

# Package ‘paws.cost.management’

March 9, 2021

**Title** Amazon Web Services Cost Management Services

**Version** 0.1.11

**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 (>= 0.3.0)

**Suggests** testthat

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Collate** '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'  
'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'

**NeedsCompilation** no

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

**Maintainer** David Kretch <david.kretch@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-03-09 07:50:25 UTC

## R topics documented:

budgets . . . . .	2
costandusagereportservice . . . . .	4
costexplorer . . . . .	5
marketplacecommerceanalytics . . . . .	7
marketplaceentitlementservice . . . . .	8
marketplacemetering . . . . .	9
pricing . . . . .	11
<b>Index</b>	<b>14</b>

---

budgets	<i>AWS Budgets</i>
---------	--------------------

---

## Description

The AWS Budgets API enables you to use AWS Budgets to plan your service usage, service costs, and instance reservations. The API reference provides descriptions, syntax, and usage examples for each of the actions and data types for AWS Budgets.

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 AWS, and how much your predicted usage will accrue in charges by the end of the month
- How much of your budget has been used

AWS 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 AWS Budgets API provides the following endpoint:

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

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

**Usage**

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

**Arguments**

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

**Service syntax**

```
svc <- budgets(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

**Operations**

<a href="#">create_budget</a>	Creates a budget and, if included, notifications and subscribers
<a href="#">create_budget_action</a>	Creates a budget action
<a href="#">create_notification</a>	Creates a notification
<a href="#">create_subscriber</a>	Creates a subscriber
<a href="#">delete_budget</a>	Deletes a budget
<a href="#">delete_budget_action</a>	Deletes a budget action
<a href="#">delete_notification</a>	Deletes a notification
<a href="#">delete_subscriber</a>	Deletes a subscriber
<a href="#">describe_budget</a>	Describes a budget
<a href="#">describe_budget_action</a>	Describes a budget action detail
<a href="#">describe_budget_action_histories</a>	Describes a budget action history detail
<a href="#">describe_budget_actions_for_account</a>	Describes all of the budget actions for an account
<a href="#">describe_budget_actions_for_budget</a>	Describes all of the budget actions for a budget
<a href="#">describe_budget_performance_history</a>	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.

**Service syntax**

```

svc <- costandusagereportservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

**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)

```

**Description**

The Cost Explorer API enables you 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, such as 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 costs associated with the Cost Explorer API, see [AWS Cost Management Pricing](#).

### Usage

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

### Arguments

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

### Service syntax

```

svc <- costexplorer(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

### Operations

<a href="#">create_anomaly_monitor</a>	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, ARN, rules, definition, and effective dates of a Cost Cate
<a href="#">get_anomalies</a>	Retrieves all of the cost anomalies detected on your account, during the time
<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 w

<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
<a href="#">get_reservation_purchase_recommendation</a>	Gets recommendations for which reservations to purchase
<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
<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 period
<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 spend
<a href="#">list_cost_category_definitions</a>	Returns the name, ARN, NumberOfRules and effective dates of all Cost Categories
<a href="#">provide_anomaly_feedback</a>	Modifies the feedback property of a given cost anomaly
<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_category_definition</a>	Updates an existing Cost Category

## Examples

```
## Not run:
svc <- costexplorer()
svc$create_anomaly_monitor(
  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.

**Service syntax**

```

svc <- marketplacecommerceanalytics(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

**Operations**

[generate\\_data\\_set](#) Given a data set type and data set publication date, asynchronously publishes the requested data set.

[start\\_support\\_data\\_export](#) 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.

**Service syntax**

```
svc <- marketplaceentitlementservice(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string"  
    ),  
    endpoint = "string",  
    region = "string"  
  )  
)
```

**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 Marketing metering and entitlement API permissions](#) in the *AWS Marketplace Seller Guide*.

**Submitting Metering Records**

- *MeterUsage*- Submits the metering record for a Marketplace product. MeterUsage is called from an EC2 instance or a container running on EKS or ECS.
- *BatchMeterUsage*- Submits the metering record for a set of customers. BatchMeterUsage 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 and Product Code.

**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 RegisterUsage operation for software entitlement and metering. Free and BYOL products for Amazon ECS or Amazon EKS aren't required to call RegisterUsage, but you can do so if you want to receive usage data in your seller reports. For more information on using the RegisterUsage operation, see [Container-Based Products](#).

BatchMeterUsage 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 BatchMeterUsage. You can also use CloudTrail to audit records over time. For more information, see the [AWS CloudTrail User Guide](#).

**Usage**

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

**Arguments**

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

**Service syntax**

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

```

        session_token = "string"
    ),
    profile = "string"
),
endpoint = "string",
region = "string"
)
)

```

## Operations

<a href="#">batch_meter_usage</a>	BatchMeterUsage is called from a SaaS application listed on the 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

AWS Price List Service API (AWS Price List Service) is a centralized and convenient way to programmatically query Amazon Web Services for services, products, and pricing information. The AWS Price List Service uses standardized product attributes such as Location, Storage Class, and Operating System, and provides prices at the SKU level. You can use the AWS Price List Service 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 AWS.

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

AWS 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

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

### Service syntax

```

svc <- pricing(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

### Operations

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

### Examples

```

## Not run:
svc <- pricing()
svc$describe_services(
  FormatVersion = "aws_v1",
  MaxResults = 1L,
  ServiceCode = "AmazonEC2"
)

```

*pricing*

13

## End(Not run)

# Index

batch\_meter\_usage, [11](#)  
budgets, [2](#)

costandusagereportservice, [4](#)  
costexplorer, [5](#)  
create\_anomaly\_monitor, [6](#)  
create\_anomaly\_subscription, [6](#)  
create\_budget, [3](#)  
create\_budget\_action, [3](#)  
create\_cost\_category\_definition, [6](#)  
create\_notification, [3](#)  
create\_subscriber, [3](#)

delete\_anomaly\_monitor, [6](#)  
delete\_anomaly\_subscription, [6](#)  
delete\_budget, [3](#)  
delete\_budget\_action, [3](#)  
delete\_cost\_category\_definition, [6](#)  
delete\_notification, [3](#)  
delete\_report\_definition, [5](#)  
delete\_subscriber, [3](#)  
describe\_budget, [3](#)  
describe\_budget\_action, [3](#)  
describe\_budget\_action\_histories, [3](#)  
describe\_budget\_actions\_for\_account, [3](#)  
describe\_budget\_actions\_for\_budget, [3](#)  
describe\_budget\_performance\_history, [3](#)  
describe\_budgets, [4](#)  
describe\_cost\_category\_definition, [6](#)  
describe\_notifications\_for\_budget, [4](#)  
describe\_report\_definitions, [5](#)  
describe\_services, [12](#)  
describe\_subscribers\_for\_notification, [4](#)

execute\_budget\_action, [4](#)

generate\_data\_set, [8](#)  
get\_anomalies, [6](#)  
get\_anomaly\_monitors, [6](#)  
get\_anomaly\_subscriptions, [6](#)  
get\_attribute\_values, [11, 12](#)  
get\_cost\_and\_usage, [6](#)  
get\_cost\_and\_usage\_with\_resources, [6](#)  
get\_cost\_categories, [6](#)  
get\_cost\_forecast, [6](#)  
get\_dimension\_values, [7](#)  
get\_entitlements, [9](#)  
get\_products, [11, 12](#)  
get\_reservation\_coverage, [7](#)  
get\_reservation\_purchase\_recommendation, [7](#)  
get\_reservation\_utilization, [7](#)  
get\_rightsizing\_recommendation, [7](#)  
get\_savings\_plans\_coverage, [7](#)  
get\_savings\_plans\_purchase\_recommendation, [7](#)  
get\_savings\_plans\_utilization, [7](#)  
get\_savings\_plans\_utilization\_details, [7](#)  
get\_tags, [7](#)  
get\_usage\_forecast, [7](#)

list\_cost\_category\_definitions, [7](#)

marketplacecommerceanalytics, [7](#)  
marketplaceentitlementservice, [8](#)  
marketplacemetering, [9](#)  
meter\_usage, [11](#)  
modify\_report\_definition, [5](#)

pricing, [11](#)  
provide\_anomaly\_feedback, [7](#)  
put\_report\_definition, [5](#)

register\_usage, [11](#)  
resolve\_customer, [11](#)

start\_support\_data\_export, [8](#)  
update\_anomaly\_monitor, [7](#)

update\_anomaly\_subscription, 7  
update\_budget, 4  
update\_budget\_action, 4  
update\_cost\_category\_definition, 7  
update\_notification, 4  
update\_subscriber, 4