AWS Billing Conductor

User Guide

AWS Billing Conductor: User Guide

Table of Contents

What is AWS Billing Conductor?

AWS Billing Conductor is a fully managed service that can support the showback and chargeback workflows of AWS Solution Providers and Enterprise customers. Using AWS Billing Conductor, you can customize your monthly billing data. The console models the billing relationship between you and your customers or business units. You can also customize a pro forma version of your billing data each month to accurately show or charge back your customers. AWS Billing Conductor doesn't change the way that you're billed by Amazon Web Services each month. Instead, it provides you with a mechanism to configure, generate, and display rates to certain customers over a given billing period. You can also use it to analyze the difference between the rates you apply to your accounting groupings relative to your actual rates from AWS. As a result of your AWS Billing Conductor configuration, the payer account can also see the custom rate that's applied on the billing details page of the AWS Billing console, or configure a cost and usage report per billing group.

You can configure the billing groups and pricing plans using the AWS Billing Conductor or the AWS Billing Conductor API.

For more information about AWS Billing Conductor service quotas, see Quotas and restrictions (p. 38).

Topics

Features in AWS Billing Conductor

You can use the AWS Billing Conductor features to do the following:

- Group your accounts into a set of mutually exclusive set of accounts (billing groups). These groups are used to provide an aggregated view across a set of accounts. In addition, the bill of each billing group is computed as a stand-alone customer bill. This allows for the localized sharing of reserved instance and Savings Plans benefits, volume tiering discounts, and AWS Free Tier.
- Apply custom pricing plans: global and service-specific markups or discounts relative to On-Demand rates.
- Create and apply one-time charges or credits to your billing groups. These custom line items can be created as flat or percentage-based.
- Analyze pro forma costs based on your pricing configuration for each billing group in the AWS Billing console **billing details** page.
- Configure a pro forma cost and usage report for each billing group.
- Analyze month-to-date and historical differences between the rates that you apply to your billing groups, relative to your actual AWS rates using the **Billing group margin report**. For more information, see Analyzing your margins per billing group (p. 14).

Related services

AWS Billing console

The AWS Billing console is the portal for all AWS customers, from students and startup companies to large enterprises. You can use the console to see the resources that are running in your AWS accounts, manage billing preferences, and access billing artifacts that are needed to make payments

to AWS. The AWS Billing console also provides a high-level explanation of the spending for your account, and serves as the entry point for enrolling in products in the AWS Cost Management products.

For more information, see the *AWS Billing User Guide*.

AWS Cost and Usage Reports

The AWS Cost and Usage Reports (AWS CUR) contain 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 or day, 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 values (CSV) or Apache Parquet format. You can view the reports using spreadsheet software such as Microsoft Excel or Apache OpenOffice Calc. You can also access them from an application using the Amazon S3 or Amazon Athena APIs.

AWS Cost and Usage Reports track your AWS usage and provide 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.

AWS Identity and Access Management (IAM)

The AWS Billing Conductor service is integrated with AWS Identity and Access Management (IAM). You can use IAM with AWS Billing Conductor to ensure that other people who work in your account have only as much access as they need to get their job done.

You also use IAM to control access to all of your AWS resources. This includes but is not limited to your billing information. It's important that you familiarize yourself with the basic concepts and best practices of IAM before you get too far along with setting up the structure of your AWS account.

For more information about how to work with IAM, see What Is IAM? and Security Best Practices in IAM in the *IAM User Guide*.

AWS Organizations (Consolidated billing)

AWS products and services can accommodate every size of company, from small startups to enterprises. If your company is large or likely to grow, you might want to set up multiple AWS accounts that reflect your company's structure. For example, you can have one account for the entire company and accounts for each employee, or an account for the entire company with IAM users for each employee. You can have an account for the entire company, accounts for each department or team within the company, and accounts for each employee.

If you create multiple accounts, you can use the consolidated billing feature of AWS Organizations to combine all your member accounts under one management account and receive a single bill. For more information, see Consolidated billing for Organizations in the *AWS Billing User Guide*.

Understanding your AWS Billing Conductor dashboard

The AWS Billing Conductor dashboard provides a high-level summary of the key metrics to help you understand the impact of your custom pricing dimensions.

Key performance indicators

This section defines the key performance indicators (KPI) that are available on your AWS Billing Conductor dashboard. KPIs are all month-to-date. As you create or add accounts to your AWS Organizations, the accounts accrue to this KPI. When you delete a billing group, the accounts in that billing group also accrue to this KPI.

- **Charged amount** – The combined charges for usage that's accrued by all billing groups, based on the custom rate that's defined by the applied pricing plans. The calculation doesn't account for any commitment-based discounts that were purchased outside of the billing group, any non-public pricing, or any credit consumed in the billable domain. Examples of commitment-based discounts include reserved instances and Savings Plans.
- **AWS costs** – The combined month-to-date charge for usage that's accrued by all billing groups, according to the estimated charges on your AWS bill. The calculations include any commitment-based discounts purchased outside of the billing group if those benefits were applied in the billable domain, any non-public pricing, volume-tiered discounts, and credits. Examples of commitment-based discounts include reserved instances and Savings Plans.
- **Margin** – The aggregated month-to-date margin that's accrued by all billing groups. The margin is calculated by subtracting the AWS costs from the charged amount. Based on the factors such as the pricing plan and the applied custom line items, the margin can also be a negative.

  Note
  Post-billing period adjustments impact your historical margins. For more information, see Analyzing your margins per billing group (p. 14).
- **Billing groups** – The number of mutually exclusive groups of accounts, with a primary account and an associated pricing plan.
- **Accounts** – The number of accounts within a consolidated billing family.
- **Unmonitored accounts** – The number of accounts within a consolidated billing family that haven't been assigned to a billing group.

Other definitions for AWS Billing Conductor

This section defines other terms that are used throughout AWS Billing Conductor to help you use the service effectively.

- **Billable** – The billing output that's generated by AWS and used as the bias of calculating your AWS invoice.
- **Pro forma** – The output that's generated by AWS Billing Conductor. It's consistent with your desired changes in rate management (pricing configuration) and aggregated account visibility (billing groups).
- **Resource groups** – The inputs that are used to calculate percentage-based custom line items. Resource values include the accrued costs for the billing group and any flat custom line items that are associated with a given billing group for a billing period.

Viewing your top-five billing groups per charged amount

You can understand your top-five billing groups that generate revenue by referencing the visual and table view. To manage your existing billing groups, choose **Manage billing groups** on the dashboard page.

Creating billing groups, pricing configurations, and custom line items

This section shows how you can create billing groups, pricing configurations, and custom line items in AWS Billing Conductor. Each section also provides an overview of how you can use the billing group table, pricing rule table, and the custom line items table after you create each item.

Topics

Creating billing groups

You can use AWS Billing Conductor to create billing groups to organize your accounts. By default, payer accounts in with admin permissions can create billing groups. Each billing group is mutually exclusive. This means that an account can only belong to one billing group in a given billing period. Although you can see the billing group segmentation immediately, it takes up to 24 hours after creating a billing group to see the group's custom rates reflected.

> **Note**
> Moving accounts across billing groups in the middle of the month will initiate the recomputation of both billing groups back to the start of the billing period. Moving accounts mid-month doesn't affect previous billing periods.

Use the following steps to create a billing group.

To create a billing group

1. Sign in to the AWS Management Console and open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Billing groups**.
3. Choose **Create billing group**.
4. For **Billing group details**, enter the name of the billing group. For naming restrictions, see Quotas and restrictions (p. 38).
5. (Optional) For **Description**, enter a description for the billing group.
6. For **Pricing plan**, choose a pricing plan to associate with the billing group. To create a pricing plan, see Creating pricing plans (p. 7).
7. Under **Accounts**, choose one or more accounts to add to the billing group *or* choose **Import organizational unit** to automatically select the accounts that are within an organizational unit.

For a policy example to grant access to the import OU feature, see Granting AWS Billing Conductor access to the import organizational unit (OU) feature (p. 32).

You can use the table filter to sort by account names, account IDs, or the root email address that's associated with an account.

8. For **Primary account selection**, choose one account to be the primary account for a billing group.

   The primary account inherits the ability to see pro forma cost and usage across the billing group, and can generate a pro forma Cost and Usage Reports (AWS CUR) for the billing group.

   Note
   You must select your primary account in step 7. You can't change your primary account after the billing group is created. To assign a new primary account, delete the billing group and regroup your accounts. While a payer account can be included within a billing group, a payer account can't be assigned the role of the primary account.

9. Choose **Create billing group**.

Billing group table

After you create a billing group, you can view the details of the billing group in a filterable table. You can filter using the following dimensions:

- Billing group name
- Primary account name
- Primary account ID
- Number of accounts
- Pricing plan name

To view the details for each billing group, choose the billing group name in the table.

Creating pricing rules

You can create pricing rules in AWS Billing Conductor to customize your billing rates across your billing groups. Pricing rules can be global or service-specific in scope. By default, a payer account in with admin permissions can create pricing rules. It takes up to 24 hours after you apply a pricing rule to a billing group to see the custom rates for your billing group reflected.

A single pricing plan can be applied to multiple billing groups.

Note
Updating a pricing rule also updates all the pricing plans where the pricing rule is associated. A pricing plan is a collection of pricing rules. If the pricing plan is associated with a billing group or set of billing groups, this change affects only the current billing period. Previous billing periods remain the same.

Use the following steps to create a pricing rule.

To create a pricing rule

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Pricing configuration**.
3. Choose the **Pricing rules** tab.
4. Choose **Create pricing rules**.

5. For **Pricing rule details**, enter the name of the pricing rule. For naming restrictions, see Quotas and restrictions (p. 38).
6. (Optional) For **Description**, enter a description for the pricing rule.
7. For **Scope**, choose `Global` or `Service`.
   - Global - applies to all usage.
   - Service - only applies to a given service. When choosing service, choose a service code to configure the pricing rates for.
8. For **Type**, choose either **Discount** or **Markup**.
9. For **Percentage**, enter the percentage amount.

   If you enter **0** as the percentage, the pricing plan defaults to the AWS On-Demand rate.
10. (Optional) To create another pricing rule in the same workflow, choose **Add pricing rule**.
11. Choose **Create pricing rule**.

Pricing rule table

After you create a pricing rule, you can view the details of the pricing rule in a filterable table. You can filter by the following dimensions:

- Pricing rule name
- Scope
- Service code
- Rate

Creating pricing plans

You can create pricing plans in AWS Billing Conductor to customize the output of your billing details across your billing groups. By default, a payer account in with admin permissions can create pricing plans. It takes up to 24 hours after you apply a pricing plan to a billing group to see the custom rates for your billing group reflected.

A single pricing plan can be applied to multiple billing groups.

> **Note**
> Updating a pricing plan also affects the billing details of each billing group where the pricing plan is associated. If the pricing plan is associated with a billing group or set of billing groups, this change affects only the current billing period. Previous billing periods remain the same.

Use the following steps to create a pricing plan.

To create a pricing plan

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Pricing configuration**.
3. From the **Pricing plan** tab, choose **Create pricing plan**.
4. For **Pricing plan details**, enter the name of the pricing plan. For naming restrictions, see Quotas and restrictions (p. 38).
5. (Optional) For **Description**, enter a description for the pricing plan.
6. In the **Pricing rules table**, choose the pricing rules that you want to be associated with the pricing plan. You can filter the pricing rules by pricing rule name, scope, service code, or rate.

7. Choose **Create pricing plan**.

Pricing plan table

After you create a pricing plan, you can view the details of the pricing plan in a filterable table. You can filter by the following dimensions:

- The pricing plan name
- The description
- The number of pricing rules that's associated with the pricing plan

Creating custom line items per billing group

You can use AWS Billing Conductor to create custom line items and associate them to a billing group. Using custom line items, you can manually allocate costs and discounts at your discretion. Custom line items can be calculated using the **flat charge** or **percentage charge** values. You can configure the percentage-based custom line item to include or exclude resources. These resources might include billing group cost and other flat custom line items that are associated with a billing group for a given billing period.

Common use cases for custom line item creation include, but are not limited to the following:

- Allocating AWS Support fees
- Allocating shared service costs
- Applying managed service fees
- Applying tax
- Distributing credits
- Distributing RI and Savings Plans savings (as opposed to On-Demand)
- Adding organizational credits and discount line items

Creating a flat charge custom line item

Use the following steps to create a custom line item that applies either a credit or fee line item to an individual billing group.

To create a custom line item

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Custom line items**.
3. Choose **Create custom line item**.
4. For **Custom line item details**, enter the name of the custom line item. For naming restrictions, see Quotas and restrictions (p. 38).
5. For **Description**, enter a description for the custom line item. The character limit is 255.
6. For **Billing period**, choose either the existing billing period or the previous billing period.
7. For **Billing group**, choose a billing group. You can only associate the custom charge to one billing group at a time.
8. Choose **Flat charge** for your **charge type**.
9. Choose a **charge value** and enter an input amount.

A discount line item adds a credit. This reduces the amount that's charged to the selected billing group. A markup line item adds a charge. This increases the amount that's charged to the selected billing group. All custom line items are in USD.

10. Choose **Create**.

Creating a percentage charge custom line item

Use the following steps to create a custom line item that applies either a credit or fee line item to an individual billing group.

To create a custom line item

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Custom line items**.
3. Choose **Create custom line item**.
4. For **Custom line item details**, enter the name of the custom line item. For naming restrictions, see Quotas and restrictions (p. 38).
5. For **Description**, enter a description for the custom line item. The character limit is 255.
6. For **Billing period**, choose either the existing billing period or the previous billing period.
7. For **Billing group**, choose a billing group. You can only associate the custom charge to one billing group at a time.
8. Choose **percentage charge** for your **charge type**.
9. Choose a **charge value** and enter an input amount.

   A discount line item adds a credit. This reduces the amount that's charged to the selected billing group. A markup line item adds a charge. This increases the amount that's charged to the selected billing group. All custom line items are in USD.
10. (Optional) Choose the resource values that you want included in the calculation. By default, the `billing group cost` is selected as a resource. This excludes all flat custom line items.
11. (Optional) Include one of more flat custom line item. Choose each applicable flat custom line item from the table that you want included in the percentage-based calculation.

    Note
    Percentage custom line items can be created with no associated resources. These custom line items show as `$0.00` value in your billing data.
12. Choose **Create**.

Custom line items table

After you create a custom line item, you can view the details of the line item in a filterable table. You can filter by the following dimensions:

- The line item name
- The line item description
- The amount that's charged
- The billing group that the line item is attributed to
- The date that line item was created

To view custom line items that you created in previous billing periods, use the **date picker** dropdown list.

Editing custom line items

Use the following steps to edit your custom line items.

To edit a custom line item

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Custom line items**.
3. Choose **Create custom line item**.
4. Choose the custom line item that you want to edit.
5. Choose **Edit**.
6. Change the parameters that you want to edit.

   Note
   You can't change the billing period, billing group, charge type (flat or percentage), or charge value type (credit or fee).
7. Choose **Save changes**.

Deleting custom line items

Use the following steps to delete your custom line items.

To edit a custom line item

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Custom line items**.
3. Choose **Create custom line item**.
4. Choose the custom line item that you want to delete.
5. Choose **delete**.
6. Read how deleting the custom line item might affect you, and then choose **Delete custom line item**.

Best practices for AWS Billing Conductor

This section highlights some best practices for when you're working with AWS Billing Conductor.

Topics

Controlling access to AWS Billing Conductor

The Billing and Cost Management is only accessible to users who have access to the payer or management account. To grant IAM users permission to create billing groups and see the AWS Billing Conductor Key Performance Indicators (KPIs) in the Billing and Cost Management console, you must also grant IAM users the following:

- List accounts within Organizations

To learn more about giving users the ability to create billing groups and pricing plans in the AWS Billing Conductor console, see Identity and access management for AWS Billing Conductor (p. 20).

You can also create AWS Billing Conductor resources programmatically using the AWS Billing Conductor API. When you configure access to the AWS Billing Conductor API, we recommend creating a unique IAM user for allowing programmatic access. This helps you define more precise access controls between who in your organization has access to the AWS Billing Conductor console, and the API. To give multiple IAM users query access to the AWS Billing Conductor API, we recommend creating a programmatic access IAM role for each.

Understanding the AWS Billing Conductor data set

While the AWS Billing Conductor data models share many similarities with the standard AWS Billing data model, there are a few differences.

The AWS Billing Conductor does not include:

- Credits (redeemed at the payer or linked account level)
- Tax
- AWS Support charges

Additionally, the AWS Billing Conductor shares reserved instances and Savings Plans with the accounts placed within the same billing group, irrespective of your sharing preferences in the standard billing domain.

Understanding the AWS Billing Conductor computational logic

The AWS Billing Conductor computation is flexible to the changes that you make in a given month, while retaining the historical integrity of your prior period billing data. This is best described with an example.

In this example, we have two billing groups, `A` and `B`. Billing group `A` starts the billing period with accounts 1 through 3 in the group. At mid-month, the payer account moves `Account 3` to `Billing Group B`. At that point, the re-computation of the costs for Billing Groups `A` and `B` are required to accurately model the latest change. When `Account 3` is moved, `Billing Group A`'s usage is modeled as if `Account 3` was not a part of the billing group during the current billing period. Additionally, `Billing Group B`'s usage is modeled as if `Account 3` was a part of `Billing Group B` since the beginning of the billing period. This approach eliminates the need to calculate complex rates and chargeback models when accounts move across groups within the billing period.

Billing Group A	**Days: 1 - 15**	**Days: 16 - 30**	**End of Month**
Account 1	$ 100	$ 100	$ 200
Account 2	$ 100	$ 100	$ 200
Account 3	$ 100	N/A	N/A
Total	**$ 300**	**$ 200**	**$ 400**

Billing Group B	**Days: 1 - 15**	**Days: 16 - 30**	**End of Month**
Account 4	$ 100	$ 100	$ 200
Account 5	$ 100	$ 100	$ 200
Account 6	$ 100	$ 100	$ 200
Account 3	$ 100	$ 100	$ 200
Total	**$ 400**	**$ 400**	**$ 800**

Understanding the AWS Billing Conductor update frequency

AWS billing data is updated at least once a day. AWS Billing Conductor uses this data to compute your pro forma billing data. Custom line items that are generated to apply to the current month are reflected within 24 hours. Custom line items that are generated to apply to the prior billing period might take up to 48 hours to reflect in a billing group AWS Cost and Usage Reports, or on the bills page for a given billing group.

Understanding the differences between AWS 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 AWS 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.

Analyzing your margins per billing group

You can use the billing group margin report in AWS Billing Conductor to analyze your margins both in aggregate and with specific billing groups. All billing groups are included by default.

Use the following steps to view your margins for an individual billing group or a set of billing groups.

To view your billing group margins

1. Open AWS Billing Conductor at https://console.aws.amazon.com/billingconductor/.
2. In the navigation pane, choose **Billing group margin report**.
3. For **report type**, choose **Select billing group**.
4. Choose a single or multiple billing groups by name to see the charged amount and AWS costs for each.

The margin analysis is shown as a bar chart graph and in a table.

Negative margins are shown in red in the graph, with a negative dollar amount and negative percentage.

Billing group margin analysis table

The billing group margin analysis table is sorted in reverse chronological order by default. You can sort the table by all of the columns, which include the following:

- Month
- Charged amount
- AWS costs
- Margin amount
- Margin percentage

The graph and table returns values for the last 13 months of the billing groups selected. If the billing groups were created at different times, we assume the time range of the oldest selected billing group.

Viewing your billing group details

You can use your billing group details to monitor, analyze, and edit your billing group in AWS Billing Conductor. The billing group details provide a month-to-date margin analysis, a history of custom line items applied, and the ability to edit and delete the billing group as needed.

Viewing your billing details by custom pricing dimensions

After you create and assign your billing groups and pricing plans, you can view your custom billing dimensions with usage type granularity for each billing group under management.

Use the following steps to view your billing details in the pro forma domain.

To view your pro forma billing details

1. Open the AWS Billing console at https://console.aws.amazon.com/billing/.
2. In the navigation pane, choose **Bills**.
3. Choose **Settings** in the top-right corner of **billing details**.
4. Enable the **Pro forma data view**.
5. For **Billing group**, choose the billing to analyze.

You can analyze the billing group usage by service and AWS Region to see the cost of that usage, consistent with the rates defined in AWS Billing Conductor.

You can find the custom line items under the service **AWS Billing Conductor** on the **Billing details** page.

Configuring Cost and Usage Reports per billing group

You can create pro forma AWS Cost and Usage Reports (AWS CUR) for each billing group that you create. The pro forma AWS CUR has the same file format, granularity, and columns as the standard AWS CUR, and contains the most comprehensive set of cost and usage data available for a given period of time.

You can publish your pro forma AWS CUR to an Amazon Simple Storage Service (Amazon S3) bucket that you own.

AWS updates the report in your bucket once a day in comma-separated values (CSV) or Apache Parquet format. You can view the reports using spreadsheet software such as Microsoft Excel and Apache OpenOffice Calc. You can also access them from an application using the Amazon S3 or Amazon Athena APIs. For more information about the standard AWS CUR, see the AWS Cost and Usage Reports User Guide.

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 AWS Billing console at https://console.aws.amazon.com/billing/.
2. On the navigation pane, choose **Cost & Usage Reports**.
3. In the top right of the **report table**, choose **Settings**.
4. Enable 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. For **Billing group**, choose a billing group.
10. For **Additional report details**, select **Include resource IDs** to include the IDs of each individual resource in the report.
11. 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.
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 dropdown list, and then choose **Next**.
    - Enter a bucket name and the AWS Region where you want to create a new bucket, and choose **Next**.
15. 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 Amazon QuickSight, 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 4 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.
19. For **Enable report data integration for**, choose whether you want to upload your Cost and Usage Reports to Amazon Athena, Amazon Redshift, or Amazon QuickSight. The report is compressed in the following formats:
    - **Athena**: parquet compression
    - **Amazon Redshift or Amazon QuickSight**: .gz compression
20. Choose **Next**.
21. After reviewing the settings for your report, choose **Review and Complete**.

Using the AWS Billing Conductor API

The API is readily available in Java, Python, .NET, and Go. New capabilities released in the AWS Billing Conductor will also be available as an API.

For more information about the AWS Billing Conductor API, see the AWS Billing Conductor API reference.

Security in AWS Billing Conductor

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 Billing Conductor, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Billing Conductor. The following topics show you how to configure AWS Billing Conductor to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Billing Conductor resources.

Topics

Data protection in AWS Billing Conductor

The AWS shared responsibility model applies to data protection in AWS Billing Conductor. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.

- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with AWS Billing Conductor or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Identity and access management for AWS Billing Conductor

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Billing Conductor resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in AWS Billing Conductor.

Service user – If you use the AWS Billing Conductor service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS Billing Conductor features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in AWS Billing Conductor, see Troubleshooting AWS Billing Conductor identity and access (p. 34).

Service administrator – If you're in charge of AWS Billing Conductor resources at your company, you probably have full access to AWS Billing Conductor. It's your job to determine which AWS Billing Conductor features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with AWS Billing Conductor, see How AWS Billing Conductor works with IAM (p. 24).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS Billing Conductor. To view example AWS Billing Conductor identity-based policies that you can use in IAM, see AWS Billing Conductor identity-based policy examples (p. 28).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *AWS General Reference*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API

operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for AWS Billing Conductor in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored

in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role).

You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Billing Conductor works with IAM

Before you use IAM to manage access to AWS Billing Conductor, you should understand what IAM features are available to use with AWS Billing Conductor. To get a high-level view of how AWS Billing Conductor and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

Topics

AWS Billing Conductor identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. AWS Billing Conductor supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some

exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in AWS Billing Conductor use the following prefix before the action: `AWS Billing Conductor:`. For example, to grant someone permission to run an Amazon EC2 instance with the Amazon EC2 `RunInstances` API operation, you include the `ec2:RunInstances` action in their policy. Policy statements must include either an `Action` or `NotAction` element. AWS Billing Conductor defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "ec2:action1",
      "ec2:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "ec2:Describe*"
```

To see a list of AWS Billing Conductor actions, see Actions Defined by AWS Billing Conductor in the *IAM User Guide*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

The Amazon EC2 instance resource has the following ARN:

```
arn:${Partition}:ec2:${Region}:${Account}:instance/${InstanceId}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

For example, to specify the `i-1234567890abcdef0` instance in your statement, use the following ARN:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

To specify all instances that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

Some AWS Billing Conductor actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permissions to use the volume and the instance. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "resource1",
      "resource2"
```

To see a list of AWS Billing Conductor resource types and their ARNs, see Resources Defined by AWS Billing Conductor in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Billing Conductor.

Condition keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

AWS Billing Conductor defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see Example: Restricting Access to a Specific Region.

To see a list of AWS Billing Conductor condition keys, see Condition Keys for AWS Billing Conductor in the *IAM User Guide*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Billing Conductor.

Examples

To view examples of AWS Billing Conductor identity-based policies, see AWS Billing Conductor identity-based policy examples (p. 28).

AWS Billing Conductor resource-based policies

Resource-based policies are JSON policy documents that specify what actions a specified principal can perform on the AWS Billing Conductor resource and under what conditions. Amazon S3 supports resource-based permissions policies for Amazon S3 *buckets*. Resource-based policies let you grant usage permission to other accounts on a per-resource basis. You can also use a resource-based policy to allow an AWS service to access your Amazon S3 *buckets*.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, you must also grant the principal entity permission to access the resource. Grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM Roles Differ from Resource-based Policies in the *IAM User Guide*.

The Amazon S3 service supports only one type of resource-based policy called a *bucket policy*, which is attached to a *bucket*. This policy defines which principal entities (accounts, users, roles, and federated users) can perform actions on the *AWS Billing Conductor*.

Examples

To view examples of AWS Billing Conductor resource-based policies, see AWS Billing Conductor resource-based policy examples (p. 33),

Access control lists (ACLs)

Access control lists (ACLs) are lists of grantees that you can attach to resources. They grant accounts permissions to access the resource to which they are attached. You can attach ACLs to an Amazon S3 *bucket* resource.

With Amazon S3 access control lists (ACLs), you can manage access to *bucket* resources. Each *bucket* has an ACL attached to it as a subresource. It defines which AWS accounts, IAM users or groups of users, or IAM roles are granted access and the type of access. When a request is received for a resource, AWS checks the corresponding ACL to verify that the requester has the necessary access permissions.

When you create a *bucket* resource, Amazon S3 creates a default ACL that grants the resource owner full control over the resource. In the following example *bucket* ACL, John Doe is listed as the owner of the *bucket* and is granted full control over that *bucket*. An ACL can have up to 100 grantees.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://AWS Billing Conductor.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6</ID>
    <DisplayName>john-doe</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6</ID>
        <DisplayName>john-doe</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
```

```
  </AccessControlList>
</AccessControlPolicy>
```

The ID field in the ACL is the AWS account canonical user ID. To learn how to view this ID in an account that you own, see Finding an AWS Account Canonical User ID.

Authorization based on AWS Billing Conductor tags

You can attach tags to AWS Billing Conductor resources or pass tags in a request to AWS Billing Conductor. To control access based on tags, you provide tag information in the condition element of a policy using the `AWS Billing Conductor:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

AWS Billing Conductor IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with AWS Billing Conductor

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

AWS Billing Conductor supports using temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

AWS Billing Conductor supports service roles.

Choosing an IAM role in AWS Billing Conductor

When you create a resource in AWS Billing Conductor, you must choose a role to allow AWS Billing Conductor to access Amazon EC2 on your behalf. If you have previously created a service role or service-linked role, then AWS Billing Conductor provides you with a list of roles to choose from. It's important to choose a role that allows access to start and stop Amazon EC2 instances.

AWS Billing Conductor identity-based policy examples

By default, IAM users and roles don't have permission to create or modify AWS Billing Conductor resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Billing Conductor resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or root users in your account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

AWS Billing Conductor identity-based policy examples

This topic contains example policies that you can attach to your IAM user or group to control access to your account's information and tools.

Topics

- Granting AWS Billing Conductor access to the import organizational unit (OU) feature (p. 32)

Granting full access to the AWS Billing Conductor console

To access the AWS Billing Conductor console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Billing Conductor resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the AWS Billing Conductor console, also attach the following AWS managed policy to the entities. For more information, see Adding Permissions to a User in the *IAM User Guide*:

In addition to the `billingconductor:*` permissions, `pricing:DescribeServices` is required for pricing rule creation, and `organizations:ListAccounts` is required to list linked accounts that are linked to the payer account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "billingconductor:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "organizations:ListAccounts",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "pricing:DescribeServices",
            "Resource": "*"
        }
    ]
}
```

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

Granting full access to the AWS Billing Conductor API

In this example, you grant an IAM user full only access to the AWS Billing Conductor API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "billingconductor:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "organizations:ListAccounts",
            "Resource": "*"
        }
    ]
```

```
}
```

Granting read only access to the AWS Billing Conductor console

In this example, you grant an IAM user read only access to the AWS Billing Conductor console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "billingconductor:List*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "organizations:ListAccounts",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "pricing:DescribeServices",
            "Resource": "*"
        }
    ]
}
```

Granting AWS Billing Conductor access through the Billing console

In this example, IAM users can toggle and view pro forma billing data through the bills page in their Billing console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "billing:ListBillingViews",
                "aws-portal:ViewBilling"
            ]
            "Resource": "*"
        }
    ]
}
```

Granting AWS Billing Conductor access through AWS Cost and Usage Reports

In this example, IAM users can toggle and view pro forma billing data through the Cost and Usage Reports page in their Billing console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "billing:ListBillingViews",
                "aws-portal:ViewBilling",
                "cur:DescribeReportDefinitions"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Granting AWS Billing Conductor access to the import organizational unit (OU) feature

In this example, IAM users have read-only access to the specific AWS Organizations APIs required to import your organizational unit accounts when you're creating a billing group. The import OU feature is on the AWS Billing Conductor console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListRoots",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListChildren"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policies for AWS Billing Conductor

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AWSBillingConductorFullAccess

The AWSBillingConductorFullAccess managed policy grants complete access to AWS Billing Conductor console and APIs. Users can list, create, and delete AWS Billing Conductor resources.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "billingconductor:*",
               "organizations:ListAccounts",
               "pricing:DescribeServices",
             ]
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: AWSBillingConductorReadOnlyAccess

The AWSBillingConductorReadOnlyAccess managed policy grants read-only access to AWS Billing Conductor console and APIs. Users can view and list all AWS Billing Conductor resources. Users can't create or delete resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "billingconductor:List*",
               "organizations:ListAccounts",
               "pricing:DescribeServices"
             ]
            "Resource": "*"
        }
    ]
}
```

AWS Billing Conductor updates to AWS managed policies

View details about updates to AWS managed policies for AWS Billing Conductor since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS Billing Conductor Document history page.

Change	Description	Date
AWSBillingConductorFullAccess	Created policy	March 29, 2022
AWSBillingConductorReadOnlyAccess	Created policy	March 29, 2022
AWS Billing Conductor change log published	AWS Billing Conductor started tracking changes for its AWS managed policies.	March 29, 2022

AWS Billing Conductor resource-based policy examples

Topics

Restricting Amazon S3 bucket access to specific IP addresses

The following example grants permissions to any user to perform any Amazon S3 operations on objects in the specified bucket. However, the request must originate from the range of IP addresses specified in the condition.

The condition in this statement identifies the 54.240.143.* range of allowed Internet Protocol version 4 (IPv4) IP addresses, with one exception: 54.240.143.188.

The `Condition` block uses the `IpAddress` and `NotIpAddress` conditions and the `aws:SourceIp` condition key, which is an AWS-wide condition key. For more information about these condition keys, see Specifying Conditions in a Policy. The`aws:sourceIp` IPv4 values use the standard CIDR notation. For more information, see IP Address Condition Operators in the *IAM User Guide*.

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::examplebucket/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
         "NotIpAddress": {"aws:SourceIp": "54.240.143.188/32"}
      }
    }
  ]
}
```

Troubleshooting AWS Billing Conductor identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Billing Conductor and IAM.

Topics

I am not authorized to perform an action in AWS Billing Conductor

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a *`AWS Billing Conductor`* but does not have `AWS Billing Conductor:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: AWS Billing
 Conductor:GetWidget on resource: my-example-AWS Billing Conductor
```

In this case, Mateo asks his administrator to update his policies to allow him to access the *`my-example-AWS Billing Conductor`* resource using the `AWS Billing Conductor:`*`GetWidget`* action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS Billing Conductor.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS Billing Conductor. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I'm an administrator and want to allow others to access AWS Billing Conductor

To allow others to access AWS Billing Conductor, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in AWS Billing Conductor.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my AWS Billing Conductor resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Billing Conductor supports these features, see How AWS Billing Conductor works with IAM (p. 24).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Logging and monitoring in AWS Billing Conductor

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS account. There are several tools available to monitor your AWS Billing Conductor usage.

AWS Cost and Usage Reports

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 or by the day.

For more information about AWS Cost and Usage Reports, see the *Cost and Usage Report Guide*.

Compliance validation for AWS Billing Conductor

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs. AWS Billing Conductor is not in scope of any AWS compliance programs.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS Billing Conductor is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Billing Conductor

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Billing Conductor

As a managed service, AWS Billing Conductor is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS Billing Conductor through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Quotas and restrictions

The following table describes quotas and restrictions within AWS Billing Conductor.

Quotas

Number of accounts per billing group	1,000
Number of pricing plans	5,000
Number of pricing rules	50,000
Number of pricing rules that can associate to a pricing plan	500
Number of pricing plans that can associate with a pricing rule	1,000
Number of custom line items	50,000

Restrictions

Other restrictions in the following table cannot be increased.

Number of billing group Cost and Usage Reports per billing group	10
Billing group name	• Must be within 128 characters • Cannot contain a `space` • Cannot contain special characters
Billing group description	Must be within 1,024 characters
Pricing plan name	• Must be within 128 characters • Cannot contain a `space` • Cannot contain special characters
Pricing plan description	Must be within 1,024 characters
Custom line item name	• Must be within 128 characters • Cannot contain a `space` • Cannot contain special characters

Document history for user guide

The following table describes the documentation for this release of AWS Billing Conductor.

Change	Description	Date
Initial release (p. 39)	Initial release of AWS Billing Conductor User Guide and API Reference.	March 16, 2022

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.