

Package ‘paws.application.integration’

May 9, 2026

Title 'Amazon Web Services' Application Integration Services

Version 0.9.0

Description Interface to 'Amazon Web Services' application integration services, including 'Simple Queue Service' ('SQS') message queue, 'Simple Notification Service' ('SNS') publish/subscribe messaging, 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.application.integration>

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

Imports paws.common (>= 0.8.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.2

Collate 'eventbridge_service.R' 'eventbridge_interfaces.R'
'eventbridge_operations.R' 'eventbridgepipes_service.R'
'eventbridgepipes_interfaces.R' 'eventbridgepipes_operations.R'
'eventbridgescheduler_service.R'
'eventbridgescheduler_interfaces.R'
'eventbridgescheduler_operations.R' 'locationservice_service.R'
'locationservice_interfaces.R' 'locationservice_operations.R'
'mq_service.R' 'mq_interfaces.R' 'mq_operations.R'
'mwaa_service.R' 'mwaa_interfaces.R' 'mwaa_operations.R'
'reexports_paws.common.R' 'resourceexplorer_service.R'
'resourceexplorer_interfaces.R' 'resourceexplorer_operations.R'
'schemas_service.R' 'schemas_interfaces.R'
'schemas_operations.R' 'sfn_service.R' 'sfn_interfaces.R'
'sfn_operations.R' 'sns_service.R' 'sns_interfaces.R'
'sns_operations.R' 'sqs_service.R' 'sqs_interfaces.R'
'sqs_operations.R' 'swf_service.R' 'swf_interfaces.R'
'swf_operations.R'

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-03-14 12:20:02 UTC

Contents

| | |
|--------------------------------|----|
| eventbridge | 2 |
| eventbridgepipes | 6 |
| eventbridgescheduler | 8 |
| locationservice | 11 |
| mq | 14 |
| mwaa | 17 |
| resourceexplorer | 20 |
| schemas | 23 |
| sfn | 26 |
| sns | 29 |
| sqs | 32 |
| swf | 35 |

Index **39**

| | |
|-------------|---------------------------|
| eventbridge | <i>Amazon EventBridge</i> |
|-------------|---------------------------|

Description

Amazon EventBridge helps you to respond to state changes in your Amazon Web Services resources. When your resources change state, they automatically send events to an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a predetermined schedule. For example, you can configure rules to:

- Automatically invoke an Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the running state.
- Direct specific API records from CloudTrail to an Amazon Kinesis data stream for detailed analysis of potential security or availability risks.
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume.

For more information about the features of Amazon EventBridge, see the [Amazon EventBridge User Guide](#).

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- eventbridge(
  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"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

| | |
|---|--|
| activate_event_source | Activates a partner event source that has been deactivated |
| cancel_replay | Cancels the specified replay |
| create_api_destination | Creates an API destination, which is an HTTP invocation endpoint configured as a target |
| create_archive | Creates an archive of events with the specified settings |
| create_connection | Creates a connection |
| create_endpoint | Creates a global endpoint |
| create_event_bus | Creates a new event bus within your account |
| create_partner_event_source | Called by an SaaS partner to create a partner event source |
| deactivate_event_source | You can use this operation to temporarily stop receiving events from the specified partner |
| deauthorize_connection | Removes all authorization parameters from the connection |
| delete_api_destination | Deletes the specified API destination |
| delete_archive | Deletes the specified archive |
| delete_connection | Deletes a connection |

| | |
|---|--|
| <code>delete_endpoint</code> | Delete an existing global endpoint |
| <code>delete_event_bus</code> | Deletes the specified custom event bus or partner event bus |
| <code>delete_partner_event_source</code> | This operation is used by SaaS partners to delete a partner event source |
| <code>delete_rule</code> | Deletes the specified rule |
| <code>describe_api_destination</code> | Retrieves details about an API destination |
| <code>describe_archive</code> | Retrieves details about an archive |
| <code>describe_connection</code> | Retrieves details about a connection |
| <code>describe_endpoint</code> | Get the information about an existing global endpoint |
| <code>describe_event_bus</code> | Displays details about an event bus in your account |
| <code>describe_event_source</code> | This operation lists details about a partner event source that is shared with your account |
| <code>describe_partner_event_source</code> | An SaaS partner can use this operation to list details about a partner event source that the |
| <code>describe_replay</code> | Retrieves details about a replay |
| <code>describe_rule</code> | Describes the specified rule |
| <code>disable_rule</code> | Disables the specified rule |
| <code>enable_rule</code> | Enables the specified rule |
| <code>list_api_destinations</code> | Retrieves a list of API destination in the account in the current Region |
| <code>list_archives</code> | Lists your archives |
| <code>list_connections</code> | Retrieves a list of connections from the account |
| <code>list_endpoints</code> | List the global endpoints associated with this account |
| <code>list_event_buses</code> | Lists all the event buses in your account, including the default event bus, custom event b |
| <code>list_event_sources</code> | You can use this to see all the partner event sources that have been shared with your Am |
| <code>list_partner_event_source_accounts</code> | An SaaS partner can use this operation to display the Amazon Web Services account ID |
| <code>list_partner_event_sources</code> | An SaaS partner can use this operation to list all the partner event source names that the |
| <code>list_replays</code> | Lists your replays |
| <code>list_rule_names_by_target</code> | Lists the rules for the specified target |
| <code>list_rules</code> | Lists your Amazon EventBridge rules |
| <code>list_tags_for_resource</code> | Displays the tags associated with an EventBridge resource |
| <code>list_targets_by_rule</code> | Lists the targets assigned to the specified rule |
| <code>put_events</code> | Sends custom events to Amazon EventBridge so that they can be matched to rules |
| <code>put_partner_events</code> | This is used by SaaS partners to write events to a customer's partner event bus |
| <code>put_permission</code> | Running PutPermission permits the specified Amazon Web Services account or Amazon |
| <code>put_rule</code> | Creates or updates the specified rule |
| <code>put_targets</code> | Adds the specified targets to the specified rule, or updates the targets if they are already |
| <code>remove_permission</code> | Revokes the permission of another Amazon Web Services account to be able to put even |
| <code>remove_targets</code> | Removes the specified targets from the specified rule |
| <code>start_replay</code> | Starts the specified replay |
| <code>tag_resource</code> | Assigns one or more tags (key-value pairs) to the specified EventBridge resource |
| <code>test_event_pattern</code> | Tests whether the specified event pattern matches the provided event |
| <code>untag_resource</code> | Removes one or more tags from the specified EventBridge resource |
| <code>update_api_destination</code> | Updates an API destination |
| <code>update_archive</code> | Updates the specified archive |
| <code>update_connection</code> | Updates settings for a connection |
| <code>update_endpoint</code> | Update an existing endpoint |
| <code>update_event_bus</code> | Updates the specified event bus |

Examples

```
## Not run:
svc <- eventbridge()
svc$activate_event_source(
  Foo = 123
)

## End(Not run)
```

eventbridgepipes

Amazon EventBridge Pipes

Description

Amazon EventBridge Pipes connects event sources to targets. Pipes reduces the need for specialized knowledge and integration code when developing event driven architectures. This helps ensure consistency across your company's applications. With Pipes, the target can be any available EventBridge target. To set up a pipe, you select the event source, add optional event filtering, define optional enrichment, and select the target for the event data.

Usage

```
eventbridgepipes(
  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.

| | |
|-------------|---|
| | <ul style="list-style-type: none"> • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- eventbridgepipes(
  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"
  ),
  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_pipe | Create a pipe |
| delete_pipe | Delete an existing pipe |
| describe_pipe | Get the information about an existing pipe |
| list_pipes | Get the pipes associated with this account |
| list_tags_for_resource | Displays the tags associated with a pipe |
| start_pipe | Start an existing pipe |
| stop_pipe | Stop an existing pipe |
| tag_resource | Assigns one or more tags (key-value pairs) to the specified pipe |
| untag_resource | Removes one or more tags from the specified pipes |
| update_pipe | Update an existing pipe |

Examples

```

## Not run:
svc <- eventbridgepipes()
svc$create_pipe(
  Foo = 123
)

## End(Not run)

```

eventbridgescheduler *Amazon EventBridge Scheduler*

Description

Amazon EventBridge Scheduler is a serverless scheduler that allows you to create, run, and manage tasks from one central, managed service. EventBridge Scheduler delivers your tasks reliably, with built-in mechanisms that adjust your schedules based on the availability of downstream targets. The following reference lists the available API actions, and data types for EventBridge Scheduler.

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- eventbridgescheduler(
  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"
  ),
  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_schedule | Creates the specified schedule |
| create_schedule_group | Creates the specified schedule group |
| delete_schedule | Deletes the specified schedule |
| delete_schedule_group | Deletes the specified schedule group |
| get_schedule | Retrieves the specified schedule |
| get_schedule_group | Retrieves the specified schedule group |
| list_schedule_groups | Returns a paginated list of your schedule groups |
| list_schedules | Returns a paginated list of your EventBridge Scheduler schedules |
| list_tags_for_resource | Lists the tags associated with the Scheduler resource |
| tag_resource | Assigns one or more tags (key-value pairs) to the specified EventBridge Scheduler resource |
| untag_resource | Removes one or more tags from the specified EventBridge Scheduler schedule group |
| update_schedule | Updates the specified schedule |

Examples

```
## Not run:
svc <- eventbridgescheduler()
svc$create_schedule(
  Foo = 123
)

## End(Not run)
```

| | |
|-----------------|--------------------------------|
| locationservice | <i>Amazon Location Service</i> |
|-----------------|--------------------------------|

Description

"Suite of geospatial services including Maps, Places, Routes, Tracking, and Geofencing"

Usage

```
locationservice(
  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`.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- locationservice(
  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"
  ),
  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_tracker_consumer | Creates an association between a geofence collection and a tracker resource |
| batch_delete_device_position_history | Deletes the position history of one or more devices from a tracker resource |
| batch_delete_geofence | Deletes a batch of geofences from a geofence collection |
| batch_evaluate_geofences | Evaluates device positions against the geofence geometries from a given geofence collection |
| batch_get_device_position | Lists the latest device positions for requested devices |
| batch_put_geofence | A batch request for storing geofence geometries into a given geofence collection, or updating existing geofences |
| batch_update_device_position | Uploads position update data for one or more devices to a tracker resource (up to 10 devices per request) |
| calculate_route | Calculates a route given the following required parameters: DeparturePosition and DestinationPosition |
| calculate_route_matrix | Calculates a route matrix given the following required parameters: DeparturePosition and DestinationPosition |
| create_geofence_collection | Creates a geofence collection, which manages and stores geofences |
| create_key | Creates an API key resource in your Amazon Web Services account, which lets you grant access to your Amazon Web Services account |
| create_map | Creates a map resource in your Amazon Web Services account, which provides map tiles and labels |
| create_place_index | Creates a place index resource in your Amazon Web Services account |
| create_route_calculator | Creates a route calculator resource in your Amazon Web Services account |
| create_tracker | Creates a tracker resource in your Amazon Web Services account, which lets you retrieve and update device positions |
| delete_geofence_collection | Deletes a geofence collection from your Amazon Web Services account |
| delete_key | Deletes the specified API key |
| delete_map | Deletes a map resource from your Amazon Web Services account |
| delete_place_index | Deletes a place index resource from your Amazon Web Services account |
| delete_route_calculator | Deletes a route calculator resource from your Amazon Web Services account |
| delete_tracker | Deletes a tracker resource from your Amazon Web Services account |
| describe_geofence_collection | Retrieves the geofence collection details |
| describe_key | Retrieves the API key resource details |
| describe_map | Retrieves the map resource details |
| describe_place_index | Retrieves the place index resource details |
| describe_route_calculator | Retrieves the route calculator resource details |
| describe_tracker | Retrieves the tracker resource details |
| disassociate_tracker_consumer | Removes the association between a tracker resource and a geofence collection |
| forecast_geofence_events | Evaluates device positions against geofence geometries from a given geofence collection |
| get_device_position | Retrieves a device's most recent position according to its sample time |
| get_device_position_history | Retrieves the device position history from a tracker resource within a specified range |
| get_geofence | Retrieves the geofence details from a geofence collection |
| get_map_glyphs | Retrieves glyphs used to display labels on a map |
| get_map_sprites | Retrieves the sprite sheet corresponding to a map resource |
| get_map_style_descriptor | Retrieves the map style descriptor from a map resource |
| get_map_tile | Retrieves a vector data tile from the map resource |
| get_place | Finds a place by its unique ID |
| list_device_positions | A batch request to retrieve all device positions |
| list_geofence_collections | Lists geofence collections in your Amazon Web Services account |
| list_geofences | Lists geofences stored in a given geofence collection |

| | |
|--|---|
| list_keys | Lists API key resources in your Amazon Web Services account |
| list_maps | Lists map resources in your Amazon Web Services account |
| list_place_indexes | Lists place index resources in your Amazon Web Services account |
| list_route_calculators | Lists route calculator resources in your Amazon Web Services account |
| list_tags_for_resource | Returns a list of tags that are applied to the specified Amazon Location resource |
| list_tracker_consumers | Lists geofence collections currently associated to the given tracker resource |
| list_trackers | Lists tracker resources in your Amazon Web Services account |
| put_geofence | Stores a geofence geometry in a given geofence collection, or updates the geometry of a geofence collection |
| search_place_index_for_position | Reverse geocodes a given coordinate and returns a legible address |
| search_place_index_for_suggestions | Generates suggestions for addresses and points of interest based on partial or misspelled text |
| search_place_index_for_text | Geocodes free-form text, such as an address, name, city, or region to allow you to search for a location |
| tag_resource | Assigns one or more tags (key-value pairs) to the specified Amazon Location Service resource |
| untag_resource | Removes one or more tags from the specified Amazon Location resource |
| update_geofence_collection | Updates the specified properties of a given geofence collection |
| update_key | Updates the specified properties of a given API key resource |
| update_map | Updates the specified properties of a given map resource |
| update_place_index | Updates the specified properties of a given place index resource |
| update_route_calculator | Updates the specified properties for a given route calculator resource |
| update_tracker | Updates the specified properties of a given tracker resource |
| verify_device_position | Verifies the integrity of the device's position by determining if it was reported behind a geofence |

Examples

```
## Not run:
svc <- locationservice()
svc$associate_tracker_consumer(
  Foo = 123
)

## End(Not run)
```

mq

AmazonMQ

Description

Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ that makes it easy to set up and operate message brokers in the cloud. A message broker allows software applications and components to communicate using various programming languages, operating systems, and formal messaging protocols.

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- mq(
  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"
),
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_broker | Creates a broker |
| create_configuration | Creates a new configuration for the specified configuration name |
| create_tags | Add a tag to a resource |
| create_user | Creates an ActiveMQ user |
| delete_broker | Deletes a broker |
| delete_tags | Removes a tag from a resource |
| delete_user | Deletes an ActiveMQ user |
| describe_broker | Returns information about the specified broker |
| describe_broker_engine_types | Describe available engine types and versions |
| describe_broker_instance_options | Describe available broker instance options |
| describe_configuration | Returns information about the specified configuration |
| describe_configuration_revision | Returns the specified configuration revision for the specified configuration |
| describe_user | Returns information about an ActiveMQ user |
| list_brokers | Returns a list of all brokers |
| list_configuration_revisions | Returns a list of all revisions for the specified configuration |
| list_configurations | Returns a list of all configurations |
| list_tags | Lists tags for a resource |
| list_users | Returns a list of all ActiveMQ users |
| promote | Promotes a data replication replica broker to the primary broker role |
| reboot_broker | Reboots a broker |

| | |
|--------------------------------------|---|
| update_broker | Adds a pending configuration change to a broker |
| update_configuration | Updates the specified configuration |
| update_user | Updates the information for an ActiveMQ user |

Examples

```
## Not run:
svc <- mq()
svc$create_broker(
  Foo = 123
)

## End(Not run)
```

| | |
|------|-------------------|
| mwaa | <i>AmazonMWAA</i> |
|------|-------------------|

Description

Amazon Managed Workflows for Apache Airflow

This section contains the Amazon Managed Workflows for Apache Airflow (MWAA) API reference documentation. For more information, see [What is Amazon MWAA?](#).

Endpoints

- `api.airflow.{region}.amazonaws.com` - This endpoint is used for environment management.
 - [create_environment](#)
 - [delete_environment](#)
 - [get_environment](#)
 - [list_environments](#)
 - [list_tags_for_resource](#)
 - [tag_resource](#)
 - [untag_resource](#)
 - [update_environment](#)
- `env.airflow.{region}.amazonaws.com` - This endpoint is used to operate the Airflow environment.
 - [create_cli_token](#)
 - [create_web_login_token](#)
 - [invoke_rest_api](#)

Regions

For a list of supported regions, see [Amazon MWAA endpoints and quotas](#) in the *Amazon Web Services General Reference*.

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- mwaa(
  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"
  ),
  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_cli_token | Creates a CLI token for the Airflow CLI |
| create_environment | Creates an Amazon Managed Workflows for Apache Airflow (Amazon MWAA) environment |
| create_web_login_token | Creates a web login token for the Airflow Web UI |
| delete_environment | Deletes an Amazon Managed Workflows for Apache Airflow (Amazon MWAA) environment |
| get_environment | Describes an Amazon Managed Workflows for Apache Airflow (MWAA) environment |
| invoke_rest_api | Invokes the Apache Airflow REST API on the webserver with the specified inputs |
| list_environments | Lists the Amazon Managed Workflows for Apache Airflow (MWAA) environments |
| list_tags_for_resource | Lists the key-value tag pairs associated to the Amazon Managed Workflows for Apache Airflow (M |
| publish_metrics | Internal only |
| tag_resource | Associates key-value tag pairs to your Amazon Managed Workflows for Apache Airflow (MWAA) |
| untag_resource | Removes key-value tag pairs associated to your Amazon Managed Workflows for Apache Airflow |
| update_environment | Updates an Amazon Managed Workflows for Apache Airflow (MWAA) environment |

Examples

```
## Not run:
svc <- mwa()
svc$create_cli_token(
  Foo = 123
)

## End(Not run)
```

resourceexplorer

AWS Resource Explorer

Description

Amazon Web Services Resource Explorer is a resource search and discovery service. By using Resource Explorer, you can explore your resources using an internet search engine-like experience. Examples of resources include Amazon Relational Database Service (Amazon RDS) instances, Amazon Simple Storage Service (Amazon S3) buckets, or Amazon DynamoDB tables. You can search for your resources using resource metadata like names, tags, and IDs. Resource Explorer can search across all of the Amazon Web Services Regions in your account in which you turn the service on, to simplify your cross-Region workloads.

Resource Explorer scans the resources in each of the Amazon Web Services Regions in your Amazon Web Services account in which you turn on Resource Explorer. Resource Explorer **creates and maintains an index** in each Region, with the details of that Region's resources.

You can **search across all of the indexed Regions in your account** by designating one of your Amazon Web Services Regions to contain the aggregator index for the account. When you **promote a local index in a Region to become the aggregator index for the account**, Resource Explorer automatically replicates the index information from all local indexes in the other Regions to the aggregator index. Therefore, the Region with the aggregator index has a copy of all resource information for all Regions in the account where you turned on Resource Explorer. As a result, views in the aggregator index Region include resources from all of the indexed Regions in your account.

For more information about Amazon Web Services Resource Explorer, including how to enable and configure the service, see the [Amazon Web Services Resource Explorer User Guide](#).

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- resourceexplorer(
  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"
),
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_default_view | Sets the specified view as the default for the Amazon Web Services Region in which you are currently logged in. |
| batch_get_view | Retrieves details about a list of views. |
| create_index | Turns on Amazon Web Services Resource Explorer in the Amazon Web Services Region. |
| create_view | Creates a view that users can query by using the Search operation. |
| delete_index | Deletes the specified index and turns off Amazon Web Services Resource Explorer. |
| delete_view | Deletes the specified view. |
| disassociate_default_view | After you call this operation, the affected Amazon Web Services Region no longer has a default view. |
| get_account_level_service_configuration | Retrieves the status of your account's Amazon Web Services service access, and whether Amazon Web Services Resource Explorer is turned on. |
| get_default_view | Retrieves the Amazon Resource Name (ARN) of the view that is the default for the Amazon Web Services Region. |
| get_index | Retrieves details about the Amazon Web Services Resource Explorer index in the Amazon Web Services Region. |
| get_managed_view | Retrieves details of the specified Amazon Web Services-managed view. |
| get_view | Retrieves details of the specified view. |
| list_indexes | Retrieves a list of all of the indexes in Amazon Web Services Regions that are currently turned on. |
| list_indexes_for_members | Retrieves a list of a member's indexes in all Amazon Web Services Regions that are currently turned on. |
| list_managed_views | Lists the Amazon resource names (ARNs) of the Amazon Web Services-managed views. |
| list_resources | Returns a list of resources and their details that match the specified criteria. |
| list_supported_resource_types | Retrieves a list of all resource types currently supported by Amazon Web Services Resource Explorer. |
| list_tags_for_resource | Lists the tags that are attached to the specified resource. |
| list_views | Lists the Amazon resource names (ARNs) of the views available in the Amazon Web Services Region. |
| search | Searches for resources and displays details about all resources that match the specified criteria. |

[tag_resource](#)[untag_resource](#)[update_index_type](#)[update_view](#)

Adds one or more tag key and value pairs to an Amazon Web Services Resource E

Removes one or more tag key and value pairs from an Amazon Web Services Reso

Changes the type of the index from one of the following types to the other

Modifies some of the details of a view

Examples

```
## Not run:
svc <- resourceexplorer()
svc$associate_default_view(
  Foo = 123
)

## End(Not run)
```

schemas

Schemas

Description

Amazon EventBridge Schema Registry

Usage

```
schemas(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.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- schemas(
  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"
  ),
  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_discoverer | Creates a discoverer |
| create_registry | Creates a registry |
| create_schema | Creates a schema definition |
| delete_discoverer | Deletes a discoverer |
| delete_registry | Deletes a Registry |
| delete_resource_policy | Delete the resource-based policy attached to the specified registry |
| delete_schema | Delete a schema definition |
| delete_schema_version | Delete the schema version definition |
| describe_code_binding | Describe the code binding URI |
| describe_discoverer | Describes the discoverer |
| describe_registry | Describes the registry |
| describe_schema | Retrieve the schema definition |
| export_schema | Export schema |
| get_code_binding_source | Get the code binding source URI |
| get_discovered_schema | Get the discovered schema that was generated based on sampled events |
| get_resource_policy | Retrieves the resource-based policy attached to a given registry |
| list_discoverers | List the discoverers |
| list_registries | List the registries |
| list_schemas | List the schemas |
| list_schema_versions | Provides a list of the schema versions and related information |
| list_tags_for_resource | Get tags for resource |
| put_code_binding | Put code binding URI |
| put_resource_policy | The name of the policy |
| search_schemas | Search the schemas |
| start_discoverer | Starts the discoverer |
| stop_discoverer | Stops the discoverer |
| tag_resource | Add tags to a resource |
| untag_resource | Removes tags from a resource |
| update_discoverer | Updates the discoverer |
| update_registry | Updates a registry |
| update_schema | Updates the schema definition |

Examples

```

## Not run:
svc <- schemas()

```

```

svc$create_discoverer(
  Foo = 123
)

## End(Not run)

```

sfn

AWS Step Functions

Description

Step Functions

Step Functions coordinates the components of distributed applications and microservices using visual workflows.

You can use Step Functions to build applications from individual components, each of which performs a discrete function, or *task*, allowing you to scale and change applications quickly. Step Functions provides a console that helps visualize the components of your application as a series of steps. Step Functions automatically triggers and tracks each step, and retries steps when there are errors, so your application executes predictably and in the right order every time. Step Functions logs the state of each step, so you can quickly diagnose and debug any issues.

Step Functions manages operations and underlying infrastructure to ensure your application is available at any scale. You can run tasks on Amazon Web Services, your own servers, or any system that has access to Amazon Web Services. You can access and use Step Functions using the console, the Amazon Web Services SDKs, or an HTTP API. For more information about Step Functions, see the [Step Functions Developer Guide](#).

If you use the Step Functions API actions using Amazon Web Services SDK integrations, make sure the API actions are in camel case and parameter names are in Pascal case. For example, you could use Step Functions API action `startSyncExecution` and specify its parameter as `StateMachineArn`.

Usage

```
sfn(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.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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

```
sfn <- sfn(
  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"
```

```

    ),
    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_activity | Creates an activity |
| create_state_machine | Creates a state machine |
| create_state_machine_alias | Creates an alias for a state machine that points to one or two versions of the same state machine |
| delete_activity | Deletes an activity |
| delete_state_machine | Deletes a state machine |
| delete_state_machine_alias | Deletes a state machine alias |
| delete_state_machine_version | Deletes a state machine version |
| describe_activity | Describes an activity |
| describe_execution | Provides information about a state machine execution, such as the state machine Amazon Resource Name (ARN), the execution role ARN, and the execution progress |
| describe_map_run | Provides information about a Map Run's configuration, progress, and results |
| describe_state_machine | Provides information about a state machine's definition, its IAM role Amazon Resource Name (ARN), and its execution role ARN |
| describe_state_machine_alias | Returns details about a state machine alias |
| describe_state_machine_for_execution | Provides information about a state machine's definition, its execution role ARN, and its execution progress |
| get_activity_task | Used by workers to retrieve a task (with the specified activity ARN) which has been assigned to them |
| get_execution_history | Returns the history of the specified execution as a list of events |
| list_activities | Lists the existing activities |
| list_executions | Lists all executions of a state machine or a Map Run |
| list_map_runs | Lists all Map Runs that were started by a given state machine execution |
| list_state_machine_aliases | Lists aliases for a specified state machine ARN |
| list_state_machines | Lists the existing state machines |
| list_state_machine_versions | Lists versions for the specified state machine Amazon Resource Name (ARN) |
| list_tags_for_resource | List tags for a given resource |
| publish_state_machine_version | Creates a version from the current revision of a state machine |
| redrive_execution | Restarts unsuccessful executions of Standard workflows that didn't complete successfully |
| send_task_failure | Used by activity workers, Task states using the callback pattern, and optionally Task states using the poller pattern |
| send_task_heartbeat | Used by activity workers and Task states using the callback pattern, and optionally Task states using the poller pattern |
| send_task_success | Used by activity workers, Task states using the callback pattern, and optionally Task states using the poller pattern |
| start_execution | Starts a state machine execution |
| start_sync_execution | Starts a Synchronous Express state machine execution |
| stop_execution | Stops an execution |
| tag_resource | Add a tag to a Step Functions resource |
| test_state | Accepts the definition of a single state and executes it |

| | |
|---|---|
| untag_resource | Remove a tag from a Step Functions resource |
| update_map_run | Updates an in-progress Map Run's configuration to include changes to the settings th |
| update_state_machine | Updates an existing state machine by modifying its definition, roleArn, loggingConf |
| update_state_machine_alias | Updates the configuration of an existing state machine alias by modifying its descrip |
| validate_state_machine_definition | Validates the syntax of a state machine definition specified in Amazon States Langua |

Examples

```
## Not run:
svc <- sfn()
svc$create_activity(
  Foo = 123
)

## End(Not run)
```

 sns

Amazon Simple Notification Service

Description

Amazon Simple Notification Service (Amazon SNS) is a web service that enables you to build distributed web-enabled applications. Applications can use Amazon SNS to easily push real-time notification messages to interested subscribers over multiple delivery protocols. For more information about this product see the [Amazon SNS product page](#). For detailed information about Amazon SNS features and their associated API calls, see the [Amazon SNS Developer Guide](#).

For information on the permissions you need to use this API, see [Identity and access management in Amazon SNS](#) in the *Amazon SNS Developer Guide*.

We also provide SDKs that enable you to access Amazon SNS from your preferred programming language. The SDKs contain functionality that automatically takes care of tasks such as: cryptographically signing your service requests, retrying requests, and handling error responses. For a list of available SDKs, go to [Tools for Amazon Web Services](#).

Usage

```
sns(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

| | |
|-------------|---|
| | <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- sns(
  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"
    ),
    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_permission | Adds a statement to a topic's access control policy, granting access for the specified |
| check_if_phone_number_is_opted_out | Accepts a phone number and indicates whether the phone holder has opted out of re |
| confirm_subscription | Verifies an endpoint owner's intent to receive messages by validating the token sent |
| create_platform_application | Creates a platform application object for one of the supported push notification serv |
| create_platform_endpoint | Creates an endpoint for a device and mobile app on one of the supported push notif |
| create_sms_sandbox_phone_number | Adds a destination phone number to an Amazon Web Services account in the SMS |
| create_topic | Creates a topic to which notifications can be published |
| delete_endpoint | Deletes the endpoint for a device and mobile app from Amazon SNS |
| delete_platform_application | Deletes a platform application object for one of the supported push notification serv |
| delete_sms_sandbox_phone_number | Deletes an Amazon Web Services account's verified or pending phone number from |
| delete_topic | Deletes a topic and all its subscriptions |
| get_data_protection_policy | Retrieves the specified inline DataProtectionPolicy document that is stored in the sp |
| get_endpoint_attributes | Retrieves the endpoint attributes for a device on one of the supported push notificati |
| get_platform_application_attributes | Retrieves the attributes of the platform application object for the supported push no |
| get_sms_attributes | Returns the settings for sending SMS messages from your Amazon Web Services a |
| get_sms_sandbox_account_status | Retrieves the SMS sandbox status for the calling Amazon Web Services account in |
| get_subscription_attributes | Returns all of the properties of a subscription |
| get_topic_attributes | Returns all of the properties of a topic |
| list_endpoints_by_platform_application | Lists the endpoints and endpoint attributes for devices in a supported push notificati |
| list_origination_numbers | Lists the calling Amazon Web Services account's dedicated origination numbers an |
| list_phone_numbers_opted_out | Returns a list of phone numbers that are opted out, meaning you cannot send SMS m |
| list_platform_applications | Lists the platform application objects for the supported push notification services, s |
| list_sms_sandbox_phone_numbers | Lists the calling Amazon Web Services account's current verified and pending desti |
| list_subscriptions | Returns a list of the requester's subscriptions |
| list_subscriptions_by_topic | Returns a list of the subscriptions to a specific topic |
| list_tags_for_resource | List all tags added to the specified Amazon SNS topic |
| list_topics | Returns a list of the requester's topics |

| | |
|---|--|
| opt_in_phone_number | Use this request to opt in a phone number that is opted out, which enables you to re |
| publish | Sends a message to an Amazon SNS topic, a text message (SMS message) directly |
| publish_batch | Publishes up to ten messages to the specified topic |
| put_data_protection_policy | Adds or updates an inline policy document that is stored in the specified Amazon S |
| remove_permission | Removes a statement from a topic's access control policy |
| set_endpoint_attributes | Sets the attributes for an endpoint for a device on one of the supported push notifica |
| set_platform_application_attributes | Sets the attributes of the platform application object for the supported push notifica |
| set_sms_attributes | Use this request to set the default settings for sending SMS messages and receiving |
| set_subscription_attributes | Allows a subscription owner to set an attribute of the subscription to a new value |
| set_topic_attributes | Allows a topic owner to set an attribute of the topic to a new value |
| subscribe | Subscribes an endpoint to an Amazon SNS topic |
| tag_resource | Add tags to the specified Amazon SNS topic |
| unsubscribe | Deletes a subscription |
| untag_resource | Remove tags from the specified Amazon SNS topic |
| verify_sms_sandbox_phone_number | Verifies a destination phone number with a one-time password (OTP) for the calling |

Examples

```
## Not run:
svc <- sns()
svc$add_permission(
  Foo = 123
)

## End(Not run)
```

sqs

Amazon Simple Queue Service

Description

Welcome to the *Amazon SQS API Reference*.

Amazon SQS is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components.

For information on the permissions you need to use this API, see [Identity and access management](#) in the *Amazon SQS Developer Guide*.

You can use [Amazon Web Services SDKs](#) to access Amazon SQS using your favorite programming language. The SDKs perform tasks such as the following automatically:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

Additional information

- [Amazon SQS Product Page](#)
- [Amazon SQS Developer Guide](#)
 - [Making API Requests](#)
 - [Amazon SQS Message Attributes](#)
 - [Amazon SQS Dead-Letter Queues](#)
- [Amazon SQS in the *Command Line Interface*](#)
- [Amazon Web Services General Reference](#)
 - [Regions and Endpoints](#)

Usage

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

Arguments

| | |
|-------------|---|
| config | <p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- sqs(
  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"
  ),
  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_permission | Adds a permission to a queue for a specific principal |
| cancel_message_move_task | Cancels a specified message movement task |
| change_message_visibility | Changes the visibility timeout of a specified message in a queue to a new value |
| change_message_visibility_batch | Changes the visibility timeout of multiple messages |
| create_queue | Creates a new standard or FIFO queue |
| delete_message | Deletes the specified message from the specified queue |
| delete_message_batch | Deletes up to ten messages from the specified queue |
| delete_queue | Deletes the queue specified by the QueueUrl, regardless of the queue's contents |

| | |
|--|---|
| get_queue_attributes | Gets attributes for the specified queue |
| get_queue_url | The GetQueueUrl API returns the URL of an existing Amazon SQS queue |
| list_dead_letter_source_queues | Returns a list of your queues that have the RedrivePolicy queue attribute configured with a |
| list_message_move_tasks | Gets the most recent message movement tasks (up to 10) under a specific source queue |
| list_queues | Returns a list of your queues in the current region |
| list_queue_tags | List all cost allocation tags added to the specified Amazon SQS queue |
| purge_queue | Deletes available messages in a queue (including in-flight messages) specified by the QueueName |
| receive_message | Retrieves one or more messages (up to 10), from the specified queue |
| remove_permission | Revokes any permissions in the queue policy that matches the specified Label parameter |
| send_message | Delivers a message to the specified queue |
| send_message_batch | You can use SendMessageBatch to send up to 10 messages to the specified queue by assigning |
| set_queue_attributes | Sets the value of one or more queue attributes, like a policy |
| start_message_move_task | Starts an asynchronous task to move messages from a specified source queue to a specified destination |
| tag_queue | Add cost allocation tags to the specified Amazon SQS queue |
| untag_queue | Remove cost allocation tags from the specified Amazon SQS queue |

Examples

```
## Not run:
svc <- sqs()
svc$add_permission(
  Foo = 123
)

## End(Not run)
```

swf

Amazon Simple Workflow Service

Description

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that use Amazon's cloud to coordinate work across distributed components. In Amazon SWF, a *task* represents a logical unit of work that is performed by a component of your workflow. Coordinating tasks in a workflow involves managing intertask dependencies, scheduling, and concurrency in accordance with the logical flow of the application.

Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

This documentation serves as reference only. For a broader overview of the Amazon SWF programming model, see the *Amazon SWF Developer Guide*.

Usage

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

Arguments

| | |
|-------------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- swf(
  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"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

| | |
|--|---|
| count_closed_workflow_executions | Returns the number of closed workflow executions within the given domain that meet t |
| count_open_workflow_executions | Returns the number of open workflow executions within the given domain that meet th |
| count_pending_activity_tasks | Returns the estimated number of activity tasks in the specified task list |
| count_pending_decision_tasks | Returns the estimated number of decision tasks in the specified task list |
| delete_activity_type | Deletes the specified activity type |
| delete_workflow_type | Deletes the specified workflow type |
| deprecate_activity_type | Deprecates the specified activity type |
| deprecate_domain | Deprecates the specified domain |
| deprecate_workflow_type | Deprecates the specified workflow type |
| describe_activity_type | Returns information about the specified activity type |
| describe_domain | Returns information about the specified domain, including description and status |
| describe_workflow_execution | Returns information about the specified workflow execution including its type and som |
| describe_workflow_type | Returns information about the specified workflow type |
| get_workflow_execution_history | Returns the history of the specified workflow execution |
| list_activity_types | Returns information about all activities registered in the specified domain that match th |
| list_closed_workflow_executions | Returns a list of closed workflow executions in the specified domain that meet the filter |
| list_domains | Returns the list of domains registered in the account |
| list_open_workflow_executions | Returns a list of open workflow executions in the specified domain that meet the filterin |
| list_tags_for_resource | List tags for a given domain |
| list_workflow_types | Returns information about workflow types in the specified domain |

| | |
|--|---|
| <code>poll_for_activity_task</code> | Used by workers to get an ActivityTask from the specified activity taskList |
| <code>poll_for_decision_task</code> | Used by deciders to get a DecisionTask from the specified decision taskList |
| <code>record_activity_task_heartbeat</code> | Used by activity workers to report to the service that the ActivityTask represented by the taskToken is still running |
| <code>register_activity_type</code> | Registers a new activity type along with its configuration settings in the specified domain |
| <code>register_domain</code> | Registers a new domain |
| <code>register_workflow_type</code> | Registers a new workflow type and its configuration settings in the specified domain |
| <code>request_cancel_workflow_execution</code> | Records a WorkflowExecutionCancelRequested event in the currently running workflow execution |
| <code>respond_activity_task_canceled</code> | Used by workers to tell the service that the ActivityTask identified by the taskToken was canceled |
| <code>respond_activity_task_completed</code> | Used by workers to tell the service that the ActivityTask identified by the taskToken completed |
| <code>respond_activity_task_failed</code> | Used by workers to tell the service that the ActivityTask identified by the taskToken has failed |
| <code>respond_decision_task_completed</code> | Used by deciders to tell the service that the DecisionTask identified by the taskToken has completed |
| <code>signal_workflow_execution</code> | Records a WorkflowExecutionSignaled event in the workflow execution history and creates a new workflow execution |
| <code>start_workflow_execution</code> | Starts an execution of the workflow type in the specified domain using the provided workflow type |
| <code>tag_resource</code> | Add a tag to a Amazon SWF domain |
| <code>terminate_workflow_execution</code> | Records a WorkflowExecutionTerminated event and forces closure of the workflow execution |
| <code>undeprecate_activity_type</code> | Undeprecates a previously deprecated activity type |
| <code>undeprecate_domain</code> | Undeprecates a previously deprecated domain |
| <code>undeprecate_workflow_type</code> | Undeprecates a previously deprecated workflow type |
| <code>untag_resource</code> | Remove a tag from a Amazon SWF domain |

Examples

```
## Not run:
svc <- swf()
svc$count_closed_workflow_executions(
  Foo = 123
)

## End(Not run)
```

Index

activate_event_source, [4](#)
add_permission, [31](#), [34](#)
associate_default_view, [22](#)
associate_tracker_consumer, [13](#)

batch_delete_device_position_history, [13](#)
batch_delete_geofence, [13](#)
batch_evaluate_geofences, [13](#)
batch_get_device_position, [13](#)
batch_get_view, [22](#)
batch_put_geofence, [13](#)
batch_update_device_position, [13](#)

calculate_route, [13](#)
calculate_route_matrix, [13](#)
cancel_message_move_task, [34](#)
cancel_replay, [4](#)
change_message_visibility, [34](#)
change_message_visibility_batch, [34](#)
check_if_phone_number_is_opted_out, [31](#)
confirm_subscription, [31](#)
count_closed_workflow_executions, [37](#)
count_open_workflow_executions, [37](#)
count_pending_activity_tasks, [37](#)
count_pending_decision_tasks, [37](#)
create_activity, [28](#)
create_api_destination, [4](#)
create_archive, [4](#)
create_broker, [16](#)
create_cli_token, [17](#), [19](#)
create_configuration, [16](#)
create_connection, [4](#)
create_discoverer, [25](#)
create_endpoint, [4](#)
create_environment, [17](#), [19](#)
create_event_bus, [4](#)
create_geofence_collection, [13](#)
create_index, [22](#)
create_key, [13](#)
create_map, [13](#)
create_partner_event_source, [4](#)
create_pipe, [8](#)
create_place_index, [13](#)
create_platform_application, [31](#)
create_platform_endpoint, [31](#)
create_queue, [34](#)
create_registry, [25](#)
create_route_calculator, [13](#)
create_schedule, [10](#)
create_schedule_group, [10](#)
create_schema, [25](#)
create_sms_sandbox_phone_number, [31](#)
create_state_machine, [28](#)
create_state_machine_alias, [28](#)
create_tags, [16](#)
create_topic, [31](#)
create_tracker, [13](#)
create_user, [16](#)
create_view, [22](#)
create_web_login_token, [17](#), [19](#)

deactivate_event_source, [4](#)
deauthorize_connection, [4](#)
delete_activity, [28](#)
delete_activity_type, [37](#)
delete_api_destination, [4](#)
delete_archive, [4](#)
delete_broker, [16](#)
delete_connection, [4](#)
delete_discoverer, [25](#)
delete_endpoint, [5](#), [31](#)
delete_environment, [17](#), [19](#)
delete_event_bus, [5](#)
delete_geofence_collection, [13](#)
delete_index, [22](#)
delete_key, [13](#)
delete_map, [13](#)
delete_message, [34](#)
delete_message_batch, [34](#)

- delete_partner_event_source, 5
- delete_pipe, 8
- delete_place_index, 13
- delete_platform_application, 31
- delete_queue, 34
- delete_registry, 25
- delete_resource_policy, 25
- delete_route_calculator, 13
- delete_rule, 5
- delete_schedule, 10
- delete_schedule_group, 10
- delete_schema, 25
- delete_schema_version, 25
- delete_sms_sandbox_phone_number, 31
- delete_state_machine, 28
- delete_state_machine_alias, 28
- delete_state_machine_version, 28
- delete_tags, 16
- delete_topic, 31
- delete_tracker, 13
- delete_user, 16
- delete_view, 22
- delete_workflow_type, 37
- deprecate_activity_type, 37
- deprecate_domain, 37
- deprecate_workflow_type, 37
- describe_activity, 28
- describe_activity_type, 37
- describe_api_destination, 5
- describe_archive, 5
- describe_broker, 16
- describe_broker_engine_types, 16
- describe_broker_instance_options, 16
- describe_code_binding, 25
- describe_configuration, 16
- describe_configuration_revision, 16
- describe_connection, 5
- describe_discoverer, 25
- describe_domain, 37
- describe_endpoint, 5
- describe_event_bus, 5
- describe_event_source, 5
- describe_execution, 28
- describe_geofence_collection, 13
- describe_key, 13
- describe_map, 13
- describe_map_run, 28
- describe_partner_event_source, 5
- describe_pipe, 8
- describe_place_index, 13
- describe_registry, 25
- describe_replay, 5
- describe_route_calculator, 13
- describe_rule, 5
- describe_schema, 25
- describe_state_machine, 28
- describe_state_machine_alias, 28
- describe_state_machine_for_execution, 28
- describe_tracker, 13
- describe_user, 16
- describe_workflow_execution, 37
- describe_workflow_type, 37
- disable_rule, 5
- disassociate_default_view, 22
- disassociate_tracker_consumer, 13
- enable_rule, 5
- eventbridge, 2
- eventbridgepipes, 6
- eventbridgescheduler, 8
- export_schema, 25
- forecast_geofence_events, 13
- get_account_level_service_configuration, 22
- get_activity_task, 28
- get_code_binding_source, 25
- get_data_protection_policy, 31
- get_default_view, 22
- get_device_position, 13
- get_device_position_history, 13
- get_discovered_schema, 25
- get_endpoint_attributes, 31
- get_environment, 17, 19
- get_execution_history, 28
- get_geofence, 13
- get_index, 22
- get_managed_view, 22
- get_map_glyphs, 13
- get_map_sprites, 13
- get_map_style_descriptor, 13
- get_map_tile, 13
- get_place, 13
- get_platform_application_attributes, 31

- get_queue_attributes, 35
- get_queue_url, 35
- get_resource_policy, 25
- get_schedule, 10
- get_schedule_group, 10
- get_sms_attributes, 31
- get_sms_sandbox_account_status, 31
- get_subscription_attributes, 31
- get_topic_attributes, 31
- get_view, 22
- get_workflow_execution_history, 37

- invoke_rest_api, 17, 19

- list_activities, 28
- list_activity_types, 37
- list_api_destinations, 5
- list_archives, 5
- list_brokers, 16
- list_closed_workflow_executions, 37
- list_configuration_revisions, 16
- list_configurations, 16
- list_connections, 5
- list_dead_letter_source_queues, 35
- list_device_positions, 13
- list_discoverers, 25
- list_domains, 37
- list_endpoints, 5
- list_endpoints_by_platform_application, 31
- list_environments, 17, 19
- list_event_buses, 5
- list_event_sources, 5
- list_executions, 28
- list_geofence_collections, 13
- list_geofences, 13
- list_indexes, 22
- list_indexes_for_members, 22
- list_keys, 14
- list_managed_views, 22
- list_map_runs, 28
- list_maps, 14
- list_message_move_tasks, 35
- list_open_workflow_executions, 37
- list_origination_numbers, 31
- list_partner_event_source_accounts, 5
- list_partner_event_sources, 5
- list_phone_numbers_opted_out, 31
- list_pipes, 8
- list_place_indexes, 14
- list_platform_applications, 31
- list_queue_tags, 35
- list_queues, 35
- list_registries, 25
- list_replays, 5
- list_resources, 22
- list_route_calculators, 14
- list_rule_names_by_target, 5
- list_rules, 5
- list_schedule_groups, 10
- list_schedules, 10
- list_schema_versions, 25
- list_schemas, 25
- list_sms_sandbox_phone_numbers, 31
- list_state_machine_aliases, 28
- list_state_machine_versions, 28
- list_state_machines, 28
- list_subscriptions, 31
- list_subscriptions_by_topic, 31
- list_supported_resource_types, 22
- list_tags, 16
- list_tags_for_resource, 5, 8, 10, 14, 17, 19, 22, 25, 28, 31, 37
- list_targets_by_rule, 5
- list_topics, 31
- list_tracker_consumers, 14
- list_trackers, 14
- list_users, 16
- list_views, 22
- list_workflow_types, 37
- locationservice, 11

- mq, 14
- mwa, 17

- opt_in_phone_number, 32

- poll_for_activity_task, 38
- poll_for_decision_task, 38
- promote, 16
- publish, 32
- publish_batch, 32
- publish_metrics, 19
- publish_state_machine_version, 28
- purge_queue, 35
- put_code_binding, 25
- put_data_protection_policy, 32
- put_events, 5

- put_geofence, [14](#)
- put_partner_events, [5](#)
- put_permission, [5](#)
- put_resource_policy, [25](#)
- put_rule, [5](#)
- put_targets, [5](#)

- reboot_broker, [16](#)
- receive_message, [35](#)
- record_activity_task_heartbeat, [38](#)
- redrive_execution, [28](#)
- register_activity_type, [38](#)
- register_domain, [38](#)
- register_workflow_type, [38](#)
- remove_permission, [5](#), [32](#), [35](#)
- remove_targets, [5](#)
- request_cancel_workflow_execution, [38](#)
- resourceexplorer, [20](#)
- respond_activity_task_canceled, [38](#)
- respond_activity_task_completed, [38](#)
- respond_activity_task_failed, [38](#)
- respond_decision_task_completed, [38](#)

- schemas, [23](#)
- search, [22](#)
- search_place_index_for_position, [14](#)
- search_place_index_for_suggestions, [14](#)
- search_place_index_for_text, [14](#)
- search_schemas, [25](#)
- send_message, [35](#)
- send_message_batch, [35](#)
- send_task_failure, [28](#)
- send_task_heartbeat, [28](#)
- send_task_success, [28](#)
- set_endpoint_attributes, [32](#)
- set_platform_application_attributes, [32](#)
- set_queue_attributes, [35](#)
- set_sms_attributes, [32](#)
- set_subscription_attributes, [32](#)
- set_topic_attributes, [32](#)
- sfn, [26](#)
- signal_workflow_execution, [38](#)
- sns, [29](#)
- sq, [32](#)
- start_discoverer, [25](#)
- start_execution, [28](#)
- start_message_move_task, [35](#)
- start_pipe, [8](#)

- start_replay, [5](#)
- start_sync_execution, [28](#)
- start_workflow_execution, [38](#)
- stop_discoverer, [25](#)
- stop_execution, [28](#)
- stop_pipe, [8](#)
- subscribe, [32](#)
- swf, [35](#)

- tag_queue, [35](#)
- tag_resource, [5](#), [8](#), [10](#), [14](#), [17](#), [19](#), [23](#), [25](#), [28](#), [32](#), [38](#)
- terminate_workflow_execution, [38](#)
- test_event_pattern, [5](#)
- test_state, [28](#)

- undeprecate_activity_type, [38](#)
- undeprecate_domain, [38](#)
- undeprecate_workflow_type, [38](#)
- unsubscribe, [32](#)
- untag_queue, [35](#)
- untag_resource, [5](#), [8](#), [10](#), [14](#), [17](#), [19](#), [23](#), [25](#), [29](#), [32](#), [38](#)
- update_api_destination, [5](#)
- update_archive, [5](#)
- update_broker, [17](#)
- update_configuration, [17](#)
- update_connection, [5](#)
- update_discoverer, [25](#)
- update_endpoint, [5](#)
- update_environment, [17](#), [19](#)
- update_event_bus, [5](#)
- update_geofence_collection, [14](#)
- update_index_type, [23](#)
- update_key, [14](#)
- update_map, [14](#)
- update_map_run, [29](#)
- update_pipe, [8](#)
- update_place_index, [14](#)
- update_registry, [25](#)
- update_route_calculator, [14](#)
- update_schedule, [10](#)
- update_schema, [25](#)
- update_state_machine, [29](#)
- update_state_machine_alias, [29](#)
- update_tracker, [14](#)
- update_user, [17](#)
- update_view, [23](#)

`validate_state_machine_definition`, [29](#)
`verify_device_position`, [14](#)
`verify_sms_sandbox_phone_number`, [32](#)