Package: paws.cost.management (via r-universe)

October 28, 2026

Title 'Amazon Web Services' Cost Management Services

Version 0.11.0

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, https://paws-r.r-universe.dev/paws.cost.management, https://www.paws-r-sdk.com

BugReports https://github.com/paws-r/paws/issues

Depends R (>= 3.5.0)

Imports paws.common (>= 0.9.0)

Suggests testthat

Encoding UTF-8

Roxygen list(markdown = TRUE, roclets = c(``rd'', ``namespace'', ``collate''))

Config/roxygen2/version 8.1.0

Collate 'billing_service.R' 'billing_interfaces.R' 'billing_operations.R' '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'
'paymentcryptographycontrolplane_service.R'
'paymentcryptographycontrolplane_interfaces.R'
'paymentcryptographycontrolplane_operations.R'
'paymentcryptographydataplane_service.R'
'paymentcryptographydataplane_interfaces.R'
'paymentcryptographydataplane_operations.R' 'pricing_service.R'
'pricing_interfaces.R' 'pricing_operations.R'
'reexports_paws.common.R' 'savingsplans_service.R'
'savingsplans_interfaces.R' 'savingsplans_operations.R'

Config/pak/sysreqs libxml2-dev libssl-dev

Repository https://paws-r.r-universe.dev

Date/Publication 2026-10-28 19:14:01 UTC

RemoteUrl https://github.com/paws-r/paws

RemoteRef HEAD

RemoteSha 1341bca0616aa9ee814720f471634b968e45e133

RemoteSubdir cran/paws.cost.management

Contents

billing . . . 3
billingconductor . . . 5
budgets . . . 9
costandusagereportservice . . . 12
costexplorer . . . 15
marketplacecatalog . . . 18
marketplacecommerceanalytics . . . 21
marketplaceentitlementservice . . . 23
marketplacemetering . . . 26
paymentcryptographycontrolplane . . . 30
paymentcryptographydataplane . . . 33
pricing . . . 36
savingsplans . . . 39

Index **42**

billing *AWS Billing*

Description

You can use the Billing API to programatically list the billing views available to you for a given time period. A billing view represents a set of billing data.

The Billing API provides the following endpoint:
`https://billing.us-east-1.api.aws`

Usage

```
billing(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- billing(
  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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_source_views	Associates one or more source billing views with an existing billing view
create_billing_view	Creates a billing view with the specified billing view attributes
delete_billing_view	Deletes the specified billing view
disassociate_source_views	Removes the association between one or more source billing views and an
get_billing_preferences	Retrieves billing preferences for the specified feature
get_billing_view	Returns the metadata associated to the specified billing view ARN
get_credit_allocation_history	Returns the per-billing-month allocation history for credits applied to an A
get_credits	Returns the list of Amazon Web Services account credits for the specified a
get_enterprise_support_charge_summary	Returns a summary of Enterprise Support data aggregated across all accou
get_enterprise_support_contract_details	Returns Enterprise Support contract details
get_resource_policy	Returns the resource-based policy document attached to the resource in JS
list_billing_views	Lists the billing views available for a given time period
list_billing_view_segments	Lists the segments of a billing view over a given time period
list_business_support_account_charges	Returns Business Support charges broken down at the linked account level
list_business_support_subscription_history	Returns the history of Business Support subscription contracts across accou
list_enterprise_support_linked_account_charges	Returns Support-eligible spend broken down at linked account level
list_source_views_for_billing_view	Lists the source views (managed Amazon Web Services billing views) asso
list_tags_for_resource	Lists tags associated with the billing view resource
redeem_credits	Redeems an Amazon Web Services promotional credit code on behalf of th
tag_resource	An API operation for adding one or more tags (key-value pairs) to a resour
untag_resource	Removes one or more tags from a resource
update_billing_preferences	Updates billing preferences for the specified feature
update_billing_view	An API to update the attributes of the billing view

Examples

```
## Not run:
svc <- billing()
svc$associate_source_views(
  Foo = 123
)

## End(Not run)
```

`billingconductor` *AWSBillingConductor*

Description

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. 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 Billing Conductor configuration, the payer account can also see the custom rate applied on the billing details page of the Billing console, or configure a cost and usage report per billing group.

This documentation shows how you can configure Billing Conductor by using its API. For more information about using the Billing Conductor user interface, see the Billing Conductor User Guide.

Usage

```
billingconductor(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_accounts	Connects an array of account IDs in a consolidated billing family to a
associate_pricing_rules	Connects an array of PricingRuleArns to a defined PricingPlan
batch_associate_resources_to_custom_line_item	Associates a batch of resources to a percentage custom line item
batch_disassociate_resources_from_custom_line_item	Disassociates a batch of resources from a percentage custom line iten
create_billing_group	Creates a billing group that resembles a consolidated billing family th
create_custom_line_item	Creates a custom line item that can be used to create a one-time fixed
create_pricing_plan	Creates a pricing plan that is used for computing Amazon Web Servi
create_pricing_rule	Creates a pricing rule can be associated to a pricing plan, or a set of p
delete_billing_group	Deletes a billing group
delete_custom_line_item	Deletes the custom line item identified by the given ARN in the curre
delete_pricing_plan	Deletes a pricing plan
delete_pricing_rule	Deletes the pricing rule that's identified by the input Amazon Resour
disassociate_accounts	Removes the specified list of account IDs from the given billing grou
disassociate_pricing_rules	Disassociates a list of pricing rules from a pricing plan
get_billing_group_cost_report	Retrieves the margin summary report, which includes the Amazon W
get_billing_transfer_preference	Retrieves the auto billing group creation preference for a billing trans
list_account_associations	This is a paginated call to list linked accounts that are linked to the pa
list_billing_group_cost_reports	A paginated call to retrieve a summary report of actual Amazon Web
list_billing_groups	A paginated call to retrieve a list of billing groups for the given billin
list_custom_line_items	A paginated call to get a list of all custom line items (FFLIs) for the g
list_custom_line_item_versions	A paginated call to get a list of all custom line item versions
list_pricing_plans	A paginated call to get pricing plans for the given billing period
list_pricing_plans_associated_with_pricing_rule	A list of the pricing plans that are associated with a pricing rule
list_pricing_rules	Describes a pricing rule that can be associated to a pricing plan, or se
list_pricing_rules_associated_to_pricing_plan	Lists the pricing rules that are associated with a pricing plan
list_resources_associated_to_custom_line_item	List the resources that are associated to a custom line item
list_tags_for_resource	A list the tags for a resource
tag_resource	Associates the specified tags to a resource with the specified resource
untag_resource	Deletes specified tags from a resource
update_billing_group	This updates an existing billing group
update_billing_transfer_preference	Sets the auto billing group creation preference for a billing transfer
update_custom_line_item	Update an existing custom line item in the current or previous billing
update_pricing_plan	This updates an existing pricing plan
update_pricing_rule	Updates an existing pricing rule

Examples

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

## End(Not run)
```

budgets *AWS Budgets*

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(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_budget	Creates a budget and, if included, notifications and subscribers
create_budget_action	Creates a budget action
create_notification	Creates a notification
create_subscriber	Creates a subscriber
delete_budget	Deletes a budget
delete_budget_action	Deletes a budget action
delete_notification	Deletes a notification
delete_subscriber	Deletes a subscriber
describe_budget	Describes a budget
describe_budget_action	Describes a budget action detail
describe_budget_action_histories	Describes a budget action history detail
describe_budget_actions_for_account	Describes all of the budget actions for an account
describe_budget_actions_for_budget	Describes all of the budget actions for a budget
describe_budget_notifications_for_account	Lists the budget names and notifications that are associated with an account
describe_budget_performance_history	Describes the history for DAILY, MONTHLY, and QUARTERLY budgets
describe_budgets	Lists the budgets that are associated with an account
describe_notifications_for_budget	Lists the notifications that are associated with a budget
describe_subscribers_for_notification	Lists the subscribers that are associated with a notification
execute_budget_action	Executes a budget action
list_tags_for_resource	Lists tags associated with a budget or budget action resource
tag_resource	Creates tags for a budget or budget action resource
untag_resource	Deletes tags associated with a budget or budget action resource
update_budget	Updates a budget

update_budget_action	Updates a budget action
update_notification	Updates a notification
update_subscriber	Updates a subscriber

Examples

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

## End(Not run)
```

`costandusagereportservice`

AWS Cost and Usage Report Service

Description

You can use the Amazon Web Services Cost and Usage Report API to programmatically create, query, and delete Amazon Web Services Cost and Usage Report definitions.

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

Service Endpoint

The Amazon Web Services Cost and Usage Report API provides the following endpoint:

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

Usage

```
costandusagereportservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

delete_report_definition	Deletes the specified report
describe_report_definitions	Lists the Amazon Web Services Cost and Usage Report available to this account
list_tags_for_resource	Lists the tags associated with the specified report definition
modify_report_definition	Allows you to programmatically update your report preferences
put_report_definition	Creates a new report using the description that you provide
tag_resource	Associates a set of tags with a report definition
untag_resource	Disassociates a set of tags from a report definition

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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      sts_regional_endpoint = "string",
      use_dual_stack = "logical"
    ),
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
  )
```

Operations

create_anomaly_monitor	Creates a new cost anomaly detection monitor with the requeste
create_anomaly_subscription	Adds an alert subscription to a cost anomaly detection monitor
create_cost_category_definition	Creates a new cost category with the requested name and rules
delete_anomaly_monitor	Deletes a cost anomaly monitor
delete_anomaly_subscription	Deletes a cost anomaly subscription
delete_cost_category_definition	Deletes a cost category
describe_cost_category_definition	Returns the name, Amazon Resource Name (ARN), rules, defin
get_anomalies	Retrieves all of the cost anomalies detected on your account dur
get_anomaly_monitors	Retrieves the cost anomaly monitor definitions for your account
get_anomaly_subscriptions	Retrieves the cost anomaly subscription objects for your accoun
get_approximate_usage_records	Retrieves estimated usage records for hourly granularity or reso
get_commitment_purchase_analysis	Retrieves a commitment purchase analysis result based on the A
get_cost_and_usage	Retrieves cost and usage metrics for your account
get_cost_and_usage_comparisons	Retrieves cost and usage comparisons for your account between
get_cost_and_usage_with_resources	Retrieves cost and usage metrics with resources for your accoun
get_cost_categories	Retrieves an array of cost category names and values incurred c
get_cost_comparison_drivers	Retrieves key factors driving cost changes between two time per
get_cost_forecast	Retrieves a forecast for how much Amazon Web Services predic
get_dimension_values	Retrieves all available filter values for a specified filter over a pe
get_reservation_coverage	Retrieves the reservation coverage for your account, which you
get_reservation_purchase_recommendation	Gets recommendations for reservation purchases
get_reservation_utilization	Retrieves the reservation utilization for your account
get_rightsizing_recommendation	Creates recommendations that help you save cost by identifying
get_savings_plan_purchase_recommendation_details	Retrieves the details for a Savings Plan recommendation
get_savings_plans_coverage	Retrieves the Savings Plans covered for your account
get_savings_plans_purchase_recommendation	Retrieves the Savings Plans recommendations for your account
get_savings_plans_utilization	Retrieves the Savings Plans utilization for your account across c

get_savings_plans_utilization_details	Retrieves attribute data along with aggregate utilization and savi
get_tags	Queries for available tag keys and tag values for a specified peri
get_usage_forecast	Retrieves a forecast for how much Amazon Web Services predic
list_commitment_purchase_analyses	Lists the commitment purchase analyses for your account
list_cost_allocation_tag_backfill_history	Retrieves a list of your historical cost allocation tag backfill requ
list_cost_allocation_tags	Get a list of cost allocation tags
list_cost_category_definitions	Returns the name, Amazon Resource Name (ARN), NumberOfl
list_cost_category_resource_associations	Returns resource associations of all cost categories defined in th
list_savings_plans_purchase_recommendation_generation	Retrieves a list of your historical recommendation generations v
list_tags_for_resource	Returns a list of resource tags associated with the resource speci
provide_anomaly_feedback	Modifies the feedback property of a given cost anomaly
start_commitment_purchase_analysis	Specifies the parameters of a planned commitment purchase and
start_cost_allocation_tag_backfill	Request a cost allocation tag backfill
start_savings_plans_purchase_recommendation_generation	Requests a Savings Plans recommendation generation
tag_resource	An API operation for adding one or more tags (key-value pairs)
untag_resource	Removes one or more tags from a resource
update_anomaly_monitor	Updates an existing cost anomaly monitor
update_anomaly_subscription	Updates an existing cost anomaly subscription
update_cost_allocation_tags_status	Updates status for cost allocation tags in bulk, with maximum b
update_cost_category_definition	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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_describe_entities	Returns metadata and content for multiple entities
cancel_change_set	Used to cancel an open change request
delete_resource_policy	Deletes a resource-based policy on an entity that is identified by its resource ARN
describe_assessment	Returns the metadata and detailed results of a single assessment, including the framework that was e
describe_change_set	Provides information about a given change set
describe_entity	Returns the metadata and content of the entity
get_resource_policy	Gets a resource-based policy of an entity that is identified by its resource ARN

list_assessments	Returns a paginated list of assessments associated with an entity or change set in AWS Marketplace
list_change_sets	Returns the list of change sets owned by the account being used to make the call
list_entities	Provides the list of entities of a given type
list_tags_for_resource	Lists all tags that have been added to a resource (either an entity or change set)
put_resource_policy	Attaches a resource-based policy to an entity
start_change_set	Allows you to request changes for your entities
tag_resource	Tags a resource (either an entity or change set)
untag_resource	Removes a tag or list of tags from a resource (either an entity or change set)

Examples

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

## End(Not run)
```

`marketplacecommerceanalytics`
AWS Marketplace Commerce Analytics

Description

Provides AWS Marketplace business intelligence data on-demand.

Usage

```
marketplacecommerceanalytics(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

generate_data_set	Given a data set type and data set publication date, asynchronously publishes the requested data s
start_support_data_export	This target has been deprecated

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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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)
```

`marketplacemetering` *AWSMarketplace Metering*

Description

Amazon Web Services Marketplace Metering Service

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

Amazon Web Services Marketplace sellers can use this API to submit usage data for custom usage dimensions.

For information about the permissions that you need to use this API, see Amazon Web Services Marketplace metering and entitlement API permissions in the *Amazon Web Services Marketplace Seller Guide.*

Submitting metering records

MeterUsage

- Submits the metering record for an Amazon Web Services Marketplace product.
- Called from: Amazon Elastic Compute Cloud (Amazon EC2) instance or a container running on either Amazon Elastic Kubernetes Service (Amazon EKS) or Amazon Elastic Container Service (Amazon ECS)
- Supported product types: Amazon Machine Images (AMIs) and containers
- Vendor-metered tagging: Supported allocation tagging

BatchMeterUsage

- Submits the metering record for a set of customers. batch_meter_usage API calls are captured by CloudTrail. You can use CloudTrail to verify that the software as a service (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 CloudTrail User Guide.
- Called from: SaaS applications
- Supported product type: SaaS
- Vendor-metered tagging: Supports allocation tagging

Accepting new customers

ResolveCustomer

- Resolves the registration token that the buyer submits through the browser during the registration process. Obtains a CustomerIdentifier along with the CustomerAWSAccountId, ProductCode, and LicenseArn.
- Called from: SaaS application during the registration process
- Supported product type: SaaS
- Vendor-metered tagging: Not applicable

Entitlement and metering for paid container products

RegisterUsage

- Provides software entitlement and metering. Paid container software products sold through Amazon Web Services Marketplace must integrate with the Marketplace Metering Service and call the register_usage operation. Free and Bring Your Own License model (BYOL) products for Amazon ECS or Amazon EKS aren't required to call register_usage. However, you can do so if you want to receive usage data in your seller reports. For more information about using the register_usage operation, see Container-based products.
- Called from: Paid container software products
- Supported product type: Containers
- Vendor-metered tagging: Not applicable

Entitlement custom metering for container products

- MeterUsage API is available in GovCloud Regions but only supports AMI FCP products in GovCloud Regions. Flexible Consumption Pricing (FCP) Container products aren't supported in GovCloud Regions: us-gov-west-1 and us-gov-east-1. For more information, see Container-based products.
- Custom metering for container products are called using the MeterUsage API. The API is used for FCP AMI and FCP Container product metering.

Custom metering for Amazon EKS is available in 17 Amazon Web Services Regions

- The metering service supports Amazon ECS and EKS for Flexible Consumption Pricing (FCP) products using MeterUsage API. Amazon ECS is supported in all Amazon Web Services Regions that MeterUsage API is available except for GovCloud.

- Amazon EKS is supported in the following: us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1, eu-central-1, eu-west-2, eu-west-3, eu-north-1, ap-east-1, ap-southeast-1, ap-northeast-1, ap-southeast-2, ap-northeast-2, ap-south-1, ca-central-1, sa-east-1.

  For questions about adding Amazon Web Services Regions for metering, contact Amazon Web Services Marketplace Seller Operations.

Usage

```
marketplacemetering(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_meter_usage	Amazon Web Services Marketplace is introducing Concurrent Agreements, enabling buyers to make mul
meter_usage	As a seller, your software hosted in the buyer's Amazon Web Services account uses this API action to em
register_usage	Paid container software products sold through Amazon Web Services Marketplace must integrate with th
resolve_customer	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)
```

`paymentcryptographycontrolplane`

Payment Cryptography Control Plane

Description

Amazon Web Services Payment Cryptography Control Plane APIs manage encryption keys for use during payment-related cryptographic operations. You can create, import, export, share, manage, and delete keys. You can also manage Identity and Access Management (IAM) policies for keys. For more information, see Identity and access management in the *Amazon Web Services Payment Cryptography User Guide.*

To use encryption keys for payment-related transaction processing and associated cryptographic operations, you use the Amazon Web Services Payment Cryptography Data Plane. You can perform actions like encrypt, decrypt, generate, and verify payment-related data.

All Amazon Web Services Payment Cryptography API calls must be signed and transmitted using Transport Layer Security (TLS). We recommend you always use the latest supported TLS version for logging API requests.

Amazon Web Services Payment Cryptography supports CloudTrail for control plane operations, a service that logs Amazon Web Services API calls and related events for your Amazon Web Services account and delivers them to an Amazon S3 bucket you specify. By using the information collected by CloudTrail, you can determine what requests were made to Amazon Web Services Payment Cryptography, who made the request, when it was made, and so on. If you don't configure a trail, you can still view the most recent events in the CloudTrail console. For more information, see the CloudTrail User Guide.

Usage

```
paymentcryptographycontrolplane(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- paymentcryptographycontrolplane(
  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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

add_key_replication_regions	Adds replication Amazon Web Services Regions to an existing Amazon Web Servi
associate_mpa_team	Associates a Multi-Party Approval (MPA) team with a protected operation
create_alias	Creates an alias, or a friendly name, for an Amazon Web Services Payment Crypto
create_key	Creates an Amazon Web Services Payment Cryptography key, a logical representat
delete_alias	Deletes the alias, but doesn't affect the underlying key
delete_key	Deletes the key material and metadata associated with Amazon Web Services Payn
delete_resource_policy	Removes the resource-based policy attached to an Amazon Web Services Payment

disable_default_key_replication_regions	Disables Multi-Region key replication settings for the specified Amazon Web Serv
disassociate_mpa_team	Removes the association between a Multi-Party Approval (MPA) team and a protec
enable_default_key_replication_regions	Enables Multi-Region key replication settings for your Amazon Web Services acco
export_key	Exports a key from Amazon Web Services Payment Cryptography
get_alias	Gets the Amazon Web Services Payment Cryptography key associated with the alia
get_certificate_signing_request	Creates a certificate signing request (CSR) from a key pair
get_default_key_replication_regions	Retrieves the list of Amazon Web Services Regions where Multi-Region key replic
get_key	Gets the key metadata for an Amazon Web Services Payment Cryptography key, in
get_mpa_team_association	Returns the Multi-Party Approval (MPA) team association for a protected operatio
get_parameters_for_export	Gets the export token and the signing key certificate to initiate a TR-34 key export
get_parameters_for_import	Gets the import token and the wrapping key certificate in PEM format (base64 enc
get_public_key_certificate	Gets the public key certificate of the asymmetric key pair that exists within Amazo
get_resource_policy	Returns the resource-based policy attached to an Amazon Web Services Payment C
import_key	Imports symmetric keys and public key certificates in PEM format (base64 encode
list_aliases	Lists the aliases for all keys in the caller's Amazon Web Services account and Ama
list_keys	Lists the keys in the caller's Amazon Web Services account and Amazon Web Serv
list_tags_for_resource	Lists the tags for an Amazon Web Services resource
put_resource_policy	Attaches or replaces a resource-based policy on an Amazon Web Services Payment
remove_key_replication_regions	Removes Replication Regions from an existing Amazon Web Services Payment Cr
restore_key	Cancels a scheduled key deletion during the waiting period
start_key_usage	Enables an Amazon Web Services Payment Cryptography key, which makes it acti
stop_key_usage	Disables an Amazon Web Services Payment Cryptography key, which makes it ina
tag_resource	Adds or edits tags on an Amazon Web Services Payment Cryptography key
untag_resource	Deletes a tag from an Amazon Web Services Payment Cryptography key
update_alias	Associates an existing Amazon Web Services Payment Cryptography alias with a c

Examples

```
## Not run:
svc <- paymentcryptographycontrolplane()
svc$add_key_replication_regions(
  Foo = 123
)

## End(Not run)
```

`paymentcryptographydataplane`

Payment Cryptography Data Plane

Description

You use the Amazon Web Services Payment Cryptography Data Plane to manage how encryption keys are used for payment-related transaction processing and associated cryptographic operations.

You can encrypt, decrypt, generate, verify, and translate payment-related cryptographic operations in Amazon Web Services Payment Cryptography. For more information, see Data operations in the *Amazon Web Services Payment Cryptography User Guide*.

To manage your encryption keys, you use the Amazon Web Services Payment Cryptography Control Plane. You can create, import, export, share, manage, and delete keys. You can also manage Identity and Access Management (IAM) policies for keys.

Usage

```
paymentcryptographydataplane(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html`
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- paymentcryptographydataplane(
  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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

decrypt_data	Decrypts ciphertext data to plaintext using a symmetric (TDES, AES), asymmetric (RSA
encrypt_data	Encrypts plaintext data to ciphertext using a symmetric (TDES, AES), asymmetric (RSA
generate_as_2805_kek_validation	Generates a KekValidationRequest or a KekValidationResponse for node-to-node initiali
generate_auth_request_cryptogram	Generates an Authorization Request Cryptogram (ARQC) for an EMV chip payment ca
generate_card_validation_data	Generates card-related validation data using algorithms such as Card Verification Values
generate_mac	Generates a Message Authentication Code (MAC) cryptogram within Amazon Web Ser
generate_mac_emv_pin_change	Generates an issuer script mac for EMV payment cards that use offline PINs as the cardl
generate_pin_data	Generates pin-related data such as PIN, PIN Verification Value (PVV), PIN Block, and I
re_encrypt_data	Re-encrypts ciphertext using DUKPT, symmetric, or asymmetric data encryption keys
translate_key_material	Translates an cryptographic key between different wrapping keys without importing the
translate_pin_data	Translates encrypted PIN block from and to ISO 9564 formats 0,1,3,4
verify_auth_request_cryptogram	Verifies Authorization Request Cryptogram (ARQC) for a EMV chip payment card auth
verify_card_validation_data	Verifies card-related validation data using algorithms such as Card Verification Values (C
verify_mac	Verifies a Message Authentication Code (MAC)
verify_pin_data	Verifies pin-related data such as PIN and PIN Offset using algorithms including VISA P

Examples

```
## Not run:
svc <- paymentcryptographydataplane()
svc$decrypt_data(
  Foo = 123
)

## End(Not run)
```

`pricing` *AWS Price List Service*

Description

The 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 do the following:

- Build cost control and scenario planning tools
- Reconcile billing data
- Forecast future spend for budgeting purposes
- 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 Amazon Web Services 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.

For more information, see Using the Amazon Web Services Price List API in the *Billing User Guide*.

Usage

```
pricing(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.

credentials — Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

describe_services	Returns the metadata for one service or a list of the metadata for all services
get_attribute_values	Returns a list of attribute values

get_price_list_file_url	This feature is in preview release and is subject to change
get_products	Returns a list of all products that match the filter criteria
list_price_lists	This feature is in preview release and is subject to change

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 Amazon Web Services usage (for example, on Amazon EC2 instances). You commit to a consistent amount of usage per hour, in the specified currency, for a term of one or three years, and receive a lower price for that usage. For more information, see the Amazon Web Services Savings Plans User Guide.

Usage

```
savingsplans(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **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`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **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` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_savings_plan	Creates a Savings Plan
delete_queued_savings_plan	Deletes the queued purchase for the specified Savings Plan
describe_savings_plan_rates	Describes the rates for a specific, existing Savings Plan
describe_savings_plans	Describes the specified Savings Plans
describe_savings_plans_offering_rates	Describes the offering rates for Savings Plans you might want to purchase
describe_savings_plans_offerings	Describes the offerings for the specified Savings Plans
list_tags_for_resource	Lists the tags for the specified resource
return_savings_plan	Returns the specified Savings Plan
tag_resource	Adds the specified tags to the specified resource
untag_resource	Removes the specified tags from the specified resource

Examples

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

## End(Not run)
```

Index

add_key_replication_regions, *32*
associate_accounts, *8*
associate_mpa_team, *32*
associate_pricing_rules, *8*
associate_source_views, *5*

batch_associate_resources_to_custom_line_item, *8*
batch_describe_entities, *20*
batch_disassociate_resources_from_custom_line_item, *8*
batch_meter_usage, *27*, *30*
billing, 3
billingconductor, 5
budgets, 9

cancel_change_set, *20*
costandusagereportservice, 12
costexplorer, 15
create_alias, *32*
create_anomaly_monitor, *17*
create_anomaly_subscription, *17*
create_billing_group, *8*
create_billing_view, *5*
create_budget, *11*
create_budget_action, *11*
create_cost_category_definition, *17*
create_custom_line_item, *8*
create_key, *32*
create_notification, *11*
create_pricing_plan, *8*
create_pricing_rule, *8*
create_savings_plan, *41*
create_subscriber, *11*

decrypt_data, *36*
delete_alias, *32*
delete_anomaly_monitor, *17*
delete_anomaly_subscription, *17*
delete_billing_group, *8*
delete_billing_view, *5*
delete_budget, *11*
delete_budget_action, *11*
delete_cost_category_definition, *17*
delete_custom_line_item, *8*
delete_key, *32*
delete_notification, *11*
delete_pricing_plan, *8*
delete_pricing_rule, *8*
delete_queued_savings_plan, *41*
delete_report_definition, *14*
delete_resource_policy, *20*, *32*
delete_subscriber, *11*
describe_assessment, *20*
describe_budget, *11*
describe_budget_action, *11*
describe_budget_action_histories, *11*
describe_budget_actions_for_account, *11*
describe_budget_actions_for_budget, *11*
describe_budget_notifications_for_account, *11*
describe_budget_performance_history, *11*
describe_budgets, *11*
describe_change_set, *20*
describe_cost_category_definition, *17*
describe_entity, *20*
describe_notifications_for_budget, *11*
describe_report_definitions, *14*
describe_savings_plan_rates, *41*
describe_savings_plans, *41*
describe_savings_plans_offering_rates, *41*
describe_savings_plans_offerings, *41*
describe_services, *38*
describe_subscribers_for_notification, *11*
disable_default_key_replication_regions,

33
disassociate_accounts, *8*
disassociate_mpa_team, *33*
disassociate_pricing_rules, *8*
disassociate_source_views, *5*

enable_default_key_replication_regions, *33*
encrypt_data, *36*
execute_budget_action, *11*
export_key, *33*

generate_as_2805_kek_validation, *36*
generate_auth_request_cryptogram, *36*
generate_card_validation_data, *36*
generate_data_set, *23*
generate_mac, *36*
generate_mac_emv_pin_change, *36*
generate_pin_data, *36*
get_alias, *33*
get_anomalies, *17*
get_anomaly_monitors, *17*
get_anomaly_subscriptions, *17*
get_approximate_usage_records, *17*
get_attribute_values, *37, 38*
get_billing_group_cost_report, *8*
get_billing_preferences, *5*
get_billing_transfer_preference, *8*
get_billing_view, *5*
get_certificate_signing_request, *33*
get_commitment_purchase_analysis, *17*
get_cost_and_usage, *17*
get_cost_and_usage_comparisons, *17*
get_cost_and_usage_with_resources, *17*
get_cost_categories, *17*
get_cost_comparison_drivers, *17*
get_cost_forecast, *17*
get_credit_allocation_history, *5*
get_credits, *5*
get_default_key_replication_regions, *33*
get_dimension_values, *17*
get_enterprise_support_charge_summary, *5*
get_enterprise_support_contract_details, *5*
get_entitlements, *26*
get_key, *33*
get_mpa_team_association, *33*
get_parameters_for_export, *33*
get_parameters_for_import, *33*
get_price_list_file_url, *39*
get_products, *37*, *39*
get_public_key_certificate, *33*
get_reservation_coverage, *17*
get_reservation_purchase_recommendation, *17*
get_reservation_utilization, *17*
get_resource_policy, *5*, *20*, *33*
get_rightsizing_recommendation, *17*
get_savings_plan_purchase_recommendation_details, *17*
get_savings_plans_coverage, *17*
get_savings_plans_purchase_recommendation, *17*
get_savings_plans_utilization, *17*
get_savings_plans_utilization_details, *18*
get_tags, *18*
get_usage_forecast, *18*

import_key, *33*

list_account_associations, *8*
list_aliases, *33*
list_assessments, *21*
list_billing_group_cost_reports, *8*
list_billing_groups, *8*
list_billing_view_segments, *5*
list_billing_views, *5*
list_business_support_account_charges, *5*
list_business_support_subscription_history, *5*
list_change_sets, *21*
list_commitment_purchase_analyses, *18*
list_cost_allocation_tag_backfill_history, *18*
list_cost_allocation_tags, *18*
list_cost_category_definitions, *18*
list_cost_category_resource_associations, *18*
list_custom_line_item_versions, *8*
list_custom_line_items, *8*
list_enterprise_support_linked_account_charges, *5*
list_entities, *21*
list_keys, *33*

list_price_lists, *39*
list_pricing_plans, *8*
list_pricing_plans_associated_with_pricing_rule, *8*
list_pricing_rules, *8*
list_pricing_rules_associated_to_pricing_plan, *8*
list_resources_associated_to_custom_line_item, *8*
list_savings_plans_purchase_recommendation_generation, *18*
list_source_views_for_billing_view, *5*
list_tags_for_resource, *5*, *8*, *11*, *14*, *18*, *21*, *33*, *41*

marketplacecatalog, 18
marketplacecommerceanalytics, 21
marketplaceentitlementservice, 23
marketplacemetering, 26
meter_usage, *30*
modify_report_definition, *14*

paymentcryptographycontrolplane, 30
paymentcryptographydataplane, 33
pricing, 36
provide_anomaly_feedback, *18*
put_report_definition, *14*
put_resource_policy, *21*, *33*

re_encrypt_data, *36*
redeem_credits, *5*
register_usage, *27*, *30*
remove_key_replication_regions, *33*
resolve_customer, *30*
restore_key, *33*
return_savings_plan, *41*

savingsplans, 39
start_change_set, *21*
start_commitment_purchase_analysis, *18*
start_cost_allocation_tag_backfill, *18*
start_key_usage, *33*
start_savings_plans_purchase_recommendation_generation, *18*
start_support_data_export, *23*
stop_key_usage, *33*

tag_resource, *5*, *8*, *11*, *14*, *18*, *21*, *33*, *41*
translate_key_material, *36*
translate_pin_data, *36*

untag_resource, *5*, *8*, *11*, *14*, *18*, *21*, *33*, *41*
update_alias, *33*
update_anomaly_monitor, *18*
update_anomaly_subscription, *18*
update_billing_group, *8*
update_billing_preferences, *5*
update_billing_transfer_preference, *8*
update_billing_view, *5*
update_budget, *11*
update_budget_action, *12*
update_cost_allocation_tags_status, *18*
update_cost_category_definition, *18*
update_custom_line_item, *8*
update_notification, *12*
update_pricing_plan, *8*
update_pricing_rule, *8*
update_subscriber, *12*

verify_auth_request_cryptogram, *36*
verify_card_validation_data, *36*
verify_mac, *36*
verify_pin_data, *36*