Application Cost Profiler

User Guide

Application Cost Profiler: User Guide

Table of Contents

What is AWS Application Cost Profiler?

AWS Application Cost Profiler helps you separate your AWS billing and costs by the tenants of your service. A *tenant* can be a user, a group of users, or a project.

A *resource* is an entity that users can work with in AWS, such as an Amazon Elastic Compute Cloud (Amazon EC2) instance. Make sure that you can identify your resource usage by the tenant you choose.

Typical AWS resource usage includes shared services that support multiple tenants within your organization. Certain resources use time-based dimensions. To get cost and billing information by tenant rather than by hourly usage for the resource, you can integrate your resources with Application Cost Profiler. With this granular approach, you can understand how AWS resources are consumed across a shared software solution.

The following resources that can use either time-based dimensions or hourly usage are enabled for Application Cost Profiler:

- Amazon EC2 instances (on demand and spot instances only)
- Amazon Simple Queue Service (Amazon SQS) queues
- Amazon Simple Notification Service (Amazon SNS) topics
- Amazon DynamoDB reads and writes

> **Note**
> Amazon SQS, Amazon SNS, and DynamoDB usage is not charged by time, unlike most resources. In their case, the usage during an hour (for example, a number of reads and writes in DynamoDB), is categorized by the percentage of the hour that you allocate to different tenants, regardless of when the reads or writes happened during the hour.

You integrate your services with Application Cost Profiler in three steps:

1. **Enable and configure a report** – This step defines what you want your final output to look like.
2. **Send tenant usage data to Application Cost Profiler** – This step requires code in your service to create usage data that associates tenants with the time they use your resources, and then send that usage data to Application Cost Profiler.
3. **Get reports** – Application Cost Profiler provides reports at the cadence that you specified in your report configuration. The reports show the cost associated with each tenant's usage, giving you a granular view of your billing.

For more information about these steps, see *Getting started* (p. 2).

Getting started with Application Cost Profiler

AWS Application Cost Profiler helps you get cost information about your AWS resources by reporting resource usage by tenant, rather than for the resource as a whole. A *tenant* can be a user, a group of users, or a project. Make sure that you can identify your resource usage by the tenant you choose. To get cost reports about tenant usage, you configure a report and send usage data to Application Cost Profiler. This section discusses the prerequisites that you must complete before you use Application Cost Profiler.

Topics

Get an AWS account and your root user credentials

To access AWS, you must sign up for an AWS account.

To sign up for an AWS account

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

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Creating an IAM user

If your account already includes an AWS Identity and Access Management (IAM) user with full AWS administrative permissions, you can skip this section.

> **Note**
> For more information about using IAM with Application Cost Profiler, see Identity and access management for AWS Application Cost Profiler (p. 20).

When you first create an Amazon Web Services (AWS) account, you begin with a single sign-in identity. That identity has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user*. When you sign in, enter the email address and password that you used to create the account.

Important
We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. To view the tasks that require you to sign in as the root user, see Tasks that require root user credentials.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add users**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note
    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

Signing in as an IAM user

Sign in to the IAM console by choosing **IAM user** and entering your AWS account ID or account alias. On the next page, enter your IAM user name and your password.

Note
For your convenience, the AWS sign-in page uses a browser cookie to remember your IAM user name and account information. If you previously signed in as a different user, choose the sign-in link beneath the button to return to the main sign-in page. From there, you can enter your AWS account ID or account alias to be redirected to the IAM user sign-in page for your account.

Creating IAM user access keys

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them from the AWS Management Console. As a best practice, do not use the AWS account root user access keys for any task where it's not required. Instead, create a new administrator IAM user with access keys for yourself.

The only time that you can view or download the secret access key is when you create the keys. You cannot recover them later. However, you can create new access keys at any time. You must also have permissions to perform the required IAM actions. For more information, see Permissions required to access IAM resources in the *IAM User Guide*.

To create access keys for an IAM user

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys you want to create, and then choose the **Security credentials** tab.
4. In the **Access keys** section, choose **Create access key**.
5. To view the new access key pair, choose **Show**. You will not have access to the secret access key again after this dialog box closes. Your credentials will look something like this:
   - Access key ID: AKIAIOSFODNN7EXAMPLE
   - Secret access key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. To download the key pair, choose **Download .csv file**. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes.

   Keep the keys confidential in order to protect your AWS account and never email them. Do not share them outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.
7. After you download the `.csv` file, choose **Close**. When you create an access key, the key pair is active by default, and you can use the pair right away.

Related topics

- What is IAM? in the *IAM User Guide*
- AWS security credentials in *AWS General Reference*

Application Cost Profiler specific prerequisites

Before you get started with Application Cost Profiler, you must complete the following prerequisites:

- **Enable Cost Explorer**

Enable AWS Cost Explorer for your AWS account. Setting up an account with Cost Explorer can take up to 24 hours. You must complete Cost Explorer setup before Application Cost Profiler can generate your daily and monthly reports.

For more information, see Enabling Cost Explorer in the *AWS Billing and Cost Management User Guide*.

- **Create S3 buckets**

  Create at least two Amazon Simple Storage Service (Amazon S3) buckets. Application Cost Profiler uses one S3 bucket to provide reports to you. You use the other S3 bucket to upload usage data to Application Cost Profiler. Typically, you only need one S3 bucket to upload usage data. However, you might want to have more than one S3 bucket so that you can keep usage for different services in separate S3 buckets with different permissions, if needed for your security. You must give Application Cost Profiler permissions to these S3 buckets.

  For more information about setting up the Amazon S3 buckets for Application Cost Profiler, see Setting up Amazon S3 buckets for Application Cost Profiler (p. 5).

- **Enable tags**

  To report usage by tag, rather than by resource, you must enable those tags in the AWS Billing and Cost Management console.

  For more information about activating AWS generated tags, see Activating the AWS-Generated Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*. For more information about activating user-defined tags, see Activating User-Defined Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

Next steps

After you complete these prerequisites, you can:

- Configure your report and send usage data to Application Cost Profiler. For more information, see Creating your report (p. 10).
- Get and analyze your generated reports. For more information, see Using Application Cost Profiler reports (p. 15).

Setting up Amazon S3 buckets for Application Cost Profiler

To send usage data to and receive reports from AWS Application Cost Profiler, you must have at least one Amazon Simple Storage Service (Amazon S3) bucket in your AWS account to store data and one S3 bucket to receive your reports.

> **Note**
> For users of AWS Organizations, the Amazon S3 buckets can be either in the management account or in individual member accounts. The data in S3 buckets owned by the management account can be used to generate reports for the entire organization. In individual member accounts, the data in the S3 buckets can only be used to generate reports for that member account.

The S3 buckets you create are owned by the AWS account that you create them in. The S3 buckets are billed at standard Amazon S3 rates. For more information about how to create an Amazon S3 bucket, see Creating a bucket in the *Amazon Simple Storage Service User Guide*.

In order for Application Cost Profiler to use the S3 buckets, you must attach a policy to the buckets that gives Application Cost Profiler permissions to read and/or write to the bucket. If you modify the policy after your reports are set up, you may prevent Application Cost Profiler from being able to read your usage data or deliver your reports.

The following topics show how to set up permissions on your Amazon S3 buckets after you have created them. In addition to the ability to read and write objects, if you encrypted the buckets, Application Cost Profiler must have access to the AWS Key Management Service (AWS KMS) key for each bucket.

Topics

Giving Application Cost Profiler access to your report delivery S3 bucket

The S3 bucket that you configure for Application Cost Profiler to deliver your reports to must have a policy attached that allows Application Cost Profiler to create the report objects. In addition, the S3 bucket must be configured to enable encryption.

> **Note**
> When you create your bucket, you must choose to encrypt it. You may choose to encrypt your bucket with Amazon S3-managed keys (SSE-S3) or with your own key managed by AWS KMS (SSE-KMS). If you have already created your bucket with no encryption, you must edit your bucket to add encryption.

To give Application Cost Profiler access to your report delivery S3 bucket

1. Go to the Amazon S3 console and sign in.
2. Select **Buckets** from the left navigation, and then choose your bucket from the list.
3. Choose the **Permissions** tab, then, next to **Bucket policy**, choose **Edit**.
4. In the **Policy** section, insert the following policy. Replace *<bucket_name>* with the name of your bucket, and *<AWS account>* with the ID of your AWS account.

```
{
      "Version":"2008-10-17",
      "Statement":[
        {
          "Effect":"Allow",
          "Principal":{
            "Service":"application-cost-profiler.amazonaws.com"
          },
          "Action":[
            "s3:PutObject*",
            "s3:GetEncryptionConfiguration"
          ],
          "Resource": [
            "arn:aws:s3:::<bucket-name>",
            "arn:aws:s3:::<bucket-name>/*"
          ],
          "Condition": {
            "StringEquals": {
              "aws:SourceAccount": "<AWS account>"
            },
            "ArnEquals": {
```

```
            "aws:SourceArn": "arn:aws:application-cost-profiler:us-east-1:<AWS
account>:*"
          }
        }
      }
    ]
  }
```

In this policy you are giving the Application Cost Profiler service principal (`application-cost-profiler.amazonaws.com`) access to deliver reports to the specified bucket. It does this on your behalf, and includes a header with your AWS account and an ARN specific to your report delivery bucket. To ensure that Application Cost Profiler is accessing your bucket only when acting on your behalf, the `Condition` checks for those headers.

5. Choose **Save changes** to save your policy, attached to your bucket.

   If you have created your bucket using SSE-S3 encryption, then you are done. If you used SSE-KMS encryption, then the following steps are necessary to give Application Cost Profiler access to your bucket.

6. (Optional) Choose the **Properties** tab for your bucket, and under **Default Encryption**, select the Amazon Resource Name (ARN) for your AWS KMS key. This action displays the AWS Key Management Service console and shows your key.
7. (Optional) Add the policy to give Application Cost Profiler access to the AWS KMS key. For instructions on adding this policy, see Giving Application Cost Profiler access to SSE-KMS encrypted S3 buckets (p. 8).

Giving Application Cost Profiler access to your usage data S3 bucket

The S3 bucket that you configure for Application Cost Profiler to read your usage data from must have a policy attached to allow Application Cost Profiler to read the usage data objects.

Note
By giving Application Cost Profiler access to your usage data, you agree that we may temporarily copy such usage data objects to the US East (N. Virginia) AWS Region while processing reports. These data objects will be kept in the US East (N. Virginia) Region until the monthly report generation is complete.

To give Application Cost Profiler access to your usage data S3 bucket

1. Go to the Amazon S3 console and sign in.
2. Select **Buckets** from the left navigation, and then choose your bucket from the list.
3. Choose the **Permissions** tab, then, next to **Bucket policy**, choose **Edit**.
4. In the **Policy** section, insert the following policy. Replace *<bucket-name>* with the name of your bucket, and *<AWS account>* with the ID of your AWS account.

```
{
      "Version":"2008-10-17",
      "Statement":[
        {
          "Effect":"Allow",
          "Principal":{
            "Service":"application-cost-profiler.amazonaws.com"
          },
          "Action":[
            "s3:GetObject*"
          ],
```

```
          "Resource": [
            "arn:aws:s3:::<bucket-name>",
            "arn:aws:s3:::<bucket-name>/*"
          ],
          "Condition": {
            "StringEquals": {
              "aws:SourceAccount": "<AWS account>"
            },
            "ArnEquals": {
              "aws:SourceArn": "arn:aws:application-cost-profiler:us-east-1:<AWS
account>:*"
            }
          }
        }
      ]
    }
```

In this policy you are giving the Application Cost Profiler service principal (`application-cost-profiler.amazonaws.com`) access to get data out of the specified bucket. It does this on your behalf, and includes a header with your AWS account and an ARN specific to your usage bucket. To ensure that Application Cost Profiler is accessing your bucket only when acting on your behalf, the `Condition` checks for those headers.

5. Choose **Save changes** to save your policy, attached to your bucket.

If your bucket is encrypted with AWS KMS managed keys, then you must give Application Cost Profiler access to your bucket by following the procedure in the next section.

Giving Application Cost Profiler access to SSE-KMS encrypted S3 buckets

If you encrypt the S3 buckets that you configure for Application Cost Profiler (required for report buckets) with keys stored in AWS KMS (SSE-KMS), you must also give permissions to Application Cost Profiler to decrypt them. You do this by giving access to the AWS KMS keys used to encrypt the data.

Note
If your bucket is encrypted with Amazon S3 managed keys, then you do not need to complete this procedure.

To give Application Cost Profiler access to AWS KMS for SSE-KMS encrypted S3 buckets

1. Go to the AWS KMS console and sign in.
2. Select **Customer managed keys** from the left navigation, and then choose the key that is used to encrypt your bucket from the list.
3. Select **Switch to policy view**, then choose **Edit**.
4. In the **Policy** section, insert the following policy statement.

```
{
  "Effect": "Allow",
  "Principal": {
      "Service": "application-cost-profiler.amazonaws.com"
  },
  "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey*"
  ],
  "Resource": "*",
        "Condition": {
          "StringEquals": {
```

```
            "aws:SourceAccount": "<AWS account>"
          },
          "ArnEquals": {
            "aws:SourceArn": "arn:aws:application-cost-profiler:us-east-1:<AWS
account>:*"
          }
 }
```

5. Choose **Save changes** to save your policy, attached to your key.
6. Repeat for each key that encrypts an S3 bucket that Application Cost Profiler needs to access.

Note
The data is copied out of your S3 bucket on import into Application Cost Profiler managed buckets (that are encrypted). If you revoke the access to the keys, Application Cost Profiler can't retrieve any new objects from the bucket. However, any data already imported can still be used to generate reports.

Creating your report

After fulfilling the prerequisites, you're ready to configure the report for your AWS account and send your usage data to AWS Application Cost Profiler. This section describes how to configure the report and how to send the usage data to Application Cost Profiler.

Configure your Application Cost Profiler report

The following procedure shows how to configure the report that you want to generate based on your usage date. You configure details such as the frequency that the report is generated.

> **Note**
> If your AWS account is part of an AWS organization, you can configure the report using either the management account or an individual member account. Reports configured for individual accounts only contain data for that account. Reports configured using the management account can include data for the entire organization.
> The Amazon S3 bucket used for report output must belong to the account creating the report configuration.

To configure your Application Cost Profiler report

1. Open a web browser and sign in to the Application Cost Profiler console.
2. Choose **Get started now** to configure or modify a report.
3. Enter a **Report Name** and **Report Description** for your report.
4. Enter the name of your S3 bucket in the **Enter S3 bucket name** field and enter the S3 prefix in the **Enter S3 prefix** field. For more information about creating S3 buckets and giving Application Cost Profiler permissions, see Setting up Amazon S3 buckets for Application Cost Profiler (p. 5).
5. Select the options that you want your report to have:
   - **Time Frequency** – Choose whether the report is generated on a **Daily** or **Monthly** cadence, or **Both**.
   - **Report Output Format** – Choose the type of file to create within your Amazon S3 bucket. If you choose **CSV**, Application Cost Profiler creates a comma-separated values text file with gzip compression for the reports. If you choose **Parquet**, a Parquet file is generated for the reports.
6. Choose **Configure** to save your report configuration.

> **Note**
> You can also use the AWS Application Cost Profiler API to configure reports.

Verify the report settings by choosing **Get started now** to view the current report configuration.

> **Note**
> You can only have a single report configured. Returning to the configuration page will edit your existing report.

After you have configured your report, data ingestion is enabled. You can integrate your services with Application Cost Profiler to provide usage data for your resources.

Reporting tenant usage data from your services

After you have configured the report, you are ready to send tenant usage data from the resources or services in your account. You must inform Application Cost Profiler when your resource is being used for

a specific tenant. For example, if your service accepts API calls from different tenants, you record a start and end time for each tenant as you start and end an API call from that tenant. Application Cost Profiler uses that data to generate reports about the cost of your service, by the amount of time spent on work for each tenant.

To give Application Cost Profiler the usage data, you do the following:

- **Prepare resource usage data** – Create tables that describe when a resource is used for a specific tenant.
- **Upload usage data** – Upload the tables to an Amazon S3 bucket that you have given Application Cost Profiler permission to access.
- **Import usage data** – Call the `ImportApplicationUsage` API operation to let Application Cost Profiler know the data is ready to be processed.

The following sections describe each of these steps in more detail.

Topics

Step 1: Preparing your resource usage data

As a resource is being used in your service, you track which tenant is using it. Record this data into a table that you can later upload for Application Cost Profiler to import. Each row in the table describes a resource , the tenant that is using the resource, and the start and end times of that usage. An example of a resource is an Amazon Elastic Compute Cloud (Amazon EC2) instance that is being used.

This step requires that you integrate code into your service to output the correct information about the usage.

The fields that are in a resource usage table are listed in the following table.

Field	Description
ApplicationId	Identifies the application or product in your system that is being used. Defines the scope of the tenant metadata.
TenantId	An identifier in your system for the tenant who is consuming the specified resource. Application Cost Profiler aggregates to this level within the **ApplicationId**.
TenantDesc	(Optional) Additional data about the tenant for your own additional reporting.
UsageAccountId	The account that the resource runs in (important for accounts that are part of an organization).
StartTime	Timestamp (in milliseconds and microseconds) from Epoch, in UTC. Indicates the start time of the period for the usage by the specified tenant.

Field	Description
EndTime	Timestamp (in milliseconds and microseconds) from Epoch, in UTC. Indicates the end time of the period for the usage by the specified tenant.
ResourceId	Amazon Resource Name (ARN) for resource being used.
Name	(Optional) As an alternative to specifying a **ResourceId**, you can specify a **Name** resource tag to attribute costs to a set of resources (the field must include the value you want to use for the **Name** tag). Resource tags are enabled as part of your Cost and Usage Report. For more information about resource tags, see Resource tags details in the *Cost and Usage Report User Guide*.

The output must be in a comma-separated values (.csv) file that includes a heading row, as shown in the following example.

```
ApplicationId,TenantId,TenantDesc,UsageAccountId,StartTime,EndTime,ResourceId
MyApp,Tenant1,,123456789012,1613681437032.9001,1613681437041.5312,arn:aws:ec2:us-
east-1:123456789012:instance/1234-abcd-example-1234
MyApp,Tenant2,,123456789012,1613681245531.4426,1613681245551.1323,arn:aws:ec2:us-
east-1:123456789012:instance/1234-abcd-example-1234
MyApp,Tenant1,,123456789012,1613681904815.3381,1613681904930.0972,arn:aws:ec2:us-
east-1:123456789012:instance/1234-abcd-example-1234
MyApp,Tenant2,,123456789012,1613681904765.1956,1613681904946.574,arn:aws:ec2:us-
east-1:123456789012:instance/1234-abcd-example-1234
```

Save the data as a file, with a .csv extension (or .csv.gzip if compressed with gzip). When you upload this data to Application Cost Profiler, each time slice is assigned to the associated tenant. In this example, the report includes the time slice of the Amazon EC2 instance cost for that tenant. For EC2 instances only, slices that are not associated with a specific tenant are added to an *unattributed* tenant. Overlapping time slices are counted multiple times. It's your responsibility to ensure that the data in your usage table is accurate.

> **Note**
> Your file must represent one hour of time. If a resource is used over multiple hours, end the usage on the hour, and have a new record in the next file that starts at the same time.
> You must submit a single file containing an entire hour's data. If multiple files are submitted for the same hour's data, Application Cost Profiler only considers the data in the latest file.

For example, the following table shows how Application Cost Profiler calculates usage for three tenants, over an hour (3,600,000 milliseconds), based on provided time slices.

Tenant	Provided time slices	Calculated percent of hourly cost
Tenant1	1,200,000 ms	33.34%
Tenant2	600,000 ms	16.66%
<unattributed>		50.00%

In this example, Tenant1 is assigned one-third of the hour and Tenant2 is assigned one-sixth of the hour. The remaining half-hour (1,800,000 ms) is not attributed to either of the clients, which is 50% of the hour.

Currently, the following resources are enabled for Application Cost Profiler:

- Amazon EC2 instances (on demand and spot instances only)
- Lambda functions
- Amazon Elastic Container Service (Amazon ECS) instances
- Amazon Simple Queue Service (Amazon SQS) queues
- Amazon Simple Notification Service (Amazon SNS) topics
- Amazon DynamoDB reads and writes

> **Note**
> Amazon SQS, Amazon SNS, and DynamoDB usage is not charged by time, unlike most resources. In their case, the usage during an hour (for example, a number of reads and writes in DynamoDB), is categorized by the percentage of the hour that you allocate to different tenants, regardless of when the reads or writes happened during the hour.

Step 2: Uploading your resource usage

After you have a file of usage by tenant, upload your data file to Amazon S3 and make sure that Application Cost Profiler has permission to access it.

To learn more about creating an S3 bucket, see Application Cost Profiler specific prerequisites (p. 4).

You must make sure that Application Cost Profiler has access to your S3 bucket. This only needs to be done once per S3 bucket (you can reuse the same bucket for uploading multiple usage files). For information about giving access to the bucket, see Giving Application Cost Profiler access to your usage data S3 bucket (p. 7). If the bucket is encrypted,see Giving Application Cost Profiler access to SSE-KMS encrypted S3 buckets (p. 8).

> **Note**
> It is not required that you encrypt the S3 buckets that you use for usage data.

Upload your data to the S3 bucket as a file, with a .csv extension (or .csv.gzip if compressed with gzip), at hourly intervals. After you upload a new file, you must inform Application Cost Profiler that you have uploaded it so that the file can be imported into your report.

> **Note**
> By giving Application Cost Profiler access to your usage data, you agree that we may temporarily copy such usage data objects to the US East (N. Virginia) AWS Region while processing reports. These data objects will be kept in the US East (N. Virginia) Region until the monthly report generation is complete.

Step 3: Importing usage data into Application Cost Profiler

After you have uploaded usage data to an Amazon S3 bucket that Application Cost Profiler has access to, inform Application Cost Profiler that the data exists and to import it into your final report. You do this by using the `ImportApplicationUsage` operation in the Application Cost Profiler API.

For information about the AWS Application Cost Profiler API, including the `ImportApplicationUsage` operation, see the AWS Application Cost Profiler API Reference.

The following example shows how to call `ImportApplicationUsage`. Replace the *`input text in brackets`* with the values for your S3 bucket and uploaded object.

```
POST /ImportApplicationUsage HTTP/1.1
Content-type: application/json

{
    "sourceS3Location" : {
        "bucket": "<bucket-name>",
        "key": "<object-key>",
        "region": "<region-id>"
    }
}
```

Note
The `region` parameter is only required if your bucket is in an AWS Region that is disabled by default. For more information, see Managing AWS Regions in the *AWS General Reference*.

Application Cost Profiler generates a new report at the frequency that you requested when configuring your report (p. 10), using the data that you imported with `ImportApplicationUsage`.

After you have configured your report and automated importing your usage data into Application Cost Profiler, you are ready to view your generated reports. For more information about reports, see Using Application Cost Profiler reports (p. 15).

Using Application Cost Profiler reports

After you have integrated your usage data with AWS Application Cost Profiler and are sending the data on an hourly basis, Application Cost Profiler automatically generates your report.

Reports are generated either daily or monthly, based on the option you selected when configuring your report (p. 10). Reports are delivered to the Amazon Simple Storage Service (Amazon S3) bucket that you selected when you configured the report.

Daily reports generated on the first day of the month have the previous month's data.

Data available in an Application Cost Profiler report

The columns that are created in a usage report are shown in the following table.

Column name	Description
PayerAccountId	The management account ID in an organization, or the account ID if the account is not part of AWS Organizations.
UsageAccountId	The account ID for the account with usage.
LineItemType	The type of record. Always `Usage`.
UsageStartTime	Timestamp (in milliseconds) from Epoch, in UTC. Indicates the start time of the period for the usage by the specified tenant.
UsageEndTime	Timestamp (in milliseconds) from Epoch, in UTC. Indicates the end time of the period for the usage by the specified tenant.
ApplicationIdentifier	The **ApplicationId** specified in the usage data sent to Application Cost Profiler.
TenantIdentifier	The **TenantId** specified in the usage data sent to Application Cost Profiler. Data with no record in the usage data is collected in `unattributed`.
TenantDescription	The `TenantDesc` specified in the usage data sent to Application Cost Profiler.
ProductCode	The AWS product being billed (for example, `AmazonEC2`).
UsageType	The type of usage being billed (for example, `BoxUsage:c5.large`).

Column name	Description
Operation	The operation being billed (for example, `RunInstances`).
ResourceId	The resource ID or Amazon Resource Name (ARN) for the resource being billed.
ScaleFactor	If a resource is over-allocated for an hour, for example, the usage data reported is equal to 2 hours instead of 1 hour, a scale factor is applied to make the total equal the actual billed amount (in this case, 0.5). This column reports the scale factor used for the specific resource for that hour. The scale factor is always greater than zero (0) and less than or equal to 1.
TenantAttributionPercent	The percentage of the usage attributed to the specified tenant (between zero (0) and 1).
UsageAmount	The amount of usage attributed to the specified tenant.
CurrencyCode	The currency that the rate and cost are in (for example, `USD`).
Rate	The billing rate for the usage, per unit.
TenantCost	The total cost for that resource for the specified tenant.
Region	The AWS Region of the resource.
Name	If you created resource tags for your resources on the Cost and Usage report, or through the resource usage data, the **Name** tag is shown here. For more information about resource tags, see Resource tags details in the *Cost and Usage Report User Guide*.

The following is an example of the output report for one resource for two hours.

```
PayerAccountId,UsageAccountId,LineItemType,UsageStartTime,UsageEndTime,ApplicationIdentifier,TenantIden
123456789012,123456789012,Usage,2021-02-01T00:00:00.000Z,2021-02-01T00:30:00.000Z,Canary,unattributed,,
east-1,test-tag
123456789012,123456789012,Usage,2021-02-01T00:30:00.000Z,2021-02-01T01:00:00.000Z,Canary,Tenant1,exampl
east-1,test-tag
123456789012,123456789012,Usage,2021-02-01T01:00:00.000Z,2021-02-01T02:00:00.000Z,Canary,Tenant4,exampl
east-1,test-tag
123456789012,123456789012,Usage,2021-02-01T01:00:00.000Z,2021-02-01T02:00:00.000Z,Canary,Tenant3,exampl
east-1,test-tag
123456789012,123456789012,Usage,2021-02-01T01:00:00.000Z,2021-02-01T02:00:00.000Z,Canary,Tenant2,exampl
east-1,test-tag
123456789012,123456789012,Usage,2021-02-01T01:00:00.000Z,2021-02-01T02:00:00.000Z,Canary,Tenant1,exampl
east-1,test-tag
```

In this example, the first hour is allocated to `Tenant1` for half of the time. A half hour remains as `unattributed`. In the second hour, four tenants are all allocated the full hour. In this case, the scale

factor scales them all down by 0.25, and they are all allocated one-quarter of the hour. You can see the final cost in the `TenantCost` column.

AWS Application Cost Profiler quotas and endpoints

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is AWS Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

The following tables list the service quotas per account and the AWS Region endpoints for Application Cost Profiler.

Service quotas

Resource	Default value	Description
Rate of `PutReportDefinition` requests	5	The maximum number of `PutReportDefinition` requests per second per account.
Rate of `UpdateReportDefinition` requests	5	The maximum number of `UpdateReportDefinition` requests per second per account.
Rate of `GetReportDefinition` requests	5	The maximum number of `GetReportDefinition` requests per second per account.
Rate of `DeleteReportDefinition` requests	5	The maximum number of `DeleteReportDefinition` requests per second per account.
Rate of `ListReportDefinitions` requests	5	The maximum number of `ListReportDefinitions` requests per second per account.
Rate of `ImportApplicationUsage` requests	5	The maximum number of `ImportApplicationUsage` requests per second per account.
Maximum size of usage data file	10 MB	The maximum size of an hourly usage data file.

Service endpoints

Application Cost Profiler is a global service. All API calls must be made to the US East (N. Virginia) endpoint.

- US East (N. Virginia) – `application-cost-profiler.us-east-1.amazonaws.com`

Security in AWS Application Cost Profiler

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 Application Cost Profiler, 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 Application Cost Profiler. It shows you how to configure Application Cost Profiler to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Application Cost Profiler resources.

Contents

Data protection in AWS Application Cost Profiler

The AWS shared responsibility model applies to data protection in AWS Application Cost Profiler. 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 Application Cost Profiler 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.

Encryption at rest

AWS Application Cost Profiler always encrypts all data stored in the service at rest without requiring any additional configuration. This encryption is automatic when you use Application Cost Profiler.

For Amazon S3 buckets you provide, you must encrypt the report bucket, and can encrypt the usage data bucket and give Application Cost Profiler access. For more information, see Setting up Amazon S3 buckets for Application Cost Profiler (p. 5).

Encryption in transit

AWS Application Cost Profiler uses Transport Layer Security (TLS) and client-side encryption for encryption in transit. Communication with Application Cost Profiler is always done over HTTPS so your data is always encrypted in transit. This encryption is configured by default when you use Application Cost Profiler.

Identity and access management for AWS Application Cost Profiler

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 Application Cost Profiler 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 Application Cost Profiler.

Service user – If you use the Application Cost Profiler service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Application Cost Profiler 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 Application Cost Profiler, see Troubleshooting AWS Application Cost Profiler identity and access (p. 29).

Service administrator – If you're in charge of Application Cost Profiler resources at your company, you probably have full access to Application Cost Profiler. It's your job to determine which Application Cost Profiler features and resources your employees 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 Application Cost Profiler, see How AWS Application Cost Profiler 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 Application Cost Profiler. To view example Application Cost Profiler identity-based policies that you can use in IAM, see AWS Application Cost Profiler identity-based policy examples (p. 26).

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 first create an AWS account, you begin with a single 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, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

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:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **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 services 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 IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. 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. In other words, 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 Application Cost Profiler works with IAM

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

Topics

- Application Cost Profiler IAM roles (p. 26)

Application Cost Profiler identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources in addition to the conditions under which actions are allowed or denied. Application Cost Profiler supports specific actions. 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 Application Cost Profiler use the following prefix before the action: `application-cost-profiler:`. For example, to grant someone permission to view the details of your Application Cost Profiler report definition, you include the `application-cost-profiler:GetReportDefinition` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Application Cost Profiler 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": [
      "application-cost-profiler:ListReportDefinitions",
      "application-cost-profiler:GetReportDefinition"
```

The following are the actions available in Application Cost Profiler. Each allows the API action of the same name. For more information about the Application Cost Profiler API, see AWS Application Cost Profiler API Reference.

- `application-cost-profiler:ListReportDefinitions` – Allows listing the report definition for your AWS account, if any.
- `application-cost-profiler:GetReportDefinition` – Allows getting the details of the report definition for your Application Cost Profiler report.
- `application-cost-profiler:PutReportDefinition` – Allows creating a new report definition.
- `application-cost-profiler:UpdateReportDefinition` – Allows updating a report definition.
- `application-cost-profiler:DeleteReportDefinition` – Allows deleting a report (only available through the Application Cost Profiler API).
- `application-cost-profiler:ImportApplicationUsage` – Allows requesting Application Cost Profiler import usage data from a specified Amazon S3 bucket.

Resources

Application Cost Profiler does not support specifying resource Amazon Resource Names (ARNs) in a policy.

Condition keys

Application Cost Profiler does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

Examples

To view examples of Application Cost Profiler identity-based policies, see AWS Application Cost Profiler identity-based policy examples (p. 26).

Application Cost Profiler resource-based policies

Application Cost Profiler does not support resource-based policies.

Authorization based on Application Cost Profiler tags

Application Cost Profiler does not support tagging resources or controlling access based on tags.

Application Cost Profiler IAM roles

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

Using temporary credentials with Application Cost Profiler

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.

Application Cost Profiler 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.

Application Cost Profiler does not support 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.

Application Cost Profiler does not support service roles.

AWS Application Cost Profiler identity-based policy examples

By default, AWS Identity and Access Management (IAM) users and roles don't have permissions to create or modify AWS Application Cost Profiler resources. They also can't perform tasks using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. An IAM administrator must

create IAM policies that grant users and roles permission to perform the specific API operations that 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 are very powerful. They determine whether someone can create, access, or delete Application Cost Profiler 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 using AWS managed policies** – To start using Application Cost Profiler quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Using the Application Cost Profiler console

To access the AWS Application Cost Profiler console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Application Cost Profiler 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 use the Application Cost Profiler console to view the Application Cost Profiler report definition for your AWS account, attach the following permissions to the entities.

```
application-cost-profiler:ListReportDefinitions
application-cost-profiler:GetReportDefinition
```

For example, you could create the following policy for your read-only users.

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

```
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "application-cost-profiler:ListReportDefinitions",
            "application-cost-profiler:GetReportDefinition"
         ],
         "Resource":"*"
      }
   ]
}
```

For more information, see Adding Permissions to a User in the *IAM User Guide*.

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.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accessing one Amazon S3 bucket

In this example, you want to grant an IAM user in your AWS account access to one of your Amazon S3 buckets, `examplebucket`. You also want to allow the user to add, update, and delete objects.

In addition to granting the `s3:PutObject`, `s3:GetObject`, and `s3:DeleteObject` permissions to the user, the policy also grants the `s3:ListAllMyBuckets`, `s3:GetBucketLocation`, and `s3:ListBucket` permissions. These are the additional permissions required by the console. Also, the `s3:PutObjectAcl` and the `s3:GetObjectAcl` actions are required to be able to copy, cut, and paste objects in the console. For an example walkthrough that grants permissions to users and tests them using the console, see An Example Walkthrough: Using user policies to control access to your bucket.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListBucketsInConsole",
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      },
      {
         "Sid":"ViewSpecificBucketInfo",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource":"arn:aws:s3:::examplebucket"
      },
      {
         "Sid":"ManageBucketContents",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::examplebucket/*"
      }
   ]
}
```

Troubleshooting AWS Application Cost Profiler identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Application Cost Profiler and AWS Identity and Access Management (IAM).

Topics

I Am Not Authorized to Perform an Action in Application Cost Profiler

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 the Application Cost Profiler report but does not have `application-cost-profiler:ListReportDefinitions` permission.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 application-cost-profiler:ListReportDefinitions on resource: Report Definition
```

In this case, Mateo asks his administrator to update his policies to allow him to access the report definition resource using the `application-cost-profiler:ListReportDefinitions` 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, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Application Cost Profiler.

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 Application Cost Profiler. However, the action requires the service to have permissions 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 asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

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 Application Cost Profiler

To allow others to access Application Cost Profiler, 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 Application Cost Profiler.

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 Application Cost Profiler 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 Application Cost Profiler supports these features, see How AWS Application Cost Profiler 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*.

Compliance validation for AWS Application Cost Profiler

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, and HIPAA.

To learn whether Application Cost Profiler or other AWS services are 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 services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.

- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  Note
  Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Application Cost Profiler

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

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

Infrastructure security in AWS Application Cost Profiler

As a managed service, Application Cost Profiler 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 Application Cost Profiler 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 using an access key ID and a secret access key that is associated with an AWS Identity and Access Management (IAM) principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Monitoring Application Cost Profiler events in EventBridge

You can use Amazon EventBridge to automate your AWS services and respond automatically to system events, such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near-real time. You can write simple rules to indicate which events are of interest to you and which automated actions to take when an event matches a rule. For more information, see Amazon EventBridge User Guide.

You can monitor AWS Application Cost Profiler events in EventBridge. EventBridge routes that data to targets such as AWS Lambda and Amazon Simple Notification Service (Amazon SNS). These events are the same as those that appear in Amazon CloudWatch Events, which delivers a near-real-time stream of system events that describe changes in AWS resources.

Monitor report generation with EventBridge

With EventBridge, you can create rules that define actions to take when Application Cost Profiler sends notification of a report being generated. For example, you can create a rule that sends you an email message each time a report is generated.

To monitor for report generation

1. Log in to AWS using an account that has permissions to use both EventBridge and Application Cost Profiler.
2. Open the Amazon EventBridge console at https://console.aws.amazon.com/artifact/.
3. Choose **Create rule**.
4. Enter a **Name** for the rule, and, optionally, a description.

   A rule can't have the same name as another rule in the same AWS Region and on the same event bus.
5. For **Define pattern**, choose **Event pattern**.
6. Under **Event matching pattern**, choose **Custom pattern**.
7. For **Event pattern**, add the following pattern and then choose **Save**.

```
{
  "source": ["aws.application-cost-profiler"],
  "detail-type": ["Application Cost Profiler Report Generated"]
}
```

8. In the **Select event bus** section, choose the event bus to use. If you haven't created a custom event bus, choose **AWS default event bus**.

   Confirm that **Enable the rule on the selected event bus** is turned on.
9. For **Select targets**, choose the AWS service that you want to act when EventBridge detects an event of the selected type.
10. The fields displayed vary depending on the service you choose. Enter information specific to the target type as needed.

11. For many target types, EventBridge needs permissions to send events to the target. In these cases, EventBridge can create the AWS Identity and Access Management (IAM) role that is needed for your rule to run.

    - To create an IAM role automatically, choose **Create a new role for this specific resource**.
    - To use an IAM role that you created earlier, choose **Use existing role**.

12. For **Retry policy and dead-letter queue**, under **Retry policy**, enter the following:

    1. For **Maximum age of event**, enter a value between one minute (00:01) and 24 hours (24:00).
    2. For **Retry attempts**, enter a number between 0 and 185.

13. For **Dead-letter queue**, select **None** to not use a dead-letter queue. EventBridge sends events that match this rule to the dead-letter queue if they are not successfully delivered to the target. To use a dead-letter queue, or to learn more about them, see Using dead-letter queues in the *Amazon EventBridge User Guide*.
14. (Optional) Choose **Add target** to add another target for this rule.
15. (Optional) Enter one or more tags for the rule. For more information, see Amazon EventBridge tags in the *Amazon EventBridge User Guide*.
16. Choose **Create**.

Example of a Report Generated event

This event informs you when a report is generated and ready for you to retrieve. The `message` field gives you the Amazon Simple Storage Service (Amazon S3) bucket and key for the Amazon S3 object where the report is stored.

```
{
   "version": "0",
   "id": "01234567-EXAMPLE",
   "detail-type": "Application Cost Profiler Report Generated",
   "source": "aws.application-cost-profiler",
   "account": "123456789012",
   "time": "2021-03-31T10:23:43Z",
   "region": "us-east-1",
   "resources": [],
   "detail": {
     "message": "Application Cost Profiler report delivered in bucket: SampleBucket, key:
 SampleReport-112233445566"
   }
}
```

Document history

The following table describes the documentation releases for AWS Application Cost Profiler.

update-history-change	update-history-description	update-history-date
Updates to examples of S3 bucket policies (p. 35)	Documentation-only update to the S3 bucket policy examples. For more information, see Setting up Amazon S3 buckets for Application Cost Profiler.	December 6, 2021
General availability (p. 35)	The initial public release of Application Cost Profiler.	May 13, 2021