 months = 44,640.
  - RIFee line items (monthly recurring costs): The total number of available units in your subscription, such as the total number of Amazon EC2 hours in a specific RI subscription.

    For example, 5 RIs x 744 hours = 3,720.
- **Line items applicable:** Fee, RIFee, Refund, Credit
- **Sample values:** `26208`, `98.19`, `15796`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

U

reservation/UnitsPerReservation

- **Description:** `UnitsPerReservation` populates for both Fee and RIFee line items with distinct values.
  - Fee line items: The total number of units reserved for the subscription, such as the total number of RI hours purchased for the term of the subscription.

    For example 744 hours per month x 12 months = 8,928 total hours/units.
  - RIFee line items (monthly recurring costs): The total number of available units in your subscription, such as the total number of Amazon EC2 hours in a specific RI subscription.

    For example, 1 unit x 744 hours = 744.
- **Line items applicable:** Fee, RIFee, Refund, Credit
- **Sample values:** `334.0`, `486.72`, `18455`
- **Services:**
  - Amazon EC2

  - OpenSearch Service
  - Amazon DynamoDB
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/UnusedAmortizedUpfrontFeeForBillingPeriod

- **Description:** The amortized-upfront-fee-for-billing-period-column amortized portion of the initial upfront fee for all upfront RIs and partial upfront RIs. Because there are no upfront payments for no upfront RIs, the value for no upfront RIs is `0`. We do not provide this value for Dedicated Host reservations at this time. The change will be made in a future update.
- **Line items applicable:** RIFee
- **Sample values:** `6.05`, `1.97`, `0.17`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/UnusedNormalizedUnitQuantity

- **Description:** The number of unused normalized units for a size-flexible Regional RI that you didn't use during this billing period.
- **Line items applicable:** RIFee
- **Sample values:** `25.00`, `3.50`, `274.33`
- **Services:**
  - Amazon RDS

reservation/UnusedQuantity

- **Description:** The number of RI hours that you didn't use during this billing period.

- **Line items applicable:** RIFee line item
- **Sample values:** `209.65110408`, `191.00000000`, `176.00000000`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/UnusedRecurringFee

- **Description:** The recurring fees associated with your unused reservation hours for partial upfront and no upfront RIs. Because all upfront RIs don't have recurring fees greater than `0`, the value for All Upfront RIs is `0`.
- **Line items applicable:** RIFee
- **Sample values:** `0.02971114`, `0.19190000`, `1.37280000`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon Redshift
  - Amazon ElastiCache
  - Amazon RDS

reservation/UpfrontValue

- **Description:** The upfront price paid for your AWS Reserved Instance. For no upfront RIs, this value is `0`.
- **Line items applicable:** RIFee
- **Sample values:** `150.00`, `1000.00`, `2000.00`
- **Services:**
  - Amazon EC2
  - OpenSearch Service

- Amazon Redshift
- Amazon ElastiCache
- Amazon RDS

Pricing details

Columns under the **pricing** header in AWS Cost and Usage Reports contain the prices for a line item. The pricing columns are based off of the AWS Price List Service API. AWS Price List Service API doesn't include Spot Instances, products in AWS Marketplace, upfront annual subscription fees (`Fee`), and monthly recurring fees (`RIFee`). The columns include but are not limited to the following:

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

L

pricing/LeaseContractLength

The length of time that your RI is reserved for.

O

pricing/OfferingClass

- **Description:** Describes the offering class of the Reserved Instance. When you purchase a Reserved Instance, you can choose between a Standard or Convertible offering class.
- **Sample values:** `Standard`, `Convertible`
- **Services:**
  - Amazon DynamoDB
  - Amazon EC2
  - Amazon ElastiCache
  - OpenSearch Service
  - Amazon RDS
  - Amazon Redshift

P

pricing/publicOnDemandCost

The total cost for the line item based on public On-Demand Instance rates. If you have SKUs with multiple On-Demand public costs, the equivalent cost for the highest tier is displayed. For example, services offering free-tiers or tiered pricing.

pricing/publicOnDemandRate

The public On-Demand Instance rate in this billing period for the specific line item of usage. If you have SKUs with multiple On-Demand public rates, the equivalent rate for the highest tier is displayed. For example, services offering free-tiers or tiered pricing.

pricing/PurchaseOption

How you chose to pay for this line item. Valid values are `All Upfront`, `Partial Upfront`, and `No Upfront`.

R

pricing/RateCode

A unique code for a product/ offer/ pricing-tier combination. The product and term combinations can have multiple price dimensions, such as low-use tier, and high-use tier.

pricing/RateId

The ID of the rate for a line item.

T

pricing/term

Whether your AWS usage is Reserved or On-Demand.

U

pricing/unit

The pricing unit that AWS used for calculating your usage cost. For example, the pricing unit for Amazon EC2 instance usage is in hours.

Product details

The **product** columns provide metadata about the product that incurred the expense, and the line item. The product columns are dynamic, and their visibility in Cost and Usage Reports depends on the usage of product in the billing period. The pricing columns are based off of the AWS Price List Service API. AWS Price List Service API doesn't include Spot Instances, products in AWS Marketplace, upfront annual subscription fees (`Fee`), and monthly recurring fees (`RIFee`).

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | XYZ

A

product/APICalls

- **Description:** Describes the number of APIs the DevOps Guru service uses to enable the DevOps Guru service.
- **Sample values:** `100`, `500`, `10000`
- **Services:**
  - Amazon DevOps Guru

product/attachmentType

- **Description:** Describes the type of attachment to Transit Gateway or Cloud WAN service.
- **Sample values:** `VPC`, `AWS Site-to-Site VPN`, `AWS DirectConnect`, `Connect`, `Transit Gateway`
- **Services:**
  - Amazon Virtual Private Cloud
  - AWS Cloud WAN

product/availability

- **Description:** Describes the availability of your various AWS storage options.
- **Sample values:** `99.99%`, `99.5%`
- **Services:**
  - Amazon Glacier
  - Amazon S3

  - AWS Elemental MediaStore
  - AWS RoboMaker

C

product/cacheType

- **Description:** Describes the provision opted by the customer on HDD-based file systems for a read-only SSD cache to improve performance for the frequently read data.

  For example, RC20 indicates the presence of a read-only SSD cache that is automatically sized to 20 percent of the file system's HDD storage capacity.
- **Sample values:** `RC20`, `N/A`
- **Services:**
  - Amazon FSx

product/capacitystatus

- **Description:** Describes the status of your capacity reservations.
- **Sample values:** `UnusedCapacityReservation`, `AllocatedCapacityReservation`, `Used`
- **Services:**
  - Amazon EC2

product/clockspeed

- **Description:** Describes the operating speed of your AWS instances.
- **Sample values:** `2.4 GHz`, `2.6 GHz`
- **Services:**
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon MQ
  - Amazon Neptune
  - Amazon RDS
  - AWS Database Migration Service

product/component

- **Description:** Maps to features in SageMaker AI.

  For example, if a user is running a SageMaker AI notebook, the product will have a component attribute of Notebook. If the user has deployed and hosted their model for inference, they will see product with component attribute of Hosting.
- **Sample values:** `Notebook`, `Hosting`
- **Services:**
  - Amazon SageMaker AI

D

product/databaseedition

- **Description:** Describes the database software suitable for different development, deployment scenarios, and specific application purposes.
- **Services:**
  - Amazon RDS

product/dataTransfer

- **Description:** An AWS data transfer occurs whenever data is moved to from AWS to the internet, or moved between AWS instances across their respective Regions or Availability Zones. Interregional and inter availability zone data transfers incur costs, metered per Gigabyte.
- **Services:**
  - AWS Systems Manager

product/dedicatedEbsThroughput

- **Description:** Describes the dedicated throughput between your instances (for example, Amazon EC2 instances and Amazon EBS volumes), with options between 500 and 10,000 megabits per second (Mbps) depending on the instance type used. The dedicated throughput minimizes contention between Amazon EBS I/O and other traffic from your EC2 instance, providing the best performance for your Amazon EBS volumes.
- **Sample values:** `200 Mbps`, `Upto 5000 Mbps`

- **Services:**
  - Amazon EC2
  - Amazon Neptune
  - Amazon RDS

product/deploymentoption

- **Description:** Describes where the infrastructure for the environment is located. The deployment models for AWS cloud are `public`, `on-premise`, and `hybrid`.
- **Sample values:** `Multi-AZ`, `Single-AZ`
- **Services:**
  - Amazon MQ
  - Amazon Neptune
  - Amazon RDS

product/description

- **Description:** The description of the specific AWS service.
- **Services:**
  - AWS CodePipeline
  - AWS Device Farm
  - AWS Elemental MediaConvert
  - AWS Elemental MediaStore

product/destinationCountryISOCode

- **Description:** Describes the destination Country ISO 3166-1 alpha-2 code to which the SMS was sent to.

  For reference, visit https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2.
- **Sample values:** `FR`, `CO`, `MA`, `KN`, `PL`, `LV`, `LA`, `GB`, `ID`, `KR`, `MY`, `BR`, `MM`, `CA`, `VN`, `BD`, `BJ`, `AU`, `HK`, `AM`, `CZ`, `UA`, `PH`, `TW`, `ES`, `DE`, `NG`, `FI`, `SG`, `TH`, `IL`, `TR`, `JP`, `IT`, `PR`, `RU`, `EE`
- **Services:**
  - Amazon Simple Notification Service

product/directconnectlocation

- **Description:** Specifies the location where a private dedicated network connection from the customer to AWS exists.
- **Sample values:** `Equinix DC1 - DC6`, `Equinix DC10 - DC11`, `Global Switch Singapore`
- **Services:**
  - Direct Connect

product/directorysize

- **Description:** The space on the disk that is used to store the meta information for the directory or folder.
- **Services:**
  - Direct Connect

product/directorytype

- **Description:** Specifies if the directory is a file or another directory.
- **Services:**
  - Direct Connect

product/directorytypedescription

- **Description:** The meaningful name given to the directory.
- **Services:**
  - Direct Connect

product/disableactivationconfirmationemail

- **Description:** Active or deactivate the ability to send an email to confirm the activation of a service.

product/durability

- **Description:** Describes the durability of objects over a given year.

- **Sample values:** `99.999999999%`, `N/A`, `99.99%`
- **Services:**
  - Amazon Glacier
  - Amazon S3
  - AWS Elemental MediaStore

E

product/ebsOptimized

- **Description:** Describes whether your Amazon EC2 instances are Amazon EBS–optimized.
- **Sample values:** `Yes`, `No`
- **Services:**
  - Amazon EC2

product/ecu

- **Description:** Describes the EC2 Compute Unit (ECU) that provides the relative measure of the integer processing power of an Amazon EC2 instance.
- **Sample values:** `9`, `100`, `variable`
- **Services:**
  - Amazon EC2
  - OpenSearch Service
  - Amazon GameLift Servers
  - Amazon Redshift

product/endpointtype

- **Description:** Describes the characteristics of the remote connection that a device connects to.

  For example, `REST` (representational state transfer) endpoints. A `REST` API (or `RESTful` API) is an application programming interface that conforms to the constraints of `REST` architectural style and you can interact with `RESTful` web services.
- **Sample values:** `Ipsec`, `Amazon SQS`, `AWS Lambda`

- **Services:**
  - Amazon SNS
  - Amazon VPC
  - Storage Gateway
  - Amazon Glacier

product/enhancedNetworkingSupported

- **Description:** Describes whether your instance supports enhanced networking. Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types.
- **Sample values:** `Yes`, No
- **Services:**
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon Neptune
  - Amazon RDS
  - AWS Database Migration Service

F

product/filesystemtype

- **Description:** Describes the details of the local or remote storage device, and specifications of the operating system.

product/findingGroup

- **Description:** Specifies whether a finding stored in Security Hub is paid or free. If free, the reason may also be specified.
- **Sample values:** `FreeFindingsIngestion-CrossRegion`, `FreeFindingsIngestion-FreeTier`, `FreeFindingsIngestion-FreeTrial`, `PaidFindingsIngestion`
- **Services:**
  - AWS Security Hub

product/findingSource

- **Description:** Specifies whether a finding was generated by a Security Hub control or by other partner security product.
- **Sample values:** `SecurityHubProduct`, `OtherProduct`
- **Services:**
  - AWS Security Hub

product/freeUsageIncluded

- **Description:** Free usage under AWS Free Tier is calculated each month across all Regions, and automatically applied to your bill. For example, you receive 750 Amazon EC2 Linux Micro Instance hours for free across all of the Regions you use. Not 750 hours per Region.
- **Services:**
  - Amazon Inspector

product/fromLocation

- **Description:** Describes the location where the usage originated from.
- **Sample values:** `External`, `US East (N. Virginia)`, `Global`
- **Services:**
  - Amazon CloudFront
  - AWS DataTransfer

product/fromRegionCode

- **Description:** Describes the source Region code for the AWS service. For more information, see product/regioncode.
- **Sample values:** `ap-northeast-1`
- **Services:**
  - Amazon RDS
  - Amazon EC2
  - Amazon VPC
  - Direct Connect

product/fromLocationType

- **Description:** Describes the location type where the usage originated from.
- **Sample values:** `AWS Region`, `AWS Edge Location`
- **Services:**
  - Direct Connect
  - AWS Elemental MediaConnect
  - Amazon CloudFront
  - Amazon Lightsail
  - AWS Shield

G

product/gpu

- **Description:** Describes the number of GPUs.
- **Sample values:** `16`, `32`
- **Services:**
  - Amazon SageMaker AI
  - Amazon EC2

product/gpuMemory

- **Description:** Describes your GPU memory details.
- **Sample values:** `16`, `32`
- **Services:**
  - Amazon SageMaker AI
  - Amazon EC2

product/group

- **Description:** A construct of several products that are similar by definition, or grouped together. For example, the Amazon EC2 team can categorize their products into shared instances, dedicated host, and dedicated usage.

- **Services:**
  - AWS Certificate Manager
  - AWS CodeCommit
  - AWS Glue
  - AWS IoT Analytics
  - AWS Lambda

product/groupdescription

- **Description:** A simplified name given to a product group.
- **Services:**
  - AWS Budgets
  - AWS Certificate Manager
  - AWS Lambda
  - Amazon SQS

I

product/insightsType

- **Description:** Indicates the type of Insight event generated.
- **Sample values:** `APICallVolume`
- **Services:**
  - CloudTrail

product/instance

- **Description:** An Amazon EC2 instance is a virtual server in Amazon Elastic Compute Cloud (Amazon EC2) for running applications on the AWS infrastructure. You can choose an AMI provided by AWS, the user community, or through the AWS Marketplace.
- **Sample values:** `T3`
- **Services:**
  - Amazon EC2

product/instanceFamily

- **Description:** Describes your Amazon EC2 instance family. Amazon EC2 provides you with a large number of options across 10 different instance types, each with one or more size options, organized into distinct instance families optimized for different types of applications.
- **Sample values:** `General Purpose`, `Memory Optimized`, `Accelerated Computing`
- **Services:**
  - Amazon EC2
  - Amazon RDS
  - OpenSearch Service
  - Amazon ElastiCache
  - Amazon EMR

  and more. For the full service list, download Column_Attribute_Service.zip.

product/instanceSize

- **Description:** Indicates the instance size of a resource.
- **Sample values:** `2vCPU`, `4vCPU`, `8vCPU`, `16vCPU`
- **Services:**
  - Amazon CodeCatalyst

product/instanceType

- **Description:** Describes the instance type, size, and family, which define the CPU, networking, and storage capacity of your instance.
- **Sample values:** `t2.small`, `m4.xlarge`, `t2.micro`, `m4.large`, `t2.large`
- **Services:**
  - Amazon EC2
  - Amazon RDS
  - OpenSearch Service
  - Amazon ElastiCache
  - Amazon EMR

and more. For the full service list, download Column_Attribute_Service.zip.

product/instanceTypeFamily

- **Description:** The instance family that is associated with the given usage.
- **Sample values:** `t2`, `m4`, `m3`
- **Services:**
  - Amazon DocumentDB
  - Amazon RDS

product/integratingApi

- **Description:** Application integration on AWS using service like Amazon API Gateway or no-code integration using Amazon AppFlow.

product/integratingService

- **Description:** Application integration on AWS is a suite of services used to communicate between decoupled components within micro services, distributed systems, and serverless applications. You don’t need to refactor your entire architecture. Decoupling applications at any scale can reduce the impact of changes, making it easier to update, and faster to release new features.

product/intelAvxAvailable

- **Description:** Describes whether your process has the Intel Advanced Vector Extension instruction set.
- **Sample values:** `Yes`, No.
- **Services:**
  - Amazon EC2

product/intelAvx2Available

- **Description:** Describes whether your process has the Intel Advanced Vector Extension instruction set two.

- **Sample values:** `Yes`, `No`
- **Services:**
  - Amazon EC2

product/intelTurboAvailable

- **Description:** Describes whether your core is allowed to use Intel Turbo Technology to increase frequency.
- **Sample values:** `Yes`, `No`
- **Services:**
  - Amazon EC2

product/invocation

- **Description:** Describes the invocations that EventBridge Scheduler makes to an API or service.
- **Sample values:** `Scheduled Invocation`
- **Services:**
  - Amazon CloudWatch Events

L

product/licenseModel

- **Description:** Describes the license model for your instance.
- **Sample value:** `license-included`, `bring-your-own-license`, `general-public-license`
- **Services:**
  - Amazon AppStream
  - Amazon EC2
  - Amazon MQ
  - Amazon Neptune
  - Amazon RDS

product/location

- **Description:** Describes the Region that your Amazon S3 bucket resides in.
- **Sample values:** `Asia Pacific (Mumbai)`, `Asia Pacific (Seoul)`, `Canada (Central)`, `EU (London)`, `US West (Oregon)`
- **Services:**
  - Amazon EC2
  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

product/locationType

- **Description:** Describes the endpoint of your task.
- **Sample values:** `AWS Region`, `AWS Edge Location`, `Other`
- **Services:**
  - Amazon EC2
  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

product/logsDestination

- **Description:** The `AWS::Logs::Destination` resource specifies a CloudWatch logs destination. A destination includes a physical resource (for example, Amazon Kinesis data stream) and you can subscribe the resource to a stream of log events.
- **Sample values:** `AWS Region`, `AWS Edge Location`, `Other`
- **Services:**

- Amazon EC2
- AWS Certificate Manager
- Amazon S3
- Amazon RDS
- Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

M

product/maxIopsBurstPerformance

- **Description:** Describes the max IOPS burst performance of your Amazon EBS volume.
- **Sample value:** `3000 IOPS for volumes <= 1TB`
- **Services:**
  - Amazon EC2

product/maxIopsvolume

- **Description:** Describes maximum input/output per second of your Amazon EBS volume.
- **Sample value:** `16,000 (maxiops for a General Purpose SSD (gp2))`
- **Services:**
  - Amazon EC2

product/maxThroughputvolume

- **Description:** Describes the max network throughput volume of your Amazon EBS volume.
- **Sample values:** `500 MiB/s`, `250 MiB/s`, `1000 MiB/s`, `40 - 90 MB/sec`
- **Services:**
  - Amazon EC2
  - Amazon SageMaker AI

product/memory

- **Description:** The placeholder electronics for instructions and data a computer needs to respond quickly. Computer bytes indicate the storage units.
- **Services:**
  - AWS Database Migration Service
  - DynamoDB Accelerator
  - Amazon DocumentDB
  - Amazon EC2

product/messageCountfee

- **Description:** Describes the type of metering usage, denoting whether the usage represents the number of messages or fees charged.
- **Sample values:** `CarrierFeeCount`, `MessageFees`, `MessageCount`, `CarrierFees`
- **Services:**
  - Amazon Simple Notification Service

product/messageType

- **Description:** Describes the type of SMS message. Note that SNS supports only Outbound SMS.
- **Sample values:** `OutboundSMS`
- **Services:**
  - Amazon Simple Notification Service

N

product/networkPerformance

- **Description:** Describes the network throughput of your Amazon EC2 instances.
- **Sample values:** `moderate`, `high`, `up to 10 GB`
- **Services:**
  - Amazon EC2
  - Amazon RDS

  - Amazon ElastiCache
  - Amazon SageMaker AI
  - AWS Database Migration Service

  and more. For the full service list, download Column_Attribute_Service.zip.

product/normalizationSizeFactor

- **Description:** Describes the normalization factor of the instance size.
- **Sample values:** `nano - 0.25`, `micro - 0.5`, `medium - 2`, `xlarge - 8`, `16xlarge - 128`
- **Services:**
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon MQ
  - Amazon Neptune
  - Amazon RDS

O

product/operatingSystem

- **Description:** Describes the operating system of your Amazon EC2 instance.
- **Sample values:** `Amazon Linux`, `Ubuntu`, `Windows Server`, `Oracle Linux`, `FreeBSD`
- **Services:**
  - Amazon AppStream
  - Amazon EC2
  - Amazon GameLift Servers
  - Amazon Lightsail
  - Amazon WorkSpaces
  - AWS CodeBuild

product/operation

- **Description:** Describes the specific AWS operation that this line item covers.
- **Sample values:** `RunInstances` (indicates the operation of an Amazon EC2 instance)
- **Services:**
  - Amazon EC2
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB
  - Amazon CloudWatch
  - Amazon Redshift

  and more. For the full service list, download Column_Attribute_Service.zip.

product/originationIdType

- **Description:** Describes the type of origination ID used when sending SMS messages.
- **Sample values:** `Sharedroute`
- **Services:**
  - Amazon Simple Notification Service

product/osType

- **Description:** Describes the operating system of the resource.
- **Sample values:** `Dev Environment`, `Linux`, `Linux ARM64`, `Windows`
- **Services:**
  - Amazon CodeCatalyst

P

product/parameterType

- **Description:** Use parameters in CloudFormation to enter custom values to your template when you create or update a stack. For Example, `InstanceTypeParameter`. You can use this parameter to specify the Amazon EC2 instance type when you create or update the stack.

product/physicalCores

- **Description:** Describes the number of physical cores an instance provides.
- **Sample values:** `4`, `8`
- **Services:**
  - Amazon EC2

product/physicalProcessor

- **Description:** Describes the processor on your Amazon EC2 instance.
- **Sample values:** `High Frequency Intel Xeon E7-8880 v3 (Haswell)`, `Intel Xeon E5-2670`, `AMD EPYC 7571`
- **Services:**
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon Neptune
  - Amazon RDS
  - AWS Database Migration Service

product/platoClassificationType

- **Description:** Tiered per object pricing for data annotation workflow routing.
- **Sample values:** `LabeledObject`, `3DLabeledObjectMultiFrame`, `3DLabeledObject`, `Processing:VolumeUsage`
- **Services:**
  - Amazon SageMaker AI

product/pricingUnit

- **Description:** The smallest billing unit for an AWS service. For example, 0.01c per API call.
- **Services:**
  - Directory Service

product/primaryplaceofuse

- **Description:** The primary business or residential street address location where a customer's use of the service primarily occurs.

product/processorArchitecture

- **Description:** Describes your processor architecture.
- **Sample values:** `32-bit`, `64-bit`
- **Services:**
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon Neptune
  - Amazon RDS
  - AWS Database Migration Service

product/processorFeatures

- **Description:** Describes the processor features of your instances.
- **Sample values:** `Intel AVX`, `Intel AVX2`, `Intel AVX512`, `Intel Turbo`
- **Services:**
  - AWS Database Migration Service
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon Neptune
  - Amazon RDS

product/ProductFamily

- **Description:** The category for the type of product.
- **Sample values:** `Alarm`, `AWS Budgets`, `Stopped Instance`, `Storage Snapshot`, `Compute`
- **Services:**
  - Amazon EC2

  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

product/ProductName

- **Description:** The full name of the AWS service. Use this column to filter AWS usage by AWS service.
- **Sample values:** `AWS Backup`, `AWS Config`, `Amazon Registrar`, `Amazon Elastic File System`, `Amazon Elastic Compute Cloud`

product/productSchemaDescription

- **Description:** A blueprint of how your product is constructed. This contains the various attributes that make up your product.

product/provisioned

- **Description:** Indicates whether Amazon EBS usage was related to provisioned Amazon EBS storage.
- **Sample values:** `Yes`, `No`
- **Services:**
  - Amazon EC2
  - Amazon MQ

product/provisioningType

- **Description:** Describes whether the resources were deployed on-demand or pre-provisioned.
- **Sample values:** `On-Demand`, `Pre-Provisioned`
- **Services:**
  - Amazon CodeCatalyst

product/PurchaseOption

- **Description:** Describes the available purchasing models for an AWS service. For example, AWS provides four main Amazon EC2 instance purchasing options: `On-Demand`, `Reserved Instances`, `Spot Instances`, with the added option of `Dedicated Hosts`.

product/purchaseterm

- **Description:** In Amazon EC2, this specifies a commitment to a consistent instance configuration. This includes instance type and Region for a period of 1 to 3 years.

R

product/region

- **Description:** The geographical area that hosts your AWS services. Use this field to analyze spend across a particular Region.
- **Sample values:** `eu-west-3`, `us-west-1`, `us-east-1`, `ap-northeast-2`, `sa-east-1`
- **Services:**
  - Amazon EC2
  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

product/regioncode

- **Description:** A Region is a physical location around the world where data centers are clustered. AWS calls each group of logical data centers an Availability Zone (AZ). Each AWS Region consists of multiple, isolates, and physically separate AZs within a geographical area. The Region code attribute has the same name as an AWS Region, and specifies where the AWS service is available.
- **Sample values:** `us-west-2`, `us-east-1`, `ap-southeast-2`
- **Services:**

  - Amazon SageMaker AI

product/replicationType

- **Description:** Specifies that the service is free to use. For example, AWS Server Migration Service is free to use, and you only pay for the storage resources used during the migration process.
- **Sample values:** `Free`
- **Services:**
  - AWS Application Migration Service

product/resourceAssessment

- **Description:** A process that collects, stores, and manages evidence. You can use this to assess risk and compliance with industry standards and regulations.
- **Sample values:** `All assessment`
- **Services:**
  - AWS Audit Manager

product/resourcePriceGroup

- **Description:** Describes the resource type, the resource, and the price group (the price we charge for monitoring; there are currently two price classes: A and B). Therefore, as an example, if we were monitoring an RDS resource, the resource type would be RDS (the "product"), the resource would be instance, and the price group would be B.
- **Sample values:** `RDS-DBInstance-GroupB`
- **Services:**
  - Amazon DevOps Guru

product/routeType

- **Description:** Describes the type of SMS route used. Only Standard applies for now.
- **Sample values:** `Standard`
- **Services:**
  - Amazon Simple Notification Service

S

product/servicecode

- **Description:** This identifies the specific AWS service to the customer as a unique short abbreviation.
- **Sample values:** `Amazon EC2`, `AWS KMS`
- **Services:**
  - AWS Budgets
  - AWS Backup
  - AWS Certificate Manager
  - AWS Cloud Map
  - AWS CloudTrail

product/servicename

- **Description:** A simplified description about the AWS service.
- **Services:**
  - Amazon EC2 Budgets
  - Amazon ECR
  - Amazon ECS
  - Amazon EFS
  - Amazon Elastic Inference
  - Amazon EKS

product/singleOrDualPass

- **Description:** Terms used to decide the type of encoding that happens for videos. In one-pass encoding, the encoding is done in the first pass itself. For 2-pass encoding, the file is analyzed thoroughly in the first pass, and a intermediate file is created. In the second pass the encoder finds the intermediate file and allocates bits. The actual encoding takes place in the second pass.
- **Services:**
  - AWS Elemental MediaConvert

product/sizeFlex

- **Description:** Describes whether a normalized benefit of the RI can be applied to other instance sizes within the Region and instance family.
- **Sample values:** `true`, `false`
- **Services:**
  - Amazon Elastic Compute Cloud

product/sku

- **Description:** A unique code for a product. The SKU is created by combining the `ProductCode`, `UsageType`, and `Operation`. For size-flexible RIs, the SKU uses the instance that was used. For example, if you used a `t2.micro` instance and AWS applied a `t2.small` RI discount to the usage, the line item SKU is created with the `t2.micro`.
- **Sample values:** `FFNT87MQSCR328W6`, `VBYCEU494XUAHCA7`
- **Services:**
  - Amazon EC2
  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

product/storage

- **Description:** Describes the disk storage attached to your instance.
- **Sample values:** `60GB`, `True`, `EBS Only`, `1 x 900 NVMe SSD`, `1 x 150 NVMe SSD`
- **Services:**
  - Amazon EC2
  - Amazon RDS
  - Amazon Redshift
  - OpenSearch Service
  - Amazon WorkSpaces

and more. For the full service list, download Column_Attribute_Service.zip.

product/storageclass

- **Description:** Describes the storage class of your Amazon S3 bucket.
- **Sample values:** `Archive`, `General Purpose`, `Infrequent Access`, `Intelligent-Tiering`, `Non-Critical Data`
- **Services:**
  - AWS Elemental MediaStore
  - AWS Storage Gateway
  - Amazon Cloud Directory
  - Amazon EFS
  - Amazon MQ
  - Amazon S3

product/storagemedia

- **Description:** A storage medium is any technology, including device and material used to place, keep, and retrieve electronic data.
- **Services:**
  - AWS Database Migration Service
  - Amazon CloudWatch
  - Amazon DocumentDB
  - Amazon EC2
  - Amazon ES

product/storagetype

- **Description:** Describes how and where the information is stored by a computer. This might be internal or external to a computer, server, or computing device.
- **Sample values:** `Amazon S3`, `SSD`, `SSD-backed`
- **Services:**

  - AWS Backup
  - Amazon ECR

T

product/tenancy

- **Description:** The type of tenancy allowed on the Amazon EC2 instance.
- **Sample values:** `Dedicated`, `Reserved`, `Shared`, `NA`, `Host`
- **Services:**
  - Amazon EC2
  - Amazon ECS

product/throughputCapacity

- **Description:** Describes the Speed at which the file server hosting the file system can serve file data. For Amazon FileCache, the value will be 1000 only.
- **Sample values:** `12`, `40`, `50`, `100`, `125`, `250`, `500`, `1000`
- **Services:**
  - Amazon FileCache
  - Amazon FSx

product/tier

- **Description:** With AWS, you can get volume based discounts and savings as your usage increases. For services like Amazon S3, pricing is tiered. This means the more you use, the less you pay per GB. AWS provides options to acquire services that assist your business needs.
- **Services:**
  - AWS Elemental MediaConvert

product/toLocation

- **Description:** Describes the location usage destination.
- **Sample values:** `External`, `US East (N. Virginia)`

- **Services:**
  - Amazon CloudFront
  - AWS Data Transfer

product/toLocationType

- **Description:** Describes the destination location of the service usage.
- **Sample values:** `AWS Region`, `AWS Edge Location`
- **Services:**
  - Direct Connect
  - AWS Elemental MediaConnect
  - AWS Shield
  - Amazon CloudFront
  - Amazon Lightsail

  and more. For the full service list, download Column_Attribute_Service.zip.

product/toRegionCode

- **Description:** Describes the source Region code for the AWS service. For more information, see product/regioncode.
- **Sample values:** `eu-west-1`
- **Services:**
  - Amazon RDS
  - Amazon EC2
  - Amazon VPC
  - Direct Connect

product/transcodingResult

- **Description:** The output of decoding an encoded video source to an intermediate uncompressed format, and re-encoding it into the target format.
- **Services:**

  - AWS Elemental MediaConvert

product/trialProduct

- **Description:** Describes if AWS CloudHSM allows free hours.
- **Services:**
  - AWS CloudHSM

U

product/upfrontCommitment

- **Description:** Describes if any usage commitment is required for AWS CloudHSM. You will be charged an hourly fee for each hour (or partial hour) that an HSM is provisioned to a AWS CloudHSM cluster. A cluster with no HSMs is not billed, and you aren't billed for automatic storage of encrypted backups. For more information, see AWS CloudHSM Pricing.

  Network data transfers to and from your HSMs are charged separately. For more information, see Amazon EC2 Pricing.
- **Services:**
  - AWS CloudHSM

product/usagetype

- **Description:** Describes the usage details of the line item.
- **Sample values:** `EU-BoxUsage:c5d.9xlarge`, `EU-BoxUsage:m4.16xlarge`, `SAE1-InstanceUsage:db.t2.medium`, `USW2-AW-SW-19`, `SAE1-BoxUsage:c4.large`,
- **Services:**
  - Amazon EC2
  - AWS Certificate Manager
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB

  and more. For the full service list, download Column_Attribute_Service.zip.

V

product/vcpu

- **Description:** Describes the number of threads concurrently running on a single CPU core. Amazon EC2 instances support multithreading, which enables multiple threads to run concurrently on a single CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance.
- **Sample values:** 8, 16, 36, 72, 128
- **Services:**
  - Amazon EC2
  - Amazon RDS
  - Amazon Redshift
  - OpenSearch Service
  - Amazon ElastiCache

  and more. For the full service list, download Column_Attribute_Service.zip.

product/videoCodec

- **Description:** A software or hardware that compresses and decompresses digital video. In the context of video compression, codec is a blending of encoder and decoder. A device that only compresses is typically called an encoder, and one that only decompresses is a decoder.
- **Services:**
  - AWS Elemental MediaConvert;

product/videoFrameRate

- **Description:** A video frame rate (shown as frames per second (FPS)) is the frequency rate which consecutive images (frames) are captured or displayed by video cameras, computer graphics, and motion capture systems.
- **Services:**
  - AWS Elemental MediaConvert;

product/videoQualitySetting

- **Description:** Describes the quality setting used for the encode, which impacts the compression efficiency and, therefore, the video quality at a given bitrate.
- **Sample values:** `Multi-pass`, `Multi-pass HQ`, NA, `Single-pass`, `Single-pass HQ`,
- **Services:**
  - AWS Elemental MediaConvert

product/volumeType

- **Description:** Describes your Amazon EBS volume types.
- **Sample values:** `Standard`, `General Purpose`, `General Purpose-Aurora`, `Amazon Glacier`, `Amazon SimpleDB - Standard`,
- **Services:**
  - Amazon EC2
  - Amazon S3
  - Amazon RDS
  - Amazon DynamoDB
  - Amazon Glacier

    and more. For the full service list, download Column_Attribute_Service.zip.

W

product/workforceType

- **Description:** The segmentation of the employed or unemployed labour pool. For example, `Full Time Employees` (FTE), or `Temporary`.
- **Services:**
  - Amazon SageMaker AI

Resource tags details

You can use the **resource** columns in AWS Cost and Usage Reports to find information about the specific resources covered by a line item. These columns include user-defined cost allocation tags. Examples include the following:

resourceTags/user:creator

Use a **user:Creator** tag to track which user created a resource.

resourceTags/user:name

Use a **user:Name** tag to track which resources are associated with a specific user.

resourceTags/user:owner

Use a **user:Owner** tag to track which user owns a resource.

resourceTags/user:purpose

Use a **user:Purpose** tag to track why a resource was created.

Savings Plans details

The **SavingsPlan** columns in AWS Cost and Usage Reports provide details about the Savings Plans. For more information about Savings Plans, see What are Savings Plans? in the *Savings Plans User Guide*.

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

A

savingsPlan/AmortizedUpfrontCommitmentForBillingPeriod

- **Description:** The amount of upfront fee a Savings Plans subscription is costing you for the billing period. The initial upfront payment for **All Upfront Savings Plans** and **Partial Upfront Savings Plans** amortized over the current month. For **No Upfront Savings Plans**, the value is 0.
- **Line items applicable:** SavingsPlanRecurringFee
- **Services:**

  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

E

savingsPlan/EndTime

- **Description:** The expiration date for the Savings Plans agreement.
- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

I

savingsPlan/InstanceTypeFamily

- **Description:** The instance family that is associated with the specified usage.
- **Line items applicable:** SavingsPlanCoveredUsage
- **Sample values:** `m4`, `g2`
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

N

savingsPlan/NetAmortizedUpfrontCommitmentForBillingPeriod

The cost of a Savings Plans subscription upfront fee for the billing period. This column is included in your report only when your account has a discount in the applicable billing period.

savingsPlan/NetRecurringCommitmentForBillingPeriod

The net unblended cost of the Savings Plans fee. This column is included in your report only when your account has a discount in the applicable billing period.

savingsPlan/NetSavingsPlanEffectiveCost

The effective cost for Savings Plans, which is your usage divided by the fees. This column is included in your report only when your account has a discount in the applicable billing period.

O

savingsPlan/OfferingType

- **Description:** Describes the type of Savings Plans purchased.
- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Sample values:** `ComputeSavingsPlans`, `EC2InstanceSavingsPlans`, `SageMakerSavingsPlans`
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

P

savingsPlan/PaymentOption

- **Description:** The payment options available for your Savings Plans.

- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Sample values:** `Partial Upfront`, `All Upfront`, `No Upfront`
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

savingsPlan/PurchaseTerm

- **Description:** Describes the duration, or term, of the Savings Plans.
- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Sample values:** `1yr`, `3yr`
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

R

savingsPlan/RecurringCommitmentForBillingPeriod

- **Description:** The monthly recurring fee for your Savings Plans subscriptions. For example, the recurring monthly fee for a **Partial Upfront Savings Plans** or **No Upfront Savings Plans**.
- **Line items applicable:** SavingsPlanRecurringFee
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

savingsPlan/Region

- **Description:** The AWS Region (geographic area) that hosts your AWS services. You can use this field to analyze spend across a particular AWS Region.
- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Sample values:** `US East (N. Virginia)`, `US West (N. California)`, `US East (Ohio)`, `Asia Pacific (Mumbai)`, `Europe (Ireland)`
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

S

savingsPlan/SavingsPlanArn

- **Description:** The unique Savings Plans identifier.
- **Line items applicable:** SavingsPlanUpfrontFee
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

savingsPlan/SavingsPlanEffectiveCost

- **Description:** The proportion of the Savings Plans monthly commitment amount (upfront and recurring) that is allocated to each usage line.
- **Line items applicable:** SavingsPlanCoveredUsage
- **Services:**
  - Amazon EC2
  - Fargate

  - AWS Lambda
  - Amazon SageMaker AI

savingsPlan/SavingsPlanRate

- **Description:** The Savings Plans rate for the usage.
- **Line items applicable:** SavingsPlanCoveredUsage
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

savingsPlan/StartTime

- **Description:** The start date of the Savings Plans agreement.
- **Line items applicable:** SavingsPlanCoveredUsage, SavingsPlanNegation, SavingsPlanUpfrontFee, SavingsPlanRecurringFee
- **Services:**
  - Amazon EC2
  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

T

savingsPlan/TotalCommitmentToDate

- **Description:** The total amortized upfront commitment and recurring commitment to date, for that hour.
- **Line items applicable:** SavingsPlanRecurringFee
- **Services:**
  - Amazon EC2

  - Fargate
  - AWS Lambda
  - Amazon SageMaker AI

U

savingsPlan/UsedCommitment

- **Description:** The total dollar amount of the Savings Plans commitment used. (SavingsPlanRate multiplied by usage)
- **Line items applicable:** SavingsPlanRecurringFee
- **Services:**
  - Amazon EC2
  - AWS Lambda
  - Fargate
  - Amazon SageMaker AI

Cost Categories details

You can use the **Cost Category** columns in AWS Cost and Usage Reports to find information about the specific line items categorized by the Cost Categories feature. Cost Category columns are automatically populated when you create a Cost Category and categorization rule. These columns include user-defined Cost Category names as the column headers, and corresponding Cost Category values. Examples include the following:

costCategory/Project

Use a **Project** Cost Category to categorize cost and usage information associated with the project. Project names such as “alpha” or “beta” will show as values under **costCategory/Project**.

costCategory/Team

Use a **Team** Cost Category to categorize cost and usage information associated with the team. Team names such as “Marketing” will show as values under **costCategory/Team**.

costCategory/Environment

Use a **Environment** Cost Category to categorize cost and usage information associated with the environment. Environment names such as “Development” will show as values under **costCategory/Environment**.

Discount details

Columns under the **discount** header are included in AWS Cost and Usage Reports only when the account has a discount applied during the report's billing period. This includes the following columns:

discount/BundledDiscount

The bundled discount applied to the line item. A bundled discount is a usage-based discount that provides free or discounted usage of a service or feature based on the usage of another service or feature. Examples of bundled discounts include:

- If you use AWS Shield Advanced, then you don't have to pay for AWS WAF separately. AWS WAF usage is bundled with AWS Shield Advanced. For more information about AWS Shield Advanced, see Amazon CloudFront pricing.
- If you create a NAT gateway with AWS Network Firewall, then the standard NAT gateway processing and per-hour usage charges are waived on a one-to-one basis with the firewall's processing per GB and usage hours. For more information, see AWS Network Firewall pricing.
- With Amazon Interactive Video Service (IVS) Chat, for every hour of video input sent, you get 2,700 sent messages and 270,000 delivered messages at no additional cost. For more information, see Amazon Interactive Video Service pricing.

discount/TotalDiscount

The sum of all the discount columns for the corresponding line item.

Split line item details

Columns under the **splitLineItem** header in AWS Cost and Usage Reports are fields that appear in Cost and Usage Reports if you've opted in to the split cost allocation data feature. For more information, see Understanding split cost allocation data. The feature is limited to Amazon ECS (including Fargate), AWS Batch, and Amazon EKS only.

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

A

splitLineItem/ActualUsage

- **Description:** The usage for vCPU or memory (based on lineItem/UsageType) you incurred for the specified time period for the Amazon ECS task or Kubernetes pod.
- **Line items applicable:** Usage
- **Sample values:** 0.1, 0.5, 1.3
- **Services:** Amazon ECS, Fargate, Amazon EKS

> **Note**
>
> Fargate costs are calculated based on vCPU and memory reservations and reflected in the lineItem/UsageAmount column. Split cost allocation data computes actual Fargate vCPU and memory usage by observing low latency telemetry data and then aggregating it to the hour, day, and month. Therefore splitLineItem/ActualUsage might not be the same as lineItem/UsageAmount.

N

splitLineItem/NetSplitCost

- **Description:** The effective cost for Amazon ECS tasks or Kubernetes pods after all discounts have been applied. This column is included in your report only when your account has a discount in the applicable billing period.
- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

splitLineItem/NetUnusedCost

- **Description:** The effective unused cost for Amazon ECS tasks or Kubernetes pods after all discounts have been applied. This column is included in your report only when your account has a discount in the applicable billing period.

- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

> **Note**
>
> Unused costs are proportionately applied to the Amazon ECS task or Kubernetes pod based on splitLineItem/SplitUsage.

P

splitLineItem/ParentResourceId

- **Description:** The resource ID of the parent EC2 instance associated with the Amazon ECS task or Amazon EKS pod (referenced in the lineItem/ResourceId column). The parent resource ID implies that the ECS task or Kubernetes pod workload for the specified time period ran on the parent EC2 instance. This applies only for Amazon ECS tasks or Kubernetes pods with EC2 launch type.
- **Line items applicable:** Usage
- **Services:** Amazon ECS, Amazon EKS

> **Note**
>
> splitLineItem/ParentResourceId is available only when resource IDs are included in AWS Cost and Usage Reports.

splitLineItem/PublicOnDemandSplitCost

- **Description:** The cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod based on public On-Demand Instance rates (referenced in the pricing/publicOnDemandRate column).
- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

splitLineItem/PublicOnDemandUnusedCost

- **Description:** The unused cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod based on public On-Demand Instance rates. Unused costs are costs associated with resources (CPU or memory) on the EC2 instance (referenced in the splitLineItem/ParentResourceIdcolumn) that were not utilized for the specified time period.
- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

R

splitLineItem/ReservedUsage

- **Description:** The usage for vCPU or memory (based on lineItem/UsageType) that you configured for the specified time period for the Amazon ECS task or Kubernetes pod.
- **Line items applicable:** Usage
- **Sample values:** 1, 2, 4
- **Services:** Amazon ECS, Fargate, Amazon EKS

S

splitLineItem/SplitCost

- **Description:** The cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod. This includes amortized costs if the EC2 instance (referenced in the splitLineItem/parentResourceId column) has upfront or partial upfront charges for reservations or Savings Plans.
- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

splitLineItem/SplitUsage

- **Description:** The usage for vCPU or memory (based on lineItem/UsageType) allocated for the specified time period to the Amazon ECS task or Kubernetes pod. This is defined as the maximum usage of splitLineItem/ReservedUsage or splitLineItem/ActualUsage.
- **Line items applicable:** Usage
- **Sample values:** 1, 1.3
- **Services:** Amazon ECS, Fargate, Amazon EKS

splitLineItem/SplitUsageRatio

- **Description:** The ratio of vCPU or memory (based on lineItem/UsageType) allocated to the Amazon ECS task or Kubernetes pod compared to the overall CPU or memory available on the EC2 instance (referenced in the splitLineItem/ParentResourceId column).
- **Line items applicable:** Usage
- **Sample values:** 0.25, 0.75
- **Services:** Amazon ECS, Fargate, Amazon EKS

> **Note**
>
> splitLineItem/SplitUsageRatio is only available for AWS Cost and Usage Reports with a time granularity preference of hourly data.

U

splitLineItem/UnusedCost

- **Description:** The unused cost for vCPU or memory (based on lineItem/UsageType) allocated for the time period to the Amazon ECS task or Kubernetes pod. Unused costs are costs associated with resources (CPU or memory) on the EC2 instance (referenced in the splitLineItem/ParentResourceId column) that were not utilized for the specified time period. This includes amortized costs if the EC2 instance (splitLineItem/parentResourceId) has upfront or partial upfront charges for reservations or Savings Plans.
- **Line items applicable:** Usage
- **Sample values:** 1.35, 1.75

- **Services:** Amazon ECS, Fargate, Amazon EKS

> **Note**
>
> Unused costs are proportionately applied to the Amazon ECS task or Kubernetes pod based on splitLineItem/SplitUsage.

Use cases

You can use the AWS Cost and Usage Reports (AWS CUR) to suit your reports management needs. This section goes in-depth to help you understand use cases such as tracking your Savings Plans and Reserved Instance (RI) utilization, charges, and allocations.

Topics

Understanding Savings Plans

You can use Cost and Usage Reports (AWS CUR) to track your Savings Plans utilization, charges, and allocations.

Savings Plans line items

Savings Plans provide a flexible pricing model that offers low prices on Amazon EC2, AWS Fargate, AWS Lambda, and Amazon SageMaker AI in exchange for a commitment to a consistent amount of usage (measured in $/hour) for a 1-year or 3-year term.

The following line items in AWS CUR help you track and manage your spend with Savings Plans.

> **Note**
>
> In the following tables, the columns and rows from AWS CUR are transposed for clarity. The values in the first column represent the headers of a report. These examples include only

a few key AWS CUR columns. To learn more about other AWS CUR columns, see the Data dictionary.

Upfront fee

The **SavingsPlanUpfrontFee** line item is added to your bill when you purchase an `All Upfront` or `Partial Upfront` Savings Plans. The following table shows how this one-time fee appears in some AWS CUR columns.

lineItem/LineItemType	SavingsPlanUpfrontFee	SavingsPlanUpfrontFee
lineItem/ProductCode	ComputeSavingsPlans	ComputeSavingsPlans
lineItem/UsageStartDate	2019-10-10T00:03:54Z	2019-10-10T00:12:15Z
lineItem/LineItemDescription	USD $43.80 one-time fee for 1 year All Upfront Compute Savings Plans ID: 70352035	USD $43.80 one-time fee for 3-year Partial Upfront Any Region M5 Instance Type EC2 Savings Plans ID: 12355516
lineItem/UnblendedCost	43.8	43.8
savingsPlan/SavingsPlanARN	arn:aws:savingsplans::5555555555555:savingsplan/ bc1d08fd	arn:aws:savingsplans::5555555555555:savingsplan/ 67b0ef20

Savings Plans recurring monthly fee

The **SavingsPlanRecurringFee** line item describes the recurring hourly charges that correspond to `No Upfront` or `Partial Upfront` Savings Plans. Initially, the **SavingsPlanRecurringFee** is added to your bill on the day of purchase and hourly thereafter.

The **SavingsPlanRecurringFee** allocated to the hour (applicable to Hourly cost and usage) or day (applicable to Daily cost and usage) is added to your bill at the hour of purchase. It is added every hour/day of the billing period subsequently.

For an `All Upfront` Savings Plans, the line item indicates the portion of the Savings Plans unused during the billing period.

The following table shows how the recurring hourly charges appear in some AWS CUR columns.

lineItem/LineItemType	SavingsPlanRecurringFee	SavingsPlanRecurringFee
lineItem/UsageStartDate	2019-20-10T00:00:00Z	2019-20-10T00:00:00Z
lineItem/ProductCode	Compute Savings Plans	Compute Savings Plans
lineItem/UsageType	ComputeSP:1yrPartialUpfront	USE2-EC2SP:t3.1yrPartialUpfront
lineItem/UnblendedCost	0.01	0.01
lineItem/LineItemDescription	1 year Partial Upfront Compute Savings Plan	1 year Partial Upfront t3 EC2 Instance Savings Plan in us-east-2
savingsPlan/SavingsPlanARN	arn:aws:savingsplans::5555555555555:savingsplan/ bc1d08fd	arn:aws:savingsplans::5555555555555:savingsplan/ bc1d08fd
savingsPlan/PaymentOption	Partial Upfront	Partial Upfront
savingsPlan/OfferingType	ComputeSavingsPlans	EC2InstanceSavingsPlans
savingsPlan/PurchaseTerm	1yr	1yr

Note

The SavingsPlanRecurringFee is calculated differently than the recurring RI fee. The recurring RI fee is a monthly charge while the SavingsPlanRecurringFee is an hourly charge. For information on the recurring RI fee, see Recurring monthly RI fee.

Savings Plans discount benefits

The **SavingsPlanCoveredUsage** line item describes the instance usage that received Savings Plans benefits. A **SavingsPlanCoveredUsage** line item shows an unblended cost of what the On-Demand charge would have been without the Savings Plans benefit. This unblended cost is offset by the corresponding **SavingsPlanNegation** line item.

In each **SavingsPlanCoveredUsage** line item, you can see how that usage was billed against your Savings Plans hourly commitment by using the **savingsPlan/SavingsPlanRate** and **savingsPlan/SavingsPlanEffectiveCost** fields.

You'll see a corresponding **SavingsPlanNegation** for each **SavingsPlanCoveredUsage** line item. **SavingsPlanNegation** line items offset the unblended cost of **SavingsPlanCoveredUsage**, and grouped at the hourly level by SavingsPlanARN, Operation, Usage Type, and Availability Zone. Therefore, one **SavingsPlanNegation** line item might correspond to multiple **SavingsPlanCoveredUsage** line items.

The following table shows how the covered usage and the negation line items appear in some AWS CUR columns.

lineItem/LineItemType	SavingsPlanCoveredUsage	SavingsPlanCoveredUsage	SavingsPlanNegation
lineItem/UsageStartDate	2019-10-10T00:00:00Z	2019-10-10T00:00:00Z	2019-10-10T00:00:00Z
lineItem/UsageEndDate	2019-10-10T01:00:00Z	2019-10-10T01:00:00Z	2019-10-10T01:00:00Z
lineItem/ProductCode	AmazonEC2	AmazonEC2	AmazonEC2
lineItem/UsageType	BoxUsage:t3.nano	BoxUsage:t3.nano	BoxUsage:t3.nano
lineItem/UsageAmount	1	0.5	-1.5

lineItem/UnblendedCost	0.0052	0.0026	-0.0078
lineItem/LineItemDescription	$0.0052 per On Demand Linux t3.nano Instance Hour	$0.0052 per On Demand Linux t3.nano Instance Hour	SavingsPlanNegation used by AccountId : 5555555555555 and UsageSku : K7ERD2Q28HHU97DT
SavingsPlan/SavingPlanARN	arn:aws:savingsplans:: 5555555555555: savingsplan/bc1d08fd	arn:aws:savingsplans:: 5555555555555: savingsplan/bc1d08fd	arn:aws:savingsplans:: 5555555555555: savingsplan/bc1d08fd
savingsPlan/SavingsPlanRate	0.0026	0.0026	
savingsPlan/SavingsPlanEffectiveCost	0.0026	0.0013	

When you have more usage than your Savings Plans commitment can cover, your uncovered usage still appears as a Usage Line Item and the covered usage appears as **SavingsPlanCoveredUsage** with the corresponding **SavingsPlanNegation** line items.

Understanding your reservations

You can use the AWS Cost and Usage Reports (AWS CUR) to track your Reserved Instance (RI) utilization, charges, and allocations. This section is an in-depth description to understand your reservations.

Topics

- Understanding your reservation line items
- Understanding your amortized reservation data
- Monitoring your size flexible reservations for Amazon EC2
- Monitoring your On-Demand capacity reservations

Understanding your reservation line items

RIs provide you a significant discount compared to On-Demand Instance pricing. RIs aren't physical instances. They're a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes to benefit from the billing discount.

Topics

> **Note**
>
> In the following tables, the columns and rows from AWS CUR are transposed for clarity. The values in the first column represent the headers of a report. These examples include only a few key AWS CUR columns. To learn more about other AWS CUR columns, see the Data dictionary.

Upfront fee

The **Fee** line item is added to your bill when you purchase an `All Upfront` or `Partial Upfront` RI.

The following table shows how this one-time fee appears in some AWS CUR columns.

lineItem/LineItemType	Fee
lineItem/ProductCode	AmazonEC2
lineItem/UsageStartDate	2016-01-01T00:00:00Z
lineItem/LineItemDescription	Sign up charge for subscription: 363836886, planId: 1026576

lineItem/UnblendedCost	68
Reservation/ReservationARN	arn:aws:ec2:us-east-1:123456789012:reserved-instances/f8c204c1-dd48-43f1-adb8-f88aa61e0dea

True-up fee

If you exchange a Convertible Reserved Instance, any cost associated with the exchange of the original Reserved Instance and the new Reserved instance (true-up fee) is also added to your bill as a **Fee** line item. For a true-up fee, the **reservation/ReservationARN** column contains **reserved-instance-exchange/riex**.

The following table shows a true-up fee from exchanging a Convertible Reserved Instance.

lineItem/LineItemType	lineItem/ProductCode	lineItem/UsageStartDate	lineItem/LineItemDescription	lineItem/UnblendedCost	Reservation/ReservationARN
Fee	AmazonEC2	2016-01-01T00:00:00Z			arn:aws:ec2:eu-west-1:012345678901:reserved-instance-exchange/riex-examplef-5d71-4215-886f-17a3f64ea972

Recurring monthly RI fee

The **RI Fee** line item describes the recurring monthly charges that are associated RIs applied that month. The **RI Fee** initially is added to your bill on the day of purchase and on the first day of each billing period thereafter.

The **RI Fee** is calculated by multiplying your discounted hourly rate and the number of hours in the month.

The following table shows how the recurring monthly charges appear in the report.

lineItem/LineItemType	RI fee
lineItem/ProductCode	AmazonEC2
lineItem/UsageStartDate	2016-01-01T00:00:00Z
lineItem/UsageType	HeavyUsage: m4.large
lineItem/LineItemDescription	USD 0.0309 hourly fee per Linux/UNIX (Amazon VPC), m4.large instance
lineItem/NormalizationFactor	4
lineItem/UnblendedCost	23
Reservation/AvailabilityZone	
Reservation/ReservationARN	arn:aws:ec2:us-east-1:123456789012:reserved-instances/f8c204c1-dd48-43f1-adb8-f88aa61e0dea
Reservation/TotalReservedunits	744
Reservation/TotalReservedNormalizedUnits	2976

Recurring monthly charges are recorded differently for RIs that have an Availability Zone or AWS Region Region scope. For RIs that have an Availability Zone scope, the corresponding Availability Zone is shown in the **reservation/AvailabilityZone** column. For RIs that have a Region scope, the **reservation/AvailabilityZone** column is empty. RIs with a Region scope have values for the **lineitem/NormalizationFactor** and **reservation/TotalReservedNormalizedUnits** columns that show the instance size.

> **Note**
>
> The recurring RI fee is calculated differently than the SavingsPlanRecurringFee. The recurring RI fee is a monthly charge while the SavingsPlanRecurringFee is an hourly charge. For information on the SavingsPlanRecurringFee, see Understanding Savings Plans.

RI discount benefits

The **Discounted Usage** line item describes the instance usage that received a matching RI discount benefit, and is added to your bill when you have usage that matches one of your RIs. AWS calculates RI discount benefits based on matching usage: for example, the use of an instance that matches the instance reservation. If you have matching usage, the cost associated with the usage line item is always zero because the charges associated with RIs are already accounted for in the two other line items (the upfront fee and the recurring monthly charges).

The following table shows an example of usage that received an RI discount benefit.

lineItem/LineItemType	DiscountedUsage
lineItem/ProductCode	AmazonEC2
lineItem/UsageStartDate	2016-01-01T00:00:00Z
lineItem/UsageType	BoxUsage:m4.large
lineItem/LineItemDescription	Linux/UNIX (Amazon VPC), m4.large Reserved Instance applied
lineItem/ResourceId	i-1bd250bc
lineItem/AvailabilityZone	us-east-1b
lineItem/NormalizationFactor	4
lineItem/NormalizedUsageAmount	4
lineItem/UnblendedRate	0
lineItem/UnblendedCost	0

Reservation/ReservationARN	arn:aws:ec2:us-east-1:123456789012:reserved-instances/f8c204c1-dd48-43f1-adb8-f88aa61e0dea

The value for **UsageAmount** in the Amazon EC2 **DiscountedUsage** line is the actual number of hours used. The value for **NormalizedUsageAmount** is the value for **UsageAmount** multiplied by the value for **NormalizationFactor**. The value for **NormalizationFactor** is determined by the instance size. When an RI benefit discount is applied to a matching line item of usage, the Amazon Resource Name (ARN) value in the **reservation/ReservationARN** column for the initial upfront fees and recurring monthly charges matches the ARN value in the discounted usage line items.

For more information about mapping instance size to normalization factor, see Support for modifying instance sizes in the *Amazon EC2 User Guide*.

Reserved Instance type

To determine if your report line items are associated with a Standard Reserved Instance or a Convertible Reserved Instance, filter the **lineItem/LineItemType** column by **Fee** or **RI fee**. Then, review the **product/OfferingClass** column, which indicates the Reserved Instance type.

To determine if your report line items are associated with a zonal or regional Reserved Instance, review the **reservation/AvailabilityZone** column. For zonal Reserved Instances, this column shows the corresponding Availability Zone. For regional Reserved Instances, this column is empty.

Reserved Instance benefits applied to instance usage

To understand which instance usage line items benefitted from which Reserved Instances, you can filter your report by one or more of the following columns:

- **reservation/reservationARN**: Filter this column by a reservation ARN to identify which Reserved Instance lease is associated with each line item.
- **lineitem/ResourceId**: Review this column for the ID of the resource that's covered by the Reserved Instance.
- **lineitem/LineItemType**: Filter this column by **Fee**, **RI fee**, or **DiscountedUsage** to determine the associated fees or benefits.
- **lineitem/UsageType**: Filter this column by **HeavyUsage** to identify **RI fee** line items. Or, filter this column by **BoxUsage** to identify **DiscountedUsage** line items.

- **lineitem/UsageAmount**: For **RI fee** line items, this column shows the total number of hours in the month that the Reserved Instance was applied. For **DiscountedUsage** line items, this column shows the total number of hours that the Reserved Instance was applied to a specific instance at the daily or monthly level, depending on how you configured your report.

To understand a size flexible Reserved Instance's number of normalized units applied to instance usage, review the **lineitem/NormalizedUsageAmount** column in your report. The value in this column equals the product of the following columns:

- **lineitem/UsageAmount**: This column shows the metered instance usage measured in hours.
- **lineItem/NormalizationFactor**: For **DiscountedUsage** and **RI fee** line items, this column shows the associated normalization factor of the instance. For more information on the normalization factor, see Instance size flexibility determined by normalization factor in the *Amazon EC2 User Guide*.

For AWS Organizations with multiple accounts, to see which accounts purchased or benefitted from a Reserved Instance, review the following columns:

- **reservation/reservationARN**: Review the reservation ARNs to see which accounts purchased the Reserved Instance. The ARN includes the account ID.
- **lineitem/UsageAccountId**: For **DiscountedUsage** line items, this column identifies the account IDs that received benefits from the purchased Reserved Instances.

Note

A Reserved Instance is a billing subscription and not a resource like an Amazon EC2 instance. Because of this, Reserved Instances that are tagged don't populate line items like a tagged resource. For line items with **DiscountedUsage**, tags populate for the tagged resources and not for the Reserved Instance.
To identify costs associated with a specific Reserved Instance lease, you can filter **Fee** or **RI fee** line items by the Reserved Instance ARN, which is the lease ID. To organize your cost data for Reserved Instances, consider using AWS Cost Categories. For more information, see Managing your costs with AWS Cost Categories in the *AWS Billing User Guide*

Understanding your amortized reservation data

Amortizing is when you distribute one-time reservation costs across the billing period that is affected by that cost. Amortizing enables you to see your costs in accrual-based accounting as opposed to cash-based accounting. For example, if you pay $365 for an All Upfront RI for one year and you have a matching instance that uses that RI, that instance costs you $1 a day, amortized.

You can see the data that Billing and Cost Management uses to calculate your amortized costs in the following Cost and Usage Reports columns.

Topics

- Reserved Instance inventory
- Amortization data for the billing period
- Reserved Instance effective costs

> **Note**
>
> Not all **reservation/** columns are populated for every Reserved Instance line item. The **reservation/** columns in your report are populated based on the line item type. For example, **RI fee** line items populate the **reservation/UnusedAmortizedUpfrontFeeForBillingPeriod** column. Meanwhile, **DiscountedUsage** line items populate the **reservation/effectivecost** column.

Reserved Instance inventory

You can use the following columns to track your RI inventory. The values for these columns appear only for RI subscription line items (also known as `RI Fee` line items) and not for the actual instances using the RIs.

For more information about column descriptions and sample values, see the section called "Reservation details".

- reservation/UpfrontValue
- reservation/startTime
- reservation/endTime
- reservation/modificationStatus

Amortization data for the billing period

You can use the following columns to understand the amortized costs of your RIs for the billing period. The values for these columns appear only for RI subscription line items (also known as `RI Fee` line items) and not for the actual instances using the RIs.

For more information about column descriptions and sample values, see the section called “Reservation details”.

- reservation/amortizedUpfrontFeeForBillingPeriod
- reservation/unusedQuantity
- reservation/unusedNormalizedUnitQuantity
- reservation/unusedRecurringFee
- reservation/unusedAmortizedUpfrontFeeForBillingPeriod

Reserved Instance effective costs

You can use the following columns to understand your effective cost at the instance level. The values for these columns appear only for instance usage line items (also known as `Discounted Usage boxUsage` line items).

For more information about column descriptions and sample values, see the section called “Reservation details”.

- reservation/amortizedUpfrontCostForUsage
- reservation/recurringFeeForUsage
- reservation/effectiveCost

Monitoring your size flexible reservations for Amazon EC2

Amazon EC2 Reserved Instances that apply to a Region provide Availability Zone flexibility and instance size flexibility. Reserved Instances that provide Availability Zone flexibility provide a discount on usage in any Availability Zone in the Region. Reserved Instances that provide instance size flexibility provide a discount on usage, regardless of instance size in that family. Size flexible Reserved Instances apply to the smallest instance sizes first. For more information, see How Reserved Instances are applied in the *Amazon EC2 User Guide*.

To understand how instance size flexibility provided by your Reserved Instance is applied to your usage, refer to the **lineItem/NormalizationFactor** and **lineItem/NormalizedUsageAmount** columns.

> **Note**
>
> Instance size flexibility is supported only by Linux or Unix Reserved Instances with default tenancy that are assigned to a Region. For more information on the limitations of instance size flexibility for Regional Reserved Instances, see How regional Reserved Instances are applied in the *Amazon EC2 User Guide*.

In a Cost and Usage Report, the Reserved Instance usage is applied by default to the account that purchased the Reserved Instance. Any available Reserved Instance benefit that the purchasing account can't use within the hour is then applied to other linked accounts based on the available matching On-Demand Instance usage.

Example

You purchase one `m4.xlarge` RI in a given Region. This `m4.xlarge` RI can be applied automatically to all `m4` instance usage in the same Region. In the following table, AWS applied the `m4.xlarge` to two separate `m4.large` instances.

lineItem/LineItemType	RIFee	Discounted Usage	Discounted Usage
lineItem/ProductCode	AmazonEC2	AmazonEC2	AmazonEC2
lineItem/UsageStartDate	2016-01-01T00:00:00Z	2016-01-01T00:00:00Z	2016-01-01T00:00:00Z
lineItem/UsageType	HeavyUsage:m4.xlarge	BoxUsage:m4.large	BoxUsage:m4.large
lineItem/LineItemDescription	USD 0.0618 hourly fee per Linux/UNIX (Amazon VPC), m4.xlarge instance	Linux/UNIX (Amazon VPC), m4.large Reserved Instance applied	Linux/UNIX (Amazon VPC), m4.large Reserved Instance applied

lineItem/ResourceId		i-1bd250bc	i-1df340ed
lineItem/UsageAmount		1	1
lineItem/NormalizationFactor	4	4	4
lineItem/NormalizedUsageAmount		4	4
lineItem/UnblendedRate		0	0
lineItem/UnblendedCost	46	0	0
Reservation/ReservationARN	arn:aws:ec2:us-east-1: 123456789012:reserved-instances /f8c204c1	arn:aws:ec2:us-east-1: 123456789012:reserved-instances /f8c204c1	arn:aws:ec2:us-east-1: 123456789012:reserved-instances /f8c204c1
Reservation/TotalReservedUnits	744		
Reservation/TotalReserved NormalizedUnits	5952		

The two `m4.large` usage line items have different **ResourceId**s, and both received a discount benefit from the single `m4.xlarge` RI. This is shown by matching the **reservationARN** value across the usage and recurring monthly charge line items.

For more information about RI purchase options, see How you are billed in the *Amazon EC2 User Guide*.

Monitoring your On-Demand capacity reservations

Capacity reservations enable you to reserve capacity for your Amazon EC2 instances for any duration in a specific Availability Zone. This enables you to create and manage capacity reservations separately from the billing discounts offered by Regional Reserved Instances (RI). To benefit from billing discounts, you can use Regional RIs in combination with capacity reservations.

Capacity reservation line items

You can use some columns defined in the AWS CUR data dictionary to track your capacity reservations. The following columns are also used for capacity reservations.

This section defines these line items with supplementary definitions specific to capacity reservations.

For more information about Cost and Usage Reports column descriptions, see the section called "Line item details".

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | VWXYZ

B

lineItem/BlendedRate

For capacity reservations with a **UsageType** of **Reservation** or **DedicatedRes**, the **BlendedRate** is `0`. This is because the capacity reservation costs are associated with the instance that provides the capacity, instead of with the capacity reservation itself.

R

lineItem/ResourceId

If you included `lineItem/ResourceId` when you created your Cost and Usage Reports, you can identify and track your capacity reservations using the **ResourceId** column. The capacity reservation **ResourceId** is captured only for the **UnusedBox, UnusedDed**, **Reservation**, and **DedicatedRes UsageTypes**.

Capacity reservations always include a `cr-` in their resource ID, and the resource ID has the following format:

```
arn:aws:ec2:<region>:<account id>:<capacity-reservation>/cr-0be443example1db6f
```

U

lineItem/UnblendedCost

The `BlendedRate` multiplied by the `UsageAmount`.

lineItem/UnblendedRate

For capacity reservations with a **UsageType** of **Reservation** or **DedicatedRes**, the **UnblendedRate** is `0`. This is because the costs for capacity reservations are associated with the instance that provides the capacity, instead of with the capacity reservation itself.

lineItem/UsageAmount

How much of a capacity reservation you've used. Each capacity reservation can have multiple slots for an hour, enabling you to run more than one instance that uses the reservation during an hour. Therefore, it's possible to use more than one instance-hour in an hour. **UsageAmount** is calculated by multiplying the number of instance slots covered by the line item with the number of hours covered by the line item.

lineItem/UsageType

How much of a specific reservation you've used. For Amazon EC2, the options are as follows:

lineItem/lineitemtype = BoxUsage

For this `UsageType`, the `UsageAmount` column is the amount of instance-hours of an instance you've used.

For example, a report covers 1 hour and has a capacity reservation line item that can cover 10 instances. If you use two instance-slots during the time period covered by the report, the **BoxUsage UsageAmount** covers the number of instance hours that you reserved and used. In this case, this is two (the number of used instance slots) multiplied by 1 hour (the time covered by the report) for a total of two. For a report that covers 1 day, the **UsageAmount** is two multiplied by 24, for a total of 48.

DedicatedRes

For a **UsageType** of **DedicatedRes**, the **UsageAmount** column describes how many instance-hours of a dedicated capacity reservation you reserved.

Reservation

For a **UsageType** of **Reservation**, the **UsageAmount** column describes how many instance-hours of a capacity reservation you reserved.

For example, if a report covers one hour and has a capacity reservation line item that can cover 10 instances, the **Reservation UsageAmount** covers the number of instance slots that you reserved. In this case, that's 10 (the number of available instance slots) multiplied by 1 hour (the time covered by the report) for a total of 10. For a report that covers 1 day, the **UsageAmount** would be 10 multiplied by 24, for a total of 240.

UnusedBox

For a **UsageType** of **UnusedBox**, the **UsageAmount** column describes how many instance-hours of a capacity reservation you reserved, but didn't use.

For example, a report covers 1 hour and has a capacity reservation line item that can cover 10 instances. If you didn't use eight instance-slots during the time period covered by the report, the **UnusedBox UsageAmount** covers the number of instance hours that you reserved but didn't use. In this case, that's eight (the number of unused instance slots) multiplied by 1 hour (the time covered by the report) for a total of eight. For a report that covers 1 day, the **UsageAmount** is eight multiplied by 24, for a total of 192.

UnusedDed

For a **UsageType** of **UnusedDed**, the **UsageAmount** column describes how many instance-hours of a dedicated capacity reservation that you reserved, but didn't use.

Understanding data transfer charges

You can identify your AWS data transfer charges using the lineItem/UsageType column of your AWS CUR.

> **Note**
>
> Data transfer charges can vary depending on the services used and the source AWS Region. For detailed pricing information, refer to the service’s pricing page. For example, see Amazon EC2 On-Demand Pricing for detailed pricing information about Amazon EC2 data transfer.

Data transfer within an AWS Region

Data transfer between Availability Zones in the same AWS Region have a **UsageType** of *`Region`*`-DataTransfer-Regional-Bytes`. For example, the `USE2-DataTransfer-Regional-Bytes` usage type identifies charges for data transfer between Availability Zones in the US East (Ohio) Region.

For a given resource, you're charged for both inbound and outbound traffic in a data transfer within an AWS Region. This means for each resource metered, you'll see two `DataTransfer-Regional-Bytes` line items for each data transfer. Confirm the service's pricing page for more information, because some services have in-Region traffic at no cost.

Data transfer between AWS Regions

Data transfer between different AWS Regions can have the following usage types:

- *`Source Region`*`-`*`Destination Region`*`-AWS-In-Bytes`: Measures incoming data transfer TO the destination Region FROM another specific AWS Region.
- *`Source Region`*`-`*`Destination Region`*`-AWS-Out-Bytes`: Measures outgoing data transfer FROM the source Region TO another specific AWS Region.
- *`Source Region`*`-AWS-In-Bytes`: This usage type appears when traffic flows via VPC Peering.
- *`Source Region`*`-AWS-Out-Bytes`: This usage type appears when traffic flows via VPC Peering.

For each resource, data transfer between AWS Regions corresponds to two line items in your report:

- A line item for the data transferred into the destination Region
- A line item for the data transferred out from the source Region

There's no charge for the data transferred into the destination Region. The data transfer charge is determined by the data transferred out from the source Region.

For example, a data transfer from the `USE2` Region to the `APS3` Region will have both a `APS3-USE2-AWS-In-Bytes` line item and a `USE2-APS3-AWS-Out-Bytes` line item. The `APS3-USE2-AWS-In-Bytes` line item has no corresponding charge. The data transfer charge is associated with the `USE2-APS3-AWS-Out-Bytes` line item.

Data transfer out to the internet

Data transfer from AWS to the internet have a **UsageType** of `Region-DataTransfer-Out-Bytes`. For example, the `USE2-DataTransfer-Out-Bytes` usage type identifies charges for data transfer from the `USE2` Region to the internet.

There's no charge for data transfer from the internet to AWS.

> **Note**
>
> Data transfer usage types that don't have the Region prefix, such as `DataTransfer-Regional-Bytes` or `DataTransfer-Out-Bytes`, represent data transfer from the US East (N. Virginia) Region.

Direct Connect traffic

Direct Connect data transfer over a public virtual interface have usage types that end with `DataXfer-In` or `DataXfer-Out`.

Direct Connect data transfer over a private or transit virtual interface have usage types that end with `DataXfer-In:dc.3` or `DataXfer-Out:dc.3`.

S3 Transfer Acceleration traffic

Amazon S3 data transfer using S3 Transfer Acceleration have usage types that contain `ABytes`:

- Between Amazon S3 and Amazon EC2: Usage types that end with `C3DataTransfer-In-ABytes` or `C3DataTransfer-Out-ABytes`
- Between Amazon S3 and the internet: Usage types that end with `DataTransfer-In-ABytes` or `DataTransfer-Out-ABytes`
- Between Amazon S3 and CloudFront: Usage types that end with `CloudFront-In-ABytes` or `CloudFront-Out-ABytes`
- Between Amazon S3 buckets in different AWS Regions: Usage type of `Source Region-Destination Region-AWS-Out-ABytes`

CloudFront traffic

CloudFront data transfer have a usage type of `*Region*-DataTransfer-Out-Bytes` or `*Region*-DataTransfer-Out-OBytes` coupled with the product code `AmazonCloudFront`. The Region prefix in the usage type refers to the CloudFront Edge location used in the data transfer. For example, the `AP-DataTransfer-Out-Bytes` usage type identifies charges for data transfer from the AP Region to the internet.

> **Tip**
>
> Use the lineItem/ProductCode column to distinguish CloudFront data transfer from data transfer out to the internet. The usage types for these data transfer types look similar.

Understanding split cost allocation data

You can use Cost and Usage Reports (AWS CUR) to track your Amazon ECS and Amazon EKS container costs. Using split cost allocation data, you can allocate your container costs to individual business units and teams, based on how your container workloads consume shared compute and memory resources. Split cost allocation data introduces cost and usage data for new container-level resources (that is, ECS tasks and Kubernetes pods) to AWS CUR. Previously, AWS CUR only supported costs at the EC2 instance level. Split cost allocation data generates container-level costs by looking at each container's EC2 instance resource consumption, and generates cost based on the amortized cost of the instance and the percentage of CPU and memory resources consumed by the containers that ran on the instance.

For accelerated computing instances used with Amazon EKS, split cost allocation data includes resource allocation for specialized processors alongside CPU and memory. This covers NVIDIA and AMD GPUs, AWS Trainium, and AWS Inferentia accelerators. The feature is available only for Amazon EKS environments and provides pod-level resource reservation data for these accelerated computing resources. This allows you to track and allocate costs for workloads that use these specialized processors, such as AI/ML applications and other computationally intensive tasks. For a current list of accelerated computing instances, see Accelerated Computing.

Split cost allocation data introduces new usage records and new cost metric columns for each containerized resource ID (that is, ECS task and Kubernetes pod) in AWS CUR. For more information, see Split line item details.

When including split cost allocation data in AWS CUR, two new usage records are added for each ECS task and Kubernetes pod per hour in order to reflect the CPU and memory costs. To estimate the number of new line items in AWS CUR per day, use the following formula:

For ECS: `(number of tasks * average task lifetime * 2) * 24`

For EKS: `(number of pods * average pod lifetime * 2) * 24`

For example, if you have 1,000 pods running each hour across a cluster of 10 EC2 instances and the lifetime for the pod is less than 1 hour, then:

`(1000 * 1 * 2) * 24 = 48,000 new usage records in AWS CUR`

For accelerated computing instances in Amazon EKS, three new usage records are added for each Kubernetes pod per hour in order to reflect the accelerator, CPU, and memory costs. To estimate the number of new line items in AWS CUR per day, use the following formula:

For EKS with accelerated computing: `(number of pods * average pod lifetime * 3) * 24`

For example, if you have 1,000 pods running each hour across a cluster of 10 EC2 instances and the lifetime for each pod is less than one hour, then: `(1000 * 1 * 3) * 24 = 72,000 new usage records in AWS CUR`

Note

For ECS: When it comes to AWS cost allocation tags, you can use Amazon ECS-managed tags or user-added tags for your Cost and Usage Reports. These tags apply to all new ECS split cost allocation data usage records. For more information, see Tagging your ECS resources for billing.
For EKS: Split cost allocation data creates new cost allocation tags for some Kubernetes attributes. These tags include `aws:eks:cluster-name`, `aws:eks:deployment`, `aws:eks:namespace`, `aws:eks:node`, `aws:eks:workload-name`, and `aws:eks:workload-type`.

- `aws:eks:cluster-name`, `aws:eks:namespace`, and `aws:eks:node` are populated retrospectively with the name of the cluster, namespace, and node.
- `aws:eks:workload-type` is only populated if there is exactly one workload managing the pod, and is one of the built in workloads. Workload types include `ReplicaSet`, `StatefulSet`, `Job`, `DaemonSet`, or `ReplicationController`, and

`aws:eks:workload-name` includes the name of the workload. For more information, see Workloads in the *Kubernetes Documentation*.

- `aws:eks:deployment` is only populated for the workload type `ReplicaSet`. It is the deployment that creates a `ReplicaSet`.

These tags apply to all new EKS split cost allocation data usage records. These tags are enabled for cost allocation by default. If you previously used and disabled the `aws:eks:cluster-name` tag, then split cost allocation data keeps this setting and doesn't enable the tag. You can enable it from the Cost allocation tags console page.

Enabling split cost allocation data

Note

Split cost allocation data is not available in Cost Explorer. It is available in legacy Cost and Usage Reports (CUR) and Cost and Usage Report 2.0 (CUR 2.0) with Data Exports.

It is a prerequisite to opt in to split cost allocation data through the Cost Management preferences.

To opt in to split cost allocation data

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Cost Management preferences**.
3. Under **General**, in the **Split cost allocation data** section, choose between the following:

   - **Amazon Elastic Container Service (Amazon ECS)** to opt in to Amazon ECS only.
   - **Amazon Elastic Kubernetes Service (Amazon EKS)** to opt in to Amazon EKS only. For Amazon EKS, choose between the following:
     - **Resource requests**: This allocates your Amazon EC2 by Kubernetes pod CPU and memory resources only. This will encourage application teams to only provision what they need.
     - **Amazon Managed Service for Prometheus**: This allocates your Amazon EC2 costs by the higher of Kubernetes pod CPU and memory resource requests and actual utilization. This ensures each application team pays for what they use. To learn more about setting up

Amazon Managed Service for Prometheus, see Setting up in the *Amazon Managed Service for Prometheus user guide*.

Prerequisite: You must enable all features in AWS Organizations. To learn more, see Enabling all features in your organization in the *Organizations user guide*.

- **Amazon CloudWatch Container Insights**: This provides more granular cost visibility for your clusters running multiple application containers using shared EC2 instances, enabling better cost allocation for the shared costs of your EKS clusters.

> **Note**
>
> - Only regular and payer accounts have access to the AWS Cost Management preferences and can opt in to split cost allocation data. Once opted in, member accounts can view the data in the Cost and Usage Reports.
> - If you choose resource requests, only the pods configured with memory and CPU requests are used by split cost allocation data. Pods that haven't requested any usage won't see any split cost data.
> - If you choose Amazon Managed Service for Prometheus, you need to enable all features in AWS Organizations. For more information, see Enabling all features in your organization. In addition, split cost allocation data creates a new service-linked role, which enables access to AWS services and resources used or managed by split cost allocation data.
> - For accelerated computing instances, only the Resource request option is supported. Neither Amazon Managed Service for Prometheus nor Amazon CloudWatch Container Insights are supported for these instances. When using accelerated computing instances, the system will default to Resource request to compute accelerator, CPU, and memory costs, even if other measurement options are enabled.

Once you’ve opted in, you can choose to have cost and usage data for container-level resources included in your report during step one of report creation or later by editing the report details.

To include cost and usage data in your report

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.

2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. Whether creating a new report or editing an existing report, in the **Specify report details** page, under **Report content**, select **Split cost allocation data**.

> **Note**
>
> You can also use the AWS CUR API or the AWS Command Line Interface (CLI) to manage your split cost allocation data preferences.

Split cost allocation data enables cost visibility for all Amazon ECS and Amazon EKS container objects across your entire consolidated billing family (payer and linked accounts). Once activated, split cost allocation data automatically scans for tasks and containers. It ingests the telemetry usage data for your container workloads and prepares the granular cost data for the current month.

> **Note**
>
> It can take up to 24 hours for the data to be visible in AWS CUR.

For information about managing access to Billing and Cost Management console pages, see Overview of managing access permissions.

For information regarding AWS Cost Management preferences and controlling access to Cost Explorer, see Controlling access to Cost Explorer.

Example of split cost allocation data

The purpose of the following example is to show you how split cost allocation data is calculated by computing the cost of individual Amazon ECS services, tasks in Amazon ECS clusters, and Kubernetes namespace and pods in Amazon EKS clusters. The rates used throughout the example are for illustrative purposes only.

Note

The example demonstrates Kubernetes namespace and pods running in Amazon EKS clusters. We can then apply the same cost model to Amazon ECS service and tasks running in a Amazon ECS cluster.

You have the following usage in a single hour:

- Single instance (m5.xlarge) shared cluster with two namespaces and four pods, running for the duration of a full hour.
- Instance configuration is 4 vCPU and 16 GB of memory.
- Amortized cost of the instance is $1/hr.

Split cost allocation data uses relative unit weights for CPU and memory based on a 9:1 ratio. This is derived from per vCPU per hour and per GB per hour prices in AWS Fargate.

Step 1: Compute the unit cost for CPU and memory

```
Unit-cost-per-resource = Hourly-instance-cost/((Memory-weight * Memory-available) + (CPU-weight * CPU-available))
```

= $1/((1 * 16GB) + (9 * 4vCPU)) = $0.02

```
Cost-per-vCPU-hour = CPU-weight * Unit-cost-per-resource
```

= 9 * $0.02 = $0.17

```
Cost-per-GB-hour = Memory-weight * Unit-cost-per-resource
```

= 1 * $0.02 = $0.02

Instance	Instance type	vCPU-available	Memory-available	Amortized-cost-per-hour	Cost-per-vCPU-hour	Cost-per-GB-hour
Instance1	m5.xlarge	4	16	$1	$0.17	$0.02

Step 2: Compute the allocated capacity and instance unused capacity

- Allocated capacity: The memory and vCPU allocated to the Kubernetes pod from the parent EC2 instance, defined as the maximum of used and reserved capacity.

> **Note**
>
> If memory or vCPU usage data is unavailable, reservation data will be used instead. For more information, see Amazon ECS usage reports, or Amazon EKS cost monitoring.

- Instance unused capacity: The unused capacity of vCPU and memory.

`Pod1-Allocated-vCPU = Max (1 vCPU, 0.1 vCPU)` = 1 vCPU

`Pod1-Allocated-memory = Max (4 GB, 3 GB)` = 4 GB

`Instance-Unused-vCPU = Max (CPU-available - SUM(Allocated-vCPU), 0)` = Max (4 – 4.9, 0) = 0

`Instance-Unused-memory = Max (Memory-available - SUM(Allocated-memory), 0)` = Max (16 – 14, 0) = 2 GB

In this example, the instance has CPU over subscription, attributed to Pod2 that used more vCPU than what was reserved.

Pod name	Namespace	Reserved-vCPU	Used-vCPU	Allocated-vCPU	Reserved-memory	Used-memory	Allocated-memory
Pod1	Namespace 1	1	0.1	1	4	3	4
Pod2	Namespace 2	1	1.9	1.9	4	6	6
Pod3	Namespace 1	1	0.5	1	2	2	2

Pod name	Namespace	Reserved-vCPU	Used-vCPU	Allocated-vCPU	Reserved-memory	Used-memory	Allocated-memory
Pod4	Namespace 2	1	0.5	1	2	2	2
Unused	Unused			0			2
				4.9			16

Step 3: Compute the split usage ratios

- Split usage ratio: The percentage of CPU or memory used by the Kubernetes pod compared to the overall CPU or memory available on the EC2 instance.
- Unused ratio: The percentage of CPU or memory used by the Kubernetes pod compared to the overall CPU or memory used on the EC2 instance (that is, not factoring in the unused CPU or memory on the instance).

`Pod1-vCPU-split-usage-ratio = Allocated-vCPU / Total-vCPU`

= 1 vCPU / 4.9vCPU = 0.204

`Pod1-Memory-split-usage-ratio = Allocated-GB / Total-GB`

= 4 GB/ 16GB = 0.250

`Pod1-vCPU-unused-ratio = Pod1-vCPU-split-usage-ratio / (Total-CPU-split-usage-ratio – Instance-unused-CPU)` (set to 0 if Instance-unused-CPU is 0)

= 0 (since Instance-unused-CPU is 0)

`Pod1-Memory-unused-ratio = Pod1-Memory-split-usage-ratio / (Total-Memory-split-usage-ratio – Instance-unused-memory)` (set to 0 if Instance-unused-memory is 0)

= 0.250 / (1-0.125) = 0.286

Pod name	Namespace	vCPU-split-usage-ratio	vCPU-unused-ratio	Memory-split-usage-ratio	Memory-unused-ratio
Pod1	Namespace1	0.204	0	0.250	0.286
Pod2	Namespace2	0.388	0	0.375	0.429
Pod3	Namespace1	0.204	0	0.125	0.143
Pod4	Namespace2	0.204	0	0.125	0.143
Unused	Unused	0		0.125	
		1		1	

Step 4: Compute the split cost and unused costs

- Split cost: The pay per use cost allocation of the EC2 instance cost based on allocated CPU and memory usage by the Kubernetes pod.
- Unused instance cost: The cost of unused CPU or memory resources on the instance.

```
Pod1-Split-cost = (Pod1-vCPU-split-usage-ratio * vCPU-available * Cost-per-
vCPU-hour) + (Pod1-Memory-split-usage-ratio * Memory-available * Cost-per-
GB-hour)
```

= (0.204 * 4 vCPU * $0.17) + (0.25 * 16GB * $0.02) = $0.22

```
Pod1-Unused-cost = (Pod1-vCPU-unused-ratio * Instance-vCPU-unused-ratio
* vCPU-available * Cost-per-VCPU-hour) + (Pod1-Memory-unused-ratio *
Instance-Memory-unused ratio * Memory-available * Cost-per-GB-hour)
```

= (0 * 0 * 4 * $0.17) + (0.286 * 0.125 * 16 * $0.02) = $0.01

```
Pod1-Total-split-cost = Pod1-Split-cost + Pod1-Unused-cost
```

= $0.23

Pod name	Namespace	Split-cost	Unused-cost	Total-split-cost
Pod1	Namespace1	$0.22	$0.01	$0.23
Pod2	Namespace2	$0.38	$0.02	$0.40
Pod3	Namespace1	$0.18	$0.01	$0.19
Pod4	Namespace2	$0.18	$0.01	$0.19
Unused	Unused	$0.04		
		$1	$0.04	$1

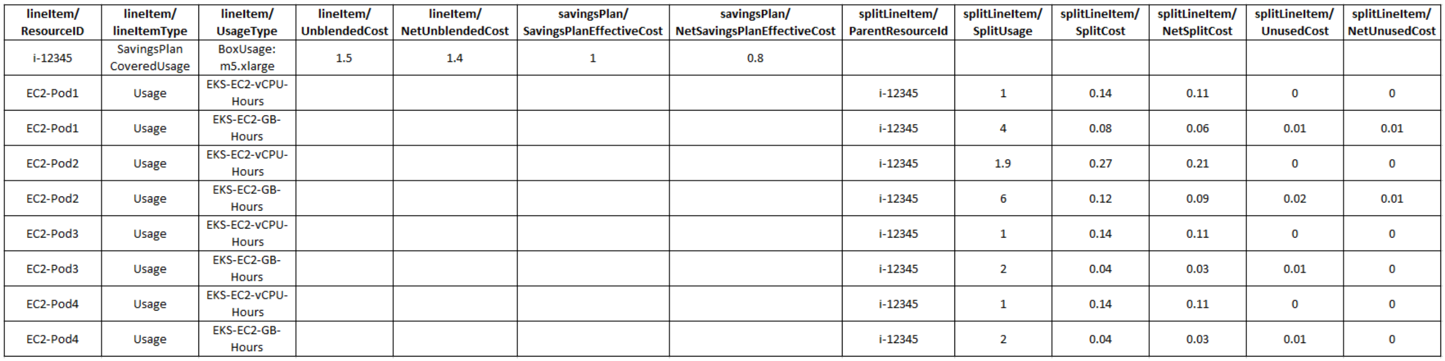
The cost of the service is the sum of the cost of pods associated with each namespace.

Total cost of Namespace1 = $0.23 + $0.19 = $0.42

Total cost of Namespace2 = $0.40 + $0.19 = $0.59

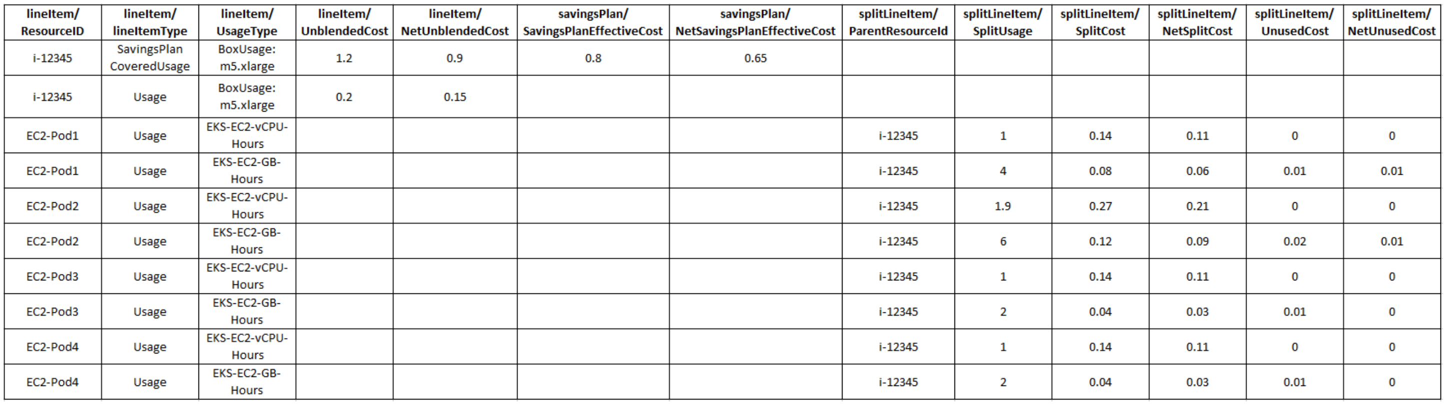
Sample AWS CUR

If you have a Savings Plans covering the entire usage for the EC2 instance in the billing period, amortized costs are computed using savingsPlan/SavingsPlanEffectiveCost.



lineItem/ ResourceID	lineItem/ lineItemType	lineItem/ UsageType	lineItem/ UnblendedCost	lineItem/ NetUnblendedCost	savingsPlan/ SavingsPlanEffectiveCost	savingsPlan/ NetSavingsPlanEffectiveCost	splitLineItem/ ParentResourceId	splitLineItem/ SplitUsage	splitLineItem/ SplitCost	splitLineItem/ NetSplitCost	splitLineItem/ UnusedCost	splitLineItem/ NetUnusedCost
i-12345	SavingsPlan CoveredUsage	BoxUsage: m5.xlarge	1.5	1.4	1	0.8						
EC2-Pod1	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod1	Usage	EKS-EC2-GB-Hours					i-12345	4	0.08	0.06	0.01	0.01
EC2-Pod2	Usage	EKS-EC2-vCPU-Hours					i-12345	1.9	0.27	0.21	0	0
EC2-Pod2	Usage	EKS-EC2-GB-Hours					i-12345	6	0.12	0.09	0.02	0.01
EC2-Pod3	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod3	Usage	EKS-EC2-GB-Hours					i-12345	2	0.04	0.03	0.01	0
EC2-Pod4	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod4	Usage	EKS-EC2-GB-Hours					i-12345	2	0.04	0.03	0.01	0

If you have a Savings Plans covering partial usage for the EC2 instance in the billing period and the rest of the EC2 instance usage is billed at On-Demand rates, EC2 instance amortized costs are computed using savingsPlan/SavingsPlanEffectiveCost (for SavingsPlanCoveredUsage) + lineItem/ UnblendedCost (for On-Demand usage).



lineItem/ ResourceID	lineItem/ lineItemType	lineItem/ UsageType	lineItem/ UnblendedCost	lineItem/ NetUnblendedCost	savingsPlan/ SavingsPlanEffectiveCost	savingsPlan/ NetSavingsPlanEffectiveCost	splitLineItem/ ParentResourceId	splitLineItem/ SplitUsage	splitLineItem/ SplitCost	splitLineItem/ NetSplitCost	splitLineItem/ UnusedCost	splitLineItem/ NetUnusedCost
i-12345	SavingsPlan CoveredUsage	BoxUsage: m5.xlarge	1.2	0.9	0.8	0.65						
i-12345	Usage	BoxUsage: m5.xlarge	0.2	0.15								
EC2-Pod1	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod1	Usage	EKS-EC2-GB-Hours					i-12345	4	0.08	0.06	0.01	0.01
EC2-Pod2	Usage	EKS-EC2-vCPU-Hours					i-12345	1.9	0.27	0.21	0	0
EC2-Pod2	Usage	EKS-EC2-GB-Hours					i-12345	6	0.12	0.09	0.02	0.01
EC2-Pod3	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod3	Usage	EKS-EC2-GB-Hours					i-12345	2	0.04	0.03	0.01	0
EC2-Pod4	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod4	Usage	EKS-EC2-GB-Hours					i-12345	2	0.04	0.03	0.01	0

Example of split cost allocation data for accelerated instances

The purpose of the following example is to show you how split cost allocation data is calculated by computing the cost of Kubernetes namespace and pods in Amazon EKS clusters. The rates used throughout the example are for illustrative purposes only.

You have the following usage in a single hour:

- Single EC2 instance that is running four pods across two namespaces, and you want to understand the costs of each namespace.
- The EC2 instance is p3.16xlarge with 8 GPU, 64 vCPU, and 488 GB RAM.
- Amortized cost of the instance is $10/hr.

Split cost allocation data normalizes the cost per resource based on a relative ratio of GPU:(cpu: memory) of 9:1. This implies that a unit of GPU costs 9x as much as a unit of CPU and memory. CPU and memory are then assigned a weight of 9:1. For a non-accelerated EC2 instance, the current default behavior will be adopted which is cpu: memory weight defaults to 9:1.

Step 1: Compute the unit cost

Based on the cpu and memory resources on the EC2 instance and using the ratio of mentioned above, Split Cost Allocation data first calculates the unit cost per GPU, vCPU-hr and GB-hr.

```
GPU-Weight =9
```

```
GPU+Memory-Weight =1
```

```
CPU-Weight=1*.9=.9
```

```
Memory-Weight=1*0.1=0.1

Hourly-Instance-Cost=$10

GPU-Available=8

Memory-Available=488

CPU-Available=64

UnitCostPerResource = Hourly-Instance-Cost/(( GPU-Weight * GPU-Available)
+ (Memory-Weight * Memory-Available) + (CPU-Weight * CPU-Available)) = $10/
((9*8gpu)+ (0.1 * 488GB) + (.9 * 64vcpu)) = $0.056

Cost-per-GPU-Hour = GPU-Weight * UnitCostPerResource = 9 * $0.056 = $0.504

Cost-per-vcpu-Hour = CPU-Weight * UnitCostPerResource = .9 * $0.056 = $0.05

Cost-per-GB-Hour = Memory-Weight * UnitCostPerResource = .1 * $0.056 =
$0.00506
```

Table 1: Unit cost calculation

Instanc	Instanc Type	vCPU Availab	GPU Availab	**	Memor Availab	Amorti Cost per Hour	Cost per vCPU-Hour	Cost per GPU-Hour	Cost per GB-Hour
Instance 1	p3.16xl ge	64	8		488	$10	$0.05	$0.50	0.005

Step 2: Calculate allocated and unused capacity

Allocated Capacity

The GPU, vcpu and Memory allocated to the Kubernetes Pod from the parent EC2 Instance, defined as the Maximum of (reserved, used) capacity

Instance Unused Capacity

The unused capacity of GPU, vcpu and Memory

```
Pod1-Allocated-GPU = Max (1 GPU, 1 GPU) = 1 GPU

Pod1-Allocated-vcpu = Max (16 vcpu, 4 vcpu) = 16 vcpu

Pod1-Allocated-Memory = Max (100 GB, 60 GB) = 100 GB

Instance-Unused-GPU = Max (GPU-Available - SUM(Allocated-vcpu), 0)

= Max (8 – 8, 0) = 0

Instance-Unused-vcpu = Max (CPU-Available - SUM(Allocated-vcpu), 0)

= Max (16 – 18, 0) = 0

Instance-Unused-Memory = Max (Memory-Available - SUM(Allocated-Memory), 0)

= Max (488 – 440, 0) = 48 GB
```

In this example, the instance has CPU over subscription, attributed to Pod 2 that used more GPU and vcpu that what was reserved.

Table 2: Calculate Allocated and Unused Capacity

Pod Name	Namesp	vcpu Reserve	vcpu Used	vcpu Allocat	GPU Reserve	GPU used	GPU Allocat	Memor Reserve	Memor Used	Memory Allocated
Pod 1	Namesp 1	16	4	16	1	1	1	100	60	100
Pod 2	Namesp 2	16	18	18	2	3	3	100	140	140
Pod 3	Namesp 1	16	4	16	2	1	2	100	60	100
Pod 4	Namesp 2	16	4	16	2	2	2	100	40	100
Unused	Unused	0	34	0	1	1	0	88	188	48
***		64	32	66	8	8	8	488	488	488

Step 3: Compute the split usage and utilization ratios

Split usage ratio

> The percentage of CPU or memory used by the Kubernetes pod compared to the overall CPU or memory available on the EC2 instance.

Unused ratio

> The percentage of CPU or memory used by the Kubernetes pod compared to the overall CPU or memory used on the EC2 instance (that is, not factoring in the unused CPU or memory on the instance).

The percentage of CPU or memory used by the Kubernetes Pod compared to the overall CPU or memory available on the EC2 Instance.

```
Pod1-GPU-Utilization-Ratio = Allocated-GPU / Total-GPU

= 1 gpu / 8 gpu = 0.125

Pod1-vcpu-Utilization-Ratio = Allocated-vcpu / Total-vcpu

= 16 vcpu / 66 vcpu = 0.24

Pod1-Memory-Utilization-Ratio = Allocated-GB / Total-GB

= 100 GB/ 488GB = 0.205

Pod1-GPU-Split-Ratio = Pod1-GPU-Utilization-Ratio / (Total-GPU-Utilization-Ratio – Instance-Unused-GPU). Set to 0 if Instance-Unused-GPU = 0

= 0 since Instance-Unused-GPU is 0

Pod1-vcpu-Split-Ratio = Pod1-CPU-Utilization-Ratio / (Total-CPU-Utilization-Ratio – Instance-Unused-CPU). Set to 0 if Instance-Unused-CPU = 0

= 0 since Instance-Unused-CPU is 0

Pod1-Memory-Split-Ratio = Pod-Memory-Utilization-Ratio / (Total-Utilization-Ratio – Instance-Unused-Memory). Set to 0 if Instance-Unused-Memory = 0

= 0.204/ (1-0.102) = 0.227
```

Table 3: Compute Utilization ratios

Pod Name	Namespace	vcpu Utilization	vcpu Split Ratio	GPU Utilization	GPU Split Ratio	Memory Utilization	Memory Split Ratio
Pod 1	Namespace 1	0.242	0	0.125	0	0.205	0.227
Pod 2	Namespace 2	0.277	0	0.375	0	0.287	0.318
Pod 3	Namespace 1	0.242	0	0.25	0	0.205	0.227
Pod 4	Namespace 2	0.242	0	0.25	0	0.205	0.227
Unused	Unused	0				0.098	
		1	0	1	0	1	1

Step 4: Compute the split cost and unused costs

Split Cost

> The pay per use cost allocation of the EC2 Instance cost based on allocated CPU and memory usage by the Kubernetes Pods

Unused Instance Cost

> The cost of unused CPU or memory resources on the instance

```
Pod1-Split-Cost = (Pod1-GPU-Utilization-Ratio * GPU-Available * Cost per GPU-Hour) + (Pod1-vcpu-Utilization-Ratio * vcpu-Available * Cost per vcpu-Hour) + (Pod1-Memory-Utilization-Ratio * Memory-Available * Cost per GB-Hour)
```

```
= (.125*8gpu*$0.504) + (0.242 * 64 vcpu * $0.05) + (0.204 * 488GB * $0.00506) = 0.504+ 0.774 + 0.503 = $1.85
```

```
Pod1-Unused-Cost = (GPU-Split-Ratio * Unused-Cost) + (vcpu-Split-Ratio *
Unused-Cost) + (Memory-Split-Ratio * Unused-Cost)
```

```
= (0*0*8*$0.504) + (0 * $0.05) + (0.227 *.102*488GB*$.00506) = $0.06
```

```
Pod1-Total-Split-Cost = Pod1-Split-Cost + Pod1-Unused-Cost = $1.85 + $0.06
= $1.91
```

[Note: Unused cost = Unused util ratio * Total resource * resource hourly cost]

Table 4 - Summary of the Split and Unused costs calculated each hour for all Pods running within the cluster

Pod Name	Namespace	Split Cost	Unused Cost	Total Cost
Pod 1	Namespace 1	$1.85	$0.06	$1.91
Pod 2	Namespace 2	$3.18	$0.09	$3.26
Pod 3	Namespace 1	$2.35	$0.06	$2.41
Pod 4	Namespace 2	$2.35	$0.06	$2.41
Total				$10

Using Kubernetes labels for cost allocation in EKS

Split cost allocation data supports Kubernetes labels as cost allocation tags for Amazon EKS clusters. While these labels are automatically imported as user-defined cost allocation tags, they require activation at the management account level. Once activated, you can use them to attribute pod-level costs in your Cost and Usage Reports (CUR) using custom attributes such as cost center, application, business unit, and environment.

This feature helps organizations accurately track and allocate costs in shared EKS environments across teams, projects, or departments. Using Kubernetes labels, you can allocate your Kubernetes costs based on your specific business requirements and organizational design.

Prerequisites

As prerequisites for using Kubernetes labels with split cost allocation data:

- You need to enable split cost allocation data in the AWS Billing and Cost Management console. This must be enabled at the management account level. For details, see Enabling split cost allocation data.
- You need an EKS cluster for which you want to track split cost allocation data. This can be an existing cluster, or you can create a new one. For more information, see Create an Amazon EKS cluster in the *Amazon EKS User Guide*.
- You must have labels assigned to your pods in the EKS cluster. For more information on how to create labels in Kubernetes, see Labels and Selectors in the *Kubernetes Documentation*.

Working with Kubernetes labels in EKS

Split cost allocation data supports up to 50 Kubernetes labels per pod, which are sorted alphabetically before being imported as cost allocation tags. Any labels beyond the first 50 are automatically discarded. If you need to add a new cost allocation tag after reaching the 50-label limit, you must first remove an existing label and ensure your new label falls within the first 50 when alphabetically sorted.

> **Note**
>
> Some AWS managed services automatically add labels to EKS pods. These labels count toward the 50-label limit per pod and will appear on your cost allocation tags page. While Kubernetes labels have no size restrictions, cost allocation tags have specific character limits: 128 characters for tag keys and 256 characters for tag values. Labels that exceed these character limits will be discarded and not presented as cost allocation tags. It's recommended to create labels that follow these character limits for cost allocation purposes.

The imported Kubernetes labels appear as cost allocation tags and must be activated at the payer account level. For more information on cost allocation tags and activation, see Using user-defined cost allocation tags. The following cost allocation tag limits apply: 50 user-defined tags per resource and 500 user-defined tags per payer account. System-generated tags do not count toward these limits.

Note

After you create and apply user-defined tags to your resources, it can take up to 24 hours for the tag keys to appear on your cost allocation tags page. Once you activate the tags, it can take an additional 24 hours for them to become active.

Managing Kubernetes labels and cost allocation tags

You can add, delete, and edit Kubernetes labels in EKS, as well as deactivate the associated cost allocation tags. The following describes the expected behavior for each action.

Adding a new label

You can add a new Kubernetes label to a pod. If the label limit of 50 has not been reached, the new label will be imported and offered as a cost allocation tag, which can then be activated. However, if the limit of 50 has been reached, the new label will not be imported even if it falls within the alphabetical sort order of first 50 labels. You must first deactivate an existing cost allocation tag to import a new label.

Editing a label

Kubernetes does not allow you to edit a label key. To change a label key, you must remove it and add a new label. However, you can edit label values, which will be reflected in your next CUR.

Deleting a label

You can remove a label from EKS pods. Note that removing a label does not automatically deactivate its associated cost allocation tag. Split cost allocation data will continue to populate in CUR until you explicitly deactivate the cost allocation tag.

Deactivating a cost allocation tag

You can deactivate any cost allocation tag created from Kubernetes labels. Once deactivated, data will no longer populate in the respective columns, and the column will be deleted from the next month’s CUR.

Best practices for managing Kubernetes labels for cost allocation

Kubernetes labels provide significant flexibility in shared cost allocation modeling. To maximize the potential of this capability, we recommend following these best practices to optimize your cost management approach.

Understanding label limits

The 50-label-per-pod limit is based on alphabetical sorting. Only the first 50 alphabetically ordered labels will be imported for cost allocation. To ensure critical labels are included, carefully plan your label naming to ensure important labels appear within the first 50 when alphabetically sorted.

Following character constraints

AWS cost allocation tags have the following character limits:

- Tag keys: 128 characters
- Tag values: 256 characters

While Kubernetes allows longer labels, any labels exceeding these limits will not be imported. Design your labels within these limits to ensure successful cost allocation tracking.

Adding new labels when at capacity

When a pod has reached the 50-label limit and you need to add a new cost allocation label, follow these steps:

1. Review existing labels and identify a cost allocation tag to deactivate.
2. Deactivate the selected tag.
3. Add the new cost allocation label.
4. Verify the new label falls within the first 50 alphabetically sorted labels.

> **Note**
>
> Remember that only the first 50 alphabetically sorted labels are used for cost allocation.

Using split cost allocation data with Amazon Managed Service for Prometheus

Splitting the cost data for Amazon EKS requires that you collect and store metrics from your clusters, including memory and CPU usage. Amazon Managed Service for Prometheus can be used for this purpose.

Once you're opted in to split cost allocation data and your Amazon Managed Service for Prometheus workspace starts receiving the two required metrics (`container_cpu_usage_seconds_total` and `container_memory_working_set_bytes`), split cost allocation data recognizes the metrics and uses them automatically.

> **Note**
>
> The two required metrics (`container_cpu_usage_seconds_total` and `container_memory_working_set_bytes`) are present in the default Prometheus scrape configuration and the default configuration provided with an AWS managed collector. However, if you customize these configurations, do not relabel, modify, or remove the following labels from the `container_cpu_usage_seconds_total` and `container_memory_working_set_bytes` metrics: `name`, `namespace`, and `pod`. If you relabel, modify, or remove these labels, it can impact the ingestion of your metrics.

You can use Amazon Managed Service for Prometheus to collect EKS metrics from a single usage account, in a single Region. The Amazon Managed Service for Prometheus workspace must be in that account and Region. You need one Amazon Managed Service for Prometheus instance for each usage account and Region for which you want to monitor the costs. You can collect metrics for multiple clusters in the Amazon Managed Service for Prometheus workspace, as long as they're in the same usage account and Region.

The following sections describe how to send the correct metrics from your EKS cluster to the Amazon Managed Service for Prometheus workspace.

Prerequisites

As prerequisites for using Amazon Managed Service for Prometheus with split cost allocation data:

- You need to enable split cost allocation data in the AWS Billing and Cost Management console. For details, see Enabling split cost allocation data. Opting in to split cost allocation data creates a service-linked role in each usage account to query Amazon Managed Service for Prometheus for the Amazon EKS cluster metrics in that account. For more information, see Service-linked roles for split cost allocation data.
- You need an EKS cluster for which you want to track split cost allocation data. This can be an existing cluster, or you can create a new one. For more information, see Create an Amazon EKS cluster in the *Amazon EKS User Guide*.

Note

You will need the `EKS cluster ARN`, `security group IDs`, and at least two `subnet IDs` (in different availability zones) for use in later steps.
(optional) Set your EKS cluster's authentication mode to either `API` or `API_AND_CONFIG_MAP`.

- You need an Amazon Managed Service for Prometheus instance in the same account and Region as your EKS cluster. If you do not already have one, you can create one. For more information on creating an Amazon Managed Service for Prometheus instance, see Create a workspace in the *Amazon Managed Service for Prometheus User Guide*.

Note

You will need the `Amazon Managed Service for Prometheus workspace ARN` for use in later steps.

Forwarding EKS metrics to Amazon Managed Service for Prometheus

Once you have an EKS cluster and an Amazon Managed Service for Prometheus instance, you can forward the metrics from the cluster to the instance. You can send metrics in two ways.

- Option 1: Use an AWS managed collector. This is the simplest way to send metrics from an EKS cluster to Amazon Managed Service for Prometheus. However, it does have a limit of only scraping metrics every 30 seconds at most.
- Option 2: Create your own Prometheus agent. In this case, you have more control over the scraping configuration, but you must manage the agent after creating it.

Option 1: Using an AWS managed collector

Using an AWS managed collector (a *scraper*) is the simplest way to send metrics from an EKS cluster to an Amazon Managed Service for Prometheus instance. The following procedure steps you through creating an AWS managed collector. For more detailed information, see AWS managed collectors in the *Amazon Managed Service for Prometheus User Guide*.

Note

AWS managed collectors have a minimum scrape interval of 30 seconds. If you have short-lived pods, the recommendation is to set your scraper interval to 15 seconds. To use a 15 second scraper interval, use option 2 to create your own Prometheus agent.

There are three steps to create an AWS managed collector:

1. Create a scraper configuration.
2. Create the scraper.
3. Configure your EKS cluster to allow the scraper to access metrics.

Step 1: Create a scraper configuration

In order to create a scraper, you must have a scraper configuration. You can use a default configuration, or create your own. The following are three ways to get a scraper configuration:

- Get the default configuration using the AWS CLI, by calling:

```
aws amp get-default-scraper-configuration
```

- Create your own configuration. For details, see the Scraper configuration instructions in the *Amazon Managed Service for Prometheus User Guide*.
- Copy the sample configuration provided in that same Scraper configuration instructions in the *Amazon Managed Service for Prometheus User Guide*.

You can edit the scraper configuration, to modify the scrape interval or to filter the metrics that are scraped, for example.

To filter the metrics that are scraped to just include the two that are needed for split cost allocation data, use the following scraper configuration:

```
global:
   scrape_interval: 30s
   #external_labels:
     #clusterArn: <REPLACE_ME>
scrape_configs:
  - job_name: kubernetes-nodes-cadvisor
    scrape_interval: 30s
    scrape_timeout: 10s
    scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
    - role: node
    relabel_configs:
    - regex: (.+)
      replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
      source_labels:
      - __meta_kubernetes_node_name
      target_label: __metrics_path__
    - replacement: kubernetes.default.svc:443
      target_label: __address__
    metric_relabel_configs:
    - source_labels: [__name__]
      regex: 'container_cpu_usage_seconds_total|container_memory_working_set_bytes'
      action: keep
```

Once you have the scraper configuration, you must base64 encode it for use in *step 2*. The configuration is a text YAML file. To encode the file, use a website such as https://www.base64encode.org/.

Step 2: Create the scraper

Now that you have a configuration file, you need to create your scraper. Create a scraper using the following AWS CLI command, based on the variables outlined in the prerequisites section. You must use information from your EKS cluster for the *<EKS-CLUSTER-ARN>*, *<SG-SECURITY-GROUP-ID>*, and *<SUBNET-ID>* fields, replace *<BASE64-CONFIGURATION-BLOB>* with the scraper configuration you created in the previous step, and replace *<AMP_WORKSPACE_ARN>* with your Amazon Managed Service for Prometheus workspace ARN.

```
aws amp create-scraper \
--source eksConfiguration="{clusterArn=<EKS-CLUSTER-ARN>,securityGroupIds=[<SG-
SECURITY-GROUP-ID>],subnetIds=[<SUBNET-ID>]}" \
--scrape-configuration configurationBlob=<BASE64-CONFIGURATION-BLOB> \
--destination ampConfiguration={workspaceArn="<AMP_WORKSPACE_ARN>"}
```

Note down the `scraperId` that is returned for use in *step 3*.

Step 3: Configure your EKS cluster to allow the scraper to access metrics

If your EKS cluster's authentication mode is set to either `API` or `API_AND_CONFIG_MAP`, then your scraper will automatically have the correct in-cluster access policy, and the scrapers will have access to your cluster. No further configuration is required, and metrics should be flowing to Amazon Managed Service for Prometheus.

If your EKS cluster's authentication mode is not set to `API` or `API_AND_CONFIG_MAP`, you will need to manually configure the cluster to allow the scraper to access your metrics through a ClusterRole and ClusterRoleBinding. To learn how to enable these permissions, see Manually configuring an EKS cluster for scraper access in the *Amazon Managed Service for Prometheus User Guide*.

Once the scraper is active, verify that both metrics (`container_cpu_usage_seconds_total` and `container_memory_working_set_bytes`) are being pushed to your Amazon Managed Service for Prometheus workspace.

```
awscurl --service="aps" --region="<REGION>" "https://aps-
workspaces.<REGION>.amazonaws.com/workspaces/<WorkSpace_ID>/api/v1/label/__name__/
values"
```

Output:

```
{
"status": "success",
"data": [
"container_cpu_usage_seconds_total",
"container_memory_working_set_bytes",
"scrape_duration_seconds",
"scrape_samples_post_metric_relabeling",
"scrape_samples_scraped",
"scrape_series_added",
"up"
]
}
```

Option 2: Creating your own Prometheus agent

If you can't use the AWS managed collector, or already have your own Prometheus server, you can use your own Prometheus instance as an agent to scrape metrics from your EKS cluster and send them to Amazon Managed Service for Prometheus.

For detailed instructions on how to use your own Prometheus instance as an agent, see Using a Prometheus instance as a collector in the *Amazon Managed Service for Prometheus User Guide*.

The following is a sample Prometheus scrape configuration that includes the Prometheus server scrape interval and the container metrics required for split cost allocation data. If you have short-lived pods, the recommendation is to lower the default Prometheus server scrape interval from 30 seconds to 15 seconds. Note that this can result in high Prometheus server memory usage.

```
global:
   scrape_interval: 30s
   #external_labels:
     #clusterArn: <REPLACE_ME>
scrape_configs:
  - job_name: kubernetes-nodes-cadvisor
    scrape_interval: 30s
    scrape_timeout: 10s
    scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
    - role: node
    relabel_configs:
    - regex: (.+)
      replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
      source_labels:
      - __meta_kubernetes_node_name
      target_label: __metrics_path__
    - replacement: kubernetes.default.svc:443
      target_label: __address__
    metric_relabel_configs:
    - source_labels: [__name__]
      regex: 'container_cpu_usage_seconds_total|container_memory_working_set_bytes'
      action: keep
```

If you followed Set up ingestion from a new Prometheus server using Helm in the in the *Amazon Managed Service for Prometheus User Guide*, then you can update your scrape configuration.

To update your scrape configuration

1. Edit `my_prometheus_values_yaml` from the guide and include the sample scrape config in the `server` block.

2. Run the following command, using `prometheus-chart-name` and `prometheus-namespace` from the *Amazon Managed Service for Prometheus User Guide*.

```
helm upgrade prometheus-chart-name prometheus-community/prometheus -n prometheus-
namespace -f my_prometheus_values_yaml
```

To learn more about `scrape_interval`or how to use a non-global scrape_interval, refer to Prometheus scrape configuration.

Alternatively, you can use the AWS Distro for OpenTelemetry collector that has a Prometheus Receiver, a Prometheus Remote Write Exporter, and the AWS Sigv4 Authentication Extension to achieve remote write access to Amazon Managed Service for Prometheus.

> **Note**
>
> Once you have set up your Prometheus agent, unlike AWS managed collectors, you are responsible for keeping the agent up to date and running to collect metrics.

Estimating your Amazon Managed Service for Prometheus costs

You can use AWS Pricing Calculator to estimate the cost of using Amazon Managed Service for Prometheus for split cost allocation data.

To configure Amazon Managed Service for Prometheus for your estimate

1. Open AWS Pricing Calculator at https://calculator.aws/#/.
2. Choose **Create estimate**.
3. On the **Add service** page, enter **Amazon Managed Service for Prometheus** in the search field, and then choose **Configure**.
4. In the **Description** field, enter a description for your estimate.
5. Choose a **Region**.
6. Select **Calculate the cost using your infrastructure details**. This option allows you to estimate your ingestion, storage, and query sample costs based on your current or proposed infrastructure setup.

7. For **Number of EC2 instances**, enter the total number of EC2 instances across all your clusters for your entire consolidated billing family (including all accounts and Regions). If you use AWS Fargate, use the number of Fargate tasks as a proxy for your EC2 instance count.
8. Split cost allocation data requires two metrics: `container_cpu_usage_seconds_total` and `container_memory_working_set_bytes`. For **Prometheus metrics per EC2 instances**, enter 2.
9. Split cost allocation data suggests a scrape interval of 15 seconds. For **Metric collection interval (in seconds)**, enter 15. If you used a different interval (for example, 30 seconds), change this to the interval you set up.
10. Split cost allocation data does not impose any specific requirements for the other parameters so enter appropriate values for the rest of the input parameters as per your business requirements.
11. Choose **Save and add service**.

Using split cost allocation data with Amazon CloudWatch Container Insights

Splitting the cost data for Amazon EKS requires that you collect and store metrics from your clusters, including memory and CPU usage. Amazon CloudWatch Container Insights can be used for this purpose.

Once you've opted in to split cost allocation data and set up the CloudWatch agent with EKS observability add-on on your EKS cluster, split cost allocation data starts receiving the two required metrics (`pod_cpu_usage_total` and `pod_memory_working_set`) in the `ContainerInsights` namespace and uses them automatically. To view the full set of container metrics for EKS, see Amazon EKS and Kubernetes Container Insights metrics in the *Amazon CloudWatch User Guide*.

The following sections describe how to send the correct metrics from your EKS cluster to split cost allocation data.

Prerequisites

As prerequisites for using Amazon CloudWatch Container Insights with split cost allocation data:

- You need to enable split cost allocation data in the AWS Billing and Cost Management console. For details, see Enabling split cost allocation data.
- You need an EKS cluster for which you want to track split cost allocation data. This can be an existing cluster, or you can create a new one. For more information, see Create an Amazon EKS cluster in the *Amazon EKS User Guide*.

Setting up Amazon CloudWatch Container Insights to forward EKS metrics

You need to set up and configure the CloudWatch agent in order to forward EKS metrics. You can use either the Amazon CloudWatch Observability EKS add-on or the Amazon CloudWatch Observability Helm chart to install the CloudWatch agent and the Fluent-bit agent on an EKS cluster. For more information on how to install and set up the CloudWatch agent, see Install the Amazon CloudWatch Observability EKS add-on in the *Amazon CloudWatch User Guide*.

The following are the minimum versions required for the CloudWatch agent and EKS add-on:

- CloudWatch agent version: v1.300045.0
- CloudWatch Observability EKS add-on version: v2.0.1-eksbuild.1

Estimating your Amazon CloudWatch costs

Enabling the feature to use Amazon CloudWatch Container Insights with split cost allocation data adds two new metrics to Amazon CloudWatch Container Insights: `pod_cpu_usage_total` and `pod_memory_working_set`. For details on these metrics, see Amazon EKS and Kubernetes Container Insights metrics in the *Amazon CloudWatch User Guide*.

To understand the costs associated with the feature

1. Open Amazon CloudWatch Pricing at https://aws.amazon.com/cloudwatch/pricing/.
2. Navigate to the **Paid tier** section.
3. Choose the **Container Insights** tab.
4. For a detailed calculation of the costs, navigate to the **Pricing examples** section, and refer to **Example 13 - Container Insights for Amazon EKS and Kubernetes**.

Understanding legacy billing reports

This section describes legacy billing reports offered outside of AWS Data Exports and AWS Cost and Usage Reports. These pages are available for reference. But, we recommend you use AWS Data Exports because these reporting methods will be unavailable at a later date.

Topics

Using Detailed Billing Reports

> ⚠ **Important**
>
> The Detailed Billing Reports feature is unavailable for new customers as of July 8, 2019.

Detailed Billing Reports (DBR) contain similar information to AWS Cost and Usage Reports (AWS CUR) regarding your charges, but calculates the individual line items differently. If you've signed up for both the DBR and AWS CUR, the line items don't match. However, when the reports are finalized at the end of the month, the total cost will align.

AWS stores DBR in Amazon S3 as CSV files using the following naming convention:

```
AWS account number-aws-billing-detailed-line-items-yyyy-mm.csv.zip
```

AWS recreates Detailed Billing Reports (DBR) multiple times a day, overwriting the reports. When AWS overwrites reports, line items might be in a different order than they were in previous reports. A final report is created at the end of the month. For the next month, AWS creates a new report file instead of overwriting the final report from the previous month. Reports for previous months remain in your S3 bucket until you delete them.

For information on how to migrate your DBR to AWS CUR, see the section called “Migrating From DBR to AWS CUR”.

Migrating from Detailed Billing Reports to Cost and Usage Reports

Detailed Billing Reports (DBR) and AWS Cost and Usage Reports (AWS CUR) both provide information about your charges. However, if you're using DBR, we recommend you transfer your report to Cost and Usage Reports.

Topics

- Comparing benefits of the Cost and Usage Reports (AWS CUR)
- Key differences between Detailed Billing Reports and Cost and Usage Reports
- Reporting on advanced charge types

Comparing benefits of the Cost and Usage Reports (AWS CUR)

AWS CUR provides the most comprehensive source of information. You can use AWS CUR to understand individual costs in depth, and to analyze them in greater detail. This is especially useful at an enterprise scale. AWS CUR is helpful if you have complex cost management needs and require dedicated query or analytic-based systems. AWS CUR also provides detailed information about Reserved Instances (RI), including amortized costs.

Comprehensive reservation information

Reserved Instances (RI), or reservations, offer a discounted hourly rate compared to On-Demand usage in exchange for committing to a one- or three-year term of service. This can result in significant savings. You can use AWS CUR to monitor and manage your reservation portfolio. AWS CUR provides you with detailed information, such as reservation Amazon Resource Numbers (ARNs), numbers of reservations, and total RIs. You can track your reservation-related discounts to specific resources to build a better understanding of your savings.

Detail Billing Reports (DBR) provide a subset of this metadata, but work is required to transform the required columns.

AWS CUR provides additional columns that are not available in DBR, such as information regarding your amortized reservation costs. For more information, see the section called “Understanding your amortized reservation data”.

On-Demand pricing availability

AWS CUR provides information regarding the On-Demand rates for each individual line item of usage. You can use this information to quantify your savings by subtracting the amount you paid from the On-Demand rate. This also gives you the flexibility of choosing to allocate your costs using public On-Demand rates.

DBR doesn’t contain information for On-Demand rates, but only the billed amount. This makes it difficult to calculate your overall savings or to allocate costs using On-Demand rates.

Granular breakdown of discounts

AWS CUR can access a granular view of the usage-based discounts. If discounts were applied, you can use AWS CUR to view the following:

- Cost before being discounted
- Discounted amount
- Total cost after the discount was applied at the line item level

DBR does not contain a granular breakdown of your discounts.

Automated data ingestion at scale

When you use AWS CUR, you can easily configure an event to trigger an automated data ingestion process, streamlining the process of refreshing the billing data in your in-house systems. AWS CUR data can automatically be refreshed when charges related to previous months are detected.

Additionally, AWS CUR is generated as multiple files, providing the added benefit of segmenting the data into smaller pieces. This way, you can ingest the data according to the processes used by multiple workers. Moreover, you can retry data downloads in smaller pieces.

AWS CUR is formatted in a way that enables you to locate and extract data quickly. This report is modeled from a manifest file that contains information for the overall structure of the data. This includes a list of every column that's contained in the report. Using this information, you can extend the report and include new information regarding your usage when it becomes available.

Cross-product integration

AWS CUR is integrated with Amazon Redshift, Quick, and Amazon Athena. You can use AWS CUR to build an AWS-based cost management solution. AWS CUR also provides data in Parquet format. This provides you with more options for building out your own cost and usage reporting system. For more information, see AWS Cost and Usage Reports Manifest Files in the *AWS Billing User Guide*.

Key differences between Detailed Billing Reports and Cost and Usage Reports

There are a few differences between DBR and AWS CUR to consider after you migrate to AWS CUR. For example, you might need to adjust how you ingest the data into your systems.

File structure

Detailed Billing Reports (DBR) are delivered as a single file. In contrast, AWS CUR are a consolidated set of files. In AWS CUR, you can view the following files in your Amazon S3 bucket:

- A set of data files that contain all of your usage line items
- A separate data file that contains all of your discounts (if applicable)
- A manifest file that lists all of the data files that belong to a single report

Column structure

DBR have a fixed list of columns, limiting its flexibility. AWS CUR don't have a fixed column structure, and instead allow you to freely add or remove columns as needed. When you begin using a new AWS service, AWS CUR can dynamically start to include new data in the report that might be useful in your case. The manifest file provides a map of all columns present in the report.

Equivalent Column Names for DBR and AWS CUR

DBR column name	AWS CUR column name
InvoiceId	bill/InvoiceId
PayerAccountId	bill/PayerAccountId
LinkedAccountId	lineItem/UsageAccountId
ProductName	product/ProductName
SubscriptionId	reservation/subscriptionid
UsageType	lineItem/UsageType
Operation	lineItem/Operation
AvailabilityZone	lineItem/AvailabilityZone
ReservedInstance	Not Supported
ItemDescription	lineItem/LineItemDescription
UsageStartDate	lineItem/UsageStartDate

DBR column name	AWS CUR column name
UsageEndDate	lineItem/UsageEndDate
UsageQuantity	lineItem/UsageAmount
BlendedRate	lineItem/BlendedRate
BlendedCost	lineItem/BlendedCost
UnBlendedRate	lineItem/UnblendedRate
UnBlendedCost	lineItem/UnblendedCost
ResourceId	lineItem/ResourceId
RecordType	Not Supported
PricingplanId	Not Supported
RateID	pricing/RateId

> **Note**
>
> There's no equivalent for RecordId in AWS CUR. But, you can gather this information by combining identity/LineItemId, identity/TimeInterval, and bill/BillType.

Retrieving DBR RecordType values through AWS CUR

RecordType values in DBR	Syntax to retrieve RecordType through AWS CUR	Use case
LineItem	SELECT SUM(line_item_unblended_cost) FROM [CUR] WHERE line_item_line_item_type = 'Usage'	Usage line item partitions out usage costs from one-time charges (for example, upfront RI payment).
InvoiceTotal	SELECT (bill_invoice_id), sum(line_item_unbl	You can use invoice total to reconcile your costs between

RecordType values in DBR	Syntax to retrieve RecordType through AWS CUR	Use case
	ended_cost) FROM [CUR] GROUP BY bill_invoice_id	Invoices and Cost and Usage Reports.
AccountTotal	SELECT line_item_usage_account_id, sum(line_item_unblended_cost) FROM [CUR] GROUP BY line_item_usage_account_id	You can use account total to isolate costs related to your member accounts for charge back purposes.
StatementTotal	SELECT SUM(line_item_unblended_cost) FROM [CUR]	You can use statement total to understand your costs for the billing period.
Discount	SELECT SUM(line_item_unblended_cost) FROM [CUR] WHERE line_item_line_item_type = 'Discount'	You can use discount line items to identify all of your discount-related line items.
Rounding	Not yet supported	Not yet supported

Reporting on advanced charge types

Refunds

AWS CUR: Refunds are identified by filtering for the `lineItem/LineItemDescription = ‘Refund’` string.

DBR: Refunds are identified by checking the ItemDescription column for the `‘Refund’` substring.

Credits

AWS CUR: Credits are identified by filtering for the `lineItem/LineItemDescription = ‘Credit’` string.

DBR: Credits are identified by checking the ItemDescription column for the `‘Credit’` substring.

Taxes

AWS CUR: Taxes are identified by filtering for the `lineItem/LineItemDescription = ‘Tax’` string.

DBR: Taxes are identified by checking the ItemDescription column for the `‘Tax’` substring.

Identifying reservation-related upfront costs

AWS CUR: Reservation-related upfront costs are identified by filtering for the `"lineItem/LineItemType" = 'Fee'` string.

DBR: Reservation-related upfront costs are identified by checking the UsageType column for the `'HeavyUsage'` substring, and whether the `'SubscriptionId'` is null.

Identifying reservation-related monthly fees

AWS CUR: Reservation-related monthly fees are identified by filtering for the `"lineItem/LineItemType" = 'RIfee'` string.

DBR: Reservation-related monthly fees are identified by checking the UsageType column for the `'HeavyUsage'` substring.

Identifying instances that received reserved instance benefits

AWS CUR: Reservation-related upfront fees are identified by filtering for the `"lineItem/LineItemType" = 'DiscountedUsage'` string.

DBR: Reservation-related upfront fees are identified by checking the ReservedInstance column for the `'Y'` substring.

Understanding unused reservation costs

You can use AWS Cost and Usage Reports (AWS CUR) to understand unused RI costs. The following four scenarios show how.

Note

In the following tables, the columns and rows from AWS CUR and DBR/DBR-RT are transposed for clarity. The values in the first column represent the headers of a report.

Scenario 1: RI usage is 100%

RI Fee line item has $0 unused cost and 0 usage hours.

Using the DBR/DBR-RT, you can understand your unused RI usage and costs by referring to the fields UsageQuantity and UnblendedCosts for RI Fee line items. RI Fee line items can be identified by the existence of ‘purchased hours’ information in the ItemDescription field. Table 1 illustrates the columns and information used to manage unused RI costs in the DBR and DBR-RT report.

Table 1 – Unused RI costs for a 100% RI usage in DBR and DBR-RT before June 17, 2019

ProductName	Amazon Elastic Compute Cloud	Amazon Elastic Compute Cloud
UsageType	HeavyUsage:c3.8xlarge	HeavyUsage:c3.8xlarge
Operation	RunInstances	RunInstances
Availability Zone	us-east-1a	us-east-1a
Reserved Instance	Y	Y
ItemDescription	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge (744 hours purchased, 744 hours used)	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge
Usage Quantity	0	744
Unblended Rate	0.1	0.1
Unblended Cost	0	74.4

Using AWS CUR, you can understand your unused RI usage and costs by referring to the fields ‘reservation/ UnusedQuantity’ and ‘reservation/ UnusedRecurringFee’ for RI Fee line items. Table 4 below illustrates the current columns and information utilized to manage unused RI costs in AWS CUR.

Table 2 – Unused RI costs for a 100% RI usage in AWS CUR

lineitem/Productcode	Amazon EC2	Amazon EC2
UsageType	`HeavyUsage:c3.8xlarge`	`USW2-BoxUsage:c3.8xlarge`
lineitem/LineItemType	`RI Fee`	`DiscountedUsage`
lineitem/LineItemDescription	`USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge`	`USD 0.00 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge`
lineitem/UsageAmount	`744`	`744`
lineitem/NormalizedUsageAmount	`47,616`	`47,616`
lineitem/UnblendedRate	`0.1`	`0`
lineitem/UnblendedCost	`74.4`	`0`
reservation/UnusedQuantity	`0`	
reservation/UnusedRecurringFee	`0`	
reservation/UnusedAmortizedUpfrontFeeForBillingPeriod	`0`	
reservation/RecurringFeeForUsage		`74.4`
reservation/AmortizedUpfrontCostForUsage		`5`

reservation/EffectiveCost		79.4

In addition to matching the current functionality supported by DBR/DBR-RT, AWS CUR has the following advantages:

- Using AWS CUR, you are able to access information regarding the EffectiveCost for the DiscountedUsage line item, which includes both the recurring and upfront fees. The DBR only accounts for recurring fees.
- In AWS CUR, the UsageType field is not transformed for the DiscountedUsage line items whereas DBR replaces the information with RI Fee line item information. This is because the user can group line items in AWS CUR by ReservationARN in order to understand what usage was discounted by which RI.
- In AWS CUR, the LineItemDescription field is not transformed for the RI Fee line item. DBR appends the hours purchased and hours used.

Scenario 2: Partial RI usage

RI Fee line item has unused cost and usage.

Using the DBR/DBR-RT, you can understand your unused RI usage and costs by referring to fields UsageQuantity and UnblendedCosts for RI Fee line items. Table 3 illustrates the columns and information used to manage unused RI costs in the DBR and DBR-RT report.

Table 3 – Unused RI costs for a partial RI usage in DBR and DBR-RT before June 17, 2019

ProductName	Amazon Elastic Compute Cloud	Amazon Elastic Compute Cloud
UsageType	HeavyUsage:c3.8xlarge	HeavyUsage:c3.8xlarge
Operation	RunInstances	RunInstances
Availability Zone	us-east-1a	us-east-1a
Reserved Instance	Y	Y

ItemDescription	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge (744 hours purchased, 644 hours used)	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge
Usage Quantity	100	644
Unblended Rate	0.1	0.1
Unblended Cost	10	64.4

Using AWS CUR, you can understand your unused RI usage and costs by referring to fields ‘reservation/ UnusedQuantity’ and ‘reservation/ UnusedRecurringFee’ for RI Fee line items. Table 4 illustrates the current columns and information utilized to manage unused RI costs in AWS CUR.

Table 4 – Unused RI costs for a partial RI usage in AWS CUR

lineitem/Productcode	Amazon EC2	Amazon EC2
UsageType	HeavyUsage:c3.8xlarge	USW2-BoxUsage:c3.8xlarge
lineitem/LineItemType	RI Fee	DiscountedUsage
lineitem/LineItemDescription	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge	USD 0.00 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge
lineitem/UsageAmount	744	644
lineitem/NormalizedUsageAmount	47,616	47,216
lineitem/UnblendedRate	0.1	0

lineitem/UnblendedCost	74.4	0
reservation/UnusedQuantity	100	
reservation/UnusedRecurringFee	0	
reservation/UnusedAmortizedUpfrontFeeForBillingPeriod	10	
reservation/RecurringFeeForUsage		64.4
reservation/AmortizedUpfrontCostForUsage		5
reservation/EffectiveCost		69.4

In addition to matching the current functionality supported by DBR/DBR-RT, AWS CUR has the following advantages:

- AWS CUR has a separate column representing UnusedQuantity for the RI Fee line item vs. DBR / DBR-RT which overloads the UsageQuantity column with the unused hours

Scenario 3: Capacity reservation

DBR/DBR-RT filters out Capacity Reservations related UnusedBox and UnusedDed usage type line items when covered by an RI because the RI Fee line item already covers the unused amount in the UsageQuantity and UnblendedCost fields. Table 5 illustrates the columns and information utilized to manage unused RI costs in the DBR and DBR-RT report.

Table 5 – Unused RI costs for Capacity Reservation scenario in DBR and DBR-RT prior to June 17 2019

ProductName	Amazon Elastic Compute Cloud	Amazon Elastic Compute Cloud

UsageType	HeavyUsage:c3.8xlarge	HeavyUsage:c3.8xlarge
Operation	RunInstances	RunInstances
Availability Zone	us-east-1a	us-east-1a
Reserved Instance	Y	Y
ItemDescription	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge (744 hours purchased, 734 hours used)	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge
Usage Quantity	10	734
Unblended Rate	0.1	0.1
Unblended Cost	1	73.4

AWS CUR shows these line items as DiscountedUsage. Table 6 illustrates the current columns and information utilized to manage unused RI costs in AWS CUR.

Table 6 – Unused RI costs for the Capacity Reservation scenario in AWS CUR

lineitem/Productcode	Amazon EC2	Amazon EC2	Amazon EC2
UsageType	HeavyUsage:c3.8xlarge	USW2-Reservation:c3.8xlarge	USW2-BoxUsage:c3.8xlarge
lineitem/LineItemType	RI Fee	Usage	DiscountedUsage
lineitem/LineItemDescription	USD 0.10 hourly fee per Linux/	USD 0.00 per Reservation	USD 0.00 hourly fee per Linux/

	UNIX (Amazon VPC), c3:8xlarge	Linux/UNIX (Amazon VPC), c3:8xlarge Instance Hour	UNIX (Amazon VPC), c3:8xlarge
lineitem/UsageAmount	744	744	744
lineitem/NormalizedUsageAmount	47,616		47,216
lineitem/UnblendedRate	0.1	0	0
lineitem/UnblendedCost	74.4	0	0
reservation/RecurringFeeForUsage			64.4
reservation/AmortizedUpfrontCostForUsage			5
reservation/EffectiveCost			69.4

Scenario 4: Size flexible reservations

Utilizing the DBR/DBR-RT, you can understand your unused RI usage and costs by referring to fields UsageQuantity and UnblendedCosts for RI Fee line items. RI Fee line items can be identified by the existence of ‘purchased hours’ information in the ItemDescription field. Table 9 illustrates the columns and information utilized to manage unused RI costs in the DBR and DBR-RT report.

Table 7 – Unused RI costs for a size flex RI scenario in DBR and DBR-RT before June 17, 2019

ProductName	Amazon Elastic Compute Cloud	Amazon Elastic Compute Cloud

UsageType	HeavyUsage:c3.8xlarge	HeavyUsage:c3.8xlarge
Operation	RunInstances	RunInstances
Availability Zone	us-east-1a	us-east-1a
Reserved Instance	Y	Y
ItemDescription	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge (744 hours purchased, 644 hours used)	USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge; UsageType : BoxUsage:c3.large
Usage Quantity	100	644
Unblended Rate	0.1	0.1
Unblended Cost	10	64.4

Using AWS CUR, you can understand your unused RI usage and costs by referring to fields ‘reservation/ UnusedQuantity’ and ‘reservation/ UnusedRecurringFee’ for RI Fee line items. Table 8 illustrates the current columns and information utilized to manage unused RI costs in the AWS CUR.

Table 8 – Unused RI costs for a size flex RI scenario in AWS CUR

lineitem/Productcode	Amazon EC2	Amazon EC2
UsageType	HeavyUsage:c3.8xlarge	USW2-BoxUsage:c3.8xlarge
lineitem/LineItemType	RI Fee	DiscountedUsage
lineitem/LineItemDescription	USD 0.10 hourly fee per Linux/UNI	USD 0.00 hourly fee per Linux/UNI

	X (Amazon VPC), c3:8xlarge	X (Amazon VPC), c3:8large
lineitem/UsageAmount	744	644
lineitem/NormalizedUsageAmount	47,616	2,576
lineitem/UnblendedRate	0.1	0
lineitem/UnblendedCost	74.4	0
reservation/UnusedQuantity	100	
reservation/UnusedRecurringFee	70.37	
reservation/UnusedAmortizedUpfrontFeeForBillingPeriod	5.5	
reservation/RecurringFeeForUsage		4.03
reservation/AmortizedUpfrontCostForUsage		0.5
reservation/EffectiveCost		4.53

In addition to matching the current functionality supported by DBR/DBR-RT, AWS CUR has the following advantages:

- AWS CUR has the NormalizedUsageAmount and quantity. The DBR / DBR-RT do not have columns representing this.
- AWS CUR UsageType and Operation are not transformed for the DiscountedUsage lineitem. The DBR / DBR-RT replaces these values with the RI Fee line item.
- AWS CUR LineItemDescription is not transformed for the DiscountedUsage line item. In DBR / DBR-RT, which replaces with the RI Fee line item description and appends the DiscountedUsage

line item Usage Type to the end of the string i.e. “USD 0.10 hourly fee per Linux/UNIX (Amazon VPC), c3:8xlarge; UsageType: BoxUsage:c3.large”

Downloading a monthly report

You can download a monthly report of your estimated AWS charges from the **Bills** page of the Billing and Cost Management console.

Suppose that you use the consolidated billing feature in AWS Organizations. Then, this report is available only for a management account and includes activity for all the member accounts. Member account owners can obtain the monthly report only from the management account. For more information, see Consolidated Billing for Organizations in the *AWS Billing User Guide*.

The report contains line items for each unique combination of AWS product, usage type, and operation that the account uses. The estimated report is updated several times per day. You can get reports for previous months by selecting the statement period. Start with the report for the month when you signed up for monthly reports. Reports from before you signed up are not available.

Downloading a monthly cost allocation report

⚠ **Important**

The monthly cost allocation report feature will be unavailable at a later date. We recommend that you use the AWS Cost and Usage Reports instead.

You can create custom cost allocation tag sets for your AWS resources that can describe the business dimensions of your AWS usage. You can use these tag sets to organize and track your AWS costs. Many AWS services expose tagging in their feature sets. You create the tags within those services by using the console, API, or the AWS command line interface (CLI). For more information, see Using Cost Allocation Tags in the *AWS Billing User Guide*.

After you create your tags, you can obtain a monthly cost allocation report. This is essentially a monthly report that includes your cost allocation tag sets.

Downloading an AWS Usage Report

⚠ **Important**

On September 15, 2023, the AWS Usage Report will no longer provide access to usage data older than March 1, 2019. To access such usage data, download historical usage and save it locally before September 15, 2023. The AWS Usage Report feature will be unavailable at a later date. We recommend that you use AWS Cost and Usage Reports instead.

You can download a usage report in XML or CSV format. Your report covers a single service, based on usage type, operation, and time period. You can also choose how the data is aggregated.

To download a usage report

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, under **Legacy Pages**, choose **Cost and Usage Reports**.
3. Under the **AWS Usage Report** section, choose **Create a Usage Report**.
4. On the **Download usage report** page, under **Services**, choose the service that you want to view usage for.
5. Choose the **Usage type**.
6. Choose the **Operation**.
7. Choose the **Time period** for the report. If you choose **Custom date range**, you need to specify the **Date range** for the report manually.
8. Under **Report granularity**, choose **Hourly**, **Daily**, or **Monthly**.
9. Choose **Download**, and then choose **XML Report** or **CSV Report**.

ⓘ **Note**

If you download a large report, the content of the report might be truncated. Check the last row of the downloaded file for warnings or error messages. If the report is truncated, download smaller reports by choosing a shorter time period. Another option is to decrease the report granularity from hourly to daily or monthly.

Troubleshooting Cost and Usage Reports

Use the following topics to help you troubleshoot common issues with Cost and Usage Reports.

Topics

There are no report files in the Amazon S3 bucket

Confirm that the Amazon S3 bucket policy grants the **billingreports.amazonaws.com** service permission to put files in the bucket. For more information on the required bucket policy, see Setting up an Amazon S3 bucket for Cost and Usage Reports.

One of my report data partitions is empty

If a report is larger than most applications can handle, then AWS splits the report into multiple files. A report update might have fewer individual file partitions than an earlier report version.

Review the report's manifest file to find any empty files that you don't need to ingest.

My Cost and Usage Report data doesn't match the data in other Billing and Cost Management features

Other Billing and Cost Management features (Cost Explorer, Detailed Billing Reports, Billing and Cost Management console) might present your costs differently for the following reasons:

- The billing features round cost data in different ways.
- The billing features might have different data refresh settings. For example, you can choose whether or not your Cost and Usage Report automatically refreshes a previously closed bill with any refunds, credits, or Support fees applied after the bill is finalized. Cost Explorer automatically reflects the same items. In this scenario, if you don't activate the automatic refresh on your Cost and Usage Report, then the Cost and Usage Report data won't match the Cost Explorer data.
- The billing features can group charges differently. For example, the **Bills** page in the Billing and Cost Management console shows data transfer charges as a separate **Data Transfer** grouping within your **AWS Service Charges**. Meanwhile, Cost and Usage Reports and Cost Explorer show data transfer charges as a usage type for each service.

If after reviewing these reasons you still believe you're seeing discrepancies between your Cost and Usage Report and other Billing and Cost Management features, open a support case to request a review of your cost data. In your support case, make sure to provide the report name and the billing period that you would like reviewed. For more information on opening a case, see Getting help with your exports and reports.

I want to backfill data because I changed the settings of my report

Open a support case to request a backfill of your cost data. In your support case, make sure to provide the report name and the billing period that you want backfilled. For more information on opening a case, see Getting help with your exports and reports.

Note that you can't get a backfill of cost data for the following scenarios:

- You can't get a backfill for cost data from before the date that you created the account.
- If you use AWS Organizations and the structure of your organization changed, such as which account is designated the management account, then you can't get a backfill of data with the previous organization structure.
- If you use AWS Organizations and you change organizations, then you can't get a backfill of data from prior to joining your current organization.

My report file folder in Amazon S3 is in an unnamed folder

Any **/** character in the **Report path prefix** of your report generates an unnamed folder in your Amazon S3 bucket. To remove the unnamed folder in your next report update, edit your report settings and remove the **/** character from the **Report path prefix**. For instructions, see Editing your Cost and Usage Reports configuration.

I can’t select the option to include resource IDs on my report

When you create your report, you can select the option to **Include resource ID**. If you create your report with **Report versioning** set to **Overwrite existing report**, then you can’t modify your **Include resource ID** selection after you create your report. To include resource IDs, you must create a new report and select the **Include resource ID** option.

My Cost and Usage Report queries for Amazon Athena don’t work on Amazon Redshift, or my Amazon Redshift queries don’t work on Amazon Athena

Amazon Athena and Amazon Redshift databases format Cost and Usage Report columns differently. Amazon Athena adds an underscore between words in the column name (line_item_normalized_usage_amount). Amazon Redshift adds an underscore between the column type and the attribute (lineitem_normalizedusageamount). Make sure to modify your queries to match the column name format in Amazon Athena or Amazon Redshift.

The columns included in my report have changed from a previous month

The columns that AWS includes in your report depend on your AWS usage. Every report includes columns with the **identity/**, **bill/**, and **lineItem/** prefixes:

- identity/LineItemId
- identity/TimeInterval
- bill/InvoiceId
- bill/BillingEntity
- bill/BillType
- bill/PayerAccountId
- bill/BillingPeriodStartDate
- bill/BillingPeriodEndDate
- lineItem/UsageAccountId
- lineItem/LineItemType
- lineItem/UsageStartDate
- lineItem/UsageEndDate
- lineItem/ProductCode
- lineItem/UsageType
- lineItem/Operation
- lineItem/AvailabilityZone
- lineItem/ResourceId
- lineItem/UsageAmount
- lineItem/NormalizationFactor
- lineItem/NormalizedUsageAmount
- lineItem/CurrencyCode
- lineItem/UnblendedRate
- lineItem/UnblendedCost
- lineItem/BlendedRate
- lineItem/BlendedCost
- lineItem/LineItemDescription
- lineItem/TaxType
- lineItem/LegalEntity

All other columns are included only if your monthly AWS usage generates data to populate those columns.

For example, your report includes **savingsPlan/** columns only if you used Savings Plans during that month.

The queries or tables based on my report don't work because the columns in my report have changed

The columns that AWS includes in your report depend on your AWS usage for the month. Because the columns included in your report can change, it's a best practice to reference column names instead of column numbers in any custom queries or tables based on your report.

I need help querying my report

For detailed information about querying your Cost and Usage Report, see CUR Query Library Help in the AWS Well-Architected Labs website.

I can't find the billing data for my Amazon EC2 Dedicated Host

In the **ResourceID** column, look for the Dedicated Host ID rather than the instance ID. Because Dedicated Hosts are metered by Dedicated Host running hours, your report shows Dedicated Host usage by metered hours associated with the host ID.

I don't understand the billing data for my Amazon EC2 Elastic IP addresses

Amazon EC2 Elastic IP addresses are metered in aggregate. This means that each line item in your report doesn't correspond with an individual Elastic IP address. Each line item represents the total number of chargeable hours. You can have one Elastic IP address assigned to a running instance at no charge. You're charged per hour on a pro-rata basis for each additional Elastic IP address that you assign to the instance. Additionally, AWS charges an hourly fee for unassigned Elastic IP addresses.

I use consolidated billing and I don't understand the difference between unblended and blended rates or costs

With consolidated billing for AWS Organizations, unblended and blended rates or costs can help you understand how much an account's usage would cost for a standalone account versus a linked

The queries or tables based on my report don't work because the columns in my report have changed 348

account in an organization. Some services offer pricing tiers that can lower unit costs as usage increases. Because AWS aggregates all usage for a service in an organization, individual accounts might access lower-priced tiers sooner when their usage is aggregated in an organization's monthly usage.

Unblended rates are the rates associated with an individual account's service usage. For a line item, the unblended cost is usage multiplied by the unblended rate. The unblended cost would be the cost of the account's usage if it were a standalone account. Blended rates are the rates associated with total usage in an organization averaged across accounts. For a line item, the blended cost is usage multiplied by the blended rate. The blended cost is the cost attributed to the account's usage as a linked account in an organization.

For more information and examples of calculating unblended and blended costs, see Understanding Consolidated Bills in the *AWS Billing User Guide*

Some line items in my report have a blended rate or blended cost of 0

Amazon EC2 line items with a Reserved Instance discount have an blended rate of zero. For these line items, the **LineItemType** is **Discounted Usage**.

The blended cost is the usage multiplied by the blended rate. If the value for blended rate or usage is zero, then the blended cost is also zero.

I don't understand how All Upfront Reserved Instances are amortized in my report

Because All Upfront Reserved Instances are paid in full upfront, the amortized costs are reflected in your report as the upfront payment divided over the associated time period (one year or three years).

reservation/AmortizedUpfrontCostForUsage and **reservation/EffectiveCost** are the same rate for All Upfront Reserved Instances. This is because both columns are an equal division of the upfront payment for the Reserved Instance over the total hours of its term.

It's expected that your report has **RIFee** line items populated for All Upfront Reserved Instances, even though the **RIFee** is $0.00. These line items represent the recurring hourly costs for the month, and they have additional usage data in other columns. All Reserved Instances generate **RIFee** line items.

Security in AWS Cost and Usage Reports

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.

AWS Cost and Usage Reports is a feature in the AWS Billing and Cost Management console. For details about security considerations, see Security in AWS Billing and Cost Management in the *AWS Billing User Guide*.

For more information on access control and IAM permissions to use AWS CUR, see Overview of Managing Access Permissions.

Quotas and restrictions

The following table describes the current quotas and restrictions within AWS Cost and Usage Reports.

Cost and Usage Reports

Number of Cost and Usage Reports	10 per account
Rates	Cost and Usage Reports are free of charge, but standard Amazon S3 rates apply.
Number of Free Tier AWS Cost and Usage Reports	10

Getting help with your exports and reports

There are many resources available for you to get help with your AWS Billing and Cost Management, AWS Data Exports, and AWS Cost and Usage Reports questions.

- AWS Knowledge Center: This is the quickest way to find answers to questions about your exports and reports. We recommend you start here.
- Account and billing support: If you're an AWS account owner, you have access to account and billing support free of charge. Only personalized technical support requires a support plan. For more information, visit Support.
- Open a support case: You can contact AWS Support and open a support case for your inquiry. This is the most direct method for communicating with AWS Support. Support does not publish a direct phone number to reach representatives, but instead will call you through the following procedure.

> **Note**
>
> To open an Support case and specify *Regarding: Account and Billing Support*, you must either be signed into AWS as the root account owner, or have IAM permissions to open a support case. For more information, see Getting started with Support in the *Support User Guide*.
> If you have closed your AWS account, you can still sign in to Support and view past bills.

To contact AWS Support

1. Sign in and navigate to Support Center.
2. Choose **Create case**.
3. On the **Create case** page, select **Account and billing** and fill in the required fields on the form.
4. Choose **Next step: Additional information**.
5. On the **Additional information** page, for **Subject**, enter a title about your issue.
6. For **Description**, describe your question or issue in detail.
7. (Optional) Choose **Attach files** to add any relevant files to your case, such as error logs or screenshots. You can attach up to three files. Each file can be up to 5 MB.

8. Choose **Next step: Solve now or contact us**.
9. On the **Contact us** page, choose your preferred language.
10. Choose your preferred contact method. You can choose one of the following options:

    - **Web**: Receive a reply in Support Center.
    - **Phone**: Receive a phone call from an Support representative.

    > **Note**
    >
    > Instant messaging support is not available for billing inquiries.

11. Review your case details and then choose **Submit**. Your case ID number and summary appear.

Document history

The following table describes the documentation for this release of AWS Data Exports.

Change	Description	Date
Support for Capacity Reservation in Data exports	Data exports now supports information about the capacity reservation that applies to a given line item.	November 13, 2025
Support for Kubernetes labels in split cost allocation data for Amazon EKS	Split cost allocation data now supports Kubernetes labels as cost allocation tags for Amazon EKS clusters.	October 27, 2025
Added GPU support in split cost allocation data for Amazon EKS	Split cost allocation data now includes GPU resource reservation data, enabling customers to track costs for GPU-accelerated workloads.	September 1, 2025
Added support for account names in cost optimization opportunities	You can use account names to easily view, filter, consolidate, and prioritize cost optimizat ion recommendations.	July 23, 2025
Added location-based emissions support	You can see your carbon emissions calculated using the location-based method (LBM) in Data Exports, alongside the existing market-based method (MBM).	June 24, 2025
Added exports of carbon emissions data	You can create exports of carbon emissions data in Data Exports.	April 23, 2025

Added exports using the FinOps foundation open source (FOCUS 1.0 with AWS columns) in GA	You can create cost and usage exports in Data Exports using the FinOps Open Cost and Usage Specification (FOCUS) in General Availability (GA).	November 25, 2024
Split cost allocation data and Amazon CloudWatch Container Insights	Split cost allocation data for Amazon EKS now supports metrics from Amazon CloudWatch Container Insights.	November 14, 2024
Added exports using the FinOps foundation open source (FOCUS 1.0 with AWS columns - *preview*)	You can create cost and usage exports in Data Exports using the FinOps Open Cost and Usage Specification (FOCUS).	June 20, 2024
Added exports of cost optimization recommendations (from Cost Optimization Hub)	You can create exports of cost optimization recommendations (from Cost Optimization Hub) in Data Exports.	June 20, 2024
Split cost allocation data and Amazon Managed Service for Prometheus	Split cost allocation data for Amazon EKS now supports metrics from Amazon Managed Service for Prometheus.	June 10, 2024
Split cost allocation data integration with Amazon EKS	Extended the split cost allocation data feature to integrate with Amazon EKS.	April 16, 2024

AWS Data Exports launched	AWS Data Exports enables you to create exports of the Cost and Usage Report (CUR) 2.0 using SQL for column selections and row filtering. This is the new and recommended way to receive your most detailed cost and usage data from AWS.	November 26, 2023