

Package ‘paws.database’

September 11, 2023

Title 'Amazon Web Services' Database Services

Version 0.4.0

Description Interface to 'Amazon Web Services' database services, including 'Relational Database Service' ('RDS'), 'DynamoDB' 'NoSQL' database, and more <<https://aws.amazon.com/>>.

License Apache License (>= 2.0)

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

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

Imports paws.common (>= 0.6.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.2.3

Collate 'dax_service.R' 'dax_interfaces.R' 'dax_operations.R'
'docdb_service.R' 'docdb_interfaces.R' 'docdb_operations.R'
'dynamodb_service.R' 'dynamodb_interfaces.R'
'dynamodb_operations.R' 'dynamodbstreams_service.R'
'dynamodbstreams_interfaces.R' 'dynamodbstreams_operations.R'
'elasticache_service.R' 'elasticache_interfaces.R'
'elasticache_operations.R' 'keyspaces_service.R'
'keyspaces_interfaces.R' 'keyspaces_operations.R'
'lakeformation_service.R' 'lakeformation_interfaces.R'
'lakeformation_operations.R' 'memorydb_service.R'
'memorydb_interfaces.R' 'memorydb_operations.R'
'neptune_service.R' 'neptune_interfaces.R'
'neptune_operations.R' 'qldb_service.R' 'qldb_interfaces.R'
'qldb_operations.R' 'qldb_session_service.R'
'qldb_session_interfaces.R' 'qldb_session_operations.R'
'rds_service.R' 'rds_operations.R' 'rds_custom.R'
'rds_interfaces.R' 'rdsdataservice_service.R'
'rdsdataservice_interfaces.R' 'rdsdataservice_operations.R'
'redshift_service.R' 'redshift_interfaces.R'
'redshift_operations.R' 'redshiftdataapiservice_service.R'

'redshiftdataapiservice_interfaces.R'
 'redshiftdataapiservice_operations.R'
 'redshiftserverless_service.R'
 'redshiftserverless_interfaces.R'
 'redshiftserverless_operations.R' 'reexports_paws.common.R'
 'simplifiedb_service.R' 'simplifiedb_interfaces.R'
 'simplifiedb_operations.R' 'timestreamquery_service.R'
 'timestreamquery_interfaces.R' 'timestreamquery_operations.R'
 'timestreamwrite_service.R' 'timestreamwrite_interfaces.R'
 'timestreamwrite_operations.R'

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-09-11 18:10:05 UTC

R topics documented:

| | |
|----------------------------------|----|
| dax | 3 |
| docdb | 5 |
| dynamodb | 9 |
| dynamodbstreams | 13 |
| elasticsearch | 15 |
| keyspaces | 19 |
| lakeformation | 22 |
| memorydb | 25 |
| neptune | 28 |
| qldb | 32 |
| qldbsession | 34 |
| rds | 37 |
| rdsdataservice | 43 |
| redshift | 45 |
| redshiftdataapiservice | 50 |
| redshiftserverless | 53 |
| simplifiedb | 56 |
| timestreamquery | 58 |
| timestreamwrite | 61 |

Index

65

dax

*Amazon DynamoDB Accelerator (DAX)***Description**

DAX is a managed caching service engineered for Amazon DynamoDB. DAX dramatically speeds up database reads by caching frequently-accessed data from DynamoDB, so applications can access that data with sub-millisecond latency. You can create a DAX cluster easily, using the AWS Management Console. With a few simple modifications to your code, your application can begin taking advantage of the DAX cluster and realize significant improvements in read performance.

Usage

```
dax(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>

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

Operations

- | | |
|--|-------------------------------|
| create_cluster | Creates a DAX cluster |
| create_parameter_group | Creates a new parameter group |
| create_subnet_group | Creates a new subnet group |

| | |
|--|--|
| <code>decrease_replication_factor</code> | Removes one or more nodes from a DAX cluster |
| <code>delete_cluster</code> | Deletes a previously provisioned DAX cluster |
| <code>delete_parameter_group</code> | Deletes the specified parameter group |
| <code>delete_subnet_group</code> | Deletes a subnet group |
| <code>describe_clusters</code> | Returns information about all provisioned DAX clusters if no cluster identifier is specified, or a |
| <code>describe_default_parameters</code> | Returns the default system parameter information for the DAX caching software |
| <code>describe_events</code> | Returns events related to DAX clusters and parameter groups |
| <code>describe_parameter_groups</code> | Returns a list of parameter group descriptions |
| <code>describe_parameters</code> | Returns the detailed parameter list for a particular parameter group |
| <code>describe_subnet_groups</code> | Returns a list of subnet group descriptions |
| <code>increase_replication_factor</code> | Adds one or more nodes to a DAX cluster |
| <code>list_tags</code> | List all of the tags for a DAX cluster |
| <code>reboot_node</code> | Reboots a single node of a DAX cluster |
| <code>tag_resource</code> | Associates a set of tags with a DAX resource |
| <code>untag_resource</code> | Removes the association of tags from a DAX resource |
| <code>update_cluster</code> | Modifies the settings for a DAX cluster |
| <code>update_parameter_group</code> | Modifies the parameters of a parameter group |
| <code>update_subnet_group</code> | Modifies an existing subnet group |

Examples

```
## Not run:
svc <- dax()
svc$create_cluster(
  Foo = 123
)

## End(Not run)
```

docdb

Amazon DocumentDB with MongoDB compatibility

Description

Amazon DocumentDB is a fast, reliable, and fully managed database service. Amazon DocumentDB makes it easy to set up, operate, and scale MongoDB-compatible databases in the cloud. With Amazon DocumentDB, you can run the same application code and use the same drivers and tools that you use with MongoDB.

Usage

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

Arguments

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

Value

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

Service syntax

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

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
    creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

| | |
|---|--|
| add_source_identifier_to_subscription | Adds a source identifier to an existing event notification subscription |
| add_tags_to_resource | Adds metadata tags to an Amazon DocumentDB resource |
| apply_pending_maintenance_action | Applies a pending maintenance action to a resource (for example, to an Amazon DocumentDB instance) |
| copy_db_cluster_parameter_group | Copies the specified cluster parameter group |
| copy_db_cluster_snapshot | Copies a snapshot of a cluster |
| create_db_cluster | Creates a new Amazon DocumentDB cluster |
| create_db_cluster_parameter_group | Creates a new cluster parameter group |
| create_db_cluster_snapshot | Creates a snapshot of a cluster |
| create_db_instance | Creates a new instance |
| create_db_subnet_group | Creates a new subnet group |
| create_event_subscription | Creates an Amazon DocumentDB event notification subscription |
| create_global_cluster | Creates an Amazon DocumentDB global cluster that can span multiple multiple Availability Zones |
| delete_db_cluster | Deletes a previously provisioned cluster |
| delete_db_cluster_parameter_group | Deletes a specified cluster parameter group |
| delete_db_cluster_snapshot | Deletes a cluster snapshot |
| delete_db_instance | Deletes a previously provisioned instance |
| delete_db_subnet_group | Deletes a subnet group |
| delete_event_subscription | Deletes an Amazon DocumentDB event notification subscription |
| delete_global_cluster | Deletes a global cluster |
| describe_certificates | Returns a list of certificate authority (CA) certificates provided by Amazon DocumentDB |

| | |
|---|--|
| <code>describe_db_cluster_parameter_groups</code> | Returns a list of DBClusterParameterGroup descriptions |
| <code>describe_db_cluster_parameters</code> | Returns the detailed parameter list for a particular cluster parameter group |
| <code>describe_db_clusters</code> | Returns information about provisioned Amazon DocumentDB clusters |
| <code>describe_db_cluster_snapshot_attributes</code> | Returns a list of cluster snapshot attribute names and values for a manual DB cluster snapshot |
| <code>describe_db_cluster_snapshots</code> | Returns information about cluster snapshots |
| <code>describe_db_engine_versions</code> | Returns a list of the available engines |
| <code>describe_db_instances</code> | Returns information about provisioned Amazon DocumentDB instances |
| <code>describe_db_subnet_groups</code> | Returns a list of DBSubnetGroup descriptions |
| <code>describe_engine_default_cluster_parameters</code> | Returns the default engine and system parameter information for the cluster default parameter group |
| <code>describe_event_categories</code> | Displays a list of categories for all event source types, or, if specified, for a specific event source type |
| <code>describe_events</code> | Returns events related to instances, security groups, snapshots, and DB parameter groups |
| <code>describe_event_subscriptions</code> | Lists all the subscription descriptions for a customer account |
| <code>describe_global_clusters</code> | Returns information about Amazon DocumentDB global clusters |
| <code>describe_orderable_db_instance_options</code> | Returns a list of orderable instance options for the specified engine |
| <code>describe_pending_maintenance_actions</code> | Returns a list of resources (for example, instances) that have at least one pending maintenance action |
| <code>failover_db_cluster</code> | Forces a failover for a cluster |
| <code>list_tags_for_resource</code> | Lists all tags on an Amazon DocumentDB resource |
| <code>modify_db_cluster</code> | Modifies a setting for an Amazon DocumentDB cluster |
| <code>modify_db_cluster_parameter_group</code> | Modifies the parameters of a cluster parameter group |
| <code>modify_db_cluster_snapshot_attribute</code> | Adds an attribute and values to, or removes an attribute and values from, a manual DB cluster snapshot |
| <code>modify_db_instance</code> | Modifies settings for an instance |
| <code>modify_db_subnet_group</code> | Modifies an existing subnet group |
| <code>modify_event_subscription</code> | Modifies an existing Amazon DocumentDB event notification subscription |
| <code>modify_global_cluster</code> | Modify a setting for an Amazon DocumentDB global cluster |
| <code>reboot_db_instance</code> | You might need to reboot your instance, usually for maintenance reasons |
| <code>remove_from_global_cluster</code> | Detaches an Amazon DocumentDB secondary cluster from a global cluster |
| <code>remove_source_identifier_from_subscription</code> | Removes a source identifier from an existing Amazon DocumentDB event notification subscription |
| <code>remove_tags_from_resource</code> | Removes metadata tags from an Amazon DocumentDB resource |
| <code>reset_db_cluster_parameter_group</code> | Modifies the parameters of a cluster parameter group to the default value |
| <code>restore_db_cluster_from_snapshot</code> | Creates a new cluster from a snapshot or cluster snapshot |
| <code>restore_db_cluster_to_point_in_time</code> | Restores a cluster to an arbitrary point in time |
| <code>start_db_cluster</code> | Restarts the stopped cluster that is specified by DBClusterIdentifier |
| <code>stop_db_cluster</code> | Stops the running cluster that is specified by DBClusterIdentifier |

Examples

```
## Not run:
svc <- docdb()
svc$add_source_identifier_to_subscription(
  Foo = 123
)

## End(Not run)
```


dynamodb

*Amazon DynamoDB***Description**

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation, and use the Amazon Web Services Management Console to monitor resource utilization and performance metrics.

DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements, while maintaining consistent and fast performance. All of your data is stored on solid state disks (SSDs) and automatically replicated across multiple Availability Zones in an Amazon Web Services Region, providing built-in high availability and data durability.

Usage

```
dynamodb(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>

| | |
|-------------|---|
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

    region = "string"
)

```

Operations

| | |
|--|---|
| batch_execute_statement | This operation allows you to perform batch reads or writes on data stored in DynamoDB |
| batch_get_item | The BatchGetItem operation returns the attributes of one or more items from one or more tables |
| batch_write_item | The BatchWriteItem operation puts or deletes multiple items in one or more tables |
| create_backup | Creates a backup for an existing table |
| create_global_table | Creates a global table from an existing table |
| create_table | The CreateTable operation adds a new table to your account |
| delete_backup | Deletes an existing backup of a table |
| delete_item | Deletes a single item in a table by primary key |
| delete_table | The DeleteTable operation deletes a table and all of its items |
| describe_backup | Describes an existing backup of a table |
| describe_continuous_backups | Checks the status of continuous backups and point in time recovery on the specified table |
| describe_contributor_insights | Returns information about contributor insights for a given table or global secondary index |
| describe_endpoints | Returns the regional endpoint information |
| describe_export | Describes an existing table export |
| describe_global_table | Returns information about the specified global table |
| describe_global_table_settings | Describes Region-specific settings for a global table |
| describe_import | Represents the properties of the import |
| describe_kinesis_streaming_destination | Returns information about the status of Kinesis streaming |
| describe_limits | Returns the current provisioned-capacity quotas for your Amazon Web Services account |
| describe_table | Returns information about the table, including the current status of the table, when in a pending state |
| describe_table_replica_auto_scaling | Describes auto scaling settings across replicas of the global table at once |
| describe_time_to_live | Gives a description of the Time to Live (TTL) status on the specified table |
| disable_kinesis_streaming_destination | Stops replication from the DynamoDB table to the Kinesis data stream |
| enable_kinesis_streaming_destination | Starts table data replication to the specified Kinesis data stream at a timestamp chosen by the user |
| execute_statement | This operation allows you to perform reads and singleton writes on data stored in DynamoDB |
| execute_transaction | This operation allows you to perform transactional reads or writes on data stored in DynamoDB |
| export_table_to_point_in_time | Exports table data to an S3 bucket |
| get_item | The GetItem operation returns a set of attributes for the item with the given primary key |
| import_table | Imports table data from an S3 bucket |
| list_backups | List backups associated with an Amazon Web Services account |
| list_contributor_insights | Returns a list of ContributorInsightsSummary for a table and all its global secondary indexes |
| list_exports | Lists completed exports within the past 90 days |
| list_global_tables | Lists all global tables that have a replica in the specified Region |
| list_imports | Lists completed imports within the past 90 days |
| list_tables | Returns an array of table names associated with the current account and endpoint |
| list_tags_of_resource | List all tags on an Amazon DynamoDB resource |
| put_item | Creates a new item, or replaces an old item with a new item |
| query | You must provide the name of the partition key attribute and a single value for that attribute |
| restore_table_from_backup | Creates a new table from an existing backup |
| restore_table_to_point_in_time | Restores the specified table to the specified point in time within EarliestRestorableTime to LatestRestorableTime |
| scan | The Scan operation returns one or more items and item attributes by accessing every item in the table |
| tag_resource | Associate a set of tags with an Amazon DynamoDB resource |
| transact_get_items | TransactGetItems is a synchronous operation that atomically retrieves multiple items from a table |

| | |
|---|---|
| transact_write_items | TransactWriteItems is a synchronous write operation that groups up to 100 action requests into a single atomic operation. |
| untag_resource | Removes the association of tags from an Amazon DynamoDB resource. |
| update_continuous_backups | UpdateContinuousBackups enables or disables point in time recovery for the specified table or index. |
| update_contributor_insights | Updates the status for contributor insights for a specific table or index. |
| update_global_table | Adds or removes replicas in the specified global table. |
| update_global_table_settings | Updates settings for a global table. |
| update_item | Edits an existing item's attributes, or adds a new item to the table if it does not already exist. |
| update_table | Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams on a table. |
| update_table_replica_auto_scaling | Updates auto scaling settings on your global tables at once. |
| update_time_to_live | The UpdateTimeToLive method enables or disables Time to Live (TTL) for the specified table. |

Examples

```
## Not run:
svc <- dynamodb()
# This example reads multiple items from the Music table using a batch of
# three GetItem requests. Only the AlbumTitle attribute is returned.
svc$batch_get_item(
  RequestItems = list(
    Music = list(
      Keys = list(
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Call Me Today"
          )
        ),
        list(
          Artist = list(
            S = "Acme Band"
          ),
          SongTitle = list(
            S = "Happy Day"
          )
        ),
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Scared of My Shadow"
          )
        )
      ),
      ProjectionExpression = "AlbumTitle"
    )
  )
)
```

```
)
## End(Not run)
```

dynamodbstreams *Amazon DynamoDB Streams*

Description

Amazon DynamoDB

Amazon DynamoDB Streams provides API actions for accessing streams and processing stream records. To learn more about application development with Streams, see [Capturing Table Activity with DynamoDB Streams](#) in the Amazon DynamoDB Developer Guide.

Usage

```
dynamodbstreams(
  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>

| | |
|-------------|---|
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```
    region = "string"
  )
```

Operations

| | |
|------------------------------------|--|
| describe_stream | Returns information about a stream, including the current status of the stream, its Amazon Resource Name |
| get_records | Retrieves the stream records from a given shard |
| get_shard_iterator | Returns a shard iterator |
| list_streams | Returns an array of stream ARNs associated with the current account and endpoint |

Examples

```
## Not run:
svc <- dynamodbstreams()
# The following example describes a stream with a given stream ARN.
svc$describe_stream(
  StreamArn = "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/stream/2..."
)

## End(Not run)
```

elasticache

Amazon ElastiCache

Description

Amazon ElastiCache is a web service that makes it easier to set up, operate, and scale a distributed cache in the cloud.

With ElastiCache, customers get all of the benefits of a high-performance, in-memory cache with less of the administrative burden involved in launching and managing a distributed cache. The service makes setup, scaling, and cluster failure handling much simpler than in a self-managed cache deployment.

In addition, through integration with Amazon CloudWatch, customers get enhanced visibility into the key performance statistics associated with their cache and can receive alarms if a part of their cache runs hot.

Usage

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

Arguments

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

Value

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

Service syntax

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



```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

| | |
|--|--|
| add_tags_to_resource | A tag is a key-value pair where the key and value are case-sensitive |
| authorize_cache_security_group_ingress | Allows network ingress to a cache security group |
| batch_apply_update_action | Apply the service update |
| batch_stop_update_action | Stop the service update |
| complete_migration | Complete the migration of data |
| copy_snapshot | Makes a copy of an existing snapshot |
| create_cache_cluster | Creates a cluster |
| create_cache_parameter_group | Creates a new Amazon ElastiCache cache parameter group |
| create_cache_security_group | Creates a new cache security group |
| create_cache_subnet_group | Creates a new cache subnet group |
| create_global_replication_group | Global Datastore for Redis offers fully managed, fast, reliable and secure |
| create_replication_group | Creates a Redis (cluster mode disabled) or a Redis (cluster mode enabled) |
| create_snapshot | Creates a copy of an entire cluster or replication group at a specific moment |
| create_user | For Redis engine version 6 |
| create_user_group | For Redis engine version 6 |
| decrease_node_groups_in_global_replication_group | Decreases the number of node groups in a Global datastore |
| decrease_replica_count | Dynamically decreases the number of replicas in a Redis (cluster mode enabled) |
| delete_cache_cluster | Deletes a previously provisioned cluster |
| delete_cache_parameter_group | Deletes the specified cache parameter group |
| delete_cache_security_group | Deletes a cache security group |

| | |
|---|--|
| <code>delete_cache_subnet_group</code> | Deletes a cache subnet group |
| <code>delete_global_replication_group</code> | Deleting a Global datastore is a two-step process: |
| <code>delete_replication_group</code> | Deletes an existing replication group |
| <code>delete_snapshot</code> | Deletes an existing snapshot |
| <code>delete_user</code> | For Redis engine version 6 |
| <code>delete_user_group</code> | For Redis engine version 6 |
| <code>describe_cache_clusters</code> | Returns information about all provisioned clusters if no cluster identifier is provided |
| <code>describe_cache_engine_versions</code> | Returns a list of the available cache engines and their versions |
| <code>describe_cache_parameter_groups</code> | Returns a list of cache parameter group descriptions |
| <code>describe_cache_parameters</code> | Returns the detailed parameter list for a particular cache parameter group |
| <code>describe_cache_security_groups</code> | Returns a list of cache security group descriptions |
| <code>describe_cache_subnet_groups</code> | Returns a list of cache subnet group descriptions |
| <code>describe_engine_default_parameters</code> | Returns the default engine and system parameter information for the specified engine |
| <code>describe_events</code> | Returns events related to clusters, cache security groups, and cache parameter groups |
| <code>describe_global_replication_groups</code> | Returns information about a particular global replication group |
| <code>describe_replication_groups</code> | Returns information about a particular replication group |
| <code>describe_reserved_cache_nodes</code> | Returns information about reserved cache nodes for this account, or about all reserved cache nodes |
| <code>describe_reserved_cache_nodes_offerings</code> | Lists available reserved cache node offerings |
| <code>describe_service_updates</code> | Returns details of the service updates |
| <code>describe_snapshots</code> | Returns information about cluster or replication group snapshots |
| <code>describe_update_actions</code> | Returns details of the update actions |
| <code>describe_user_groups</code> | Returns a list of user groups |
| <code>describe_users</code> | Returns a list of users |
| <code>disassociate_global_replication_group</code> | Remove a secondary cluster from the Global datastore using the Global Datastore ID |
| <code>failover_global_replication_group</code> | Used to failover the primary region to a secondary region |
| <code>increase_node_groups_in_global_replication_group</code> | Increase the number of node groups in the Global datastore |
| <code>increase_replica_count</code> | Dynamically increases the number of replicas in a Redis (cluster mode) replication group |
| <code>list_allowed_node_type_modifications</code> | Lists all available node types that you can scale your Redis cluster's or replication group's nodes to |
| <code>list_tags_for_resource</code> | Lists all tags currently on a named resource |
| <code>modify_cache_cluster</code> | Modifies the settings for a cluster |
| <code>modify_cache_parameter_group</code> | Modifies the parameters of a cache parameter group |
| <code>modify_cache_subnet_group</code> | Modifies an existing cache subnet group |
| <code>modify_global_replication_group</code> | Modifies the settings for a Global datastore |
| <code>modify_replication_group</code> | Modifies the settings for a replication group |
| <code>modify_replication_group_shard_configuration</code> | Modifies a replication group's shards (node groups) by allowing you to add or remove shards |
| <code>modify_user</code> | Changes user password(s) and/or access string |
| <code>modify_user_group</code> | Changes the list of users that belong to the user group |
| <code>purchase_reserved_cache_nodes_offering</code> | Allows you to purchase a reserved cache node offering |
| <code>rebalance_slots_in_global_replication_group</code> | Redistribute slots to ensure uniform distribution across existing shards |
| <code>reboot_cache_cluster</code> | Reboots some, or all, of the cache nodes within a provisioned cluster |
| <code>remove_tags_from_resource</code> | Removes the tags identified by the TagKeys list from the named resource |
| <code>reset_cache_parameter_group</code> | Modifies the parameters of a cache parameter group to the engine or system default |
| <code>revoke_cache_security_group_ingress</code> | Revokes ingress from a cache security group |
| <code>start_migration</code> | Start the migration of data |
| <code>test_failover</code> | Represents the input of a TestFailover operation which tests automatic failover |
| <code>test_migration</code> | Async API to test connection between source and target replication groups |

Examples

```
## Not run:
svc <- elasticache()
svc$add_tags_to_resource(
  Foo = 123
)

## End(Not run)
```

keyspaces

Amazon Keyspaces

Description

Amazon Keyspaces (for Apache Cassandra) is a scalable, highly available, and managed Apache Cassandra-compatible database service. Amazon Keyspaces makes it easy to migrate, run, and scale Cassandra workloads in the Amazon Web Services Cloud. With just a few clicks on the Amazon Web Services Management Console or a few lines of code, you can create keyspaces and tables in Amazon Keyspaces, without deploying any infrastructure or installing software.

In addition to supporting Cassandra Query Language (CQL) requests via open-source Cassandra drivers, Amazon Keyspaces supports data definition language (DDL) operations to manage keyspaces and tables using the Amazon Web Services SDK and CLI, as well as infrastructure as code (IaC) services and tools such as CloudFormation and Terraform. This API reference describes the supported DDL operations in detail.

For the list of all supported CQL APIs, see [Supported Cassandra APIs, operations, and data types in Amazon Keyspaces](#) in the *Amazon Keyspaces Developer Guide*.

To learn how Amazon Keyspaces API actions are recorded with CloudTrail, see [Amazon Keyspaces information in CloudTrail](#) in the *Amazon Keyspaces Developer Guide*.

For more information about Amazon Web Services APIs, for example how to implement retry logic or how to sign Amazon Web Services API requests, see [Amazon Web Services APIs](#) in the *General Reference*.

Usage

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

Arguments

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

Value

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

Service syntax

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

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

| | |
|--|--|
| create_keyspace | The CreateKeyspace operation adds a new keyspace to your account |
| create_table | The CreateTable operation adds a new table to the specified keyspace |
| delete_keyspace | The DeleteKeyspace operation deletes a keyspace and all of its tables |
| delete_table | The DeleteTable operation deletes a table and all of its data |
| get_keyspace | Returns the name and the Amazon Resource Name (ARN) of the specified table |
| get_table | Returns information about the table, including the table's name and current status, the keyspace name |
| list_keyspaces | Returns a list of keyspaces |
| list_tables | Returns a list of tables for a specified keyspace |
| list_tags_for_resource | Returns a list of all tags associated with the specified Amazon Keyspaces resource |
| restore_table | Restores the specified table to the specified point in time within the earliest_restorable_timestamp and |
| tag_resource | Associates a set of tags with a Amazon Keyspaces resource |
| untag_resource | Removes the association of tags from a Amazon Keyspaces resource |
| update_table | Adds new columns to the table or updates one of the table's settings, for example capacity mode, enc |

Examples

```

## Not run:
svc <- keyspaces()

```

```

svc$create_keyspace(
  Foo = 123
)

## End(Not run)

```

lakeformation

AWS Lake Formation

Description

Lake Formation

Defines the public endpoint for the Lake Formation service.

Usage

```

lakeformation(
  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>

| | |
|-------------|---|
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

    region = "string"
)

```

Operations

| | |
|--|---|
| add_lf_tags_to_resource | Attaches one or more LF-tags to an existing resource |
| assume_decorated_role_with_saml | Allows a caller to assume an IAM role decorated as the SAML user specified in the request |
| batch_grant_permissions | Batch operation to grant permissions to the principal |
| batch_revoke_permissions | Batch operation to revoke permissions from the principal |
| cancel_transaction | Attempts to cancel the specified transaction |
| commit_transaction | Attempts to commit the specified transaction |
| create_data_cells_filter | Creates a data cell filter to allow one to grant access to certain columns on certain tables |
| create_lf_tag | Creates an LF-tag with the specified name and values |
| delete_data_cells_filter | Deletes a data cell filter |
| delete_lf_tag | Deletes the specified LF-tag given a key name |
| delete_objects_on_cancel | For a specific governed table, provides a list of Amazon S3 objects that will be written to the table |
| deregister_resource | Deregisters the resource as managed by the Data Catalog |
| describe_resource | Retrieves the current data access role for the given resource registered in Lake Formation |
| describe_transaction | Returns the details of a single transaction |
| extend_transaction | Indicates to the service that the specified transaction is still active and should not be canceled |
| get_data_cells_filter | Returns a data cells filter |
| get_data_lake_settings | Retrieves the list of the data lake administrators of a Lake Formation-managed data lake |
| get_effective_permissions_for_path | Returns the Lake Formation permissions for a specified table or database resource |
| get_lf_tag | Returns an LF-tag definition |
| get_query_state | Returns the state of a query previously submitted |
| get_query_statistics | Retrieves statistics on the planning and execution of a query |
| get_resource_lf_tags | Returns the LF-tags applied to a resource |
| get_table_objects | Returns the set of Amazon S3 objects that make up the specified governed table |
| get_temporary_glue_partition_credentials | This API is identical to GetTemporaryTableCredentials except that this is used with partitions |
| get_temporary_glue_table_credentials | Allows a caller in a secure environment to assume a role with permission to access the table |
| get_work_unit_results | Returns the work units resulting from the query |
| get_work_units | Retrieves the work units generated by the StartQueryPlanning operation |
| grant_permissions | Grants permissions to the principal to access metadata in the Data Catalog and data |
| list_data_cells_filter | Lists all the data cell filters on a table |
| list_lf_tags | Lists LF-tags that the requester has permission to view |
| list_permissions | Returns a list of the principal permissions on the resource, filtered by the permissions |
| list_resources | Lists the resources registered to be managed by the Data Catalog |
| list_table_storage_optimizers | Returns the configuration of all storage optimizers associated with a specified table |
| list_transactions | Returns metadata about transactions and their status |
| put_data_lake_settings | Sets the list of data lake administrators who have admin privileges on all resources |
| register_resource | Registers the resource as managed by the Data Catalog |
| remove_lf_tags_from_resource | Removes an LF-tag from the resource |
| revoke_permissions | Revokes permissions to the principal to access metadata in the Data Catalog and data |
| search_databases_by_lf_tags | This operation allows a search on DATABASE resources by TagCondition |
| search_tables_by_lf_tags | This operation allows a search on TABLE resources by LFTags |
| start_query_planning | Submits a request to process a query statement |
| start_transaction | Starts a new transaction and returns its transaction ID |
| update_data_cells_filter | Updates a data cell filter |

| | |
|--|---|
| update_lf_tag | Updates the list of possible values for the specified LF-tag key |
| update_resource | Updates the data access role used for vending access to the given (registered) resource |
| update_table_objects | Updates the manifest of Amazon S3 objects that make up the specified governed table |
| update_table_storage_optimizer | Updates the configuration of the storage optimizers for a table |

Examples

```
## Not run:
svc <- lakeformation()
svc$add_lf_tags_to_resource(
  Foo = 123
)

## End(Not run)
```

memorydb

Amazon MemoryDB

Description

MemoryDB for Redis is a fully managed, Redis-compatible, in-memory database that delivers ultra-fast performance and Multi-AZ durability for modern applications built using microservices architectures. MemoryDB stores the entire database in-memory, enabling low latency and high throughput data access. It is compatible with Redis, a popular open source data store, enabling you to leverage Redis' flexible and friendly data structures, APIs, and commands.

Usage

```
memorydb(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.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

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

```

Operations

| | |
|---|--|
| batch_update_cluster | Apply the service update to a list of clusters supplied |
| copy_snapshot | Makes a copy of an existing snapshot |
| create_acl | Creates an Access Control List |
| create_cluster | Creates a cluster |
| create_parameter_group | Creates a new MemoryDB parameter group |
| create_snapshot | Creates a copy of an entire cluster at a specific moment in time |
| create_subnet_group | Creates a subnet group |
| create_user | Creates a MemoryDB user |
| delete_acl | Deletes an Access Control List |
| delete_cluster | Deletes a cluster |
| delete_parameter_group | Deletes the specified parameter group |
| delete_snapshot | Deletes an existing snapshot |
| delete_subnet_group | Deletes a subnet group |
| delete_user | Deletes a user |
| describe_ac_ls | Returns a list of ACLs |
| describe_clusters | Returns information about all provisioned clusters if no cluster identifier is specified, or |
| describe_engine_versions | Returns a list of the available Redis engine versions |
| describe_events | Returns events related to clusters, security groups, and parameter groups |
| describe_parameter_groups | Returns a list of parameter group descriptions |
| describe_parameters | Returns the detailed parameter list for a particular parameter group |
| describe_reserved_nodes | Returns information about reserved nodes for this account, or about a specified reserved |
| describe_reserved_nodes_offerings | Lists available reserved node offerings |
| describe_service_updates | Returns details of the service updates |
| describe_snapshots | Returns information about cluster snapshots |
| describe_subnet_groups | Returns a list of subnet group descriptions |
| describe_users | Returns a list of users |
| failover_shard | Used to failover a shard |
| list_allowed_node_type_updates | Lists all available node types that you can scale to from your cluster's current node type |
| list_tags | Lists all tags currently on a named resource |
| purchase_reserved_nodes_offering | Allows you to purchase a reserved node offering |
| reset_parameter_group | Modifies the parameters of a parameter group to the engine or system default value |
| tag_resource | A tag is a key-value pair where the key and value are case-sensitive |
| untag_resource | Use this operation to remove tags on a resource |
| update_acl | Changes the list of users that belong to the Access Control List |

| | |
|-------------------------------------|---|
| <code>update_cluster</code> | Modifies the settings for a cluster |
| <code>update_parameter_group</code> | Updates the parameters of a parameter group |
| <code>update_subnet_group</code> | Updates a subnet group |
| <code>update_user</code> | Changes user password(s) and/or access string |

Examples

```
## Not run:
svc <- memorydb()
svc$batch_update_cluster(
  Foo = 123
)

## End(Not run)
```

neptune

Amazon Neptune

Description

Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications that work with highly connected datasets. The core of Amazon Neptune is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Amazon Neptune supports popular graph models Property Graph and W3C's RDF, and their respective query languages Apache TinkerPop Gremlin and SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets. Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security.

This interface reference for Amazon Neptune contains documentation for a programming or command line interface you can use to manage Amazon Neptune. Note that Amazon Neptune is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

Usage

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

Arguments

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

Value

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

Service syntax

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

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
    creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

| | |
|---|--|
| add_role_to_db_cluster | Associates an Identity and Access Management (IAM) role with an Neptune DB cluster |
| add_source_identifier_to_subscription | Adds a source identifier to an existing event notification subscription |
| add_tags_to_resource | Adds metadata tags to an Amazon Neptune resource |
| apply_pending_maintenance_action | Applies a pending maintenance action to a resource (for example, to a DB instance) |
| copy_db_cluster_parameter_group | Copies the specified DB cluster parameter group |
| copy_db_cluster_snapshot | Copies a snapshot of a DB cluster |
| copy_db_parameter_group | Copies the specified DB parameter group |
| create_db_cluster | Creates a new Amazon Neptune DB cluster |
| create_db_cluster_endpoint | Creates a new custom endpoint and associates it with an Amazon Neptune DB cluster |
| create_db_cluster_parameter_group | Creates a new DB cluster parameter group |
| create_db_cluster_snapshot | Creates a snapshot of a DB cluster |
| create_db_instance | Creates a new DB instance |
| create_db_parameter_group | Creates a new DB parameter group |
| create_db_subnet_group | Creates a new DB subnet group |
| create_event_subscription | Creates an event notification subscription |
| create_global_cluster | Creates a Neptune global database spread across multiple Amazon Regions |
| delete_db_cluster | The DeleteDBCluster action deletes a previously provisioned DB cluster |
| delete_db_cluster_endpoint | Deletes a custom endpoint and removes it from an Amazon Neptune DB cluster |
| delete_db_cluster_parameter_group | Deletes a specified DB cluster parameter group |
| delete_db_cluster_snapshot | Deletes a DB cluster snapshot |

| | |
|---|---|
| <code>delete_db_instance</code> | The DeleteDBInstance action deletes a previously provisioned DB instance |
| <code>delete_db_parameter_group</code> | Deletes a specified DBParameterGroup |
| <code>delete_db_subnet_group</code> | Deletes a DB subnet group |
| <code>delete_event_subscription</code> | Deletes an event notification subscription |
| <code>delete_global_cluster</code> | Deletes a global database |
| <code>describe_db_cluster_endpoints</code> | Returns information about endpoints for an Amazon Neptune DB cluster |
| <code>describe_db_cluster_parameter_groups</code> | Returns a list of DBClusterParameterGroup descriptions |
| <code>describe_db_cluster_parameters</code> | Returns the detailed parameter list for a particular DB cluster parameter group |
| <code>describe_db_clusters</code> | Returns information about provisioned DB clusters, and supports pagination |
| <code>describe_db_cluster_snapshot_attributes</code> | Returns a list of DB cluster snapshot attribute names and values for a manual I |
| <code>describe_db_cluster_snapshots</code> | Returns information about DB cluster snapshots |
| <code>describe_db_engine_versions</code> | Returns a list of the available DB engines |
| <code>describe_db_instances</code> | Returns information about provisioned instances, and supports pagination |
| <code>describe_db_parameter_groups</code> | Returns a list of DBParameterGroup descriptions |
| <code>describe_db_parameters</code> | Returns the detailed parameter list for a particular DB parameter group |
| <code>describe_db_subnet_groups</code> | Returns a list of DBSubnetGroup descriptions |
| <code>describe_engine_default_cluster_parameters</code> | Returns the default engine and system parameter information for the cluster da |
| <code>describe_engine_default_parameters</code> | Returns the default engine and system parameter information for the specified |
| <code>describe_event_categories</code> | Displays a list of categories for all event source types, or, if specified, for a spe |
| <code>describe_events</code> | Returns events related to DB instances, DB security groups, DB snapshots, and |
| <code>describe_event_subscriptions</code> | Lists all the subscription descriptions for a customer account |
| <code>describe_global_clusters</code> | Returns information about Neptune global database clusters |
| <code>describe_orderable_db_instance_options</code> | Returns a list of orderable DB instance options for the specified engine |
| <code>describe_pending_maintenance_actions</code> | Returns a list of resources (for example, DB instances) that have at least one p |
| <code>describe_valid_db_instance_modifications</code> | You can call DescribeValidDBInstanceModifications to learn what modificatio |
| <code>failover_db_cluster</code> | Forces a failover for a DB cluster |
| <code>failover_global_cluster</code> | Initiates the failover process for a Neptune global database |
| <code>list_tags_for_resource</code> | Lists all tags on an Amazon Neptune resource |
| <code>modify_db_cluster</code> | Modify a setting for a DB cluster |
| <code>modify_db_cluster_endpoint</code> | Modifies the properties of an endpoint in an Amazon Neptune DB cluster |
| <code>modify_db_cluster_parameter_group</code> | Modifies the parameters of a DB cluster parameter group |
| <code>modify_db_cluster_snapshot_attribute</code> | Adds an attribute and values to, or removes an attribute and values from, a mar |
| <code>modify_db_instance</code> | Modifies settings for a DB instance |
| <code>modify_db_parameter_group</code> | Modifies the parameters of a DB parameter group |
| <code>modify_db_subnet_group</code> | Modifies an existing DB subnet group |
| <code>modify_event_subscription</code> | Modifies an existing event notification subscription |
| <code>modify_global_cluster</code> | Modify a setting for an Amazon Neptune global cluster |
| <code>promote_read_replica_db_cluster</code> | Not supported |
| <code>reboot_db_instance</code> | You might need to reboot your DB instance, usually for maintenance reasons |
| <code>remove_from_global_cluster</code> | Detaches a Neptune DB cluster from a Neptune global database |
| <code>remove_role_from_db_cluster</code> | Disassociates an Identity and Access Management (IAM) role from a DB clus |
| <code>remove_source_identifier_from_subscription</code> | Removes a source identifier from an existing event notification subscription |
| <code>remove_tags_from_resource</code> | Removes metadata tags from an Amazon Neptune resource |
| <code>reset_db_cluster_parameter_group</code> | Modifies the parameters of a DB cluster parameter group to the default value |
| <code>reset_db_parameter_group</code> | Modifies the parameters of a DB parameter group to the engine/system default |
| <code>restore_db_cluster_from_snapshot</code> | Creates a new DB cluster from a DB snapshot or DB cluster snapshot |
| <code>restore_db_cluster_to_point_in_time</code> | Restores a DB cluster to an arbitrary point in time |
| <code>start_db_cluster</code> | Starts an Amazon Neptune DB cluster that was stopped using the Amazon con |

`stop_db_cluster`

Stops an Amazon Neptune DB cluster

Examples

```
## Not run:
svc <- neptune()
svc$add_role_to_db_cluster(
  Foo = 123
)

## End(Not run)
```

qldb

Amazon QLDB

Description

The resource management API for Amazon QLDB

Usage

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

Arguments

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

- **credentials:**
 - **creds:**
 - * **access_key_id:** AWS access key ID
 - * **secret_access_key:** AWS secret access key
 - * **session_token:** AWS temporary session token
 - **profile:** The name of a profile to use. If not given, then the default profile is used.
 - **anonymous:** Set anonymous credentials.
 - **endpoint:** The complete URL to use for the constructed client.
 - **region:** The AWS Region used in instantiating the client.
- **close_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style:** Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

| | |
|---|--|
| cancel_journal_kinesis_stream | Ends a given Amazon QLDB journal stream |
| create_ledger | Creates a new ledger in your Amazon Web Services account in the current Region |
| delete_ledger | Deletes a ledger and all of its contents |
| describe_journal_kinesis_stream | Returns detailed information about a given Amazon QLDB journal stream |
| describe_journal_s3_export | Returns information about a journal export job, including the ledger name, export ID |
| describe_ledger | Returns information about a ledger, including its state, permissions mode, encryption |
| export_journal_to_s3 | Exports journal contents within a date and time range from a ledger into a specified |
| get_block | Returns a block object at a specified address in a journal |
| get_digest | Returns the digest of a ledger at the latest committed block in the journal |
| get_revision | Returns a revision data object for a specified document ID and block address |
| list_journal_kinesis_streams_for_ledger | Returns all Amazon QLDB journal streams for a given ledger |
| list_journal_s3_exports | Returns all journal export jobs for all ledgers that are associated with the current A |
| list_journal_s3_exports_for_ledger | Returns all journal export jobs for a specified ledger |
| list_ledgers | Returns all ledgers that are associated with the current Amazon Web Services accou |
| list_tags_for_resource | Returns all tags for a specified Amazon QLDB resource |
| stream_journal_to_kinesis | Creates a journal stream for a given Amazon QLDB ledger |
| tag_resource | Adds one or more tags to a specified Amazon QLDB resource |
| untag_resource | Removes one or more tags from a specified Amazon QLDB resource |
| update_ledger | Updates properties on a ledger |
| update_ledger_permissions_mode | Updates the permissions mode of a ledger |

Examples

```

## Not run:
svc <- qldb()
svc$cancel_journal_kinesis_stream(
  Foo = 123
)

## End(Not run)

```

Description

The transactional data APIs for Amazon QLDB

Instead of interacting directly with this API, we recommend using the QLDB driver or the QLDB shell to execute data transactions on a ledger.

- If you are working with an AWS SDK, use the QLDB driver. The driver provides a high-level abstraction layer above this *QLDB Session* data plane and manages `send_command` API calls for you. For information and a list of supported programming languages, see [Getting started with the driver](#) in the *Amazon QLDB Developer Guide*.
- If you are working with the AWS Command Line Interface (AWS CLI), use the QLDB shell. The shell is a command line interface that uses the QLDB driver to interact with a ledger. For information, see [Accessing Amazon QLDB using the QLDB shell](#).

Usage

```
qldb-session(
  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>

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

Operations

`send_command` Sends a command to an Amazon QLDB ledger

Examples

```
## Not run:
svc <- qldbsession()
svc$send_command(
  Foo = 123
)

## End(Not run)
```

rds

Amazon Relational Database Service

Description

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizeable capacity for an industry-standard relational database and manages common database administration tasks, freeing up developers to focus on what makes their applications and businesses unique.

Amazon RDS gives you access to the capabilities of a MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, Oracle, or Amazon Aurora database server. These capabilities mean that the code, applications, and tools you already use today with your existing databases work with Amazon RDS without modification. Amazon RDS automatically backs up your database and maintains the database software that powers your DB instance. Amazon RDS is flexible: you can scale your DB instance's compute resources and storage capacity to meet your application's demand. As with all Amazon Web Services, there are no up-front investments, and you pay only for the resources you use.

This interface reference for Amazon RDS contains documentation for a programming or command line interface you can use to manage Amazon RDS. Amazon RDS is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

Amazon RDS API Reference

- For the alphabetical list of API actions, see [API Actions](#).
- For the alphabetical list of data types, see [Data Types](#).
- For a list of common query parameters, see [Common Parameters](#).
- For descriptions of the error codes, see [Common Errors](#).

Amazon RDS User Guide

- For a summary of the Amazon RDS interfaces, see [Available RDS Interfaces](#).
- For more information about how to use the Query API, see [Using the Query API](#).

Usage

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

Arguments

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

Value

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

Service syntax

```

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

```

Operations

[add_role_to_db_cluster](#)
[add_role_to_db_instance](#)
[add_source_identifier_to_subscription](#)
[add_tags_to_resource](#)
[apply_pending_maintenance_action](#)
[authorize_db_security_group_ingress](#)
[backtrack_db_cluster](#)
[build_auth_token](#)
[cancel_export_task](#)
[copy_db_cluster_parameter_group](#)
[copy_db_cluster_snapshot](#)
[copy_db_parameter_group](#)
[copy_db_snapshot](#)
[copy_option_group](#)

Associates an Identity and Access Management (IAM) role with a DB cl
 Associates an Amazon Web Services Identity and Access Management (A
 Adds a source identifier to an existing RDS event notification subscrip
 Adds metadata tags to an Amazon RDS resource
 Applies a pending maintenance action to a resource (for example, to a D
 Enables ingress to a DBSecurityGroup using one of two forms of authori
 Backtracks a DB cluster to a specific time, without creating a new DB cl
 Return an authentication token for a database connection
 Cancels an export task in progress that is exporting a snapshot or cluster
 Copies the specified DB cluster parameter group
 Copies a snapshot of a DB cluster
 Copies the specified DB parameter group
 Copies the specified DB snapshot
 Copies the specified option group

| | |
|--|---|
| <code>create_blue_green_deployment</code> | Creates a blue/green deployment |
| <code>create_custom_db_engine_version</code> | Creates a custom DB engine version (CEV) |
| <code>create_db_cluster</code> | Creates a new Amazon Aurora DB cluster or Multi-AZ DB cluster |
| <code>create_db_cluster_endpoint</code> | Creates a new custom endpoint and associates it with an Amazon Aurora DB cluster |
| <code>create_db_cluster_parameter_group</code> | Creates a new DB cluster parameter group |
| <code>create_db_cluster_snapshot</code> | Creates a snapshot of a DB cluster |
| <code>create_db_instance</code> | Creates a new DB instance |
| <code>create_db_instance_read_replica</code> | Creates a new DB instance that acts as a read replica for an existing source instance |
| <code>create_db_parameter_group</code> | Creates a new DB parameter group |
| <code>create_db_proxy</code> | Creates a new DB proxy |
| <code>create_db_proxy_endpoint</code> | Creates a DBProxyEndpoint |
| <code>create_db_security_group</code> | Creates a new DB security group |
| <code>create_db_snapshot</code> | Creates a snapshot of a DB instance |
| <code>create_db_subnet_group</code> | Creates a new DB subnet group |
| <code>create_event_subscription</code> | Creates an RDS event notification subscription |
| <code>create_global_cluster</code> | Creates an Aurora global database spread across multiple Amazon Web Services Regions |
| <code>create_option_group</code> | Creates a new option group |
| <code>delete_blue_green_deployment</code> | Deletes a blue/green deployment |
| <code>delete_custom_db_engine_version</code> | Deletes a custom engine version |
| <code>delete_db_cluster</code> | The DeleteDBCluster action deletes a previously provisioned DB cluster |
| <code>delete_db_cluster_automated_backup</code> | Deletes automated backups using the DbClusterResourceId value of the source instance |
| <code>delete_db_cluster_endpoint</code> | Deletes a custom endpoint and removes it from an Amazon Aurora DB cluster |
| <code>delete_db_cluster_parameter_group</code> | Deletes a specified DB cluster parameter group |
| <code>delete_db_cluster_snapshot</code> | Deletes a DB cluster snapshot |
| <code>delete_db_instance</code> | The DeleteDBInstance action deletes a previously provisioned DB instance |
| <code>delete_db_instance_automated_backup</code> | Deletes automated backups using the DbiResourceId value of the source instance |
| <code>delete_db_parameter_group</code> | Deletes a specified DB parameter group |
| <code>delete_db_proxy</code> | Deletes an existing DB proxy |
| <code>delete_db_proxy_endpoint</code> | Deletes a DBProxyEndpoint |
| <code>delete_db_security_group</code> | Deletes a DB security group |
| <code>delete_db_snapshot</code> | Deletes a DB snapshot |
| <code>delete_db_subnet_group</code> | Deletes a DB subnet group |
| <code>delete_event_subscription</code> | Deletes an RDS event notification subscription |
| <code>delete_global_cluster</code> | Deletes a global database cluster |
| <code>delete_option_group</code> | Deletes an existing option group |
| <code>deregister_db_proxy_targets</code> | Remove the association between one or more DBProxyTarget data structures and a DB proxy |
| <code>describe_account_attributes</code> | Lists all of the attributes for a customer account |
| <code>describe_blue_green_deployments</code> | Describes one or more blue/green deployments |
| <code>describe_certificates</code> | Lists the set of CA certificates provided by Amazon RDS for this Amazon Web Services Region |
| <code>describe_db_cluster_automated_backups</code> | Displays backups for both current and deleted DB clusters |
| <code>describe_db_cluster_backtracks</code> | Returns information about backtracks for a DB cluster |
| <code>describe_db_cluster_endpoints</code> | Returns information about endpoints for an Amazon Aurora DB cluster |
| <code>describe_db_cluster_parameter_groups</code> | Returns a list of DBClusterParameterGroup descriptions |
| <code>describe_db_cluster_parameters</code> | Returns the detailed parameter list for a particular DB cluster parameter group |
| <code>describe_db_clusters</code> | Describes existing Amazon Aurora DB clusters and Multi-AZ DB clusters |
| <code>describe_db_cluster_snapshot_attributes</code> | Returns a list of DB cluster snapshot attribute names and values for a particular snapshot |
| <code>describe_db_cluster_snapshots</code> | Returns information about DB cluster snapshots |
| <code>describe_db_engine_versions</code> | Returns a list of the available DB engines |

| | |
|--|--|
| describe_db_instance_automated_backups | Displays backups for both current and deleted instances |
| describe_db_instances | Describes provisioned RDS instances |
| describe_db_log_files | Returns a list of DB log files for the DB instance |
| describe_db_parameter_groups | Returns a list of DBParameterGroup descriptions |
| describe_db_parameters | Returns the detailed parameter list for a particular DB parameter group |
| describe_db_proxies | Returns information about DB proxies |
| describe_db_proxy_endpoints | Returns information about DB proxy endpoints |
| describe_db_proxy_target_groups | Returns information about DB proxy target groups, represented by DBProxyTargetGroup objects |
| describe_db_proxy_targets | Returns information about DBProxyTarget objects |
| describe_db_security_groups | Returns a list of DBSecurityGroup descriptions |
| describe_db_snapshot_attributes | Returns a list of DB snapshot attribute names and values for a manual DB snapshot |
| describe_db_snapshots | Returns information about DB snapshots |
| describe_db_subnet_groups | Returns a list of DBSubnetGroup descriptions |
| describe_engine_default_cluster_parameters | Returns the default engine and system parameter information for the cluster |
| describe_engine_default_parameters | Returns the default engine and system parameter information for the specified DB instance |
| describe_event_categories | Displays a list of categories for all event source types, or, if specified, for a particular event source type |
| describe_events | Returns events related to DB instances, DB clusters, DB parameter groups, and DB snapshots |
| describe_event_subscriptions | Lists all the subscription descriptions for a customer account |
| describe_export_tasks | Returns information about a snapshot or cluster export to Amazon S3 |
| describe_global_clusters | Returns information about Aurora global database clusters |
| describe_option_group_options | Describes all available options |
| describe_option_groups | Describes the available option groups |
| describe_orderable_db_instance_options | Returns a list of orderable DB instance options for the specified DB engine |
| describe_pending_maintenance_actions | Returns a list of resources (for example, DB instances) that have at least one pending maintenance action |
| describe_reserved_db_instances | Returns information about reserved DB instances for this account, or about all reserved DB instances |
| describe_reserved_db_instances_offerings | Lists available reserved DB instance offerings |
| describe_source_regions | Returns a list of the source Amazon Web Services Regions where the current DB instance is replicated |
| describe_valid_db_instance_modifications | You can call DescribeValidDBInstanceModifications to learn what modifications are valid for a particular DB instance |
| download_db_log_file_portion | Downloads all or a portion of the specified log file, up to 1 MB in size |
| failover_db_cluster | Forces a failover for a DB cluster |
| failover_global_cluster | Promotes the specified secondary DB cluster to be the primary DB cluster |
| list_tags_for_resource | Lists all tags on an Amazon RDS resource |
| modify_activity_stream | Changes the audit policy state of a database activity stream to either locked or unlocked |
| modify_certificates | Override the system-default Secure Sockets Layer/Transport Layer Security (SSL) certificates |
| modify_current_db_cluster_capacity | Set the capacity of an Aurora Serverless v1 DB cluster to a specific value |
| modify_custom_db_engine_version | Modifies the status of a custom engine version (CEV) |
| modify_db_cluster | Modifies the settings of an Amazon Aurora DB cluster or a Multi-AZ DB instance |
| modify_db_cluster_endpoint | Modifies the properties of an endpoint in an Amazon Aurora DB cluster |
| modify_db_cluster_parameter_group | Modifies the parameters of a DB cluster parameter group |
| modify_db_cluster_snapshot_attribute | Adds an attribute and values to, or removes an attribute and values from, a DB cluster snapshot |
| modify_db_instance | Modifies settings for a DB instance |
| modify_db_parameter_group | Modifies the parameters of a DB parameter group |
| modify_db_proxy | Changes the settings for an existing DB proxy |
| modify_db_proxy_endpoint | Changes the settings for an existing DB proxy endpoint |
| modify_db_proxy_target_group | Modifies the properties of a DBProxyTargetGroup |
| modify_db_snapshot | Updates a manual DB snapshot with a new engine version |
| modify_db_snapshot_attribute | Adds an attribute and values to, or removes an attribute and values from, a DB snapshot |
| modify_db_subnet_group | Modifies an existing DB subnet group |

| | |
|--|---|
| <code>modify_event_subscription</code> | Modifies an existing RDS event notification subscription |
| <code>modify_global_cluster</code> | Modifies a setting for an Amazon Aurora global database cluster |
| <code>modify_option_group</code> | Modifies an existing option group |
| <code>promote_read_replica</code> | Promotes a read replica DB instance to a standalone DB instance |
| <code>promote_read_replica_db_cluster</code> | Promotes a read replica DB cluster to a standalone DB cluster |
| <code>purchase_reserved_db_instances_offering</code> | Purchases a reserved DB instance offering |
| <code>reboot_db_cluster</code> | You might need to reboot your DB cluster, usually for maintenance reasons |
| <code>reboot_db_instance</code> | You might need to reboot your DB instance, usually for maintenance reasons |
| <code>register_db_proxy_targets</code> | Associate one or more DBProxyTarget data structures with a DBProxyTarget |
| <code>remove_from_global_cluster</code> | Detaches an Aurora secondary cluster from an Aurora global database cluster |
| <code>remove_role_from_db_cluster</code> | Removes the association of an Amazon Web Services Identity and Access Management role from a DB cluster |
| <code>remove_role_from_db_instance</code> | Disassociates an Amazon Web Services Identity and Access Management role from a DB instance |
| <code>remove_source_identifier_from_subscription</code> | Removes a source identifier from an existing RDS event notification subscription |
| <code>remove_tags_from_resource</code> | Removes metadata tags from an Amazon RDS resource |
| <code>reset_db_cluster_parameter_group</code> | Modifies the parameters of a DB cluster parameter group to the default values |
| <code>reset_db_parameter_group</code> | Modifies the parameters of a DB parameter group to the engine/system default values |
| <code>restore_db_cluster_from_s3</code> | Creates an Amazon Aurora DB cluster from MySQL data stored in an Amazon S3 bucket |
| <code>restore_db_cluster_from_snapshot</code> | Creates a new DB cluster from a DB snapshot or DB cluster snapshot |
| <code>restore_db_cluster_to_point_in_time</code> | Restores a DB cluster to an arbitrary point in time |
| <code>restore_db_instance_from_db_snapshot</code> | Creates a new DB instance from a DB snapshot |
| <code>restore_db_instance_from_s3</code> | Amazon Relational Database Service (Amazon RDS) supports importing data from an Amazon S3 bucket |
| <code>restore_db_instance_to_point_in_time</code> | Restores a DB instance to an arbitrary point in time |
| <code>revoke_db_security_group_ingress</code> | Revokes ingress from a DBSecurityGroup for previously authorized IP ranges |
| <code>start_activity_stream</code> | Starts a database activity stream to monitor activity on the database |
| <code>start_db_cluster</code> | Starts an Amazon Aurora DB cluster that was stopped using the Amazon Web Services console |
| <code>start_db_instance</code> | Starts an Amazon RDS DB instance that was stopped using the Amazon Web Services console |
| <code>start_db_instance_automated_backups_replication</code> | Enables replication of automated backups to a different Amazon Web Services Region |
| <code>start_export_task</code> | Starts an export of DB snapshot or DB cluster data to Amazon S3 |
| <code>stop_activity_stream</code> | Stops a database activity stream that was started using the Amazon Web Services console |
| <code>stop_db_cluster</code> | Stops an Amazon Aurora DB cluster |
| <code>stop_db_instance</code> | Stops an Amazon RDS DB instance |
| <code>stop_db_instance_automated_backups_replication</code> | Stops automated backup replication for a DB instance |
| <code>switchover_blue_green_deployment</code> | Switches over a blue/green deployment |
| <code>switchover_global_cluster</code> | Switches over the specified secondary DB cluster to be the new primary |
| <code>switchover_read_replica</code> | Switches over an Oracle standby database in an Oracle Data Guard environment |

Examples

```
## Not run:
svc <- rds()
svc$add_role_to_db_cluster(
  Foo = 123
)

## End(Not run)
```

| | |
|----------------|----------------------------|
| rdsdataservice | <i>AWS RDS DataService</i> |
|----------------|----------------------------|

Description

Amazon RDS Data Service

Amazon RDS provides an HTTP endpoint to run SQL statements on an Amazon Aurora Serverless v1 DB cluster. To run these statements, you work with the Data Service API.

The Data Service API isn't supported on Amazon Aurora Serverless v2 DB clusters.

For more information about the Data Service API, see [Using the Data API](#) in the *Amazon Aurora User Guide*.

Usage

```
rdsdataservice(
  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>

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

Operations

| | |
|--------------------------------------|--|
| <code>batch_execute_statement</code> | Runs a batch SQL statement over an array of data |
| <code>begin_transaction</code> | Starts a SQL transaction |
| <code>commit_transaction</code> | Ends a SQL transaction started with the BeginTransaction operation and commits the changes |
| <code>execute_sql</code> | Runs one or more SQL statements |
| <code>execute_statement</code> | Runs a SQL statement against a database |
| <code>rollback_transaction</code> | Performs a rollback of a transaction |

Examples

```
## Not run:
svc <- rdsdataservice()
svc$batch_execute_statement(
  Foo = 123
)
## End(Not run)
```

redshift

Amazon Redshift

Description

Overview

This is an interface reference for Amazon Redshift. It contains documentation for one of the programming or command line interfaces you can use to manage Amazon Redshift clusters. Note that Amazon Redshift is asynchronous, which means that some interfaces may require techniques, such as polling or asynchronous callback handlers, to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a change is applied immediately, on the next instance reboot, or during the next maintenance window. For a summary of the Amazon Redshift cluster management interfaces, go to [Using the Amazon Redshift Management Interfaces](#).

Amazon Redshift manages all the work of setting up, operating, and scaling a data warehouse: provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine. You can focus on using your data to acquire new insights for your business and customers.

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the [Amazon Redshift Getting Started Guide](#).

If you are a database developer, the [Amazon Redshift Database Developer Guide](#) explains how to design, build, query, and maintain the databases that make up your data warehouse.

Usage

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

Arguments

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

Value

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

Service syntax

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

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
    creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

[accept_reserved_node_exchange](#)
[add_partner](#)
[associate_data_share_consumer](#)
[authorize_cluster_security_group_ingress](#)
[authorize_data_share](#)
[authorize_endpoint_access](#)
[authorize_snapshot_access](#)
[batch_delete_cluster_snapshots](#)
[batch_modify_cluster_snapshots](#)
[cancel_resize](#)
[copy_cluster_snapshot](#)
[create_authentication_profile](#)
[create_cluster](#)
[create_cluster_parameter_group](#)
[create_cluster_security_group](#)
[create_cluster_snapshot](#)
[create_cluster_subnet_group](#)
[create_custom_domain_association](#)
[create_endpoint_access](#)
[create_event_subscription](#)

Exchanges a DC1 Reserved Node for a DC2 Reserved Node with no cl
 Adds a partner integration to a cluster
 From a datashare consumer account, associates a datashare with the ac
 Adds an inbound (ingress) rule to an Amazon Redshift security group
 From a data producer account, authorizes the sharing of a datashare wi
 Grants access to a cluster
 Authorizes the specified Amazon Web Services account to restore the s
 Deletes a set of cluster snapshots
 Modifies the settings for a set of cluster snapshots
 Cancels a resize operation for a cluster
 Copies the specified automated cluster snapshot to a new manual cluste
 Creates an authentication profile with the specified parameters
 Creates a new cluster with the specified parameters
 Creates an Amazon Redshift parameter group
 Creates a new Amazon Redshift security group
 Creates a manual snapshot of the specified cluster
 Creates a new Amazon Redshift subnet group
 Used to create a custom domain name for a cluster
 Creates a Redshift-managed VPC endpoint
 Creates an Amazon Redshift event notification subscription

| | |
|--|---|
| <code>create_hsm_client_certificate</code> | Creates an HSM client certificate that an Amazon Redshift cluster will use to connect to an HSM. |
| <code>create_hsm_configuration</code> | Creates an HSM configuration that contains the information required by an Amazon Redshift cluster to connect to an HSM. |
| <code>create_scheduled_action</code> | Creates a scheduled action. |
| <code>create_snapshot_copy_grant</code> | Creates a snapshot copy grant that permits Amazon Redshift to use an Amazon S3 bucket to store snapshots. |
| <code>create_snapshot_schedule</code> | Create a snapshot schedule that can be associated to a cluster and which can be used to create snapshots. |
| <code>create_tags</code> | Adds tags to a cluster. |
| <code>create_usage_limit</code> | Creates a usage limit for a specified Amazon Redshift feature on a cluster. |
| <code>deauthorize_data_share</code> | From a datashare producer account, removes authorization from the specified datashare. |
| <code>delete_authentication_profile</code> | Deletes an authentication profile. |
| <code>delete_cluster</code> | Deletes a previously provisioned cluster without its final snapshot being created. |
| <code>delete_cluster_parameter_group</code> | Deletes a specified Amazon Redshift parameter group. |
| <code>delete_cluster_security_group</code> | Deletes an Amazon Redshift security group. |
| <code>delete_cluster_snapshot</code> | Deletes the specified manual snapshot. |
| <code>delete_cluster_subnet_group</code> | Deletes the specified cluster subnet group. |
| <code>delete_custom_domain_association</code> | Contains information about deleting a custom domain association for a cluster. |
| <code>delete_endpoint_access</code> | Deletes a Redshift-managed VPC endpoint. |
| <code>delete_event_subscription</code> | Deletes an Amazon Redshift event notification subscription. |
| <code>delete_hsm_client_certificate</code> | Deletes the specified HSM client certificate. |
| <code>delete_hsm_configuration</code> | Deletes the specified Amazon Redshift HSM configuration. |
| <code>delete_partner</code> | Deletes a partner integration from a cluster. |
| <code>delete_scheduled_action</code> | Deletes a scheduled action. |
| <code>delete_snapshot_copy_grant</code> | Deletes the specified snapshot copy grant. |
| <code>delete_snapshot_schedule</code> | Deletes a snapshot schedule. |
| <code>delete_tags</code> | Deletes tags from a resource. |
| <code>delete_usage_limit</code> | Deletes a usage limit from a cluster. |
| <code>describe_account_attributes</code> | Returns a list of attributes attached to an account. |
| <code>describe_authentication_profiles</code> | Describes an authentication profile. |
| <code>describe_cluster_db_revisions</code> | Returns an array of ClusterDbRevision objects. |
| <code>describe_cluster_parameter_groups</code> | Returns a list of Amazon Redshift parameter groups, including parameter group names. |
| <code>describe_cluster_parameters</code> | Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group. |
| <code>describe_clusters</code> | Returns properties of provisioned clusters including general cluster properties. |
| <code>describe_cluster_security_groups</code> | Returns information about Amazon Redshift security groups. |
| <code>describe_cluster_snapshots</code> | Returns one or more snapshot objects, which contain metadata about your snapshots. |
| <code>describe_cluster_subnet_groups</code> | Returns one or more cluster subnet group objects, which contain metadata about your cluster subnet groups. |
| <code>describe_cluster_tracks</code> | Returns a list of all the available maintenance tracks. |
| <code>describe_cluster_versions</code> | Returns descriptions of the available Amazon Redshift cluster versions. |
| <code>describe_custom_domain_associations</code> | Contains information for custom domain associations for a cluster. |
| <code>describe_data_shares</code> | Shows the status of any inbound or outbound datashares available in the specified Amazon Redshift account. |
| <code>describe_data_shares_for_consumer</code> | Returns a list of datashares where the account identifier being called is the consumer. |
| <code>describe_data_shares_for_producer</code> | Returns a list of datashares when the account identifier being called is the producer. |
| <code>describe_default_cluster_parameters</code> | Returns a list of parameter settings for the specified parameter group family. |
| <code>describe_endpoint_access</code> | Describes a Redshift-managed VPC endpoint. |
| <code>describe_endpoint_authorization</code> | Describes an endpoint authorization. |
| <code>describe_event_categories</code> | Displays a list of event categories for all event source types, or for a specific event source type. |
| <code>describe_events</code> | Returns events related to clusters, security groups, snapshots, and parameter groups. |
| <code>describe_event_subscriptions</code> | Lists descriptions of all the Amazon Redshift event notification subscriptions. |
| <code>describe_hsm_client_certificates</code> | Returns information about the specified HSM client certificate. |
| <code>describe_hsm_configurations</code> | Returns information about the specified Amazon Redshift HSM configuration. |

| | |
|---|--|
| <code>describe_logging_status</code> | Describes whether information, such as queries and connection attempts, is logged |
| <code>describe_node_configuration_options</code> | Returns properties of possible node configurations such as node type, number of nodes, and availability zone |
| <code>describe_orderable_cluster_options</code> | Returns a list of orderable cluster options |
| <code>describe_partners</code> | Returns information about the partner integrations defined for a cluster |
| <code>describe_reserved_node_exchange_status</code> | Returns exchange status details and associated metadata for a reserved node exchange |
| <code>describe_reserved_node_offerings</code> | Returns a list of the available reserved node offerings by Amazon Redshift |
| <code>describe_reserved_nodes</code> | Returns the descriptions of the reserved nodes |
| <code>describe_resize</code> | Returns information about the last resize operation for the specified cluster |
| <code>describe_scheduled_actions</code> | Describes properties of scheduled actions |
| <code>describe_snapshot_copy_grants</code> | Returns a list of snapshot copy grants owned by the Amazon Web Services account |
| <code>describe_snapshot_schedules</code> | Returns a list of snapshot schedules |
| <code>describe_storage</code> | Returns account level backups storage size and provisional storage |
| <code>describe_table_restore_status</code> | Lists the status of one or more table restore requests made using the Redshift console |
| <code>describe_tags</code> | Returns a list of tags |
| <code>describe_usage_limits</code> | Shows usage limits on a cluster |
| <code>disable_logging</code> | Stops logging information, such as queries and connection attempts, for the specified cluster |
| <code>disable_snapshot_copy</code> | Disables the automatic copying of snapshots from one region to another region |
| <code>disassociate_data_share_consumer</code> | From a datashare consumer account, remove association for the specified datashare |
| <code>enable_logging</code> | Starts logging information, such as queries and connection attempts, for the specified cluster |
| <code>enable_snapshot_copy</code> | Enables the automatic copy of snapshots from one region to another region |
| <code>get_cluster_credentials</code> | Returns a database user name and temporary password with temporary validity |
| <code>get_cluster_credentials_with_iam</code> | Returns a database user name and temporary password with temporary validity |
| <code>get_reserved_node_exchange_configuration_options</code> | Gets the configuration options for the reserved-node exchange |
| <code>get_reserved_node_exchange_offerings</code> | Returns an array of DC2 ReservedNodeOfferings that matches the payment method |
| <code>modify_aqua_configuration</code> | This operation is retired |
| <code>modify_authentication_profile</code> | Modifies an authentication profile |
| <code>modify_cluster</code> | Modifies the settings for a cluster |
| <code>modify_cluster_db_revision</code> | Modifies the database revision of a cluster |
| <code>modify_cluster_iam_roles</code> | Modifies the list of Identity and Access Management (IAM) roles that are associated with the cluster |
| <code>modify_cluster_maintenance</code> | Modifies the maintenance settings of a cluster |
| <code>modify_cluster_parameter_group</code> | Modifies the parameters of a parameter group |
| <code>modify_cluster_snapshot</code> | Modifies the settings for a snapshot |
| <code>modify_cluster_snapshot_schedule</code> | Modifies a snapshot schedule for a cluster |
| <code>modify_cluster_subnet_group</code> | Modifies a cluster subnet group to include the specified list of VPC subnets |
| <code>modify_custom_domain_association</code> | Contains information for changing a custom domain association |
| <code>modify_endpoint_access</code> | Modifies a Redshift-managed VPC endpoint |
| <code>modify_event_subscription</code> | Modifies an existing Amazon Redshift event notification subscription |
| <code>modify_scheduled_action</code> | Modifies a scheduled action |
| <code>modify_snapshot_copy_retention_period</code> | Modifies the number of days to retain snapshots in the destination Amazon S3 bucket |
| <code>modify_snapshot_schedule</code> | Modifies a snapshot schedule |
| <code>modify_usage_limit</code> | Modifies a usage limit in a cluster |
| <code>pause_cluster</code> | Pauses a cluster |
| <code>purchase_reserved_node_offering</code> | Allows you to purchase reserved nodes |
| <code>reboot_cluster</code> | Reboots a cluster |
| <code>reject_data_share</code> | From a datashare consumer account, rejects the specified datashare |
| <code>reset_cluster_parameter_group</code> | Sets one or more parameters of the specified parameter group to their default values |
| <code>resize_cluster</code> | Changes the size of the cluster |
| <code>restore_from_cluster_snapshot</code> | Creates a new cluster from a snapshot |

| | |
|--|--|
| <code>restore_table_from_cluster_snapshot</code> | Creates a new table from a table in an Amazon Redshift cluster snapshot |
| <code>resume_cluster</code> | Resumes a paused cluster |
| <code>revoke_cluster_security_group_ingress</code> | Revokes an ingress rule in an Amazon Redshift security group for a principal |
| <code>revoke_endpoint_access</code> | Revokes access to a cluster |
| <code>revoke_snapshot_access</code> | Removes the ability of the specified Amazon Web Services account to restore a snapshot |
| <code>rotate_encryption_key</code> | Rotates the encryption keys for a cluster |
| <code>update_partner_status</code> | Updates the status of a partner integration |

Examples

```
## Not run:
svc <- redshift()
svc$accept_reserved_node_exchange(
  Foo = 123
)

## End(Not run)
```

redshiftdataapiservice

Redshift Data API Service

Description

You can use the Amazon Redshift Data API to run queries on Amazon Redshift tables. You can run SQL statements, which are committed if the statement succeeds.

For more information about the Amazon Redshift Data API and CLI usage examples, see [Using the Amazon Redshift Data API](#) in the *Amazon Redshift Management Guide*.

Usage

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

Arguments

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

- **credentials:**
 - **creds:**
 - * **access_key_id:** AWS access key ID

| | |
|-------------|--|
| | <ul style="list-style-type: none"> * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. – endpoint: The complete URL to use for the constructed client. – region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

```
svc <- redshiftdataapiservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
```

```

    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

| | |
|---|--|
| batch_execute_statement | Runs one or more SQL statements, which can be data manipulation language (DML) or data definition language (DDL) |
| cancel_statement | Cancel a running query |
| describe_statement | Describe the details about a specific instance when a query was run by the Amazon Redshift Data API |
| describe_table | Describe the detailed information about a table from metadata in the cluster |
| execute_statement | Runs an SQL statement, which can be data manipulation language (DML) or data definition language (DDL) |
| get_statement_result | Fetches the temporarily cached result of an SQL statement |
| list_databases | List the databases in a cluster |
| list_schemas | List the schemas in a database |
| list_statements | List of SQL statements |
| list_tables | List the tables in a database |

Examples

```

## Not run:
svc <- redshiftdataapiservice()
svc$batch_execute_statement(
  Foo = 123
)

## End(Not run)

```

Description

This is an interface reference for Amazon Redshift Serverless. It contains documentation for one of the programming or command line interfaces you can use to manage Amazon Redshift Serverless.

Amazon Redshift Serverless automatically provisions data warehouse capacity and intelligently scales the underlying resources based on workload demands. Amazon Redshift Serverless adjusts capacity in seconds to deliver consistently high performance and simplified operations for even the most demanding and volatile workloads. Amazon Redshift Serverless lets you focus on using your data to acquire new insights for your business and customers.

To learn more about Amazon Redshift Serverless, see [What is Amazon Redshift Serverless](#).

Usage

```
redshiftserverless(
    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>

| | |
|-------------|---|
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

    region = "string"
)

```

Operations

| | |
|--|--|
| convert_recovery_point_to_snapshot | Converts a recovery point to a snapshot |
| create_endpoint_access | Creates an Amazon Redshift Serverless managed VPC endpoint |
| create_namespace | Creates a namespace in Amazon Redshift Serverless |
| create_snapshot | Creates a snapshot of all databases in a namespace |
| create_usage_limit | Creates a usage limit for a specified Amazon Redshift Serverless usage type |
| create_workgroup | Creates a workgroup in Amazon Redshift Serverless |
| delete_endpoint_access | Deletes an Amazon Redshift Serverless managed VPC endpoint |
| delete_namespace | Deletes a namespace from Amazon Redshift Serverless |
| delete_resource_policy | Deletes the specified resource policy |
| delete_snapshot | Deletes a snapshot from Amazon Redshift Serverless |
| delete_usage_limit | Deletes a usage limit from Amazon Redshift Serverless |
| delete_workgroup | Deletes a workgroup |
| get_credentials | Returns a database user name and temporary password with temporary authorization token |
| get_endpoint_access | Returns information, such as the name, about a VPC endpoint |
| get_namespace | Returns information about a namespace in Amazon Redshift Serverless |
| get_recovery_point | Returns information about a recovery point |
| get_resource_policy | Returns a resource policy |
| get_snapshot | Returns information about a specific snapshot |
| get_table_restore_status | Returns information about a TableRestoreStatus object |
| get_usage_limit | Returns information about a usage limit |
| get_workgroup | Returns information about a specific workgroup |
| list_endpoint_access | Returns an array of EndpointAccess objects and relevant information |
| list_namespaces | Returns information about a list of specified namespaces |
| list_recovery_points | Returns an array of recovery points |
| list_snapshots | Returns a list of snapshots |
| list_table_restore_status | Returns information about an array of TableRestoreStatus objects |
| list_tags_for_resource | Lists the tags assigned to a resource |
| list_usage_limits | Lists all usage limits within Amazon Redshift Serverless |
| list_workgroups | Returns information about a list of specified workgroups |
| put_resource_policy | Creates or updates a resource policy |
| restore_from_recovery_point | Restore the data from a recovery point |
| restore_from_snapshot | Restores a namespace from a snapshot |
| restore_table_from_snapshot | Restores a table from a snapshot to your Amazon Redshift Serverless instance |
| tag_resource | Assigns one or more tags to a resource |
| untag_resource | Removes a tag or set of tags from a resource |
| update_endpoint_access | Updates an Amazon Redshift Serverless managed endpoint |
| update_namespace | Updates a namespace with the specified settings |
| update_snapshot | Updates a snapshot |
| update_usage_limit | Update a usage limit in Amazon Redshift Serverless |
| update_workgroup | Updates a workgroup with the specified configuration settings |

Examples

```
## Not run:
svc <- redshiftserverless()
svc$convert_recovery_point_to_snapshot(
  Foo = 123
)

## End(Not run)
```

simpledb

Amazon SimpleDB

Description

Amazon SimpleDB is a web service providing the core database functions of data indexing and querying in the cloud. By offloading the time and effort associated with building and operating a web-scale database, SimpleDB provides developers the freedom to focus on application development.

A traditional, clustered relational database requires a sizable upfront capital outlay, is complex to design, and often requires extensive and repetitive database administration. Amazon SimpleDB is dramatically simpler, requiring no schema, automatically indexing your data and providing a simple API for storage and access. This approach eliminates the administrative burden of data modeling, index maintenance, and performance tuning. Developers gain access to this functionality within Amazon's proven computing environment, are able to scale instantly, and pay only for what they use.

Visit <http://aws.amazon.com/simpledb/> for more information.

Usage

```
simpledb(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.

| | |
|-------------|--|
| | <ul style="list-style-type: none"> • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

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

```

Operations

| | |
|---|---|
| batch_delete_attributes | Performs multiple DeleteAttributes operations in a single call, which reduces round trips and latency |
| batch_put_attributes | The BatchPutAttributes operation creates or replaces attributes within one or more items |
| create_domain | The CreateDomain operation creates a new domain |
| delete_attributes | Deletes one or more attributes associated with an item |
| delete_domain | The DeleteDomain operation deletes a domain |
| domain_metadata | Returns information about the domain, including when the domain was created, the number of items |
| get_attributes | Returns all of the attributes associated with the specified item |
| list_domains | The ListDomains operation lists all domains associated with the Access Key ID |
| put_attributes | The PutAttributes operation creates or replaces attributes in an item |
| select | The Select operation returns a set of attributes for ItemNames that match the select expression |

Examples

```

## Not run:
svc <- simpledb()
svc$batch_delete_attributes(
  Foo = 123
)

## End(Not run)

```

timestreamquery

Amazon Timestream Query

Description

Amazon Timestream Query

Usage

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

Arguments

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

Value

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

Service syntax

```

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

```

Operations

| | |
|--|--|
| cancel_query | Cancel a query that has been issued |
| create_scheduled_query | Create a scheduled query that will be run on your behalf at the configured schedule |
| delete_scheduled_query | Deletes a given scheduled query |
| describe_endpoints | DescribeEndpoints returns a list of available endpoints to make Timestream API calls against |
| describe_scheduled_query | Provides detailed information about a scheduled query |
| execute_scheduled_query | You can use this API to run a scheduled query manually |
| list_scheduled_queries | Gets a list of all scheduled queries in the caller's Amazon account and Region |
| list_tags_for_resource | List all tags on a Timestream query resource |
| prepare_query | A synchronous operation that allows you to submit a query with parameters to be stored by Time |
| query | Query is a synchronous operation that enables you to run a query against your Amazon Timestre |
| tag_resource | Associate a set of tags with a Timestream resource |
| untag_resource | Removes the association of tags from a Timestream query resource |
| update_scheduled_query | Update a scheduled query |

Examples

```
## Not run:
svc <- timestreamquery()
svc$cancel_query(
  Foo = 123
)

## End(Not run)
```

timestreamwrite

Amazon Timestream Write

Description

Amazon Timestream is a fast, scalable, fully managed time-series database service that makes it easy to store and analyze trillions of time-series data points per day. With Timestream, you can easily store and analyze IoT sensor data to derive insights from your IoT applications. You can analyze industrial telemetry to streamline equipment management and maintenance. You can also store and analyze log data and metrics to improve the performance and availability of your applications.

Timestream is built from the ground up to effectively ingest, process, and store time-series data. It organizes data to optimize query processing. It automatically scales based on the volume of data ingested and on the query volume to ensure you receive optimal performance while inserting and querying data. As your data grows over time, Timestream's adaptive query processing engine spans across storage tiers to provide fast analysis while reducing costs.

Usage

```
timestreamwrite(
  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

| | |
|-------------|--|
| | <ul style="list-style-type: none"> – 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. <ul style="list-style-type: none"> • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. |
| endpoint | Optional shorthand for complete URL to use for the constructed client. |
| region | Optional shorthand for AWS Region used in instantiating the client. |

Value

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

Service syntax

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

```

    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

| | |
|--|---|
| create_batch_load_task | Creates a new Timestream batch load task |
| create_database | Creates a new Timestream database |
| create_table | Adds a new table to an existing database in your account |
| delete_database | Deletes a given Timestream database |
| delete_table | Deletes a given Timestream table |
| describe_batch_load_task | Returns information about the batch load task, including configurations, mappings, progress, and |
| describe_database | Returns information about the database, including the database name, time that the database was |
| describe_endpoints | Returns a list of available endpoints to make Timestream API calls against |
| describe_table | Returns information about the table, including the table name, database name, retention duration |
| list_batch_load_tasks | Provides a list of batch load tasks, along with the name, status, when the task is resumable until, |
| list_databases | Returns a list of your Timestream databases |
| list_tables | Provides a list of tables, along with the name, status, and retention properties of each table |
| list_tags_for_resource | Lists all tags on a Timestream resource |
| resume_batch_load_task | Resume batch load task |
| tag_resource | Associates a set of tags with a Timestream resource |
| untag_resource | Removes the association of tags from a Timestream resource |
| update_database | Modifies the KMS key for an existing database |
| update_table | Modifies the retention duration of the memory store and magnetic store for your Timestream table |
| write_records | Enables you to write your time-series data into Timestream |

Examples

```

## Not run:
svc <- timestreamwrite()
svc$create_batch_load_task(
  Foo = 123
)

```

End(Not run)

Index

accept_reserved_node_exchange, [47](#)
add_terraform_tags_to_resource, [24](#)
add_partner, [47](#)
add_role_to_db_cluster, [30, 39](#)
add_role_to_db_instance, [39](#)
add_source_identifier_to_subscription,
[7, 30, 39](#)
add_tags_to_resource, [7, 17, 30, 39](#)
apply_pending_maintenance_action, [7, 30, 39](#)
associate_data_share_consumer, [47](#)
assume_decorated_role_with_saml, [24](#)
authorize_cache_security_group_ingress,
[17](#)
authorize_cluster_security_group_ingress,
[47](#)
authorize_data_share, [47](#)
authorize_db_security_group_ingress,
[39](#)
authorize_endpoint_access, [47](#)
authorize_snapshot_access, [47](#)

backtrack_db_cluster, [39](#)
batch_apply_update_action, [17](#)
batch_delete_attributes, [58](#)
batch_delete_cluster_snapshots, [47](#)
batch_execute_statement, [11, 45, 52](#)
batch_get_item, [11](#)
batch_grant_permissions, [24](#)
batch_modify_cluster_snapshots, [47](#)
batch_put_attributes, [58](#)
batch_revoke_permissions, [24](#)
batch_stop_update_action, [17](#)
batch_update_cluster, [27](#)
batch_write_item, [11](#)
begin_transaction, [45](#)
build_auth_token, [39](#)

cancel_export_task, [39](#)
cancel_journal_kinesis_stream, [34](#)

cancel_query, [60](#)
cancel_resize, [47](#)
cancel_statement, [52](#)
cancel_transaction, [24](#)
commit_transaction, [24, 45](#)
complete_migration, [17](#)
convert_recovery_point_to_snapshot, [55](#)
copy_cluster_snapshot, [47](#)
copy_db_cluster_parameter_group, [7, 30, 39](#)
copy_db_cluster_snapshot, [7, 30, 39](#)
copy_db_parameter_group, [30, 39](#)
copy_db_snapshot, [39](#)
copy_option_group, [39](#)
copy_snapshot, [17, 27](#)
create_acl, [27](#)
create_authentication_profile, [47](#)
create_backup, [11](#)
create_batch_load_task, [63](#)
create_blue_green_deployment, [40](#)
create_cache_cluster, [17](#)
create_cache_parameter_group, [17](#)
create_cache_security_group, [17](#)
create_cache_subnet_group, [17](#)
create_cluster, [4, 27, 47](#)
create_cluster_parameter_group, [47](#)
create_cluster_security_group, [47](#)
create_cluster_snapshot, [47](#)
create_cluster_subnet_group, [47](#)
create_custom_db_engine_version, [40](#)
create_custom_domain_association, [47](#)
create_data_cells_filter, [24](#)
create_database, [63](#)
create_db_cluster, [7, 30, 40](#)
create_db_cluster_endpoint, [30, 40](#)
create_db_cluster_parameter_group, [7, 30, 40](#)
create_db_cluster_snapshot, [7, 30, 40](#)
create_db_instance, [7, 30, 40](#)

- create_db_instance_read_replica, 40
- create_db_parameter_group, 30, 40
- create_db_proxy, 40
- create_db_proxy_endpoint, 40
- create_db_security_group, 40
- create_db_snapshot, 40
- create_db_subnet_group, 7, 30, 40
- create_domain, 58
- create_endpoint_access, 47, 55
- create_event_subscription, 7, 30, 40, 47
- create_global_cluster, 7, 30, 40
- create_global_replication_group, 17
- create_global_table, 11
- create_hsm_client_certificate, 48
- create_hsm_configuration, 48
- create_keyspace, 21
- create_ledger, 34
- create_lf_tag, 24
- create_namespace, 55
- create_option_group, 40
- create_parameter_group, 4, 27
- create_replication_group, 17
- create_scheduled_action, 48
- create_scheduled_query, 60
- create_snapshot, 17, 27, 55
- create_snapshot_copy_grant, 48
- create_snapshot_schedule, 48
- create_subnet_group, 4, 27
- create_table, 11, 21, 63
- create_tags, 48
- create_usage_limit, 48, 55
- create_user, 17, 27
- create_user_group, 17
- create_workgroup, 55

- dax, 3
- deauthorize_data_share, 48
- decrease_node_groups_in_global_replication_group, 17
- decrease_replica_count, 17
- decrease_replication_factor, 5
- delete_acl, 27
- delete_attributes, 58
- delete_authentication_profile, 48
- delete_backup, 11
- delete_blue_green_deployment, 40
- delete_cache_cluster, 17
- delete_cache_parameter_group, 17
- delete_cache_security_group, 17
- delete_cache_subnet_group, 18
- delete_cluster, 5, 27, 48
- delete_cluster_parameter_group, 48
- delete_cluster_security_group, 48
- delete_cluster_snapshot, 48
- delete_cluster_subnet_group, 48
- delete_custom_db_engine_version, 40
- delete_custom_domain_association, 48
- delete_data_cells_filter, 24
- delete_database, 63
- delete_db_cluster, 7, 30, 40
- delete_db_cluster_automated_backup, 40
- delete_db_cluster_endpoint, 30, 40
- delete_db_cluster_parameter_group, 7, 30, 40
- delete_db_cluster_snapshot, 7, 30, 40
- delete_db_instance, 7, 31, 40
- delete_db_instance_automated_backup, 40
- delete_db_parameter_group, 31, 40
- delete_db_proxy, 40
- delete_db_proxy_endpoint, 40
- delete_db_security_group, 40
- delete_db_snapshot, 40
- delete_db_subnet_group, 7, 31, 40
- delete_domain, 58
- delete_endpoint_access, 48, 55
- delete_event_subscription, 7, 31, 40, 48
- delete_global_cluster, 7, 31, 40
- delete_global_replication_group, 18
- delete_hsm_client_certificate, 48
- delete_hsm_configuration, 48
- delete_item, 11
- delete_keyspace, 21
- delete_ledger, 34
- delete_lf_tag, 24
- delete_namespace, 55
- delete_objects_on_cancel, 24
- delete_option_group, 40
- delete_parameter_group, 5, 27
- delete_partner, 48
- delete_replication_group, 18
- delete_resource_policy, 55
- delete_scheduled_action, 48
- delete_scheduled_query, 60
- delete_snapshot, 18, 27, 55
- delete_snapshot_copy_grant, 48
- delete_snapshot_schedule, 48

- delete_subnet_group, [5](#), [27](#)
- delete_table, [11](#), [21](#), [63](#)
- delete_tags, [48](#)
- delete_usage_limit, [48](#), [55](#)
- delete_user, [18](#), [27](#)
- delete_user_group, [18](#)
- delete_workgroup, [55](#)
- deregister_db_proxy_targets, [40](#)
- deregister_resource, [24](#)
- describe_ac_ls, [27](#)
- describe_account_attributes, [40](#), [48](#)
- describe_authentication_profiles, [48](#)
- describe_backup, [11](#)
- describe_batch_load_task, [63](#)
- describe_blue_green_deployments, [40](#)
- describe_cache_clusters, [18](#)
- describe_cache_engine_versions, [18](#)
- describe_cache_parameter_groups, [18](#)
- describe_cache_parameters, [18](#)
- describe_cache_security_groups, [18](#)
- describe_cache_subnet_groups, [18](#)
- describe_certificates, [7](#), [40](#)
- describe_cluster_db_revisions, [48](#)
- describe_cluster_parameter_groups, [48](#)
- describe_cluster_parameters, [48](#)
- describe_cluster_security_groups, [48](#)
- describe_cluster_snapshots, [48](#)
- describe_cluster_subnet_groups, [48](#)
- describe_cluster_tracks, [48](#)
- describe_cluster_versions, [48](#)
- describe_clusters, [5](#), [27](#), [48](#)
- describe_continuous_backups, [11](#)
- describe_contributor_insights, [11](#)
- describe_custom_domain_associations, [48](#)
- describe_data_shares, [48](#)
- describe_data_shares_for_consumer, [48](#)
- describe_data_shares_for_producer, [48](#)
- describe_database, [63](#)
- describe_db_cluster_automated_backups, [40](#)
- describe_db_cluster_backtracks, [40](#)
- describe_db_cluster_endpoints, [31](#), [40](#)
- describe_db_cluster_parameter_groups, [8](#), [31](#), [40](#)
- describe_db_cluster_parameters, [8](#), [31](#), [40](#)
- describe_db_cluster_snapshot_attributes, [8](#), [31](#), [40](#)
- describe_db_cluster_snapshots, [8](#), [31](#), [40](#)
- describe_db_clusters, [8](#), [31](#), [40](#)
- describe_db_engine_versions, [8](#), [31](#), [40](#)
- describe_db_instance_automated_backups, [41](#)
- describe_db_instances, [8](#), [31](#), [41](#)
- describe_db_log_files, [41](#)
- describe_db_parameter_groups, [31](#), [41](#)
- describe_db_parameters, [31](#), [41](#)
- describe_db_proxies, [41](#)
- describe_db_proxy_endpoints, [41](#)
- describe_db_proxy_target_groups, [41](#)
- describe_db_proxy_targets, [41](#)
- describe_db_security_groups, [41](#)
- describe_db_snapshot_attributes, [41](#)
- describe_db_snapshots, [41](#)
- describe_db_subnet_groups, [8](#), [31](#), [41](#)
- describe_default_cluster_parameters, [48](#)
- describe_default_parameters, [5](#)
- describe_endpoint_access, [48](#)
- describe_endpoint_authorization, [48](#)
- describe_endpoints, [11](#), [60](#), [63](#)
- describe_engine_default_cluster_parameters, [8](#), [31](#), [41](#)
- describe_engine_default_parameters, [18](#), [31](#), [41](#)
- describe_engine_versions, [27](#)
- describe_event_categories, [8](#), [31](#), [41](#), [48](#)
- describe_event_subscriptions, [8](#), [31](#), [41](#), [48](#)
- describe_events, [5](#), [8](#), [18](#), [27](#), [31](#), [41](#), [48](#)
- describe_export, [11](#)
- describe_export_tasks, [41](#)
- describe_global_clusters, [8](#), [31](#), [41](#)
- describe_global_replication_groups, [18](#)
- describe_global_table, [11](#)
- describe_global_table_settings, [11](#)
- describe_hsm_client_certificates, [48](#)
- describe_hsm_configurations, [48](#)
- describe_import, [11](#)
- describe_journal_kinesis_stream, [34](#)
- describe_journal_s3_export, [34](#)
- describe_kinesis_streaming_destination, [11](#)
- describe_ledger, [34](#)
- describe_limits, [11](#)

- describe_logging_status, [49](#)
- describe_node_configuration_options, [49](#)
- describe_option_group_options, [41](#)
- describe_option_groups, [41](#)
- describe_orderable_cluster_options, [49](#)
- describe_orderable_db_instance_options, [8, 31, 41](#)
- describe_parameter_groups, [5, 27](#)
- describe_parameters, [5, 27](#)
- describe_partners, [49](#)
- describe_pending_maintenance_actions, [8, 31, 41](#)
- describe_replication_groups, [18](#)
- describe_reserved_cache_nodes, [18](#)
- describe_reserved_cache_nodes_offerings, [18](#)
- describe_reserved_db_instances, [41](#)
- describe_reserved_db_instances_offerings, [41](#)
- describe_reserved_node_exchange_status, [49](#)
- describe_reserved_node_offerings, [49](#)
- describe_reserved_nodes, [27, 49](#)
- describe_reserved_nodes_offerings, [27](#)
- describe_resize, [49](#)
- describe_resource, [24](#)
- describe_scheduled_actions, [49](#)
- describe_scheduled_query, [60](#)
- describe_service_updates, [18, 27](#)
- describe_snapshot_copy_grants, [49](#)
- describe_snapshot_schedules, [49](#)
- describe_snapshots, [18, 27](#)
- describe_source_regions, [41](#)
- describe_statement, [52](#)
- describe_storage, [49](#)
- describe_stream, [15](#)
- describe_subnet_groups, [5, 27](#)
- describe_table, [11, 52, 63](#)
- describe_table_replica_auto_scaling, [11](#)
- describe_table_restore_status, [49](#)
- describe_tags, [49](#)
- describe_time_to_live, [11](#)
- describe_transaction, [24](#)
- describe_update_actions, [18](#)
- describe_usage_limits, [49](#)
- describe_user_groups, [18](#)
- describe_users, [18, 27](#)
- describe_valid_db_instance_modifications, [31, 41](#)
- disable_kinesis_streaming_destination, [11](#)
- disable_logging, [49](#)
- disable_snapshot_copy, [49](#)
- disassociate_data_share_consumer, [49](#)
- disassociate_global_replication_group, [18](#)
- docdb, [5](#)
- domain_metadata, [58](#)
- download_db_log_file_portion, [41](#)
- dynamodb, [9](#)
- dynamodbstreams, [13](#)
- elasticache, [15](#)
- enable_kinesis_streaming_destination, [11](#)
- enable_logging, [49](#)
- enable_snapshot_copy, [49](#)
- execute_scheduled_query, [60](#)
- execute_sql, [45](#)
- execute_statement, [11, 45, 52](#)
- execute_transaction, [11](#)
- export_journal_to_s3, [34](#)
- export_table_to_point_in_time, [11](#)
- extend_transaction, [24](#)
- failover_db_cluster, [8, 31, 41](#)
- failover_global_cluster, [31, 41](#)
- failover_global_replication_group, [18](#)
- failover_shard, [27](#)
- get_attributes, [58](#)
- get_block, [34](#)
- get_cluster_credentials, [49](#)
- get_cluster_credentials_with_iam, [49](#)
- get_credentials, [55](#)
- get_data_cells_filter, [24](#)
- get_data_lake_settings, [24](#)
- get_digest, [34](#)
- get_effective_permissions_for_path, [24](#)
- get_endpoint_access, [55](#)
- get_item, [11](#)
- get_keyspace, [21](#)
- get_lf_tag, [24](#)
- get_namespace, [55](#)
- get_query_state, [24](#)

- get_query_statistics, [24](#)
- get_records, [15](#)
- get_recovery_point, [55](#)
- get_reserved_node_exchange_configuration_options, [49](#)
- get_reserved_node_exchange_offerings, [49](#)
- get_resource_lf_tags, [24](#)
- get_resource_policy, [55](#)
- get_revision, [34](#)
- get_shard_iterator, [15](#)
- get_snapshot, [55](#)
- get_statement_result, [52](#)
- get_table, [21](#)
- get_table_objects, [24](#)
- get_table_restore_status, [55](#)
- get_temporary_glue_partition_credentials, [24](#)
- get_temporary_glue_table_credentials, [24](#)
- get_usage_limit, [55](#)
- get_work_unit_results, [24](#)
- get_work_units, [24](#)
- get_workgroup, [55](#)
- grant_permissions, [24](#)

- import_table, [11](#)
- increase_node_groups_in_global_replication_group, [18](#)
- increase_replica_count, [18](#)
- increase_replication_factor, [5](#)

- keyspaces, [19](#)

- lakeformation, [22](#)
- list_allowed_node_type_modifications, [18](#)
- list_allowed_node_type_updates, [27](#)
- list_backups, [11](#)
- list_batch_load_tasks, [63](#)
- list_contributor_insights, [11](#)
- list_data_cells_filter, [24](#)
- list_databases, [52, 63](#)
- list_domains, [58](#)
- list_endpoint_access, [55](#)
- list_exports, [11](#)
- list_global_tables, [11](#)
- list_imports, [11](#)
- list_journal_kinesis_streams_for_ledger, [34](#)
- list_journal_s3_exports, [34](#)
- list_journal_s3_exports_for_ledger, [34](#)
- list_keyspaces, [21](#)
- list_ledgers, [34](#)
- list_lf_tags, [24](#)
- list_namespaces, [55](#)
- list_permissions, [24](#)
- list_recovery_points, [55](#)
- list_resources, [24](#)
- list_scheduled_queries, [60](#)
- list_schemas, [52](#)
- list_snapshots, [55](#)
- list_statements, [52](#)
- list_streams, [15](#)
- list_table_restore_status, [55](#)
- list_table_storage_optimizers, [24](#)
- list_tables, [11, 21, 52, 63](#)
- list_tags, [5, 27](#)
- list_tags_for_resource, [8, 18, 21, 31, 34, 41, 55, 60, 63](#)
- list_tags_of_resource, [11](#)
- list_transactions, [24](#)
- list_usage_limits, [55](#)
- list_workgroups, [55](#)

- memorydb, [25](#)
- modify_activity_stream, [41](#)
- modify_aqua_configuration, [49](#)
- modify_authentication_profile, [49](#)
- modify_cache_cluster, [18](#)
- modify_cache_parameter_group, [18](#)
- modify_cache_subnet_group, [18](#)
- modify_certificates, [41](#)
- modify_cluster, [49](#)
- modify_cluster_db_revision, [49](#)
- modify_cluster_iam_roles, [49](#)
- modify_cluster_maintenance, [49](#)
- modify_cluster_parameter_group, [49](#)
- modify_cluster_snapshot, [49](#)
- modify_cluster_snapshot_schedule, [49](#)
- modify_cluster_subnet_group, [49](#)
- modify_current_db_cluster_capacity, [41](#)
- modify_custom_db_engine_version, [41](#)
- modify_custom_domain_association, [49](#)
- modify_db_cluster, [8, 31, 41](#)
- modify_db_cluster_endpoint, [31, 41](#)

- modify_db_cluster_parameter_group, [8](#),
[31](#), [41](#)
- modify_db_cluster_snapshot_attribute,
[8](#), [31](#), [41](#)
- modify_db_instance, [8](#), [31](#), [41](#)
- modify_db_parameter_group, [31](#), [41](#)
- modify_db_proxy, [41](#)
- modify_db_proxy_endpoint, [41](#)
- modify_db_proxy_target_group, [41](#)
- modify_db_snapshot, [41](#)
- modify_db_snapshot_attribute, [41](#)
- modify_db_subnet_group, [8](#), [31](#), [41](#)
- modify_endpoint_access, [49](#)
- modify_event_subscription, [8](#), [31](#), [42](#), [49](#)
- modify_global_cluster, [8](#), [31](#), [42](#)
- modify_global_replication_group, [18](#)
- modify_option_group, [42](#)
- modify_replication_group, [18](#)
- modify_replication_group_shard_configuration,
[18](#)
- modify_scheduled_action, [49](#)
- modify_snapshot_copy_retention_period,
[49](#)
- modify_snapshot_schedule, [49](#)
- modify_usage_limit, [49](#)
- modify_user, [18](#)
- modify_user_group, [18](#)

- neptune, [28](#)

- pause_cluster, [49](#)
- prepare_query, [60](#)
- promote_read_replica, [42](#)
- promote_read_replica_db_cluster, [31](#), [42](#)
- purchase_reserved_cache_nodes_offering,
[18](#)
- purchase_reserved_db_instances_offering,
[42](#)
- purchase_reserved_node_offering, [49](#)
- purchase_reserved_nodes_offering, [27](#)
- put_attributes, [58](#)
- put_data_lake_settings, [24](#)
- put_item, [11](#)
- put_resource_policy, [55](#)

- qldb, [32](#)
- qldbsession, [34](#)
- query, [11](#), [60](#)

- rds, [37](#)

- rdsdataservice, [43](#)
- rebalance_slots_in_global_replication_group,
[18](#)
- reboot_cache_cluster, [18](#)
- reboot_cluster, [49](#)
- reboot_db_cluster, [42](#)
- reboot_db_instance, [8](#), [31](#), [42](#)
- reboot_node, [5](#)
- redshift, [45](#)
- redshiftdataapiservice, [50](#)
- redshiftserverless, [53](#)
- register_db_proxy_targets, [42](#)
- register_resource, [24](#)
- reject_data_share, [49](#)
- remove_from_global_cluster, [8](#), [31](#), [42](#)
- remove_iam_tags_from_resource, [24](#)
- remove_role_from_db_cluster, [31](#), [42](#)
- remove_role_from_db_instance, [42](#)
- remove_source_identifier_from_subscription,
[8](#), [31](#), [42](#)
- remove_tags_from_resource, [8](#), [18](#), [31](#), [42](#)
- reset_cache_parameter_group, [18](#)
- reset_cluster_parameter_group, [49](#)
- reset_db_cluster_parameter_group, [8](#), [31](#),
[42](#)
- reset_db_parameter_group, [31](#), [42](#)
- reset_parameter_group, [27](#)
- resize_cluster, [49](#)
- restore_db_cluster_from_s3, [42](#)
- restore_db_cluster_from_snapshot, [8](#), [31](#),
[42](#)
- restore_db_cluster_to_point_in_time, [8](#),
[31](#), [42](#)
- restore_db_instance_from_db_snapshot,
[42](#)
- restore_db_instance_from_s3, [42](#)
- restore_db_instance_to_point_in_time,
[42](#)
- restore_from_cluster_snapshot, [49](#)
- restore_from_recovery_point, [55](#)
- restore_from_snapshot, [55](#)
- restore_table, [21](#)
- restore_table_from_backup, [11](#)
- restore_table_from_cluster_snapshot,
[50](#)
- restore_table_from_snapshot, [55](#)
- restore_table_to_point_in_time, [11](#)
- resume_batch_load_task, [63](#)

resume_cluster, [50](#)
 revoke_cache_security_group_ingress, [18](#)
 revoke_cluster_security_group_ingress, [50](#)
 revoke_db_security_group_ingress, [42](#)
 revoke_endpoint_access, [50](#)
 revoke_permissions, [24](#)
 revoke_snapshot_access, [50](#)
 rollback_transaction, [45](#)
 rotate_encryption_key, [50](#)

 scan, [11](#)
 search_databases_by_lf_tags, [24](#)
 search_tables_by_lf_tags, [24](#)
 select, [58](#)
 send_command, [35, 37](#)
 simpledb, [56](#)
 start_activity_stream, [42](#)
 start_db_cluster, [8, 31, 42](#)
 start_db_instance, [42](#)
 start_db_instance_automated_backups_replication, [42](#)
 start_export_task, [42](#)
 start_migration, [18](#)
 start_query_planning, [24](#)
 start_transaction, [24](#)
 stop_activity_stream, [42](#)
 stop_db_cluster, [8, 32, 42](#)
 stop_db_instance, [42](#)
 stop_db_instance_automated_backups_replication, [42](#)
 stream_journal_to_kinesis, [34](#)
 switchover_blue_green_deployment, [42](#)
 switchover_global_cluster, [42](#)
 switchover_read_replica, [42](#)

 tag_resource, [5, 11, 21, 27, 34, 55, 60, 63](#)
 test_failover, [18](#)
 test_migration, [18](#)
 timestreamquery, [58](#)
 timestreamwrite, [61](#)
 transact_get_items, [11](#)
 transact_write_items, [12](#)

 untag_resource, [5, 12, 21, 27, 34, 55, 60, 63](#)
 update_acl, [27](#)
 update_cluster, [5, 28](#)
 update_continuous_backups, [12](#)
 update_contributor_insights, [12](#)
 update_data_cells_filter, [24](#)
 update_database, [63](#)
 update_endpoint_access, [55](#)
 update_global_table, [12](#)
 update_global_table_settings, [12](#)
 update_item, [12](#)
 update_ledger, [34](#)
 update_ledger_permissions_mode, [34](#)
 update_lf_tag, [25](#)
 update_namespace, [55](#)
 update_parameter_group, [5, 28](#)
 update_partner_status, [50](#)
 update_resource, [25](#)
 update_scheduled_query, [60](#)
 update_snapshot, [55](#)
 update_subnet_group, [5, 28](#)
 update_table, [12, 21, 63](#)
 update_table_objects, [25](#)
 update_table_replica_auto_scaling, [12](#)
 update_table_storage_optimizer, [25](#)
 update_time_to_live, [12](#)
 update_usage_limit, [55](#)
 update_user, [28](#)
 update_workgroup, [55](#)

 write_records, [63](#)