

# Package ‘paws.cost.management’

February 8, 2023

**Title** 'Amazon Web Services' Cost Management Services

**Version** 0.2.0

**Description** Interface to 'Amazon Web Services' cost management services, including cost and usage reports, budgets, pricing, and more  
<<https://aws.amazon.com/>>.

**License** Apache License (>= 2.0)

**URL** <https://github.com/paws-r/paws>

**BugReports** <https://github.com/paws-r/paws/issues>

**Imports** paws.common (>= 0.5.4)

**Suggests** testthat

**Encoding** UTF-8

**RoxygenNote** 7.2.2

**Collate** 'billingconductor\_service.R' 'billingconductor\_interfaces.R'  
'billingconductor\_operations.R' 'budgets\_service.R'  
'budgets\_interfaces.R' 'budgets\_operations.R'  
'costandusagereportservice\_service.R'  
'costandusagereportservice\_interfaces.R'  
'costandusagereportservice\_operations.R'  
'costexplorer\_service.R' 'costexplorer\_interfaces.R'  
'costexplorer\_operations.R' 'marketplacecatalog\_service.R'  
'marketplacecatalog\_interfaces.R'  
'marketplacecatalog\_operations.R'  
'marketplacecommerceanalytics\_service.R'  
'marketplacecommerceanalytics\_interfaces.R'  
'marketplacecommerceanalytics\_operations.R'  
'marketplaceentitlementservice\_service.R'  
'marketplaceentitlementservice\_interfaces.R'  
'marketplaceentitlementservice\_operations.R'  
'marketplacemetering\_service.R'  
'marketplacemetering\_interfaces.R'  
'marketplacemetering\_operations.R' 'pricing\_service.R'  
'pricing\_interfaces.R' 'pricing\_operations.R'

'savingsplans\_service.R' 'savingsplans\_interfaces.R'  
'savingsplans\_operations.R'

**NeedsCompilation** no

**Author** David Kretch [aut],  
Adam Banker [aut],  
Dyfan Jones [cre],  
Amazon.com, Inc. [cph]

**Maintainer** Dyfan Jones <dyfan.r.jones@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-02-08 12:40:21 UTC

## R topics documented:

billingconductor . . . . .	2
budgets . . . . .	5
costandusagereportservice . . . . .	7
costexplorer . . . . .	9
marketplacecatalog . . . . .	12
marketplacecommerceanalytics . . . . .	13
marketplaceentitlementservice . . . . .	15
marketplacemetering . . . . .	16
pricing . . . . .	19
savingsplans . . . . .	21
<b>Index</b>	<b>23</b>

---

billingconductor	<i>AWSBillingConductor</i>
------------------	----------------------------

---

## Description

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

This documentation shows how you can configure Amazon Web Services Billing Conductor using its API. For more information about using the Amazon Web Services Billing Conductor user interface, see the [Amazon Web Services Enterprise Billing Console User Guide](#).

**Usage**

```
billingconductor(config = list())
```

**Arguments**

`config` Optional configuration of credentials, endpoint, and/or region.

- **access\_key\_id**: AWS access key ID
- **secret\_access\_key**: AWS secret access key
- **session\_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close\_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3\_force\_path\_style**: Set this to `true` to force the request to use path-style addressing, i.e., `http://s3.amazonaws.com/BUCKET/KEY`.

**Value**

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```
svc <- billingconductor(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

## Operations

<a href="#">associate_accounts</a>	Connects an array of account IDs in a consolidated billing family to a
<a href="#">associate_pricing_rules</a>	Connects an array of PricingRuleArns to a defined PricingPlan
<a href="#">batch_associate_resources_to_custom_line_item</a>	Associates a batch of resources to a percentage custom line item
<a href="#">batch_disassociate_resources_from_custom_line_item</a>	Disassociates a batch of resources from a percentage custom line item
<a href="#">create_billing_group</a>	Creates a billing group that resembles a consolidated billing family th
<a href="#">create_custom_line_item</a>	Creates a custom line item that can be used to create a one-time fixed
<a href="#">create_pricing_plan</a>	Creates a pricing plan that is used for computing Amazon Web Servi
<a href="#">create_pricing_rule</a>	Creates a pricing rule can be associated to a pricing plan, or a set of p
<a href="#">delete_billing_group</a>	Deletes a billing group
<a href="#">delete_custom_line_item</a>	Deletes the custom line item identified by the given ARN in the curre
<a href="#">delete_pricing_plan</a>	Deletes a pricing plan
<a href="#">delete_pricing_rule</a>	Deletes the pricing rule identified by the input Amazon Resource Na
<a href="#">disassociate_accounts</a>	Removes the specified list of account IDs from the given billing grou
<a href="#">disassociate_pricing_rules</a>	Disassociates a list of pricing rules from a pricing plan
<a href="#">list_account_associations</a>	Amazon Web Services Billing Conductor is in beta release and is sub
<a href="#">list_billing_group_cost_reports</a>	A paginated call to retrieve a summary report of actual Amazon Web
<a href="#">list_billing_groups</a>	A paginated call to retrieve a list of billing groups for the given billin
<a href="#">list_custom_line_items</a>	A paginated call to get a list of all custom line items (FFLIs) for the
<a href="#">list_pricing_plans</a>	A paginated call to get pricing plans for the given billing period
<a href="#">list_pricing_plans_associated_with_pricing_rule</a>	A list of the pricing plans associated with a pricing rule
<a href="#">list_pricing_rules</a>	Describes a pricing rule that can be associated to a pricing plan, or se
<a href="#">list_pricing_rules_associated_to_pricing_plan</a>	Lists the pricing rules associated with a pricing plan
<a href="#">list_resources_associated_to_custom_line_item</a>	List the resources associated to a custom line item
<a href="#">list_tags_for_resource</a>	A list the tags for a resource
<a href="#">tag_resource</a>	Associates the specified tags to a resource with the specified resourc
<a href="#">untag_resource</a>	Deletes specified tags from a resource
<a href="#">update_billing_group</a>	This updates an existing billing group
<a href="#">update_custom_line_item</a>	Update an existing custom line item in the current or previous billing
<a href="#">update_pricing_plan</a>	This updates an existing pricing plan
<a href="#">update_pricing_rule</a>	Updates an existing pricing rule

## Examples

```
## Not run:
svc <- billingconductor()
svc$associate_accounts(
  Foo = 123
)

## End(Not run)
```

## Description

Use the Amazon Web Services Budgets API to plan your service usage, service costs, and instance reservations. This API reference provides descriptions, syntax, and usage examples for each of the actions and data types for the Amazon Web Services Budgets feature.

Budgets provide you with a way to see the following information:

- How close your plan is to your budgeted amount or to the free tier limits
- Your usage-to-date, including how much you've used of your Reserved Instances (RIs)
- Your current estimated charges from Amazon Web Services, and how much your predicted usage will accrue in charges by the end of the month
- How much of your budget has been used

Amazon Web Services updates your budget status several times a day. Budgets track your unblended costs, subscriptions, refunds, and RIs. You can create the following types of budgets:

- **Cost budgets** - Plan how much you want to spend on a service.
- **Usage budgets** - Plan how much you want to use one or more services.
- **RI utilization budgets** - Define a utilization threshold, and receive alerts when your RI usage falls below that threshold. This lets you see if your RIs are unused or under-utilized.
- **RI coverage budgets** - Define a coverage threshold, and receive alerts when the number of your instance hours that are covered by RIs fall below that threshold. This lets you see how much of your instance usage is covered by a reservation.

Service Endpoint

The Amazon Web Services Budgets API provides the following endpoint:

- <https://budgets.amazonaws.com>

For information about costs that are associated with the Amazon Web Services Budgets API, see [Amazon Web Services Cost Management Pricing](#).

## Usage

```
budgets(config = list())
```

## Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

- **access\_key\_id**: AWS access key ID
- **secret\_access\_key**: AWS secret access key
- **session\_token**: AWS temporary session token

- **profile:** The name of a profile to use. If not given, then the default profile is used.
- **anonymous:** Set anonymous credentials.
- **endpoint:** The complete URL to use for the constructed client.
- **region:** The AWS Region used in instantiating the client.
- **close\_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3\_force\_path\_style:** Set this to true to force the request to use path-style addressing, i.e., `http://s3.amazonaws.com/BUCKET/KEY`.

### Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```
svc <- budgets(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

### Operations

<code>create_budget</code>	Creates a budget and, if included, notifications and subscribers
<code>create_budget_action</code>	Creates a budget action
<code>create_notification</code>	Creates a notification
<code>create_subscriber</code>	Creates a subscriber
<code>delete_budget</code>	Deletes a budget
<code>delete_budget_action</code>	Deletes a budget action
<code>delete_notification</code>	Deletes a notification
<code>delete_subscriber</code>	Deletes a subscriber

<code>describe_budget</code>	Describes a budget
<code>describe_budget_action</code>	Describes a budget action detail
<code>describe_budget_action_histories</code>	Describes a budget action history detail
<code>describe_budget_actions_for_account</code>	Describes all of the budget actions for an account
<code>describe_budget_actions_for_budget</code>	Describes all of the budget actions for a budget
<code>describe_budget_notifications_for_account</code>	Lists the budget names and notifications that are associated with an account
<code>describe_budget_performance_history</code>	Describes the history for DAILY, MONTHLY, and QUARTERLY budgets
<code>describe_budgets</code>	Lists the budgets that are associated with an account
<code>describe_notifications_for_budget</code>	Lists the notifications that are associated with a budget
<code>describe_subscribers_for_notification</code>	Lists the subscribers that are associated with a notification
<code>execute_budget_action</code>	Executes a budget action
<code>update_budget</code>	Updates a budget
<code>update_budget_action</code>	Updates a budget action
<code>update_notification</code>	Updates a notification
<code>update_subscriber</code>	Updates a subscriber

### Examples

```
## Not run:
svc <- budgets()
svc$create_budget(
  Foo = 123
)

## End(Not run)
```

---

costandusagereportservice

*AWS Cost and Usage Report Service*

---

### Description

The AWS Cost and Usage Report API enables you to programmatically create, query, and delete AWS Cost and Usage report definitions.

AWS Cost and Usage reports track the monthly AWS costs and usage associated with your AWS account. The report contains line items for each unique combination of AWS product, usage type, and operation that your AWS account uses. You can configure the AWS Cost and Usage report to show only the data that you want, using the AWS Cost and Usage API.

Service Endpoint

The AWS Cost and Usage Report API provides the following endpoint:

- `cur.us-east-1.amazonaws.com`

**Usage**

```
costandusagereportservice(config = list())
```

**Arguments**

config Optional configuration of credentials, endpoint, and/or region.

- **access\_key\_id**: AWS access key ID
- **secret\_access\_key**: AWS secret access key
- **session\_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close\_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3\_force\_path\_style**: Set this to `true` to force the request to use path-style addressing, i.e., `http://s3.amazonaws.com/BUCKET/KEY`.

**Value**

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```
svc <- costandusagereportservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```



## Operations

<a href="#">delete_report_definition</a>	Deletes the specified report
<a href="#">describe_report_definitions</a>	Lists the AWS Cost and Usage reports available to this account
<a href="#">modify_report_definition</a>	Allows you to programatically update your report preferences
<a href="#">put_report_definition</a>	Creates a new report using the description that you provide

## Examples

```
## Not run:
svc <- costandusagereportservice()
# The following example deletes the AWS Cost and Usage report named
# ExampleReport.
svc$delete_report_definition(
  ReportName = "ExampleReport"
)

## End(Not run)
```

---

costexplorer

*AWS Cost Explorer Service*

---

## Description

You can use the Cost Explorer API to programmatically query your cost and usage data. You can query for aggregated data such as total monthly costs or total daily usage. You can also query for granular data. This might include the number of daily write operations for Amazon DynamoDB database tables in your production environment.

### Service Endpoint

The Cost Explorer API provides the following endpoint:

- <https://ce.us-east-1.amazonaws.com>

For information about the costs that are associated with the Cost Explorer API, see [Amazon Web Services Cost Management Pricing](#).

## Usage

```
costexplorer(config = list())
```

## Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> <li>• <b>access_key_id</b>: AWS access key ID</li> <li>• <b>secret_access_key</b>: AWS secret access key</li> <li>• <b>session_token</b>: AWS temporary session token</li> <li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li> <li>• <b>anonymous</b>: Set anonymous credentials.</li> <li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li> <li>• <b>region</b>: The AWS Region used in instantiating the client.</li> <li>• <b>close_connection</b>: Immediately close all HTTP connections.</li> <li>• <b>timeout</b>: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.</li> <li>• <b>s3_force_path_style</b>: Set this to <code>true</code> to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.</li> </ul>
--------	---

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- costexplorer(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

## Operations

[create\\_anomaly\\_monitor](#)

Creates a new cost anomaly detection monitor with the requested type and m

<a href="#">create_anomaly_subscription</a>	Adds a subscription to a cost anomaly detection monitor
<a href="#">create_cost_category_definition</a>	Creates a new Cost Category with the requested name and rules
<a href="#">delete_anomaly_monitor</a>	Deletes a cost anomaly monitor
<a href="#">delete_anomaly_subscription</a>	Deletes a cost anomaly subscription
<a href="#">delete_cost_category_definition</a>	Deletes a Cost Category
<a href="#">describe_cost_category_definition</a>	Returns the name, Amazon Resource Name (ARN), rules, definition, and effective date
<a href="#">get_anomalies</a>	Retrieves all of the cost anomalies detected on your account during the time period
<a href="#">get_anomaly_monitors</a>	Retrieves the cost anomaly monitor definitions for your account
<a href="#">get_anomaly_subscriptions</a>	Retrieves the cost anomaly subscription objects for your account
<a href="#">get_cost_and_usage</a>	Retrieves cost and usage metrics for your account
<a href="#">get_cost_and_usage_with_resources</a>	Retrieves cost and usage metrics with resources for your account
<a href="#">get_cost_categories</a>	Retrieves an array of Cost Category names and values incurred cost
<a href="#">get_cost_forecast</a>	Retrieves a forecast for how much Amazon Web Services predicts that you will incur
<a href="#">get_dimension_values</a>	Retrieves all available filter values for a specified filter over a period of time
<a href="#">get_reservation_coverage</a>	Retrieves the reservation coverage for your account, which you can use to see if you are covered
<a href="#">get_reservation_purchase_recommendation</a>	Gets recommendations for reservation purchases
<a href="#">get_reservation_utilization</a>	Retrieves the reservation utilization for your account
<a href="#">get_rightsizing_recommendation</a>	Creates recommendations that help you save cost by identifying idle and underutilized resources
<a href="#">get_savings_plans_coverage</a>	Retrieves the Savings Plans covered for your account
<a href="#">get_savings_plans_purchase_recommendation</a>	Retrieves your request parameters, Savings Plan Recommendations Summary, and recommendations
<a href="#">get_savings_plans_utilization</a>	Retrieves the Savings Plans utilization for your account across date ranges with filters
<a href="#">get_savings_plans_utilization_details</a>	Retrieves attribute data along with aggregate utilization and savings data for a specified date range
<a href="#">get_tags</a>	Queries for available tag keys and tag values for a specified period
<a href="#">get_usage_forecast</a>	Retrieves a forecast for how much Amazon Web Services predicts that you will incur
<a href="#">list_cost_allocation_tags</a>	Get a list of cost allocation tags
<a href="#">list_cost_category_definitions</a>	Returns the name, Amazon Resource Name (ARN), NumberOfRules and effective date
<a href="#">list_tags_for_resource</a>	Returns a list of resource tags associated with the resource specified by the ARN
<a href="#">provide_anomaly_feedback</a>	Modifies the feedback property of a given cost anomaly
<a href="#">tag_resource</a>	An API operation for adding one or more tags (key-value pairs) to a resource
<a href="#">untag_resource</a>	Removes one or more tags from a resource
<a href="#">update_anomaly_monitor</a>	Updates an existing cost anomaly monitor
<a href="#">update_anomaly_subscription</a>	Updates an existing cost anomaly monitor subscription
<a href="#">update_cost_allocation_tags_status</a>	Updates status for cost allocation tags in bulk, with maximum batch size of 20
<a href="#">update_cost_category_definition</a>	Updates an existing Cost Category

## Examples

```
## Not run:
svc <- costexplorer()
svc$create_anomaly_monitor(
  Foo = 123
)

## End(Not run)
```

---

marketplacecatalog      *AWS Marketplace Catalog Service*

---

## Description

Catalog API actions allow you to manage your entities through list, describe, and update capabilities. An entity can be a product or an offer on AWS Marketplace.

You can automate your entity update process by integrating the AWS Marketplace Catalog API with your AWS Marketplace product build or deployment pipelines. You can also create your own applications on top of the Catalog API to manage your products on AWS Marketplace.

## Usage

```
marketplacecatalog(config = list())
```

## Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> <li>• <b>access_key_id</b>: AWS access key ID</li> <li>• <b>secret_access_key</b>: AWS secret access key</li> <li>• <b>session_token</b>: AWS temporary session token</li> <li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li> <li>• <b>anonymous</b>: Set anonymous credentials.</li> <li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li> <li>• <b>region</b>: The AWS Region used in instantiating the client.</li> <li>• <b>close_connection</b>: Immediately close all HTTP connections.</li> <li>• <b>timeout</b>: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.</li> <li>• <b>s3_force_path_style</b>: Set this to true to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.</li> </ul>
--------	--

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- marketplacecatalog(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical"
)
)

```

## Operations

<a href="#">cancel_change_set</a>	Used to cancel an open change request
<a href="#">describe_change_set</a>	Provides information about a given change set
<a href="#">describe_entity</a>	Returns the metadata and content of the entity
<a href="#">list_change_sets</a>	Returns the list of change sets owned by the account being used to make the call
<a href="#">list_entities</a>	Provides the list of entities of a given type
<a href="#">start_change_set</a>	This operation allows you to request changes for your entities

## Examples

```

## Not run:
svc <- marketplacecatalog()
svc$cancel_change_set(
  Foo = 123
)

## End(Not run)

```

---

marketplacecommerceanalytics  
*AWS Marketplace Commerce Analytics*

---

## Description

Provides AWS Marketplace business intelligence data on-demand.

## Usage

```
marketplacecommerceanalytics(config = list())
```

**Arguments**

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> <li>• <b>access_key_id</b>: AWS access key ID</li> <li>• <b>secret_access_key</b>: AWS secret access key</li> <li>• <b>session_token</b>: AWS temporary session token</li> <li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li> <li>• <b>anonymous</b>: Set anonymous credentials.</li> <li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li> <li>• <b>region</b>: The AWS Region used in instantiating the client.</li> <li>• <b>close_connection</b>: Immediately close all HTTP connections.</li> <li>• <b>timeout</b>: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.</li> <li>• <b>s3_force_path_style</b>: Set this to true to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.</li> </ul>
--------	--

**Value**

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```
svc <- marketplacecommerceanalytics(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

**Operations**

<a href="#">generate_data_set</a>	Given a data set type and data set publication date, asynchronously publishes the requested data set
<a href="#">start_support_data_export</a>	Given a data set type and a from date, asynchronously publishes the requested customer support data

## Examples

```
## Not run:
svc <- marketplacecommerceanalytics()
svc$generate_data_set(
  Foo = 123
)

## End(Not run)
```

---

marketplaceentitlementservice

*AWS Marketplace Entitlement Service*

---

## Description

This reference provides descriptions of the AWS Marketplace Entitlement Service API.

AWS Marketplace Entitlement Service is used to determine the entitlement of a customer to a given product. An entitlement represents capacity in a product owned by the customer. For example, a customer might own some number of users or seats in an SaaS application or some amount of data capacity in a multi-tenant database.

### Getting Entitlement Records

- *GetEntitlements*- Gets the entitlements for a Marketplace product.

## Usage

```
marketplaceentitlementservice(config = list())
```

## Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"><li>• <b>access_key_id</b>: AWS access key ID</li><li>• <b>secret_access_key</b>: AWS secret access key</li><li>• <b>session_token</b>: AWS temporary session token</li><li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li><li>• <b>anonymous</b>: Set anonymous credentials.</li><li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li><li>• <b>region</b>: The AWS Region used in instantiating the client.</li><li>• <b>close_connection</b>: Immediately close all HTTP connections.</li><li>• <b>timeout</b>: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.</li><li>• <b>s3_force_path_style</b>: Set this to true to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.</li></ul>
--------	---

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- marketplaceentitlementservice(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string",  
      anonymous = "logical"  
    ),  
    endpoint = "string",  
    region = "string",  
    close_connection = "logical",  
    timeout = "numeric",  
    s3_force_path_style = "logical"  
  )  
)
```

## Operations

[get\\_entitlements](#) GetEntitlements retrieves entitlement values for a given product

## Examples

```
## Not run:  
svc <- marketplaceentitlementservice()  
svc$get_entitlements(  
  Foo = 123  
)  
  
## End(Not run)
```



## Description

AWS Marketplace Metering Service

This reference provides descriptions of the low-level AWS Marketplace Metering Service API.

AWS Marketplace sellers can use this API to submit usage data for custom usage dimensions.

For information on the permissions you need to use this API, see [AWS Marketplace metering and entitlement API permissions](#) in the *AWS Marketplace Seller Guide*.

### Submitting Metering Records

- *MeterUsage* - Submits the metering record for an AWS Marketplace product. `meter_usage` is called from an EC2 instance or a container running on EKS or ECS.
- *BatchMeterUsage* - Submits the metering record for a set of customers. `batch_meter_usage` is called from a software-as-a-service (SaaS) application.

### Accepting New Customers

- *ResolveCustomer* - Called by a SaaS application during the registration process. When a buyer visits your website during the registration process, the buyer submits a Registration Token through the browser. The Registration Token is resolved through this API to obtain a CustomerIdentifier along with the CustomerAWSAccountId and ProductCode.

### Entitlement and Metering for Paid Container Products

- Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the `register_usage` operation for software entitlement and metering. Free and BYOL products for Amazon ECS or Amazon EKS aren't required to call `register_usage`, but you can do so if you want to receive usage data in your seller reports. For more information on using the `register_usage` operation, see [Container-Based Products](#).

`batch_meter_usage` API calls are captured by AWS CloudTrail. You can use Cloudtrail to verify that the SaaS metering records that you sent are accurate by searching for records with the eventName of `batch_meter_usage`. You can also use CloudTrail to audit records over time. For more information, see the [AWS CloudTrail User Guide](http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html).

## Usage

```
marketplacemetering(config = list())
```

## Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

- **access\_key\_id**: AWS access key ID
- **secret\_access\_key**: AWS secret access key
- **session\_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

- **endpoint:** The complete URL to use for the constructed client.
- **region:** The AWS Region used in instantiating the client.
- **close\_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3\_force\_path\_style:** Set this to true to force the request to use path-style addressing, i.e., `http://s3.amazonaws.com/BUCKET/KEY`.

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- marketplacemetering(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

## Operations

<a href="#">batch_meter_usage</a>	BatchMeterUsage is called from a SaaS application listed on AWS Marketplace to post metering records
<a href="#">meter_usage</a>	API to emit metering records
<a href="#">register_usage</a>	Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace
<a href="#">resolve_customer</a>	ResolveCustomer is called by a SaaS application during the registration process

## Examples

```
## Not run:
svc <- marketplacemetering()
```

```

svc$batch_meter_usage(
  Foo = 123
)

## End(Not run)

```

pricing

*AWS Price List Service***Description**

Amazon Web Services Price List API is a centralized and convenient way to programmatically query Amazon Web Services for services, products, and pricing information. The Amazon Web Services Price List uses standardized product attributes such as Location, Storage Class, and Operating System, and provides prices at the SKU level. You can use the Amazon Web Services Price List to build cost control and scenario planning tools, reconcile billing data, forecast future spend for budgeting purposes, and provide cost benefit analysis that compare your internal workloads with Amazon Web Services.

Use `GetServices` without a service code to retrieve the service codes for all AWS services, then `GetServices` with a service code to retrieve the attribute names for that service. After you have the service code and attribute names, you can use [get\\_attribute\\_values](#) to see what values are available for an attribute. With the service code and an attribute name and value, you can use [get\\_products](#) to find specific products that you're interested in, such as an AmazonEC2 instance, with a Provisioned IOPS volumeType.

Service Endpoint

Amazon Web Services Price List service API provides the following two endpoints:

- <https://api.pricing.us-east-1.amazonaws.com>
- <https://api.pricing.ap-south-1.amazonaws.com>

**Usage**

```
pricing(config = list())
```

**Arguments**

<code>config</code>	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> <li>• <b>access_key_id</b>: AWS access key ID</li> <li>• <b>secret_access_key</b>: AWS secret access key</li> <li>• <b>session_token</b>: AWS temporary session token</li> <li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li> <li>• <b>anonymous</b>: Set anonymous credentials.</li> <li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li> </ul>
---------------------	--

- **region:** The AWS Region used in instantiating the client.
- **close\_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3\_force\_path\_style:** Set this to true to force the request to use path-style addressing, i.e., `http://s3.amazonaws.com/BUCKET/KEY`.

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- pricing(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)
```

## Operations

<code>describe_services</code>	Returns the metadata for one service or a list of the metadata for all services
<code>get_attribute_values</code>	Returns a list of attribute values
<code>get_products</code>	Returns a list of all products that match the filter criteria

## Examples

```
## Not run:
svc <- pricing()
# Retrieves the service for the given Service Code.
svc$describe_services(
```

```

    FormatVersion = "aws_v1",
    MaxResults = 1L,
    ServiceCode = "AmazonEC2"
)

## End(Not run)

```

---

savingsplans

*AWS Savings Plans*


---

## Description

Savings Plans are a pricing model that offer significant savings on AWS usage (for example, on Amazon EC2 instances). You commit to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years, and receive a lower price for that usage. For more information, see the [AWS Savings Plans User Guide](#).

## Usage

```
savingsplans(config = list())
```

## Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> <li>• <b>access_key_id</b>: AWS access key ID</li> <li>• <b>secret_access_key</b>: AWS secret access key</li> <li>• <b>session_token</b>: AWS temporary session token</li> <li>• <b>profile</b>: The name of a profile to use. If not given, then the default profile is used.</li> <li>• <b>anonymous</b>: Set anonymous credentials.</li> <li>• <b>endpoint</b>: The complete URL to use for the constructed client.</li> <li>• <b>region</b>: The AWS Region used in instantiating the client.</li> <li>• <b>close_connection</b>: Immediately close all HTTP connections.</li> <li>• <b>timeout</b>: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.</li> <li>• <b>s3_force_path_style</b>: Set this to <code>true</code> to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.</li> </ul>
--------	--

## Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```

svc <- savingsplans(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical"
  )
)

```

**Operations**

<a href="#">create_savings_plan</a>	Creates a Savings Plan
<a href="#">delete_queued_savings_plan</a>	Deletes the queued purchase for the specified Savings Plan
<a href="#">describe_savings_plan_rates</a>	Describes the specified Savings Plans rates
<a href="#">describe_savings_plans</a>	Describes the specified Savings Plans
<a href="#">describe_savings_plans_offering_rates</a>	Describes the specified Savings Plans offering rates
<a href="#">describe_savings_plans_offerings</a>	Describes the specified Savings Plans offerings
<a href="#">list_tags_for_resource</a>	Lists the tags for the specified resource
<a href="#">tag_resource</a>	Adds the specified tags to the specified resource
<a href="#">untag_resource</a>	Removes the specified tags from the specified resource

**Examples**

```

## Not run:
svc <- savingsplans()
svc$create_savings_plan(
  Foo = 123
)

## End(Not run)

```

# Index

associate\_accounts, [4](#)  
associate\_pricing\_rules, [4](#)  
batch\_associate\_resources\_to\_custom\_line\_item, [4](#)  
batch\_disassociate\_resources\_from\_custom\_line\_item, [4](#)  
batch\_meter\_usage, [17, 18](#)  
billingconductor, [2](#)  
budgets, [5](#)  
cancel\_change\_set, [13](#)  
costandusagereportservice, [7](#)  
costexplorer, [9](#)  
create\_anomaly\_monitor, [10](#)  
create\_anomaly\_subscription, [11](#)  
create\_billing\_group, [4](#)  
create\_budget, [6](#)  
create\_budget\_action, [6](#)  
create\_cost\_category\_definition, [11](#)  
create\_custom\_line\_item, [4](#)  
create\_notification, [6](#)  
create\_pricing\_plan, [4](#)  
create\_pricing\_rule, [4](#)  
create\_savings\_plan, [22](#)  
create\_subscriber, [6](#)  
delete\_anomaly\_monitor, [11](#)  
delete\_anomaly\_subscription, [11](#)  
delete\_billing\_group, [4](#)  
delete\_budget, [6](#)  
delete\_budget\_action, [6](#)  
delete\_cost\_category\_definition, [11](#)  
delete\_custom\_line\_item, [4](#)  
delete\_notification, [6](#)  
delete\_pricing\_plan, [4](#)  
delete\_pricing\_rule, [4](#)  
delete\_queued\_savings\_plan, [22](#)  
delete\_report\_definition, [9](#)  
delete\_subscriber, [6](#)  
describe\_budget, [7](#)  
describe\_budget\_action, [7](#)  
describe\_budget\_action\_histories, [7](#)  
describe\_budget\_actions\_for\_account, [7](#)  
describe\_budget\_actions\_for\_budget, [7](#)  
describe\_budget\_notifications\_for\_account, [7](#)  
describe\_budget\_performance\_history, [7](#)  
describe\_budgets, [7](#)  
describe\_change\_set, [13](#)  
describe\_cost\_category\_definition, [11](#)  
describe\_entity, [13](#)  
describe\_notifications\_for\_budget, [7](#)  
describe\_report\_definitions, [9](#)  
describe\_savings\_plan\_rates, [22](#)  
describe\_savings\_plans, [22](#)  
describe\_savings\_plans\_offering\_rates, [22](#)  
describe\_savings\_plans\_offerings, [22](#)  
describe\_services, [20](#)  
describe\_subscribers\_for\_notification, [7](#)  
disassociate\_accounts, [4](#)  
disassociate\_pricing\_rules, [4](#)  
execute\_budget\_action, [7](#)  
generate\_data\_set, [14](#)  
get\_anomalies, [11](#)  
get\_anomaly\_monitors, [11](#)  
get\_anomaly\_subscriptions, [11](#)  
get\_attribute\_values, [19, 20](#)  
get\_cost\_and\_usage, [11](#)  
get\_cost\_and\_usage\_with\_resources, [11](#)  
get\_cost\_categories, [11](#)  
get\_cost\_forecast, [11](#)  
get\_dimension\_values, [11](#)  
get\_entitlements, [16](#)  
get\_products, [19, 20](#)  
get\_reservation\_coverage, [11](#)

get\_reservation\_purchase\_recommendation, [11](#)  
get\_reservation\_utilization, [11](#)  
get\_rightsizing\_recommendation, [11](#)  
get\_savings\_plans\_coverage, [11](#)  
get\_savings\_plans\_purchase\_recommendation, [11](#)  
get\_savings\_plans\_utilization, [11](#)  
get\_savings\_plans\_utilization\_details, [11](#)  
get\_tags, [11](#)  
get\_usage\_forecast, [11](#)

list\_account\_associations, [4](#)  
list\_billing\_group\_cost\_reports, [4](#)  
list\_billing\_groups, [4](#)  
list\_change\_sets, [13](#)  
list\_cost\_allocation\_tags, [11](#)  
list\_cost\_category\_definitions, [11](#)  
list\_custom\_line\_items, [4](#)  
list\_entities, [13](#)  
list\_pricing\_plans, [4](#)  
list\_pricing\_plans\_associated\_with\_pricing\_rule, [4](#)  
list\_pricing\_rules, [4](#)  
list\_pricing\_rules\_associated\_to\_pricing\_plan, [4](#)  
list\_resources\_associated\_to\_custom\_line\_item, [4](#)  
list\_tags\_for\_resource, [4](#), [11](#), [22](#)

marketplacecatalog, [12](#)  
marketplacecommerceanalytics, [13](#)  
marketplaceentitlementservice, [15](#)  
marketplacemetering, [16](#)  
meter\_usage, [17](#), [18](#)  
modify\_report\_definition, [9](#)

pricing, [19](#)  
provide\_anomaly\_feedback, [11](#)  
put\_report\_definition, [9](#)

register\_usage, [17](#), [18](#)  
resolve\_customer, [18](#)

savingsplans, [21](#)  
start\_change\_set, [13](#)  
start\_support\_data\_export, [14](#)

tag\_resource, [4](#), [11](#), [22](#)  
untag\_resource, [4](#), [11](#), [22](#)  
update\_anomaly\_monitor, [11](#)  
update\_anomaly\_subscription, [11](#)  
update\_billing\_group, [4](#)  
update\_budget, [7](#)  
update\_budget\_action, [7](#)  
update\_cost\_allocation\_tags\_status, [11](#)  
update\_cost\_category\_definition, [11](#)  
update\_custom\_line\_item, [4](#)  
update\_notification, [7](#)  
update\_pricing\_plan, [4](#)  
update\_pricing\_rule, [4](#)  
update\_subscriber, [7](#)