

Package ‘paws.application.integration’

October 7, 2026

Title 'Amazon Web Services' Application Integration Services

Version 0.11.0

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>,
<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

Config/roxygen2/version 8.1.0

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' 'location_service.R'
'location_service_interfaces.R' 'location_service_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 2026-10-07 19:10:02 UTC

Contents

eventbridge	2
eventbridgepipes	6
eventbridgescheduler	9
locationservice	11
mq	15
mwaa	18
resourceexplorer	21
schemas	24
sfn	27
sns	30
sqs	34
swf	37
Index	41

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. • 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. <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 • 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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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

[activate_event_source](#)
[cancel_replay](#)
[create_api_destination](#)
[create_archive](#)
[create_connection](#)
[create_endpoint](#)
[create_event_bus](#)

Activates a partner event source that has been deactivated
 Cancels the specified replay
 Creates an API destination, which is an HTTP invocation endpoint configured as a target
 Creates an archive of events with the specified settings
 Creates a connection
 Creates a global endpoint
 Creates a new event bus within your account

<code>create_partner_event_source</code>	Called by an SaaS partner to create a partner event source
<code>deactivate_event_source</code>	You can use this operation to temporarily stop receiving events from the specified partner event source
<code>deauthorize_connection</code>	Removes all authorization parameters from the connection
<code>delete_api_destination</code>	Deletes the specified API destination
<code>delete_archive</code>	Deletes the specified archive
<code>delete_connection</code>	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 they have shared with your account
<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 bus, and partner event bus
<code>list_event_sources</code>	You can use this to see all the partner event sources that have been shared with your Amazon account
<code>list_partner_event_source_accounts</code>	An SaaS partner can use this operation to display the Amazon Web Services account ID of the partner event source
<code>list_partner_event_sources</code>	An SaaS partner can use this operation to list all the partner event source names that they have shared with your account
<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 EventBridge rule to put events to the specified event bus
<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 associated with the rule
<code>remove_permission</code>	Revokes the permission of another Amazon Web Services account to be able to put events to the specified event bus
<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	<i>Amazon EventBridge Pipes</i>
------------------	---------------------------------

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

<code>config</code>	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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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.

- **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-endpoint.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 <- 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",
    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

<code>create_schedule</code>	Creates the specified schedule
<code>create_schedule_group</code>	Creates the specified schedule group
<code>delete_schedule</code>	Deletes the specified schedule
<code>delete_schedule_group</code>	Deletes the specified schedule group
<code>get_schedule</code>	Retrieves the specified schedule
<code>get_schedule_group</code>	Retrieves the specified schedule group
<code>list_schedule_groups</code>	Returns a paginated list of your schedule groups
<code>list_schedules</code>	Returns a paginated list of your EventBridge Scheduler schedules
<code>list_tags_for_resource</code>	Lists the tags associated with the Scheduler resource
<code>tag_resource</code>	Assigns one or more tags (key-value pairs) to the specified EventBridge Scheduler resource
<code>untag_resource</code>	Removes one or more tags from the specified EventBridge Scheduler schedule group
<code>update_schedule</code>	Updates the specified schedule

Examples

```

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

## End(Not run)

```

locationservice

Amazon Location Service

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. <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 • 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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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 updates
batch_update_device_position	Uploads position update data for one or more devices to a tracker resource (up to 10 devices)
calculate_route	This operation is no longer current and may be deprecated in the future
calculate_route_matrix	This operation is no longer current and may be deprecated in the future
cancel_job	CancelJob cancels a job that is currently running or pending
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 generate

<code>create_map</code>	This operation is no longer current and may be deprecated in the future
<code>create_place_index</code>	This operation is no longer current and may be deprecated in the future
<code>create_route_calculator</code>	This operation is no longer current and may be deprecated in the future
<code>create_tracker</code>	Creates a tracker resource in your Amazon Web Services account, which lets you retr
<code>delete_geofence_collection</code>	Deletes a geofence collection from your Amazon Web Services account
<code>delete_key</code>	Deletes the specified API key
<code>delete_map</code>	This operation is no longer current and may be deprecated in the future
<code>delete_place_index</code>	This operation is no longer current and may be deprecated in the future
<code>delete_route_calculator</code>	This operation is no longer current and may be deprecated in the future
<code>delete_tracker</code>	Deletes a tracker resource from your Amazon Web Services account
<code>describe_geofence_collection</code>	Retrieves the geofence collection details
<code>describe_key</code>	Retrieves the API key resource details
<code>describe_map</code>	This operation is no longer current and may be deprecated in the future
<code>describe_place_index</code>	This operation is no longer current and may be deprecated in the future
<code>describe_route_calculator</code>	This operation is no longer current and may be deprecated in the future
<code>describe_tracker</code>	Retrieves the tracker resource details
<code>disassociate_tracker_consumer</code>	Removes the association between a tracker resource and a geofence collection
<code>forecast_geofence_events</code>	This action forecasts future geofence events that are likely to occur within a specified
<code>get_device_position</code>	Retrieves a device's most recent position according to its sample time
<code>get_device_position_history</code>	Retrieves the device position history from a tracker resource within a specified range
<code>get_geofence</code>	Retrieves the geofence details from a geofence collection
<code>get_job</code>	GetJob retrieves detailed information about a specific job, including its current status,
<code>get_map_glyphs</code>	This operation is no longer current and may be deprecated in the future
<code>get_map_sprites</code>	This operation is no longer current and may be deprecated in the future
<code>get_map_style_descriptor</code>	This operation is no longer current and may be deprecated in the future
<code>get_map_tile</code>	This operation is no longer current and may be deprecated in the future
<code>get_place</code>	This operation is no longer current and may be deprecated in the future
<code>list_device_positions</code>	A batch request to retrieve all device positions
<code>list_geofence_collections</code>	Lists geofence collections in your Amazon Web Services account
<code>list_geofences</code>	Lists geofences stored in a given geofence collection
<code>list_jobs</code>	ListJobs retrieves a list of jobs with optional filtering and pagination support
<code>list_keys</code>	Lists API key resources in your Amazon Web Services account
<code>list_maps</code>	This operation is no longer current and may be deprecated in the future
<code>list_place_indexes</code>	This operation is no longer current and may be deprecated in the future
<code>list_route_calculators</code>	This operation is no longer current and may be deprecated in the future
<code>list_tags_for_resource</code>	Returns a list of tags that are applied to the specified Amazon Location resource
<code>list_tracker_consumers</code>	Lists geofence collections currently associated to the given tracker resource
<code>list_trackers</code>	Lists tracker resources in your Amazon Web Services account
<code>put_geofence</code>	Stores a geofence geometry in a given geofence collection, or updates the geometry o
<code>search_place_index_for_position</code>	This operation is no longer current and may be deprecated in the future
<code>search_place_index_for_suggestions</code>	This operation is no longer current and may be deprecated in the future
<code>search_place_index_for_text</code>	This operation is no longer current and may be deprecated in the future
<code>start_job</code>	StartJob starts a new asynchronous bulk processing job
<code>tag_resource</code>	Assigns one or more tags (key-value pairs) to the specified Amazon Location Service
<code>untag_resource</code>	Removes one or more tags from the specified Amazon Location resource
<code>update_geofence_collection</code>	Updates the specified properties of a given geofence collection
<code>update_key</code>	Updates the specified properties of a given API key resource
<code>update_map</code>	This operation is no longer current and may be deprecated in the future

[update_place_index](#)
[update_route_calculator](#)
[update_tracker](#)
[verify_device_position](#)

This operation is no longer current and may be deprecated in the future
 This operation is no longer current and may be deprecated in the future
 Updates the specified properties of a given tracker resource
 Verifies the integrity of the device's position by determining if it was reported behind

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.

- **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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
        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

<code>create_broker</code>	Creates a broker
<code>create_configuration</code>	Creates a new configuration for the specified configuration name
<code>create_tags</code>	Add a tag to a resource
<code>create_user</code>	Creates an ActiveMQ user
<code>delete_broker</code>	Deletes a broker
<code>delete_configuration</code>	Deletes the specified configuration
<code>delete_tags</code>	Removes a tag from a resource
<code>delete_user</code>	Deletes an ActiveMQ user
<code>describe_broker</code>	Returns information about the specified broker
<code>describe_broker_engine_types</code>	Describe available engine types and versions
<code>describe_broker_instance_options</code>	Describe available broker instance options
<code>describe_configuration</code>	Returns information about the specified configuration
<code>describe_configuration_revision</code>	Returns the specified configuration revision for the specified configuration
<code>describe_shared_resources</code>	Returns the resources shared to a broker
<code>describe_user</code>	Returns information about an ActiveMQ user
<code>list_brokers</code>	Returns a list of all brokers
<code>list_configuration_revisions</code>	Returns a list of all revisions for the specified configuration
<code>list_configurations</code>	Returns a list of all configurations
<code>list_tags</code>	Lists tags for a resource
<code>list_users</code>	Returns a list of all ActiveMQ users
<code>promote</code>	Promotes a data replication replica broker to the primary broker role
<code>reboot_broker</code>	Reboots a broker
<code>update_broker</code>	Adds a pending configuration change to a broker
<code>update_configuration</code>	Updates the specified configuration
<code>update_user</code>	Updates the information for an ActiveMQ user

Examples

```
## Not run:
```

```

svc <- mq()
svc$create_broker(
  Foo = 123
)

## End(Not run)

```

mwaa

AmazonMWAA

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` (use `api.airflow.{region}.api.aws` for IPv6)
 - 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` (use `env.airflow.{region}.api.aws` for IPv6)
 - 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

```
mwaa(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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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 (MWAA) environment
publish_metrics	Internal only
tag_resource	Associates key-value tag pairs to your Amazon Managed Workflows for Apache Airflow (MWAA) environment
untag_resource	Removes key-value tag pairs associated to your Amazon Managed Workflows for Apache Airflow (MWAA) environment
update_environment	Updates an Amazon Managed Workflows for Apache Airflow (MWAA) environment

Examples

```
## Not run:
svc <- mwaa()
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. • 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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_default_view	Sets the specified view as the default for the Amazon Web Services Region in which you are operating.
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_resource_explorer_setup	Creates a Resource Explorer setup configuration across multiple Amazon Web Services Regions.
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_resource_explorer_setup	Deletes a Resource Explorer setup configuration.
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 the status of Amazon Web Services Resource Explorer.
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.

<code>get_managed_view</code>	Retrieves details of the specified Amazon Web Services-managed view
<code>get_resource_explorer_setup</code>	Retrieves the status and details of a Resource Explorer setup operation
<code>get_service_index</code>	Retrieves information about the Resource Explorer index in the current Amazon Web Services Region
<code>get_service_view</code>	Retrieves details about a specific Resource Explorer service view
<code>get_view</code>	Retrieves details of the specified view
<code>list_indexes</code>	Retrieves a list of all of the indexes in Amazon Web Services Regions that are currently supported
<code>list_indexes_for_members</code>	Retrieves a list of a member's indexes in all Amazon Web Services Regions that are currently supported
<code>list_managed_views</code>	Lists the Amazon resource names (ARNs) of the Amazon Web Services-managed views
<code>list_resources</code>	Returns a list of resources and their details that match the specified criteria
<code>list_service_indexes</code>	Lists all Resource Explorer indexes across the specified Amazon Web Services Region
<code>list_service_views</code>	Lists all Resource Explorer service views available in the current Amazon Web Services Region
<code>list_streaming_access_for_services</code>	Returns a list of Amazon Web Services services that have been granted streaming access to the specified resource
<code>list_supported_resource_types</code>	Retrieves a list of all resource types currently supported by Amazon Web Services
<code>list_tags_for_resource</code>	Lists the tags that are attached to the specified resource
<code>list_views</code>	Lists the Amazon resource names (ARNs) of the views available in the Amazon Web Services Region
<code>search</code>	Searches for resources and displays details about all resources that match the specified criteria
<code>tag_resource</code>	Adds one or more tag key and value pairs to an Amazon Web Services Resource
<code>untag_resource</code>	Removes one or more tag key and value pairs from an Amazon Web Services Resource
<code>update_index_type</code>	Changes the type of the index from one of the following types to the other
<code>update_view</code>	Modifies some of the details of a view

Examples

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

## End(Not run)
```

schemas	<i>Schemas</i>
---------	----------------

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. • 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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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

<code>export_schema</code>	Export schema
<code>get_code_binding_source</code>	Get the code binding source URI
<code>get_discovered_schema</code>	Get the discovered schema that was generated based on sampled events
<code>get_resource_policy</code>	Retrieves the resource-based policy attached to a given registry
<code>list_discoverers</code>	List the discoverers
<code>list_registries</code>	List the registries
<code>list_schemas</code>	List the schemas
<code>list_schema_versions</code>	Provides a list of the schema versions and related information
<code>list_tags_for_resource</code>	Get tags for resource
<code>put_code_binding</code>	Put code binding URI
<code>put_resource_policy</code>	The name of the policy
<code>search_schemas</code>	Search the schemas
<code>start_discoverer</code>	Starts the discoverer
<code>stop_discoverer</code>	Stops the discoverer
<code>tag_resource</code>	Add tags to a resource
<code>untag_resource</code>	Removes tags from a resource
<code>update_discoverer</code>	Updates the discoverer
<code>update_registry</code>	Updates a registry
<code>update_schema</code>	Updates the schema definition

Examples

```
## Not run:
svc <- schemas()
svc$create_discoverer(
  Foo = 123
)

## End(Not run)
```

Description

Step Functions

With Step Functions, you can create workflows, also called *state machines*, to build distributed applications, automate processes, orchestrate microservices, and create data and machine learning pipelines.

Through the Step Functions API, you can create, list, update, and delete state machines, activities, and other data types. You can start, stop, and redrive your state machines. Your activity workers can send task success, heartbeat, and failure responses.

With API calls, you can also manage other aspects of your workflow, such as tags, versions, and aliases.

For more information about developing solutions with 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 might 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. • 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 • 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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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 associated with the execution
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 IAM role ARN
get_activity_task	Used by workers to retrieve a task (with the specified activity ARN) which has been assigned to the worker
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 that were specified in the original configuration
update_state_machine	Updates an existing state machine by modifying its definition, roleArn, loggingConfiguration, and its execution role ARN
update_state_machine_alias	Updates the configuration of an existing state machine alias by modifying its description, roleArn, and its execution role ARN
validate_state_machine_definition	Validates the syntax of a state machine definition specified in Amazon States Language

Examples

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

## End(Not run)
```

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
- * **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-endpoint.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 <- 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",
    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

<code>add_permission</code>	Adds a statement to a topic's access control policy, granting access for the specified
<code>check_if_phone_number_is_opted_out</code>	Accepts a phone number and indicates whether the phone holder has opted out of re
<code>confirm_subscription</code>	Verifies an endpoint owner's intent to receive messages by validating the token sent
<code>create_platform_application</code>	Creates a platform application object for one of the supported push notification serv
<code>create_platform_endpoint</code>	Creates an endpoint for a device and mobile app on one of the supported push notif
<code>create_sms_sandbox_phone_number</code>	Adds a destination phone number to an Amazon Web Services account in the SMS
<code>create_topic</code>	Creates a topic to which notifications can be published
<code>delete_endpoint</code>	Deletes the endpoint for a device and mobile app from Amazon SNS
<code>delete_platform_application</code>	Deletes a platform application object for one of the supported push notification serv
<code>delete_sms_sandbox_phone_number</code>	Deletes an Amazon Web Services account's verified or pending phone number from
<code>delete_topic</code>	Deletes a topic and all its subscriptions
<code>get_data_protection_policy</code>	Amazon SNS message data protection is no longer available to new customers
<code>get_endpoint_attributes</code>	Retrieves the endpoint attributes for a device on one of the supported push notificat
<code>get_platform_application_attributes</code>	Retrieves the attributes of the platform application object for the supported push no
<code>get_sms_attributes</code>	Returns the settings for sending SMS messages from your Amazon Web Services a
<code>get_sms_sandbox_account_status</code>	Retrieves the SMS sandbox status for the calling Amazon Web Services account in
<code>get_subscription_attributes</code>	Returns all of the properties of a subscription
<code>get_topic_attributes</code>	Returns all of the properties of a topic
<code>list_endpoints_by_platform_application</code>	Lists the endpoints and endpoint attributes for devices in a supported push notificati
<code>list_origination_numbers</code>	Lists the calling Amazon Web Services account's dedicated origination numbers an
<code>list_phone_numbers_opted_out</code>	Returns a list of phone numbers that are opted out, meaning you cannot send SMS m
<code>list_platform_applications</code>	Lists the platform application objects for the supported push notification services, s
<code>list_sms_sandbox_phone_numbers</code>	Lists the calling Amazon Web Services account's current verified and pending desti
<code>list_subscriptions</code>	Returns a list of the requester's subscriptions
<code>list_subscriptions_by_topic</code>	Returns a list of the subscriptions to a specific topic
<code>list_tags_for_resource</code>	List all tags added to the specified Amazon SNS topic
<code>list_topics</code>	Returns a list of the requester's topics
<code>opt_in_phone_number</code>	Use this request to opt in a phone number that is opted out, which enables you to re
<code>publish</code>	Sends a message to an Amazon SNS topic, a text message (SMS message) directly
<code>publish_batch</code>	Publishes up to 10 messages to the specified topic in a single batch
<code>put_data_protection_policy</code>	Amazon SNS message data protection is no longer available to new customers
<code>remove_permission</code>	Removes a statement from a topic's access control policy
<code>set_endpoint_attributes</code>	Sets the attributes for an endpoint for a device on one of the supported push notifica
<code>set_platform_application_attributes</code>	Sets the attributes of the platform application object for the supported push notifica
<code>set_sms_attributes</code>	Use this request to set the default settings for sending SMS messages and receiving
<code>set_subscription_attributes</code>	Allows a subscription owner to set an attribute of the subscription to a new value
<code>set_topic_attributes</code>	Allows a topic owner to set an attribute of the topic to a new value
<code>subscribe</code>	Subscribes an endpoint to an Amazon SNS topic
<code>tag_resource</code>	Add tags to the specified Amazon SNS topic
<code>unsubscribe</code>	Deletes a subscription
<code>untag_resource</code>	Remove tags from the specified Amazon SNS topic
<code>verify_sms_sandbox_phone_number</code>	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	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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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_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 parameter
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 them to a batch ID
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 queue
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. • 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. <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-endpoints.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 <code>AWS_USE_DUALSTACK_ENDPOINT</code> 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 <- 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",
    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

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

<code>describe_workflow_type</code>	Returns information about the specified workflow type
<code>get_workflow_execution_history</code>	Returns the history of the specified workflow execution
<code>list_activity_types</code>	Returns information about all activities registered in the specified domain that match the filter
<code>list_closed_workflow_executions</code>	Returns a list of closed workflow executions in the specified domain that meet the filter
<code>list_domains</code>	Returns the list of domains registered in the account
<code>list_open_workflow_executions</code>	Returns a list of open workflow executions in the specified domain that meet the filter
<code>list_tags_for_resource</code>	List tags for a given domain
<code>list_workflow_types</code>	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 history
<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 configuration
<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, [33](#), [36](#)
associate_default_view, [23](#)
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, [23](#)
batch_put_geofence, [13](#)
batch_update_device_position, [13](#)

calculate_route, [13](#)
calculate_route_matrix, [13](#)
cancel_job, [13](#)
cancel_message_move_task, [36](#)
cancel_replay, [4](#)
change_message_visibility, [36](#)
change_message_visibility_batch, [36](#)
check_if_phone_number_is_opted_out, [33](#)
confirm_subscription, [33](#)
count_closed_workflow_executions, [39](#)
count_open_workflow_executions, [39](#)
count_pending_activity_tasks, [39](#)
count_pending_decision_tasks, [39](#)
create_activity, [29](#)
create_api_destination, [4](#)
create_archive, [4](#)
create_broker, [17](#)
create_cli_token, [18](#), [20](#)
create_configuration, [17](#)
create_connection, [4](#)
create_discoverer, [26](#)
create_endpoint, [4](#)
create_environment, [18](#), [20](#)
create_event_bus, [4](#)
create_geofence_collection, [13](#)
create_index, [23](#)

create_key, [13](#)
create_map, [14](#)
create_partner_event_source, [5](#)
create_pipe, [8](#)
create_place_index, [14](#)
create_platform_application, [33](#)
create_platform_endpoint, [33](#)
create_queue, [36](#)
create_registry, [26](#)
create_resource_explorer_setup, [23](#)
create_route_calculator, [14](#)
create_schedule, [11](#)
create_schedule_group, [11](#)
create_schema, [26](#)
create_sms_sandbox_phone_number, [33](#)
create_state_machine, [29](#)
create_state_machine_alias, [29](#)
create_tags, [17](#)
create_topic, [33](#)
create_tracker, [14](#)
create_user, [17](#)
create_view, [23](#)
create_web_login_token, [18](#), [20](#)

deactivate_event_source, [5](#)
deauthorize_connection, [5](#)
delete_activity, [29](#)
delete_activity_type, [39](#)
delete_api_destination, [5](#)
delete_archive, [5](#)
delete_broker, [17](#)
delete_configuration, [17](#)
delete_connection, [5](#)
delete_discoverer, [26](#)
delete_endpoint, [5](#), [33](#)
delete_environment, [18](#), [20](#)
delete_event_bus, [5](#)
delete_geofence_collection, [14](#)
delete_index, [23](#)
delete_key, [14](#)

- delete_map, [14](#)
- delete_message, [36](#)
- delete_message_batch, [36](#)
- delete_partner_event_source, [5](#)
- delete_pipe, [8](#)
- delete_place_index, [14](#)
- delete_platform_application, [33](#)
- delete_queue, [36](#)
- delete_registry, [26](#)
- delete_resource_explorer_setup, [23](#)
- delete_resource_policy, [26](#)
- delete_route_calculator, [14](#)
- delete_rule, [5](#)
- delete_schedule, [11](#)
- delete_schedule_group, [11](#)
- delete_schema, [26](#)
- delete_schema_version, [26](#)
- delete_sms_sandbox_phone_number, [33](#)
- delete_state_machine, [29](#)
- delete_state_machine_alias, [29](#)
- delete_state_machine_version, [29](#)
- delete_tags, [17](#)
- delete_topic, [33](#)
- delete_tracker, [14](#)
- delete_user, [17](#)
- delete_view, [23](#)
- delete_workflow_type, [39](#)
- deprecate_activity_type, [39](#)
- deprecate_domain, [39](#)
- deprecate_workflow_type, [39](#)
- describe_activity, [30](#)
- describe_activity_type, [39](#)
- describe_api_destination, [5](#)
- describe_archive, [5](#)
- describe_broker, [17](#)
- describe_broker_engine_types, [17](#)
- describe_broker_instance_options, [17](#)
- describe_code_binding, [26](#)
- describe_configuration, [17](#)
- describe_configuration_revision, [17](#)
- describe_connection, [5](#)
- describe_discoverer, [26](#)
- describe_domain, [39](#)
- describe_endpoint, [5](#)
- describe_event_bus, [5](#)
- describe_event_source, [5](#)
- describe_execution, [30](#)
- describe_geofence_collection, [14](#)
- describe_key, [14](#)
- describe_map, [14](#)
- describe_map_run, [30](#)
- describe_partner_event_source, [5](#)
- describe_pipe, [8](#)
- describe_place_index, [14](#)
- describe_registry, [26](#)
- describe_replay, [5](#)
- describe_route_calculator, [14](#)
- describe_rule, [5](#)
- describe_schema, [26](#)
- describe_shared_resources, [17](#)
- describe_state_machine, [30](#)
- describe_state_machine_alias, [30](#)
- describe_state_machine_for_execution, [30](#)
- describe_tracker, [14](#)
- describe_user, [17](#)
- describe_workflow_execution, [39](#)
- describe_workflow_type, [40](#)
- disable_rule, [5](#)
- disassociate_default_view, [23](#)
- disassociate_tracker_consumer, [14](#)
- enable_rule, [5](#)
- eventbridge, [2](#)
- eventbridgepipes, [6](#)
- eventbridgescheduler, [9](#)
- export_schema, [27](#)
- forecast_geofence_events, [14](#)
- get_account_level_service_configuration, [23](#)
- get_activity_task, [30](#)
- get_code_binding_source, [27](#)
- get_data_protection_policy, [33](#)
- get_default_view, [23](#)
- get_device_position, [14](#)
- get_device_position_history, [14](#)
- get_discovered_schema, [27](#)
- get_endpoint_attributes, [33](#)
- get_environment, [18](#), [20](#)
- get_execution_history, [30](#)
- get_geofence, [14](#)
- get_index, [23](#)
- get_job, [14](#)
- get_managed_view, [24](#)
- get_map_glyphs, [14](#)

get_map_sprites, [14](#)
get_map_style_descriptor, [14](#)
get_map_tile, [14](#)
get_place, [14](#)
get_platform_application_attributes,
 [33](#)
get_queue_attributes, [36](#)
get_queue_url, [36](#)
get_resource_explorer_setup, [24](#)
get_resource_policy, [27](#)
get_schedule, [11](#)
get_schedule_group, [11](#)
get_service_index, [24](#)
get_service_view, [24](#)
get_sms_attributes, [33](#)
get_sms_sandbox_account_status, [33](#)
get_subscription_attributes, [33](#)
get_topic_attributes, [33](#)
get_view, [24](#)
get_workflow_execution_history, [40](#)

invoke_rest_api, [18, 20](#)

list_activities, [30](#)
list_activity_types, [40](#)
list_api_destinations, [5](#)
list_archives, [5](#)
list_brokers, [17](#)
list_closed_workflow_executions, [40](#)
list_configuration_revisions, [17](#)
list_configurations, [17](#)
list_connections, [5](#)
list_dead_letter_source_queues, [36](#)
list_device_positions, [14](#)
list_discoverers, [27](#)
list_domains, [40](#)
list_endpoints, [5](#)
list_endpoints_by_platform_application,
 [33](#)
list_environments, [18, 20](#)
list_event_buses, [5](#)
list_event_sources, [5](#)
list_executions, [30](#)
list_geofence_collections, [14](#)
list_geofences, [14](#)
list_indexes, [24](#)
list_indexes_for_members, [24](#)
list_jobs, [14](#)
list_keys, [14](#)

list_managed_views, [24](#)
list_map_runs, [30](#)
list_maps, [14](#)
list_message_move_tasks, [36](#)
list_open_workflow_executions, [40](#)
list_origination_numbers, [33](#)
list_partner_event_source_accounts, [5](#)
list_partner_event_sources, [5](#)
list_phone_numbers_opted_out, [33](#)
list_pipes, [8](#)
list_place_indexes, [14](#)
list_platform_applications, [33](#)
list_queue_tags, [37](#)
list_queues, [37](#)
list_registries, [27](#)
list_replays, [5](#)
list_resources, [24](#)
list_route_calculators, [14](#)
list_rule_names_by_target, [5](#)
list_rules, [5](#)
list_schedule_groups, [11](#)
list_schedules, [11](#)
list_schema_versions, [27](#)
list_schemas, [27](#)
list_service_indexes, [24](#)
list_service_views, [24](#)
list_sms_sandbox_phone_numbers, [33](#)
list_state_machine_aliases, [30](#)
list_state_machine_versions, [30](#)
list_state_machines, [30](#)
list_streaming_access_for_services, [24](#)
list_subscriptions, [33](#)
list_subscriptions_by_topic, [33](#)
list_supported_resource_types, [24](#)
list_tags, [17](#)
list_tags_for_resource, [5, 8, 11, 14, 18,](#)
 [20, 24, 27, 30, 33, 40](#)
list_targets_by_rule, [5](#)
list_topics, [33](#)
list_tracker_consumers, [14](#)
list_trackers, [14](#)
list_users, [17](#)
list_views, [24](#)
list_workflow_types, [40](#)
locationservice, [11](#)

mq, [15](#)
mwaa, [18](#)

- opt_in_phone_number, [33](#)
- poll_for_activity_task, [40](#)
- poll_for_decision_task, [40](#)
- promote, [17](#)
- publish, [33](#)
- publish_batch, [33](#)
- publish_metrics, [20](#)
- publish_state_machine_version, [30](#)
- purge_queue, [37](#)
- put_code_binding, [27](#)
- put_data_protection_policy, [33](#)
- put_events, [5](#)
- put_geofence, [14](#)
- put_partner_events, [5](#)
- put_permission, [5](#)
- put_resource_policy, [27](#)
- put_rule, [5](#)
- put_targets, [5](#)
- reboot_broker, [17](#)
- receive_message, [37](#)
- record_activity_task_heartbeat, [40](#)
- redrive_execution, [30](#)
- register_activity_type, [40](#)
- register_domain, [40](#)
- register_workflow_type, [40](#)
- remove_permission, [5](#), [33](#), [37](#)
- remove_targets, [5](#)
- request_cancel_workflow_execution, [40](#)
- resourceexplorer, [21](#)
- respond_activity_task_canceled, [40](#)
- respond_activity_task_completed, [40](#)
- respond_activity_task_failed, [40](#)
- respond_decision_task_completed, [40](#)
- schemas, [24](#)
- search, [24](#)
- search_place_index_for_position, [14](#)
- search_place_index_for_suggestions, [14](#)
- search_place_index_for_text, [14](#)
- search_schemas, [27](#)
- send_message, [37](#)
- send_message_batch, [37](#)
- send_task_failure, [30](#)
- send_task_heartbeat, [30](#)
- send_task_success, [30](#)
- set_endpoint_attributes, [33](#)
- set_platform_application_attributes, [33](#)
- set_queue_attributes, [37](#)
- set_sms_attributes, [33](#)
- set_subscription_attributes, [33](#)
- set_topic_attributes, [33](#)
- sfn, [27](#)
- signal_workflow_execution, [40](#)
- sns, [30](#)
- sqs, [34](#)
- start_discoverer, [27](#)
- start_execution, [30](#)
- start_job, [14](#)
- start_message_move_task, [37](#)
- start_pipe, [8](#)
- start_replay, [5](#)
- start_sync_execution, [30](#)
- start_workflow_execution, [40](#)
- stop_discoverer, [27](#)
- stop_execution, [30](#)
- stop_pipe, [8](#)
- subscribe, [33](#)
- swf, [37](#)
- tag_queue, [37](#)
- tag_resource, [5](#), [8](#), [11](#), [14](#), [18](#), [20](#), [24](#), [27](#), [30](#), [33](#), [40](#)
- terminate_workflow_execution, [40](#)
- test_event_pattern, [5](#)
- test_state, [30](#)
- undeprecate_activity_type, [40](#)
- undeprecate_domain, [40](#)
- undeprecate_workflow_type, [40](#)
- unsubscribe, [33](#)
- untag_queue, [37](#)
- untag_resource, [5](#), [8](#), [11](#), [14](#), [18](#), [20](#), [24](#), [27](#), [30](#), [33](#), [40](#)
- update_api_destination, [5](#)
- update_archive, [5](#)
- update_broker, [17](#)
- update_configuration, [17](#)
- update_connection, [5](#)
- update_discoverer, [27](#)
- update_endpoint, [6](#)
- update_environment, [18](#), [20](#)
- update_event_bus, [6](#)
- update_geofence_collection, [14](#)
- update_index_type, [24](#)

- update_key, [14](#)
- update_map, [14](#)
- update_map_run, [30](#)
- update_pipe, [8](#)
- update_place_index, [15](#)
- update_registry, [27](#)
- update_route_calculator, [15](#)
- update_schedule, [11](#)
- update_schema, [27](#)
- update_state_machine, [30](#)
- update_state_machine_alias, [30](#)
- update_tracker, [15](#)
- update_user, [17](#)
- update_view, [24](#)

- validate_state_machine_definition, [30](#)
- verify_device_position, [15](#)
- verify_sms_sandbox_phone_number, [33](#)