

Package ‘paws.database’

October 7, 2026

Title 'Amazon Web Services' Database Services

Version 0.11.0

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>,
<https://paws-r.r-universe.dev/paws.database>,
<https://www.paws-r-sdk.com>

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

Depends R (>= 3.5.0)

Imports paws.common (>= 0.9.0)

Suggests testthat

Encoding UTF-8

Config/roxygen2/version 8.1.0

Collate 'dax_service.R' 'dax_interfaces.R' 'dax_operations.R'
'docdb_service.R' 'docdb_interfaces.R' 'docdb_operations.R'
'docdbelastic_service.R' 'docdbelastic_interfaces.R'
'docdbelastic_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'
'neptunedata_service.R' 'neptunedata_interfaces.R'
'neptunedata_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 2026-10-07 19:30:22 UTC

Contents

dax	3
docdb	5
docdbelastic	9
dynamodb	12
dynamodbstreams	16
elasticsearch	18
keyspaces	23
lakeformation	26
memorydb	29
neptune	33
neptunedata	37
rds	40
rdsdataservice	46
redshift	49
redshiftdataapiservice	54
redshiftserverless	57
simplifiedb	61
timestreamquery	64
timestreamwrite	66

Index	70
--------------	-----------

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 Amazon Web Services 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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

credentials

Optional credentials shorthand for the config parameter

- **creds:**

- **access_key_id:** AWS access key ID

	– secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

<code>create_cluster</code>	Creates a DAX cluster
<code>create_parameter_group</code>	Creates a new parameter group
<code>create_subnet_group</code>	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)
```

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. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```

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

```

Operations

[add_source_identifier_to_subscription](#)
[add_tags_to_resource](#)
[apply_pending_maintenance_action](#)
[copy_db_cluster_parameter_group](#)
[copy_db_cluster_snapshot](#)
[create_db_cluster](#)
[create_db_cluster_parameter_group](#)
[create_db_cluster_snapshot](#)
[create_db_instance](#)
[create_db_subnet_group](#)
[create_event_subscription](#)
[create_global_cluster](#)

Adds a source identifier to an existing event notification subscription
 Adds metadata tags to an Amazon DocumentDB resource
 Applies a pending maintenance action to a resource (for example, to an Amazon DocumentDB instance)
 Copies the specified cluster parameter group
 Copies a snapshot of a cluster
 Creates a new Amazon DocumentDB cluster
 Creates a new cluster parameter group
 Creates a snapshot of a cluster
 Creates a new instance
 Creates a new subnet group
 Creates an Amazon DocumentDB event notification subscription
 Creates an Amazon DocumentDB global cluster that can span multiple multiple Availability Zones

<code>delete_db_cluster</code>	Deletes a previously provisioned cluster
<code>delete_db_cluster_parameter_group</code>	Deletes a specified cluster parameter group
<code>delete_db_cluster_snapshot</code>	Deletes a cluster snapshot
<code>delete_db_instance</code>	Deletes a previously provisioned instance
<code>delete_db_subnet_group</code>	Deletes a subnet group
<code>delete_event_subscription</code>	Deletes an Amazon DocumentDB event notification subscription
<code>delete_global_cluster</code>	Deletes a global cluster
<code>describe_certificates</code>	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>failover_global_cluster</code>	Promotes the specified secondary DB cluster to be the primary DB cluster in the global 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
<code>switchover_global_cluster</code>	Switches over the specified secondary Amazon DocumentDB cluster to be the primary DB cluster in the global cluster

Examples

Not run:

```

svc <- docdb()
svc$add_source_identifier_to_subscription(
  Foo = 123
)

## End(Not run)

```

docdbelastic

Amazon DocumentDB Elastic Clusters

Description

Amazon DocumentDB elastic clusters

Amazon DocumentDB elastic-clusters support workloads with millions of reads/writes per second and petabytes of storage capacity. Amazon DocumentDB elastic clusters also simplify how developers interact with Amazon DocumentDB elastic-clusters by eliminating the need to choose, manage or upgrade instances.

Amazon DocumentDB elastic-clusters were created to:

- provide a solution for customers looking for a database that provides virtually limitless scale with rich query capabilities and MongoDB API compatibility.
- give customers higher connection limits, and to reduce downtime from patching.
- continue investing in a cloud-native, elastic, and class leading architecture for JSON workloads.

Usage

```

docdbelastic(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)

```

Arguments

- | | |
|--------|---|
| config | Optional configuration of credentials, endpoint, and/or region. |
|--------|---|
- **credentials:**
 - **creds:**
 - * **access_key_id:** AWS access key ID
 - * **secret_access_key:** AWS secret access key
 - * **session_token:** AWS temporary session token
 - **profile:** The name of a profile to use. If not given, then the default profile is used.
 - **anonymous:** Set anonymous credentials.

- **endpoint:** The complete URL to use for the constructed client.
- **region:** The AWS Region used in instantiating the client.
- **close_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style:** Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint:** Set sts regional endpoint resolver to regional or legacy <https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html>
- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

credentials	Optional credentials shorthand for the config parameter
	• creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

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

```

Operations

apply_pending_maintenance_action	The type of pending maintenance action to be applied to the resource
copy_cluster_snapshot	Copies a snapshot of an elastic cluster
create_cluster	Creates a new Amazon DocumentDB elastic cluster and returns its cluster structure
create_cluster_snapshot	Creates a snapshot of an elastic cluster
delete_cluster	Delete an elastic cluster
delete_cluster_snapshot	Delete an elastic cluster snapshot
get_cluster	Returns information about a specific elastic cluster
get_cluster_snapshot	Returns information about a specific elastic cluster snapshot
get_pending_maintenance_action	Retrieves all maintenance actions that are pending
list_clusters	Returns information about provisioned Amazon DocumentDB elastic clusters
list_cluster_snapshots	Returns information about snapshots for a specified elastic cluster
list_pending_maintenance_actions	Retrieves a list of all maintenance actions that are pending
list_tags_for_resource	Lists all tags on a elastic cluster resource
restore_cluster_from_snapshot	Restores an elastic cluster from a snapshot
start_cluster	Restarts the stopped elastic cluster that is specified by clusterARN
stop_cluster	Stops the running elastic cluster that is specified by clusterArn
tag_resource	Adds metadata tags to an elastic cluster resource
untag_resource	Removes metadata tags from an elastic cluster resource
update_cluster	Modifies an elastic cluster

Examples

```

## Not run:
svc <- docdbelastic()
svc$apply_pending_maintenance_action(

```

```

    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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```
svc <- 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",
    use_dual_stack = "logical"
  ),
```

```

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

```

Operations

[batch_execute_statement](#)
[batch_get_item](#)
[batch_write_item](#)
[create_backup](#)
[create_global_table](#)
[create_table](#)
[delete_backup](#)
[delete_item](#)
[delete_resource_policy](#)
[delete_table](#)
[describe_backup](#)
[describe_continuous_backups](#)
[describe_contributor_insights](#)
[describe_endpoints](#)
[describe_export](#)
[describe_global_table](#)
[describe_global_table_settings](#)
[describe_import](#)
[describe_kinesis_streaming_destination](#)
[describe_limits](#)
[describe_table](#)
[describe_table_replica_auto_scaling](#)
[describe_time_to_live](#)
[disable_kinesis_streaming_destination](#)
[enable_kinesis_streaming_destination](#)
[execute_statement](#)
[execute_transaction](#)
[export_table_to_point_in_time](#)
[get_item](#)
[get_resource_policy](#)
[import_table](#)
[list_backups](#)
[list_contributor_insights](#)

This operation allows you to perform batch reads or writes on data stored in DynamoDB.
 The BatchGetItem operation returns the attributes of one or more items from one or more tables.
 The BatchWriteItem operation puts or deletes multiple items in one or more tables.
 Creates a backup for an existing table.
 Creates a global table from an existing table.
 The CreateTable operation adds a new table to your account.
 Deletes an existing backup of a table.
 Deletes a single item in a table by primary key.
 Deletes the resource-based policy attached to the resource, which can be a table or a global table.
 The DeleteTable operation deletes a table and all of its items.
 Describes an existing backup of a table.
 Checks the status of continuous backups and point in time recovery on the specified table.
 Returns information about contributor insights for a given table or global secondary index.
 Returns the regional endpoint information.
 Describes an existing table export.
 Returns information about the specified global table.
 Describes Region-specific settings for a global table.
 Represents the properties of the import.
 Returns information about the status of Kinesis streaming.
 Returns the current provisioned-capacity quotas for your Amazon Web Services account.
 Returns information about the table, including the current status of the table, when it is in a consistent state.
 Describes auto scaling settings across replicas of the global table at once.
 Gives a description of the Time to Live (TTL) status on the specified table.
 Stops replication from the DynamoDB table to the Kinesis data stream.
 Starts table data replication to the specified Kinesis data stream at a timestamp chosen by the user.
 This operation allows you to perform reads and singleton writes on data stored in DynamoDB.
 This operation allows you to perform transactional reads or writes on data stored in DynamoDB.
 Exports table data to an S3 bucket.
 The GetItem operation returns a set of attributes for the item with the given primary key.
 Returns the resource-based policy document attached to the resource, which can be a table or a global table.
 Imports table data from an S3 bucket.
 List DynamoDB backups that are associated with an Amazon Web Services account.
 Returns a list of ContributorInsightsSummary for a table and all its global secondary indexes.

<code>list_exports</code>	Lists completed exports within the past 90 days, in reverse alphanumeric order of E
<code>list_global_tables</code>	Lists all global tables that have a replica in the specified Region
<code>list_imports</code>	Lists completed imports within the past 90 days
<code>list_tables</code>	Returns an array of table names associated with the current account and endpoint
<code>list_tags_of_resource</code>	List all tags on an Amazon DynamoDB resource
<code>put_item</code>	Creates a new item, or replaces an old item with a new item
<code>put_resource_policy</code>	Attaches a resource-based policy document to the resource, which can be a table or
<code>query</code>	You must provide the name of the partition key attribute and a single value for that
<code>restore_table_from_backup</code>	Creates a new table from an existing backup
<code>restore_table_to_point_in_time</code>	Restores the specified table to the specified point in time within EarliestRestorableE
<code>scan</code>	The Scan operation returns one or more items and item attributes by accessing every
<code>search_vectors</code>	Performs a vector similarity search on a vector index associated with an Amazon D
<code>tag_resource</code>	Associate a set of tags with an Amazon DynamoDB resource
<code>transact_get_items</code>	TransactGetItems is a synchronous operation that atomically retrieves multiple item
<code>transact_write_items</code>	TransactWriteItems is a synchronous write operation that groups up to 100 action re
<code>untag_resource</code>	Removes the association of tags from an Amazon DynamoDB resource
<code>update_continuous_backups</code>	UpdateContinuousBackups enables or disables point in time recovery for the specif
<code>update_contributor_insights</code>	Updates the status for contributor insights for a specific table or index
<code>update_global_table</code>	Adds or removes replicas in the specified global table
<code>update_global_table_settings</code>	Updates settings for a global table
<code>update_item</code>	Edits an existing item's attributes, or adds a new item to the table if it does not alrea
<code>update_kinesis_streaming_destination</code>	The command to update the Kinesis stream destination
<code>update_table</code>	Modifies the provisioned throughput settings, global secondary indexes, or Dynamoc
<code>update_table_replica_auto_scaling</code>	Updates auto scaling settings on your global tables at once
<code>update_time_to_live</code>	The UpdateTimeToLive method enables or disables Time to Live (TTL) for the spec

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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

Operations

describe_stream	Returns information about a stream, including the current status of the stream, its Amazon Resource Name, and its shard information.
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)

```

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.

- **credentials:**
 - **creds:**
 - * **access_key_id:** AWS access key ID
 - * **secret_access_key:** AWS secret access key
 - * **session_token:** AWS temporary session token
 - **profile:** The name of a profile to use. If not given, then the default profile is used.
 - **anonymous:** Set anonymous credentials.
- **endpoint:** The complete URL to use for the constructed client.
- **region:** The AWS Region used in instantiating the client.
- **close_connection:** Immediately close all HTTP connections.
- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style:** Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint:** Set sts regional endpoint resolver to regional or legacy <https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html>
- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

        endpoint = "string",
        region = "string"
    )

```

Operations

add_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_serverless_cache_snapshot	Creates a copy of an existing serverless cache's snapshot
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 offers fully managed, fast, reliable and secure cross-region replication
create_replication_group	Creates a Valkey or Redis OSS (cluster mode disabled) or a Valkey or Redis Enterprise
create_serverless_cache	Creates a serverless cache
create_serverless_cache_snapshot	This API creates a copy of an entire ServerlessCache at a specific moment in time
create_snapshot	Creates a copy of an entire cluster or replication group at a specific moment in time
create_user	For Valkey engine version 7
create_user_group	For Valkey engine version 7
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 Valkey or Redis OSS
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
delete_cache_subnet_group	Deletes a cache subnet group
delete_global_replication_group	Deleting a Global datastore is a two-step process:
delete_replication_group	Deletes an existing replication group
delete_serverless_cache	Deletes a specified existing serverless cache
delete_serverless_cache_snapshot	Deletes an existing serverless cache snapshot
delete_snapshot	Deletes an existing snapshot
delete_user	For Valkey engine version 7
delete_user_group	For Valkey engine version 7
describe_cache_clusters	Returns information about all provisioned clusters if no cluster identifier is specified
describe_cache_engine_versions	Returns a list of the available cache engines and their versions
describe_cache_parameter_groups	Returns a list of cache parameter group descriptions
describe_cache_parameters	Returns the detailed parameter list for a particular cache parameter group
describe_cache_security_groups	Returns a list of cache security group descriptions
describe_cache_subnet_groups	Returns a list of cache subnet group descriptions
describe_engine_default_parameters	Returns the default engine and system parameter information for the specified engine
describe_events	Returns events related to clusters, cache security groups, and cache parameter groups
describe_global_replication_groups	Returns information about a particular global replication group
describe_replication_groups	Returns information about a particular replication group
describe_reserved_cache_nodes	Returns information about reserved cache nodes for this account, or about all reserved nodes

<code>describe_reserved_cache_nodes_offerings</code>	Lists available reserved cache node offerings
<code>describe_serverless_caches</code>	Returns information about a specific serverless cache
<code>describe_serverless_cache_snapshots</code>	Returns information about serverless cache snapshots
<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
<code>export_serverless_cache_snapshot</code>	Provides the functionality to export the serverless cache snapshot data
<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 Valkey or Redis OS
<code>list_allowed_node_type_modifications</code>	Lists all available node types that you can scale with your cluster's repl
<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
<code>modify_serverless_cache</code>	This API modifies the attributes of a serverless cache
<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 resour
<code>reset_cache_parameter_group</code>	Modifies the parameters of a cache parameter group to the engine or sy
<code>revoke_cache_security_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
<code>test_migration</code>	Async API to test connection between source and target replication gro

Examples

```
## Not run:
svc <- elasticache()
# Adds up to 10 tags, key/value pairs, to a cluster or snapshot resource.
svc$add_tags_to_resource(
  ResourceName = "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-cluster",
  Tags = list(
    list(
      Key = "APIVersion",
      Value = "20150202"
    ),
    list(
```

```

        Key = "Service",
        Value = "ElasticCache"
    )
)
)

## 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.

- **credentials:**
 - **creds:**

	* access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

Operations

create_keyspace	The CreateKeyspace operation adds a new keyspace to your account
create_table	The CreateTable operation adds a new table to the specified keyspace
create_type	The CreateType operation creates a new user-defined type in 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
delete_type	The DeleteType operation deletes a user-defined type (UDT)
get_keyspace	Returns the name of the specified keyspace, the Amazon Resource Name (ARN), the replication factor, and the number of replicas
get_table	Returns information about the table, including the table's name and current status, the keyspace name, and the table's replication factor
get_table_auto_scaling_settings	Returns auto scaling related settings of the specified table in JSON format
get_type	The GetType operation returns information about the type, for example the field definitions, the type name, and the type's namespace
list_keyspaces	The ListKeyspaces operation returns a list of keyspaces
list_tables	The ListTables operation 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
list_types	The ListTypes operation returns a list of types for a specified keyspace
restore_table	Restores the table to the specified point in time within the earliest_restorable_timestamp and latest_restorable_timestamp range
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_keyspace	Adds a new Amazon Web Services Region to the keyspace
update_table	Adds new columns to the table or updates one of the table's settings, for example capacity mode

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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```
svc <- 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",
    use_dual_stack = "logical"
  ),
```

```

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

```

Operations

add_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
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
create_lake_formation_identity_center_configuration	Creates an IAM Identity Center connection with Lake Formation to enforce permissions
create_lake_formation_opt_in	Enforce Lake Formation permissions for the given databases, tables, and views
create_lf_tag	Creates an LF-tag with the specified name and values
create_lf_tag_expression	Creates a new LF-Tag expression with the provided name, description, and values
delete_data_cells_filter	Deletes a data cell filter
delete_lake_formation_identity_center_configuration	Deletes an IAM Identity Center connection with Lake Formation
delete_lake_formation_opt_in	Remove the Lake Formation permissions enforcement of the given resources
delete_lf_tag	Deletes an LF-tag by its key name
delete_lf_tag_expression	Deletes the LF-Tag expression
delete_objects_on_cancel	For a specific governed table, provides a list of Amazon S3 objects that are deleted when a transaction is canceled
deregister_resource	Deregisters the resource as managed by the Data Catalog
describe_lake_formation_identity_center_configuration	Retrieves the instance ARN and application ARN for the connection
describe_resource	Retrieves the current data access role for the given resource registered in the Data Catalog
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_principal	Returns the identity of the invoking principal
get_data_lake_settings	Retrieves the list of the data lake administrators of a Lake Formation instance
get_effective_permissions_for_path	Returns the Lake Formation permissions for a specified table or database
get_lf_tag	Returns an LF-tag definition
get_lf_tag_expression	Returns the details about the LF-Tag expression
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_data_location_credentials	Allows a user or application in a secure environment to access data in a temporary location

<code>get_temporary_glue_partition_credentials</code>	This API is identical to <code>GetTemporaryTableCredentials</code> except that it
<code>get_temporary_glue_table_credentials</code>	Allows a caller in a secure environment to assume a role with permissions
<code>get_work_unit_results</code>	Returns the work units resulting from the query
<code>get_work_units</code>	Retrieves the work units generated by the <code>StartQueryPlanning</code> operation
<code>grant_permissions</code>	Grants permissions to the principal to access metadata in the Data Catalog
<code>list_data_cells_filter</code>	Lists all the data cell filters on a table
<code>list_lake_formation_opt_ins</code>	Retrieve the current list of resources and principals that are opted in to
<code>list_lf_tag_expressions</code>	Returns the LF-Tag expressions in caller's account filtered based on the
<code>list_lf_tags</code>	Lists LF-tags that the requester has permission to view
<code>list_permissions</code>	Returns a list of the principal permissions on the resource, filtered based
<code>list_resources</code>	Lists the resources registered to be managed by the Data Catalog
<code>list_table_storage_optimizers</code>	Returns the configuration of all storage optimizers associated with a table
<code>list_transactions</code>	Returns metadata about transactions and their status
<code>put_data_lake_settings</code>	Sets the list of data lake administrators who have admin privileges on the
<code>register_resource</code>	Registers the resource as managed by the Data Catalog
<code>remove_lf_tags_from_resource</code>	Removes an LF-tag from the resource
<code>revoke_permissions</code>	Revokes permissions to the principal to access metadata in the Data Catalog
<code>search_databases_by_lf_tags</code>	This operation allows a search on DATABASE resources by TagConfiguration
<code>search_tables_by_lf_tags</code>	This operation allows a search on TABLE resources by LFTags
<code>start_query_planning</code>	Submits a request to process a query statement
<code>start_transaction</code>	Starts a new transaction and returns its transaction ID
<code>update_data_cells_filter</code>	Updates a data cell filter
<code>update_lake_formation_identity_center_configuration</code>	Updates the IAM Identity Center connection parameters
<code>update_lf_tag</code>	Updates the list of possible values for the specified LF-tag key
<code>update_lf_tag_expression</code>	Updates the name of the LF-Tag expression to the new description and
<code>update_resource</code>	Updates the data access role used for vending access to the given (resource)
<code>update_table_objects</code>	Updates the manifest of Amazon S3 objects that make up the specified table
<code>update_table_storage_optimizer</code>	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)
```

Description

MemoryDB is a fully managed, Redis OSS-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 OSS, a popular open source data store, enabling you to leverage Redis OSS' 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. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

batch_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_multi_region_cluster	Creates a new multi-Region 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

<code>create_subnet_group</code>	Creates a subnet group
<code>create_user</code>	Creates a MemoryDB user
<code>delete_acl</code>	Deletes an Access Control List
<code>delete_cluster</code>	Deletes a cluster
<code>delete_multi_region_cluster</code>	Deletes an existing multi-Region cluster
<code>delete_parameter_group</code>	Deletes the specified parameter group
<code>delete_snapshot</code>	Deletes an existing snapshot
<code>delete_subnet_group</code>	Deletes a subnet group
<code>delete_user</code>	Deletes a user
<code>describe_ac_ls</code>	Returns a list of ACLs
<code>describe_clusters</code>	Returns information about all provisioned clusters if no cluster identifier is specified
<code>describe_engine_versions</code>	Returns a list of the available Redis OSS engine versions
<code>describe_events</code>	Returns events related to clusters, security groups, and parameter groups
<code>describe_multi_region_clusters</code>	Returns details about one or more multi-Region clusters
<code>describe_multi_region_parameter_groups</code>	Returns a list of multi-region parameter groups
<code>describe_multi_region_parameters</code>	Returns the detailed parameter list for a particular multi-region parameter group
<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_reserved_nodes</code>	Returns information about reserved nodes for this account, or about a specified reserved node
<code>describe_reserved_nodes_offerings</code>	Lists available reserved node offerings
<code>describe_service_updates</code>	Returns details of the service updates
<code>describe_snapshots</code>	Returns information about cluster snapshots
<code>describe_subnet_groups</code>	Returns a list of subnet group descriptions
<code>describe_users</code>	Returns a list of users
<code>failover_shard</code>	Used to failover a shard
<code>list_allowed_multi_region_cluster_updates</code>	Lists the allowed updates for a multi-Region cluster
<code>list_allowed_node_type_updates</code>	Lists all available node types that you can scale to from your cluster's current node type
<code>list_tags</code>	Lists all tags currently on a named resource
<code>purchase_reserved_nodes_offering</code>	Allows you to purchase a reserved node offering
<code>reset_parameter_group</code>	Modifies the parameters of a parameter group to the engine or system default values
<code>tag_resource</code>	Use this operation to add tags to a resource
<code>untag_resource</code>	Use this operation to remove tags on a resource
<code>update_acl</code>	Changes the list of users that belong to the Access Control List
<code>update_cluster</code>	Modifies the settings for a cluster
<code>update_multi_region_cluster</code>	Updates the configuration of an existing multi-Region 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. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
--------	---

	• s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```
svc <- 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",
    use_dual_stack = "logical"
  ),
```

```

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

```

Operations

add_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
delete_db_instance	The DeleteDBInstance action deletes a previously provisioned DB instance
delete_db_parameter_group	Deletes a specified DBParameterGroup
delete_db_subnet_group	Deletes a DB subnet group
delete_event_subscription	Deletes an event notification subscription
delete_global_cluster	Deletes a global database
describe_db_cluster_endpoints	Returns information about endpoints for an Amazon Neptune DB cluster
describe_db_cluster_parameter_groups	Returns a list of DBClusterParameterGroup descriptions
describe_db_cluster_parameters	Returns the detailed parameter list for a particular DB cluster parameter group
describe_db_clusters	Returns information about provisioned DB clusters, and supports pagination
describe_db_cluster_snapshot_attributes	Returns a list of DB cluster snapshot attribute names and values for a manual snapshot
describe_db_cluster_snapshots	Returns information about DB cluster snapshots
describe_db_engine_versions	Returns a list of the available DB engines
describe_db_instances	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 default parameter group
<code>describe_engine_default_parameters</code>	Returns the default engine and system parameter information for the specified 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 DB instances, DB security groups, DB snapshots, and DB clusters
<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 pending maintenance action
<code>describe_valid_db_instance_modifications</code>	You can call DescribeValidDBInstanceModifications to learn what modifications are valid for a DB instance
<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 manual snapshot attribute
<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 cluster
<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 Neptune console
<code>stop_db_cluster</code>	Stops an Amazon Neptune DB cluster
<code>switchover_global_cluster</code>	Switches over the specified secondary DB cluster to be the new primary DB cluster

Examples

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

## End(Not run)
```

neptunedata

Amazon NeptuneData

Description

Neptune Data API

The Amazon Neptune data API provides SDK support for more than 40 of Neptune's data operations, including data loading, query execution, data inquiry, and machine learning. It supports the Gremlin and openCypher query languages, and is available in all SDK languages. It automatically signs API requests and greatly simplifies integrating Neptune into your applications.

Usage

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

Arguments

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

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

	• use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

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

```

Operations

<code>cancel_gremlin_query</code>	Cancels a Gremlin query
<code>cancel_loader_job</code>	Cancels a specified load job
<code>cancel_ml_data_processing_job</code>	Cancels a Neptune ML data processing job
<code>cancel_ml_model_training_job</code>	Cancels a Neptune ML model training job
<code>cancel_ml_model_transform_job</code>	Cancels a specified model transform job
<code>cancel_open_cypher_query</code>	Cancels a specified openCypher query
<code>create_ml_endpoint</code>	Creates a new Neptune ML inference endpoint that lets you query one specific model
<code>delete_ml_endpoint</code>	Cancels the creation of a Neptune ML inference endpoint
<code>delete_propertygraph_statistics</code>	Deletes statistics for Gremlin and openCypher (property graph) data
<code>delete_sparql_statistics</code>	Deletes SPARQL statistics
<code>execute_fast_reset</code>	The fast reset REST API lets you reset a Neptune graph quickly and easily, removing a
<code>execute_gremlin_explain_query</code>	Executes a Gremlin Explain query
<code>execute_gremlin_profile_query</code>	Executes a Gremlin Profile query, which runs a specified traversal, collects various me
<code>execute_gremlin_query</code>	This commands executes a Gremlin query
<code>execute_open_cypher_explain_query</code>	Executes an openCypher explain request
<code>execute_open_cypher_query</code>	Executes an openCypher query
<code>get_engine_status</code>	Retrieves the status of the graph database on the host
<code>get_gremlin_query_status</code>	Gets the status of a specified Gremlin query
<code>get_loader_job_status</code>	Gets status information about a specified load job
<code>get_ml_data_processing_job</code>	Retrieves information about a specified data processing job
<code>get_ml_endpoint</code>	Retrieves details about an inference endpoint
<code>get_ml_model_training_job</code>	Retrieves information about a Neptune ML model training job
<code>get_ml_model_transform_job</code>	Gets information about a specified model transform job
<code>get_open_cypher_query_status</code>	Retrieves the status of a specified openCypher query
<code>get_propertygraph_statistics</code>	Gets property graph statistics (Gremlin and openCypher)
<code>get_propertygraph_stream</code>	Gets a stream for a property graph
<code>get_propertygraph_summary</code>	Gets a graph summary for a property graph
<code>get_rdf_graph_summary</code>	Gets a graph summary for an RDF graph
<code>get_sparql_statistics</code>	Gets RDF statistics (SPARQL)
<code>get_sparql_stream</code>	Gets a stream for an RDF graph
<code>list_gremlin_queries</code>	Lists active Gremlin queries
<code>list_loader_jobs</code>	Retrieves a list of the loadIds for all active loader jobs
<code>list_ml_data_processing_jobs</code>	Returns a list of Neptune ML data processing jobs
<code>list_ml_endpoints</code>	Lists existing inference endpoints
<code>list_ml_model_training_jobs</code>	Lists Neptune ML model-training jobs
<code>list_ml_model_transform_jobs</code>	Returns a list of model transform job IDs
<code>list_open_cypher_queries</code>	Lists active openCypher queries
<code>manage_propertygraph_statistics</code>	Manages the generation and use of property graph statistics

<code>manage_sparql_statistics</code>	Manages the generation and use of RDF graph statistics
<code>start_loader_job</code>	Starts a Neptune bulk loader job to load data from an Amazon S3 bucket into a Neptune graph database
<code>start_ml_data_processing_job</code>	Creates a new Neptune ML data processing job for processing the graph data exported from a Neptune graph database
<code>start_ml_model_training_job</code>	Creates a new Neptune ML model training job
<code>start_ml_model_transform_job</code>	Creates a new model transform job

Examples

```
## Not run:
svc <- neptunedata()
svc$cancel_gremlin_query(
  Foo = 123
)

## End(Not run)
```

rds	<i>Amazon Relational Database Service</i>
-----	---

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, Db2, 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. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

[add_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](#)

Associates an Identity and Access Management (IAM) role with a DB cluster
 Associates an Amazon Web Services Identity and Access Management (IAM) role with a DB instance
 Adds a source identifier to an existing RDS event notification subscription
 Adds metadata tags to an Amazon RDS resource
 Applies a pending maintenance action to a resource (for example, to a DB instance)
 Enables ingress to a DBSecurityGroup using one of two forms of authorization
 Backtracks a DB cluster to a specific time, without creating a new DB cluster

<code>build_auth_token</code>	Return an authentication token for a database connection
<code>build_auth_token_v2</code>	Generates an auth token used to connect to a db with IAM credentials
<code>cancel_export_task</code>	Cancels an export task in progress that is exporting a snapshot or cluster
<code>copy_db_cluster_parameter_group</code>	Copies the specified DB cluster parameter group
<code>copy_db_cluster_snapshot</code>	Copies a snapshot of a DB cluster
<code>copy_db_parameter_group</code>	Copies the specified DB parameter group
<code>copy_db_snapshot</code>	Copies the specified DB snapshot
<code>copy_option_group</code>	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_shard_group</code>	Creates a new DB shard group for Aurora Limitless Database
<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_integration</code>	Creates a zero-ETL integration with Amazon Redshift
<code>create_option_group</code>	Creates a new option group
<code>create_tenant_database</code>	Creates a tenant database in a DB instance that uses the multi-tenant configuration
<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>	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_shard_group</code>	Deletes an Aurora Limitless Database DB shard 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_integration</code>	Deletes a zero-ETL integration with Amazon Redshift
<code>delete_option_group</code>	Deletes an existing option group

delete_tenant_database	Deletes a tenant database from your DB instance
deregister_db_proxy_targets	Remove the association between one or more DBProxyTarget data structures and a DB proxy
describe_account_attributes	Lists all of the attributes for a customer account
describe_blue_green_deployments	Describes one or more blue/green deployments
describe_certificates	Lists the set of certificate authority (CA) certificates provided by Amazon RDS
describe_db_cluster_automated_backups	Displays backups for both current and deleted DB clusters
describe_db_cluster_backtracks	Returns information about backtracks for a DB cluster
describe_db_cluster_endpoints	Returns information about endpoints for an Amazon Aurora DB cluster
describe_db_cluster_parameter_groups	Returns a list of DBClusterParameterGroup descriptions
describe_db_cluster_parameters	Returns the detailed parameter list for a particular DB cluster parameter group
describe_db_clusters	Describes existing Amazon Aurora DB clusters and Multi-AZ DB clusters
describe_db_cluster_snapshot_attributes	Returns a list of DB cluster snapshot attribute names and values for a manual DB cluster snapshot
describe_db_cluster_snapshots	Returns information about DB cluster snapshots
describe_db_engine_versions	Describes the properties of specific versions of 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_major_engine_versions	Describes the properties of specific major versions of DB engines
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 DBProxyTarget objects
describe_db_proxy_targets	Returns information about DBProxyTarget objects
describe_db_recommendations	Describes the recommendations to resolve the issues for your DB instances
describe_db_security_groups	Returns a list of DBSecurityGroup descriptions
describe_db_shard_groups	Describes existing Aurora Limitless Database DB shard groups
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_snapshot_tenant_databases	Describes the tenant databases that exist in a DB snapshot
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 engine
describe_event_categories	Displays a list of categories for all event source types, or, if specified, for a specific event source type
describe_events	Returns events related to DB instances, DB clusters, DB parameter groups, or 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_integrations	Describe one or more zero-ETL integrations with Amazon Redshift
describe_option_group_options	Describes all available options for the specified engine
describe_option_groups	Describes the available option groups
describe_orderable_db_instance_options	Describes the orderable DB instance options for a 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_serverless_v2_platform_versions	Describes the properties of specific platform versions for Aurora Serverless
describe_source_regions	Returns a list of the source Amazon Web Services Regions where the current DB instance is provisioned
describe_tenant_databases	Describes the tenant databases in a DB instance that uses the multi-tenant architecture

describe_valid_db_instance_modifications
disable_http_endpoint
download_db_log_file_portion
enable_http_endpoint
failover_db_cluster
failover_global_cluster
list_tags_for_resource
modify_activity_stream
modify_certificates
modify_current_db_cluster_capacity
modify_custom_db_engine_version
modify_db_cluster
modify_db_cluster_endpoint
modify_db_cluster_parameter_group
modify_db_cluster_snapshot_attribute
modify_db_instance
modify_db_parameter_group
modify_db_proxy
modify_db_proxy_endpoint
modify_db_proxy_target_group
modify_db_recommendation
modify_db_shard_group
modify_db_snapshot
modify_db_snapshot_attribute
modify_db_subnet_group
modify_event_subscription
modify_global_cluster
modify_integration
modify_option_group
modify_tenant_database
promote_read_replica
promote_read_replica_db_cluster
purchase_reserved_db_instances_offering
reboot_db_cluster
reboot_db_instance
reboot_db_shard_group
register_db_proxy_targets
remove_from_global_cluster
remove_role_from_db_cluster
remove_role_from_db_instance
remove_source_identifier_from_subscription
remove_tags_from_resource
reset_db_cluster_parameter_group
reset_db_parameter_group
restore_db_cluster_from_s3
restore_db_cluster_from_snapshot
restore_db_cluster_to_point_in_time
restore_db_instance_from_db_snapshot

You can call DescribeValidDBInstanceModifications to learn what modifications are available for a DB instance.
Disables the HTTP endpoint for the specified DB cluster
Downloads all or a portion of the specified log file, up to 1 MB in size
Enables the HTTP endpoint for the DB cluster
Forces a failover for a DB cluster
Promotes the specified secondary DB cluster to be the primary DB cluster
Lists all tags on an Amazon RDS resource
Changes the audit policy state of a database activity stream to either locked or unlocked
Override the system-default Secure Sockets Layer/Transport Layer Security (SSL) mode for a DB instance
Set the capacity of an Aurora Serverless v1 DB cluster to a specific value
Modifies the status of a custom engine version (CEV)
Modifies the settings of an Amazon Aurora DB cluster or a Multi-AZ DB instance
Modifies the properties of an endpoint in an Amazon Aurora DB cluster
Modifies the parameters of a DB cluster parameter group
Adds an attribute and values to, or removes an attribute and values from, a DB cluster snapshot
Modifies settings for a DB instance
Modifies the parameters of a DB parameter group
Changes the settings for an existing DB proxy
Changes the settings for an existing DB proxy endpoint
Modifies the properties of a DBProxyTargetGroup
Updates the recommendation status and recommended action status for a DB instance
Modifies the settings of an Aurora Limitless Database DB shard group
Updates a manual DB snapshot with a new engine version
Adds an attribute and values to, or removes an attribute and values from, a DB snapshot
Modifies an existing DB subnet group
Modifies an existing RDS event notification subscription
Modifies a setting for an Amazon Aurora global database cluster
Modifies a zero-ETL integration with Amazon Redshift
Modifies an existing option group
Modifies an existing tenant database in a DB instance
Promotes a read replica DB instance to a standalone DB instance
Promotes a read replica DB cluster to a standalone DB cluster
Purchases a reserved DB instance offering
You might need to reboot your DB cluster, usually for maintenance reasons.
You might need to reboot your DB instance, usually for maintenance reasons.
You might need to reboot your DB shard group, usually for maintenance reasons.
Associate one or more DBProxyTarget data structures with a DBProxyTargetGroup
Detaches an Aurora secondary cluster from an Aurora global database cluster
Removes the association of an Amazon Web Services Identity and Access Management (IAM) role from a DB instance
Disassociates an Amazon Web Services Identity and Access Management (IAM) role from a DB instance
Removes a source identifier from an existing RDS event notification subscription
Removes metadata tags from an Amazon RDS resource
Modifies the parameters of a DB cluster parameter group to the default values
Modifies the parameters of a DB parameter group to the engine/system default values
Creates an Amazon Aurora DB cluster from MySQL data stored in an Amazon S3 bucket
Creates a new DB cluster from a DB snapshot or DB cluster snapshot
Restores a DB cluster to an arbitrary point in time
Creates a new DB instance from a DB snapshot

<code>restore_db_instance_from_s3</code>	Amazon Relational Database Service (Amazon RDS) supports importing
<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 r
<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
<code>start_db_instance</code>	Starts an Amazon RDS DB instance that was stopped using the Amazon
<code>start_db_instance_automated_backups_replication</code>	Enables replication of automated backups to a different Amazon Web Se
<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
<code>stop_db_cluster</code>	Stops an Amazon Aurora DB cluster
<code>stop_db_instance</code>	Stops an Amazon RDS DB instance temporarily
<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 enviro

Examples

```
## Not run:
svc <- rds()
# This example add a source identifier to an event notification
# subscription.
svc$add_source_identifier_to_subscription(
  SourceIdentifier = "mymysqlinstance",
  SubscriptionName = "mymysqleventssubscription"
)

## End(Not run)
```

rdsdataservice	AWS RDS DataService
----------------	---------------------

Description

RDS Data API

Amazon RDS provides an HTTP endpoint to run SQL statements on an Amazon Aurora DB cluster. To run these statements, you use the RDS Data API (Data API).

Data API is available with the following types of Aurora databases:

- Aurora PostgreSQL - Serverless and provisioned
- Aurora MySQL - Serverless and provisioned

For more information about the Data API, see [Using RDS 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

batch_execute_statement	Runs a batch SQL statement over an array of data
begin_transaction	Starts a SQL transaction
commit_transaction	Ends a SQL transaction started with the BeginTransaction operation and commits the changes
execute_sql	Runs one or more SQL statements
execute_statement	Runs a SQL statement against a database
rollback_transaction	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.

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

	* session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

Operations

[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](#)
[create_hsm_client_certificate](#)
[create_hsm_configuration](#)
[create_integration](#)

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
 Creates an HSM client certificate that an Amazon Redshift cluster will
 Creates an HSM configuration that contains the information required b
 Creates a zero-ETL integration or S3 event integration with Amazon R

create_qev_2_idc_application
 create_redshift_idc_application
 create_scheduled_action
 create_snapshot_copy_grant
 create_snapshot_schedule
 create_tags
 create_usage_limit
 deauthorize_data_share
 delete_authentication_profile
 delete_cluster
 delete_cluster_parameter_group
 delete_cluster_security_group
 delete_cluster_snapshot
 delete_cluster_subnet_group
 delete_custom_domain_association
 delete_endpoint_access
 delete_event_subscription
 delete_hsm_client_certificate
 delete_hsm_configuration
 delete_integration
 delete_partner
 delete_qev_2_idc_application
 delete_redshift_idc_application
 delete_resource_policy
 delete_scheduled_action
 delete_snapshot_copy_grant
 delete_snapshot_schedule
 delete_tags
 delete_usage_limit
 deregister_namespace
 describe_account_attributes
 describe_authentication_profiles
 describe_cluster_db_revisions
 describe_cluster_parameter_groups
 describe_cluster_parameters
 describe_clusters
 describe_cluster_security_groups
 describe_cluster_snapshots
 describe_cluster_subnet_groups
 describe_cluster_tracks
 describe_cluster_versions
 describe_custom_domain_associations
 describe_data_shares
 describe_data_shares_for_consumer
 describe_data_shares_for_producer
 describe_default_cluster_parameters
 describe_endpoint_access
 describe_endpoint_authorization

Creates an Amazon Redshift Query Editor (QEV2) IAM Identity Center application
 Creates an Amazon Redshift application for use with IAM Identity Center
 Creates a scheduled action
 Creates a snapshot copy grant that permits Amazon Redshift to use an Amazon S3 bucket
 Create a snapshot schedule that can be associated to a cluster and which can be used to create snapshots
 Adds tags to a cluster
 Creates a usage limit for a specified Amazon Redshift feature on a cluster
 From a datashare producer account, removes authorization from the specified datashare
 Deletes an authentication profile
 Deletes a previously provisioned cluster without its final snapshot being created
 Deletes a specified Amazon Redshift parameter group
 Deletes an Amazon Redshift security group
 Deletes the specified manual snapshot
 Deletes the specified cluster subnet group
 Contains information about deleting a custom domain association for a cluster
 Deletes a Redshift-managed VPC endpoint
 Deletes an Amazon Redshift event notification subscription
 Deletes the specified HSM client certificate
 Deletes the specified Amazon Redshift HSM configuration
 Deletes a zero-ETL integration or S3 event integration with Amazon RDS
 Deletes a partner integration from a cluster
 Deletes an Amazon Redshift Query Editor (QEV2) IAM Identity Center application
 Deletes an Amazon Redshift IAM Identity Center application
 Deletes the resource policy for a specified resource
 Deletes a scheduled action
 Deletes the specified snapshot copy grant
 Deletes a snapshot schedule
 Deletes tags from a resource
 Deletes a usage limit from a cluster
 Deregisters a cluster or serverless namespace from the Amazon Web Services console
 Returns a list of attributes attached to an account
 Describes an authentication profile
 Returns an array of ClusterDbRevision objects
 Returns a list of Amazon Redshift parameter groups, including parameter group names
 Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group
 Returns properties of provisioned clusters including general cluster properties
 Returns information about Amazon Redshift security groups
 Returns one or more snapshot objects, which contain metadata about your snapshots
 Returns one or more cluster subnet group objects, which contain metadata about your subnet groups
 Returns a list of all the available maintenance tracks
 Returns descriptions of the available Amazon Redshift cluster versions
 Contains information about custom domain associations for a cluster
 Shows the status of any inbound or outbound datashares available in the specified account
 Returns a list of datashares where the account identifier being called is the producer
 Returns a list of datashares when the account identifier being called is the consumer
 Returns a list of parameter settings for the specified parameter group for the specified cluster
 Describes a Redshift-managed VPC endpoint
 Describes an endpoint authorization

describe_event_categories	Displays a list of event categories for all event source types, or for a specific event source type
describe_events	Returns events related to clusters, security groups, snapshots, and parameter groups
describe_event_subscriptions	Lists descriptions of all the Amazon Redshift event notification subscriptions
describe_hsm_client_certificates	Returns information about the specified HSM client certificate
describe_hsm_configurations	Returns information about the specified Amazon Redshift HSM configuration
describe_inbound_integrations	Returns a list of inbound integrations
describe_integrations	Describes one or more zero-ETL or S3 event integrations with Amazon S3
describe_logging_status	Describes whether information, such as queries and connection attempts, is logged
describe_node_configuration_options	Returns properties of possible node configurations such as node type, number of nodes, and availability zone
describe_orderable_cluster_options	Returns a list of orderable cluster options
describe_partners	Returns information about the partner integrations defined for a cluster
describe_qev_2_idc_applications	Lists the Amazon Redshift Query Editor (QEV2) IAM Identity Center applications
describe_redshift_idc_applications	Lists the Amazon Redshift IAM Identity Center applications
describe_reserved_node_exchange_status	Returns exchange status details and associated metadata for a reserved node exchange
describe_reserved_node_offerings	Returns a list of the available reserved node offerings by Amazon Redshift
describe_reserved_nodes	Returns the descriptions of the reserved nodes
describe_resize	Returns information about the last resize operation for the specified cluster
describe_scheduled_actions	Describes properties of scheduled actions
describe_snapshot_copy_grants	Returns a list of snapshot copy grants owned by the Amazon Web Services account
describe_snapshot_schedules	Returns a list of snapshot schedules
describe_storage	Returns account level backups storage size and provisional storage
describe_table_restore_status	Lists the status of one or more table restore requests made using the REST API
describe_tags	Returns a list of tags
describe_usage_limits	Shows usage limits on a cluster
disable_logging	Stops logging information, such as queries and connection attempts, for the specified cluster
disable_snapshot_copy	Disables the automatic copying of snapshots from one region to another
disassociate_data_share_consumer	From a datashare consumer account, remove association for the specified datashare
enable_logging	Starts logging information, such as queries and connection attempts, for the specified cluster
enable_snapshot_copy	Enables the automatic copy of snapshots from one region to another region
failover_primary_compute	Fails over the primary compute unit of the specified Multi-AZ cluster to the secondary compute unit
get_cluster_credentials	Returns a database user name and temporary password with temporary validity
get_cluster_credentials_with_iam	Returns a database user name and temporary password with temporary validity
get_identity_center_auth_token	Generates an encrypted authentication token that propagates the caller's permissions
get_reserved_node_exchange_configuration_options	Gets the configuration options for the reserved-node exchange
get_reserved_node_exchange_offerings	Returns an array of DC2 ReservedNodeOfferings that matches the payment type
get_resource_policy	Get the resource policy for a specified resource
list_recommendations	List the Amazon Redshift Advisor recommendations for one or multiple clusters
modify_aqua_configuration	This operation is retired
modify_authentication_profile	Modifies an authentication profile
modify_cluster	Modifies the settings for a cluster
modify_cluster_db_revision	Modifies the database revision of a cluster
modify_cluster_iam_roles	Modifies the list of Identity and Access Management (IAM) roles that the cluster can assume
modify_cluster_maintenance	Modifies the maintenance settings of a cluster
modify_cluster_parameter_group	Modifies the parameters of a parameter group
modify_cluster_snapshot	Modifies the settings for a snapshot
modify_cluster_snapshot_schedule	Modifies a snapshot schedule for a cluster
modify_cluster_subnet_group	Modifies a cluster subnet group to include the specified list of VPC subnets
modify_custom_domain_association	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_integration</code>	Modifies a zero-ETL integration or S3 event integration with Amazon
<code>modify_lakehouse_configuration</code>	Modifies the lakehouse configuration for a cluster
<code>modify_qev_2_idc_application</code>	Modifies an Amazon Redshift Query Editor (QEV2) IAM Identity Center
<code>modify_redshift_idc_application</code>	Changes an existing Amazon Redshift IAM Identity Center application
<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
<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>put_resource_policy</code>	Updates the resource policy for a specified resource
<code>reboot_cluster</code>	Reboots a cluster
<code>register_namespace</code>	Registers a cluster or serverless namespace to the Amazon Web Services
<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 c
<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 pr
<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
<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)
```

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 | <p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset. |
| credentials | <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 <- 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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_execute_statement	Runs one or more SQL statements, which can be data manipulation language (DML) or data definition language (DDL)
cancel_statement	Cancels a running query
describe_statement	Describes the details about a specific instance when a query was run by the Amazon Redshift Data API
describe_table	Describes 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 in JSON format
get_statement_result_v2	Fetches the temporarily cached result of an SQL statement in CSV format
list_databases	List the databases in a cluster
list_schemas	Lists the schemas in a database
list_sessions	Lists the sessions that the caller created in the last 24 hours
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)
```

redshiftserverless	<i>Redshift Serverless</i>
--------------------	----------------------------

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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```

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

```

Operations

convert_recovery_point_to_snapshot	Converts a recovery point to a snapshot
create_custom_domain_association	Creates a custom domain association for Amazon Redshift Serverless
create_endpoint_access	Creates an Amazon Redshift Serverless managed VPC endpoint
create_namespace	Creates a namespace in Amazon Redshift Serverless
create_reservation	Creates an Amazon Redshift Serverless reservation, which gives you the option to con
create_scheduled_action	Creates a scheduled action
create_snapshot	Creates a snapshot of all databases in a namespace
create_snapshot_copy_configuration	Creates a snapshot copy configuration that lets you copy snapshots to another Amazon
create_usage_limit	Creates a usage limit for a specified Amazon Redshift Serverless usage type
create_workgroup	Creates an workgroup in Amazon Redshift Serverless
delete_custom_domain_association	Deletes a custom domain association for Amazon Redshift Serverless
delete_endpoint_access	Deletes an Amazon Redshift Serverless managed VPC endpoint

<code>delete_namespace</code>	Deletes a namespace from Amazon Redshift Serverless
<code>delete_resource_policy</code>	Deletes the specified resource policy
<code>delete_scheduled_action</code>	Deletes a scheduled action
<code>delete_snapshot</code>	Deletes a snapshot from Amazon Redshift Serverless
<code>delete_snapshot_copy_configuration</code>	Deletes a snapshot copy configuration
<code>delete_usage_limit</code>	Deletes a usage limit from Amazon Redshift Serverless
<code>delete_workgroup</code>	Deletes a workgroup
<code>get_credentials</code>	Returns a database user name and temporary password with temporary authorization token
<code>get_custom_domain_association</code>	Gets information about a specific custom domain association
<code>get_endpoint_access</code>	Returns information, such as the name, about a VPC endpoint
<code>get_identity_center_auth_token</code>	Returns an Identity Center authentication token for accessing Amazon Redshift Serverless
<code>get_namespace</code>	Returns information about a namespace in Amazon Redshift Serverless
<code>get_recovery_point</code>	Returns information about a recovery point
<code>get_reservation</code>	Gets an Amazon Redshift Serverless reservation
<code>get_reservation_offering</code>	Returns the reservation offering
<code>get_resource_policy</code>	Returns a resource policy
<code>get_scheduled_action</code>	Returns information about a scheduled action
<code>get_snapshot</code>	Returns information about a specific snapshot
<code>get_table_restore_status</code>	Returns information about a TableRestoreStatus object
<code>get_track</code>	Get the Redshift Serverless version for a specified track
<code>get_usage_limit</code>	Returns information about a usage limit
<code>get_workgroup</code>	Returns information about a specific workgroup
<code>list_custom_domain_associations</code>	Lists custom domain associations for Amazon Redshift Serverless
<code>list_endpoint_access</code>	Returns an array of EndpointAccess objects and relevant information
<code>list_managed_workgroups</code>	Returns information about a list of specified managed workgroups in your account
<code>list_namespaces</code>	Returns information about a list of specified namespaces
<code>list_recovery_points</code>	Returns an array of recovery points
<code>list_reservation_offerings</code>	Returns the current reservation offerings in your account
<code>list_reservations</code>	Returns a list of Reservation objects
<code>list_scheduled_actions</code>	Returns a list of scheduled actions
<code>list_snapshot_copy_configurations</code>	Returns a list of snapshot copy configurations
<code>list_snapshots</code>	Returns a list of snapshots
<code>list_table_restore_status</code>	Returns information about an array of TableRestoreStatus objects
<code>list_tags_for_resource</code>	Lists the tags assigned to a resource
<code>list_tracks</code>	List the Amazon Redshift Serverless versions
<code>list_usage_limits</code>	Lists all usage limits within Amazon Redshift Serverless
<code>list_workgroups</code>	Returns information about a list of specified workgroups
<code>put_resource_policy</code>	Creates or updates a resource policy
<code>restore_from_recovery_point</code>	Restore the data from a recovery point
<code>restore_from_snapshot</code>	Restores a namespace from a snapshot
<code>restore_table_from_recovery_point</code>	Restores a table from a recovery point to your Amazon Redshift Serverless instance
<code>restore_table_from_snapshot</code>	Restores a table from a snapshot to your Amazon Redshift Serverless instance
<code>tag_resource</code>	Assigns one or more tags to a resource
<code>untag_resource</code>	Removes a tag or set of tags from a resource
<code>update_custom_domain_association</code>	Updates an Amazon Redshift Serverless certificate associated with a custom domain
<code>update_endpoint_access</code>	Updates an Amazon Redshift Serverless managed endpoint
<code>update_lakehouse_configuration</code>	Modifies the lakehouse configuration for a namespace
<code>update_namespace</code>	Updates a namespace with the specified settings

<code>update_scheduled_action</code>	Updates a scheduled action
<code>update_snapshot</code>	Updates a snapshot
<code>update_snapshot_copy_configuration</code>	Updates a snapshot copy configuration
<code>update_usage_limit</code>	Update a usage limit in Amazon Redshift Serverless
<code>update_workgroup</code>	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. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

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

```

Operations

batch_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 <- simplifiedb()
svc$batch_delete_attributes(
  Foo = 123
)

```

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

timestreamquery	<i>Amazon Timestream Query</i>
-----------------	--------------------------------

Description

Amazon Timestream Query

Usage

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

Arguments

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

- **credentials:**

- **creds:**

- * **access_key_id:** AWS access key ID
 - * **secret_access_key:** AWS secret access key
 - * **session_token:** AWS temporary session token

- **profile:** The name of a profile to use. If not given, then the default profile is used.

- **anonymous:** Set anonymous credentials.

- **endpoint:** The complete URL to use for the constructed client.

- **region:** The AWS Region used in instantiating the client.

- **close_connection:** Immediately close all HTTP connections.

- **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

- **s3_force_path_style:** Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **sts_regional_endpoint:** Set sts regional endpoint resolver to regional or legacy <https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

credentials Optional credentials shorthand for the config parameter

- **creds:**
 - **access_key_id:** AWS access key ID
 - **secret_access_key:** AWS secret access key
 - **session_token:** AWS temporary session token
- **profile:** The name of a profile to use. If not given, then the default profile is used.
- **anonymous:** Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

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

Value

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

Service syntax

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

)

Operations

cancel_query	Cancels 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_account_settings	Describes the settings for your account that include the query pricing model and the configured n
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_account_settings	Transitions your account to use TCUs for query pricing and modifies the maximum query compu
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 – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

create_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, a
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, [51](#)
add_terraform_tags_to_resource, [28](#)
add_partner, [51](#)
add_role_to_db_cluster, [35](#), [42](#)
add_role_to_db_instance, [42](#)
add_source_identifier_to_subscription,
 [7](#), [35](#), [42](#)
add_tags_to_resource, [7](#), [21](#), [35](#), [42](#)
apply_pending_maintenance_action, [7](#), [11](#),
 [35](#), [42](#)
associate_data_share_consumer, [51](#)
assume_decorated_role_with_saml, [28](#)
authorize_cache_security_group_ingress,
 [21](#)
authorize_cluster_security_group_ingress,
 [51](#)
authorize_data_share, [51](#)
authorize_db_security_group_ingress,
 [42](#)
authorize_endpoint_access, [51](#)
authorize_snapshot_access, [51](#)

backtrack_db_cluster, [42](#)
batch_apply_update_action, [21](#)
batch_delete_attributes, [63](#)
batch_delete_cluster_snapshots, [51](#)
batch_execute_statement, [14](#), [48](#), [57](#)
batch_get_item, [14](#)
batch_grant_permissions, [28](#)
batch_modify_cluster_snapshots, [51](#)
batch_put_attributes, [63](#)
batch_revoke_permissions, [28](#)
batch_stop_update_action, [21](#)
batch_update_cluster, [31](#)
batch_write_item, [14](#)
begin_transaction, [48](#)
build_auth_token, [43](#)
build_auth_token_v2, [43](#)

cancel_export_task, [43](#)

cancel_gremlin_query, [39](#)
cancel_loader_job, [39](#)
cancel_ml_data_processing_job, [39](#)
cancel_ml_model_training_job, [39](#)
cancel_ml_model_transform_job, [39](#)
cancel_open_cypher_query, [39](#)
cancel_query, [66](#)
cancel_resize, [51](#)
cancel_statement, [57](#)
cancel_transaction, [28](#)
commit_transaction, [28](#), [48](#)
complete_migration, [21](#)
convert_recovery_point_to_snapshot, [59](#)
copy_cluster_snapshot, [11](#), [51](#)
copy_db_cluster_parameter_group, [7](#), [35](#),
 [43](#)
copy_db_cluster_snapshot, [7](#), [35](#), [43](#)
copy_db_parameter_group, [35](#), [43](#)
copy_db_snapshot, [43](#)
copy_option_group, [43](#)
copy_serverless_cache_snapshot, [21](#)
copy_snapshot, [21](#), [31](#)
create_acl, [31](#)
create_authentication_profile, [51](#)
create_backup, [14](#)
create_batch_load_task, [68](#)
create_blue_green_deployment, [43](#)
create_cache_cluster, [21](#)
create_cache_parameter_group, [21](#)
create_cache_security_group, [21](#)
create_cache_subnet_group, [21](#)
create_cluster, [5](#), [11](#), [31](#), [51](#)
create_cluster_parameter_group, [51](#)
create_cluster_security_group, [51](#)
create_cluster_snapshot, [11](#), [51](#)
create_cluster_subnet_group, [51](#)
create_custom_db_engine_version, [43](#)
create_custom_domain_association, [51](#),
 [59](#)

- create_data_cells_filter, 28
- create_database, 68
- create_db_cluster, 7, 35, 43
- create_db_cluster_endpoint, 35, 43
- create_db_cluster_parameter_group, 7, 35, 43
- create_db_cluster_snapshot, 7, 35, 43
- create_db_instance, 7, 35, 43
- create_db_instance_read_replica, 43
- create_db_parameter_group, 35, 43
- create_db_proxy, 43
- create_db_proxy_endpoint, 43
- create_db_security_group, 43
- create_db_shard_group, 43
- create_db_snapshot, 43
- create_db_subnet_group, 7, 35, 43
- create_domain, 63
- create_endpoint_access, 51, 59
- create_event_subscription, 7, 35, 43, 51
- create_global_cluster, 7, 35, 43
- create_global_replication_group, 21
- create_global_table, 14
- create_hsm_client_certificate, 51
- create_hsm_configuration, 51
- create_integration, 43, 51
- create_keyspace, 25
- create_lake_formation_identity_center_configuration, 28
- create_lake_formation_opt_in, 28
- create_lf_tag, 28
- create_lf_tag_expression, 28
- create_ml_endpoint, 39
- create_multi_region_cluster, 31
- create_namespace, 59
- create_option_group, 43
- create_parameter_group, 5, 31
- create_qev_2_idc_application, 52
- create_redshift_idc_application, 52
- create_replication_group, 21
- create_reservation, 59
- create_scheduled_action, 52, 59
- create_scheduled_query, 66
- create_serverless_cache, 21
- create_serverless_cache_snapshot, 21
- create_snapshot, 21, 31, 59
- create_snapshot_copy_configuration, 59
- create_snapshot_copy_grant, 52
- create_snapshot_schedule, 52
- create_subnet_group, 5, 32
- create_table, 14, 25, 68
- create_tags, 52
- create_tenant_database, 43
- create_type, 25
- create_usage_limit, 52, 59
- create_user, 21, 32
- create_user_group, 21
- create_workgroup, 59
- dax, 3
- deauthorize_data_share, 52
- decrease_node_groups_in_global_replication_group, 21
- decrease_replica_count, 21
- decrease_replication_factor, 5
- delete_acl, 32
- delete_attributes, 63
- delete_authentication_profile, 52
- delete_backup, 14
- delete_blue_green_deployment, 43
- delete_cache_cluster, 21
- delete_cache_parameter_group, 21
- delete_cache_security_group, 21
- delete_cache_subnet_group, 21
- delete_cluster, 5, 11, 32, 52
- delete_cluster_parameter_group, 52
- delete_cluster_security_group, 52
- delete_cluster_snapshot, 11, 52
- delete_cluster_subnet_group, 52
- delete_custom_db_engine_version, 43
- delete_custom_domain_association, 52, 59
- delete_data_cells_filter, 28
- delete_database, 68
- delete_db_cluster, 8, 35, 43
- delete_db_cluster_automated_backup, 43
- delete_db_cluster_endpoint, 35, 43
- delete_db_cluster_parameter_group, 8, 35, 43
- delete_db_cluster_snapshot, 8, 35, 43
- delete_db_instance, 8, 35, 43
- delete_db_instance_automated_backup, 43
- delete_db_parameter_group, 35, 43
- delete_db_proxy, 43
- delete_db_proxy_endpoint, 43
- delete_db_security_group, 43
- delete_db_shard_group, 43

delete_db_snapshot, [43](#)
 delete_db_subnet_group, [8](#), [35](#), [43](#)
 delete_domain, [63](#)
 delete_endpoint_access, [52](#), [59](#)
 delete_event_subscription, [8](#), [35](#), [43](#), [52](#)
 delete_global_cluster, [8](#), [35](#), [43](#)
 delete_global_replication_group, [21](#)
 delete_hsm_client_certificate, [52](#)
 delete_hsm_configuration, [52](#)
 delete_integration, [43](#), [52](#)
 delete_item, [14](#)
 delete_keyspace, [25](#)
 delete_lake_formation_identity_center_configuration, [28](#)
 delete_lake_formation_opt_in, [28](#)
 delete_lf_tag, [28](#)
 delete_lf_tag_expression, [28](#)
 delete_ml_endpoint, [39](#)
 delete_multi_region_cluster, [32](#)
 delete_namespace, [60](#)
 delete_objects_on_cancel, [28](#)
 delete_option_group, [43](#)
 delete_parameter_group, [5](#), [32](#)
 delete_partner, [52](#)
 delete_propertygraph_statistics, [39](#)
 delete_qev_2_idc_application, [52](#)
 delete_redshift_idc_application, [52](#)
 delete_replication_group, [21](#)
 delete_resource_policy, [14](#), [52](#), [60](#)
 delete_scheduled_action, [52](#), [60](#)
 delete_scheduled_query, [66](#)
 delete_serverless_cache, [21](#)
 delete_serverless_cache_snapshot, [21](#)
 delete_snapshot, [21](#), [32](#), [60](#)
 delete_snapshot_copy_configuration, [60](#)
 delete_snapshot_copy_grant, [52](#)
 delete_snapshot_schedule, [52](#)
 delete_sparql_statistics, [39](#)
 delete_subnet_group, [5](#), [32](#)
 delete_table, [14](#), [25](#), [68](#)
 delete_tags, [52](#)
 delete_tenant_database, [44](#)
 delete_type, [25](#)
 delete_usage_limit, [52](#), [60](#)
 delete_user, [21](#), [32](#)
 delete_user_group, [21](#)
 delete_workgroup, [60](#)
 deregister_db_proxy_targets, [44](#)
 deregister_namespace, [52](#)
 deregister_resource, [28](#)
 describe_ac_ls, [32](#)
 describe_account_attributes, [44](#), [52](#)
 describe_account_settings, [66](#)
 describe_authentication_profiles, [52](#)
 describe_backup, [14](#)
 describe_batch_load_task, [68](#)
 describe_blue_green_deployments, [44](#)
 describe_cache_clusters, [21](#)
 describe_cache_engine_versions, [21](#)
 describe_cache_parameter_groups, [21](#)
 describe_cache_parameters, [21](#)
 describe_cache_security_groups, [21](#)
 describe_cache_subnet_groups, [21](#)
 describe_certificates, [8](#), [44](#)
 describe_cluster_db_revisions, [52](#)
 describe_cluster_parameter_groups, [52](#)
 describe_cluster_parameters, [52](#)
 describe_cluster_security_groups, [52](#)
 describe_cluster_snapshots, [52](#)
 describe_cluster_subnet_groups, [52](#)
 describe_cluster_tracks, [52](#)
 describe_cluster_versions, [52](#)
 describe_clusters, [5](#), [32](#), [52](#)
 describe_continuous_backups, [14](#)
 describe_contributor_insights, [14](#)
 describe_custom_domain_associations, [52](#)
 describe_data_shares, [52](#)
 describe_data_shares_for_consumer, [52](#)
 describe_data_shares_for_producer, [52](#)
 describe_database, [68](#)
 describe_db_cluster_automated_backups, [44](#)
 describe_db_cluster_backtracks, [44](#)
 describe_db_cluster_endpoints, [35](#), [44](#)
 describe_db_cluster_parameter_groups, [8](#), [35](#), [44](#)
 describe_db_cluster_parameters, [8](#), [35](#), [44](#)
 describe_db_cluster_snapshot_attributes, [8](#), [35](#), [44](#)
 describe_db_cluster_snapshots, [8](#), [35](#), [44](#)
 describe_db_clusters, [8](#), [35](#), [44](#)
 describe_db_engine_versions, [8](#), [35](#), [44](#)
 describe_db_instance_automated_backups, [44](#)

- describe_db_instances, [8](#), [35](#), [44](#)
- describe_db_log_files, [44](#)
- describe_db_major_engine_versions, [44](#)
- describe_db_parameter_groups, [36](#), [44](#)
- describe_db_parameters, [36](#), [44](#)
- describe_db_proxies, [44](#)
- describe_db_proxy_endpoints, [44](#)
- describe_db_proxy_target_groups, [44](#)
- describe_db_proxy_targets, [44](#)
- describe_db_recommendations, [44](#)
- describe_db_security_groups, [44](#)
- describe_db_shard_groups, [44](#)
- describe_db_snapshot_attributes, [44](#)
- describe_db_snapshot_tenant_databases, [44](#)
- describe_db_snapshots, [44](#)
- describe_db_subnet_groups, [8](#), [36](#), [44](#)
- describe_default_cluster_parameters, [52](#)
- describe_default_parameters, [5](#)
- describe_endpoint_access, [52](#)
- describe_endpoint_authorization, [52](#)
- describe_endpoints, [14](#), [66](#), [69](#)
- describe_engine_default_cluster_parameters, [8](#), [36](#), [44](#)
- describe_engine_default_parameters, [21](#), [36](#), [44](#)
- describe_engine_versions, [32](#)
- describe_event_categories, [8](#), [36](#), [44](#), [53](#)
- describe_event_subscriptions, [8](#), [36](#), [44](#), [53](#)
- describe_events, [5](#), [8](#), [21](#), [32](#), [36](#), [44](#), [53](#)
- describe_export, [14](#)
- describe_export_tasks, [44](#)
- describe_global_clusters, [8](#), [36](#), [44](#)
- describe_global_replication_groups, [21](#)
- describe_global_table, [14](#)
- describe_global_table_settings, [14](#)
- describe_hsm_client_certificates, [53](#)
- describe_hsm_configurations, [53](#)
- describe_import, [14](#)
- describe_inbound_integrations, [53](#)
- describe_integrations, [44](#), [53](#)
- describe_kinesis_streaming_destination, [14](#)
- describe_lake_formation_identity_center_configuration, [28](#)
- describe_limits, [14](#)
- describe_logging_status, [53](#)
- describe_multi_region_clusters, [32](#)
- describe_multi_region_parameter_groups, [32](#)
- describe_multi_region_parameters, [32](#)
- describe_node_configuration_options, [53](#)
- describe_option_group_options, [44](#)
- describe_option_groups, [44](#)
- describe_orderable_cluster_options, [53](#)
- describe_orderable_db_instance_options, [8](#), [36](#), [44](#)
- describe_parameter_groups, [5](#), [32](#)
- describe_parameters, [5](#), [32](#)
- describe_partners, [53](#)
- describe_pending_maintenance_actions, [8](#), [36](#), [44](#)
- describe_qev_2_idc_applications, [53](#)
- describe_redshift_idc_applications, [53](#)
- describe_replication_groups, [21](#)
- describe_reserved_cache_nodes, [21](#)
- describe_reserved_cache_nodes_offerings, [22](#)
- describe_reserved_db_instances, [44](#)
- describe_reserved_db_instances_offerings, [44](#)
- describe_reserved_node_exchange_status, [53](#)
- describe_reserved_node_offerings, [53](#)
- describe_reserved_nodes, [32](#), [53](#)
- describe_reserved_nodes_offerings, [32](#)
- describe_resize, [53](#)
- describe_resource, [28](#)
- describe_scheduled_actions, [53](#)
- describe_scheduled_query, [66](#)
- describe_serverless_cache_snapshots, [22](#)
- describe_serverless_caches, [22](#)
- describe_serverless_v2_platform_versions, [44](#)
- describe_service_updates, [22](#), [32](#)
- describe_snapshot_copy_grants, [53](#)
- describe_snapshot_schedules, [53](#)
- describe_snapshots, [22](#), [32](#)
- describe_source_regions, [44](#)
- describe_statement, [57](#)
- describe_storage, [53](#)
- describe_stream, [18](#)

- describe_subnet_groups, [5](#), [32](#)
- describe_table, [14](#), [57](#), [69](#)
- describe_table_replica_auto_scaling, [14](#)
- describe_table_restore_status, [53](#)
- describe_tags, [53](#)
- describe_tenant_databases, [44](#)
- describe_time_to_live, [14](#)
- describe_transaction, [28](#)
- describe_update_actions, [22](#)
- describe_usage_limits, [53](#)
- describe_user_groups, [22](#)
- describe_users, [22](#), [32](#)
- describe_valid_db_instance_modifications, [36](#), [45](#)
- disable_http_endpoint, [45](#)
- disable_kinesis_streaming_destination, [14](#)
- disable_logging, [53](#)
- disable_snapshot_copy, [53](#)
- disassociate_data_share_consumer, [53](#)
- disassociate_global_replication_group, [22](#)
- docdb, [5](#)
- docdbelastic, [9](#)
- domain_metadata, [63](#)
- download_db_log_file_portion, [45](#)
- dynamodb, [12](#)
- dynamodbstreams, [16](#)
- elasticache, [18](#)
- enable_http_endpoint, [45](#)
- enable_kinesis_streaming_destination, [14](#)
- enable_logging, [53](#)
- enable_snapshot_copy, [53](#)
- execute_fast_reset, [39](#)
- execute_gremlin_explain_query, [39](#)
- execute_gremlin_profile_query, [39](#)
- execute_gremlin_query, [39](#)
- execute_open_cypher_explain_query, [39](#)
- execute_open_cypher_query, [39](#)
- execute_scheduled_query, [66](#)
- execute_sql, [48](#)
- execute_statement, [14](#), [48](#), [57](#)
- execute_transaction, [14](#)
- export_serverless_cache_snapshot, [22](#)
- export_table_to_point_in_time, [14](#)
- extend_transaction, [28](#)
- failover_db_cluster, [8](#), [36](#), [45](#)
- failover_global_cluster, [8](#), [36](#), [45](#)
- failover_global_replication_group, [22](#)
- failover_primary_compute, [53](#)
- failover_shard, [32](#)
- get_attributes, [63](#)
- get_cluster, [11](#)
- get_cluster_credentials, [53](#)
- get_cluster_credentials_with_iam, [53](#)
- get_cluster_snapshot, [11](#)
- get_credentials, [60](#)
- get_custom_domain_association, [60](#)
- get_data_cells_filter, [28](#)
- get_data_lake_principal, [28](#)
- get_data_lake_settings, [28](#)
- get_effective_permissions_for_path, [28](#)
- get_endpoint_access, [60](#)
- get_engine_status, [39](#)
- get_gremlin_query_status, [39](#)
- get_identity_center_auth_token, [53](#), [60](#)
- get_item, [14](#)
- get_keyspace, [25](#)
- get_lf_tag, [28](#)
- get_lf_tag_expression, [28](#)
- get_loader_job_status, [39](#)
- get_ml_data_processing_job, [39](#)
- get_ml_endpoint, [39](#)
- get_ml_model_training_job, [39](#)
- get_ml_model_transform_job, [39](#)
- get_namespace, [60](#)
- get_open_cypher_query_status, [39](#)
- get_pending_maintenance_action, [11](#)
- get_propertygraph_statistics, [39](#)
- get_propertygraph_stream, [39](#)
- get_propertygraph_summary, [39](#)
- get_query_state, [28](#)
- get_query_statistics, [28](#)
- get_rdf_graph_summary, [39](#)
- get_records, [18](#)
- get_recovery_point, [60](#)
- get_reservation, [60](#)
- get_reservation_offering, [60](#)
- get_reserved_node_exchange_configuration_options, [53](#)
- get_reserved_node_exchange_offerings, [53](#)
- get_resource_lf_tags, [28](#)
- get_resource_policy, [14](#), [53](#), [60](#)

- get_scheduled_action, [60](#)
- get_shard_iterator, [18](#)
- get_snapshot, [60](#)
- get_sparql_statistics, [39](#)
- get_sparql_stream, [39](#)
- get_statement_result, [57](#)
- get_statement_result_v2, [57](#)
- get_table, [25](#)
- get_table_auto_scaling_settings, [25](#)
- get_table_objects, [28](#)
- get_table_restore_status, [60](#)
- get_temporary_data_location_credentials, [28](#)
- get_temporary_glue_partition_credentials, [29](#)
- get_temporary_glue_table_credentials, [29](#)
- get_track, [60](#)
- get_type, [25](#)
- get_usage_limit, [60](#)
- get_work_unit_results, [29](#)
- get_work_units, [29](#)
- get_workgroup, [60](#)
- grant_permissions, [29](#)
- import_table, [14](#)
- increase_node_groups_in_global_replication_group, [22](#)
- increase_replica_count, [22](#)
- increase_replication_factor, [5](#)
- keyspaces, [23](#)
- lakeformation, [26](#)
- list_allowed_multi_region_cluster_updates, [32](#)
- list_allowed_node_type_modifications, [22](#)
- list_allowed_node_type_updates, [32](#)
- list_backups, [14](#)
- list_batch_load_tasks, [69](#)
- list_cluster_snapshots, [11](#)
- list_clusters, [11](#)
- list_contributor_insights, [14](#)
- list_custom_domain_associations, [60](#)
- list_data_cells_filter, [29](#)
- list_databases, [57](#), [69](#)
- list_domains, [63](#)
- list_endpoint_access, [60](#)
- list_exports, [15](#)
- list_global_tables, [15](#)
- list_gremlin_queries, [39](#)
- list_imports, [15](#)
- list_keyspaces, [25](#)
- list_lake_formation_opt_ins, [29](#)
- list_lf_tag_expressions, [29](#)
- list_lf_tags, [29](#)
- list_loader_jobs, [39](#)
- list_managed_workgroups, [60](#)
- list_ml_data_processing_jobs, [39](#)
- list_ml_endpoints, [39](#)
- list_ml_model_training_jobs, [39](#)
- list_ml_model_transform_jobs, [39](#)
- list_namespaces, [60](#)
- list_open_cypher_queries, [39](#)
- list_pending_maintenance_actions, [11](#)
- list_permissions, [29](#)
- list_recommendations, [53](#)
- list_recovery_points, [60](#)
- list_reservation_offerings, [60](#)
- list_reservations, [60](#)
- list_resources, [29](#)
- list_scheduled_actions, [60](#)
- list_scheduled_queries, [66](#)
- list_schemas, [57](#)
- list_sessions, [57](#)
- list_snapshot_copy_configurations, [60](#)
- list_snapshots, [60](#)
- list_statements, [57](#)
- list_streams, [18](#)
- list_table_restore_status, [60](#)
- list_table_storage_optimizers, [29](#)
- list_tables, [15](#), [25](#), [57](#), [69](#)
- list_tags, [5](#), [32](#)
- list_tags_for_resource, [8](#), [11](#), [22](#), [25](#), [36](#), [45](#), [60](#), [66](#), [69](#)
- list_tags_of_resource, [15](#)
- list_tracks, [60](#)
- list_transactions, [29](#)
- list_types, [25](#)
- list_usage_limits, [60](#)
- list_workgroups, [60](#)
- manage_propertygraph_statistics, [39](#)
- manage_sparql_statistics, [40](#)
- memorydb, [29](#)
- modify_activity_stream, [45](#)
- modify_aqua_configuration, [53](#)

- modify_authentication_profile, [53](#)
- modify_cache_cluster, [22](#)
- modify_cache_parameter_group, [22](#)
- modify_cache_subnet_group, [22](#)
- modify_certificates, [45](#)
- modify_cluster, [53](#)
- modify_cluster_db_revision, [53](#)
- modify_cluster_iam_roles, [53](#)
- modify_cluster_maintenance, [53](#)
- modify_cluster_parameter_group, [53](#)
- modify_cluster_snapshot, [53](#)
- modify_cluster_snapshot_schedule, [53](#)
- modify_cluster_subnet_group, [53](#)
- modify_current_db_cluster_capacity, [45](#)
- modify_custom_db_engine_version, [45](#)
- modify_custom_domain_association, [53](#)
- modify_db_cluster, [8](#), [36](#), [45](#)
- modify_db_cluster_endpoint, [36](#), [45](#)
- modify_db_cluster_parameter_group, [8](#), [36](#), [45](#)
- modify_db_cluster_snapshot_attribute, [8](#), [36](#), [45](#)
- modify_db_instance, [8](#), [36](#), [45](#)
- modify_db_parameter_group, [36](#), [45](#)
- modify_db_proxy, [45](#)
- modify_db_proxy_endpoint, [45](#)
- modify_db_proxy_target_group, [45](#)
- modify_db_recommendation, [45](#)
- modify_db_shard_group, [45](#)
- modify_db_snapshot, [45](#)
- modify_db_snapshot_attribute, [45](#)
- modify_db_subnet_group, [8](#), [36](#), [45](#)
- modify_endpoint_access, [54](#)
- modify_event_subscription, [8](#), [36](#), [45](#), [54](#)
- modify_global_cluster, [8](#), [36](#), [45](#)
- modify_global_replication_group, [22](#)
- modify_integration, [45](#), [54](#)
- modify_lakehouse_configuration, [54](#)
- modify_option_group, [45](#)
- modify_qev_2_idc_application, [54](#)
- modify_redshift_idc_application, [54](#)
- modify_replication_group, [22](#)
- modify_replication_group_shard_configuration, [22](#)
- modify_scheduled_action, [54](#)
- modify_serverless_cache, [22](#)
- modify_snapshot_copy_retention_period, [54](#)
- modify_snapshot_schedule, [54](#)
- modify_tenant_database, [45](#)
- modify_usage_limit, [54](#)
- modify_user, [22](#)
- modify_user_group, [22](#)
- neptune, [33](#)
- neptunedata, [37](#)
- pause_cluster, [54](#)
- prepare_query, [66](#)
- promote_read_replica, [45](#)
- promote_read_replica_db_cluster, [36](#), [45](#)
- purchase_reserved_cache_nodes_offering, [22](#)
- purchase_reserved_db_instances_offering, [45](#)
- purchase_reserved_node_offering, [54](#)
- purchase_reserved_nodes_offering, [32](#)
- put_attributes, [63](#)
- put_data_lake_settings, [29](#)
- put_item, [15](#)
- put_resource_policy, [15](#), [54](#), [60](#)
- query, [15](#), [66](#)
- rds, [40](#)
- rdsdataservice, [46](#)
- rebalance_slots_in_global_replication_group, [22](#)
- reboot_cache_cluster, [22](#)
- reboot_cluster, [54](#)
- reboot_db_cluster, [45](#)
- reboot_db_instance, [8](#), [36](#), [45](#)
- reboot_db_shard_group, [45](#)
- reboot_node, [5](#)
- redshift, [49](#)
- redshiftdataapiservice, [54](#)
- redshiftserverless, [57](#)
- register_db_proxy_targets, [45](#)
- register_namespace, [54](#)
- register_resource, [29](#)
- reject_data_share, [54](#)
- remove_from_global_cluster, [8](#), [36](#), [45](#)
- remove_iam_tags_from_resource, [29](#)
- remove_role_from_db_cluster, [36](#), [45](#)
- remove_role_from_db_instance, [45](#)
- remove_source_identifier_from_subscription, [8](#), [36](#), [45](#)

- remove_tags_from_resource, [8](#), [22](#), [36](#), [45](#)
- reset_cache_parameter_group, [22](#)
- reset_cluster_parameter_group, [54](#)
- reset_db_cluster_parameter_group, [8](#), [36](#), [45](#)
- reset_db_parameter_group, [36](#), [45](#)
- reset_parameter_group, [32](#)
- resize_cluster, [54](#)
- restore_cluster_from_snapshot, [11](#)
- restore_db_cluster_from_s3, [45](#)
- restore_db_cluster_from_snapshot, [8](#), [36](#), [45](#)
- restore_db_cluster_to_point_in_time, [8](#), [36](#), [45](#)
- restore_db_instance_from_db_snapshot, [45](#)
- restore_db_instance_from_s3, [46](#)
- restore_db_instance_to_point_in_time, [46](#)
- restore_from_cluster_snapshot, [54](#)
- restore_from_recovery_point, [60](#)
- restore_from_snapshot, [60](#)
- restore_table, [25](#)
- restore_table_from_backup, [15](#)
- restore_table_from_cluster_snapshot, [54](#)
- restore_table_from_recovery_point, [60](#)
- restore_table_from_snapshot, [60](#)
- restore_table_to_point_in_time, [15](#)
- resume_batch_load_task, [69](#)
- resume_cluster, [54](#)
- revoke_cache_security_group_ingress, [22](#)
- revoke_cluster_security_group_ingress, [54](#)
- revoke_db_security_group_ingress, [46](#)
- revoke_endpoint_access, [54](#)
- revoke_permissions, [29](#)
- revoke_snapshot_access, [54](#)
- rollback_transaction, [48](#)
- rotate_encryption_key, [54](#)
- scan, [15](#)
- search_databases_by_lf_tags, [29](#)
- search_tables_by_lf_tags, [29](#)
- search_vectors, [15](#)
- select, [63](#)
- simplifiedb, [61](#)
- start_activity_stream, [46](#)
- start_cluster, [11](#)
- start_db_cluster, [8](#), [36](#), [46](#)
- start_db_instance, [46](#)
- start_db_instance_automated_backups_replication, [46](#)
- start_export_task, [46](#)
- start_loader_job, [40](#)
- start_migration, [22](#)
- start_ml_data_processing_job, [40](#)
- start_ml_model_training_job, [40](#)
- start_ml_model_transform_job, [40](#)
- start_query_planning, [29](#)
- start_transaction, [29](#)
- stop_activity_stream, [46](#)
- stop_cluster, [11](#)
- stop_db_cluster, [8](#), [36](#), [46](#)
- stop_db_instance, [46](#)
- stop_db_instance_automated_backups_replication, [46](#)
- switchover_blue_green_deployment, [46](#)
- switchover_global_cluster, [8](#), [36](#), [46](#)
- switchover_read_replica, [46](#)
- tag_resource, [5](#), [11](#), [15](#), [25](#), [32](#), [60](#), [66](#), [69](#)
- test_failover, [22](#)
- test_migration, [22](#)
- timestreamquery, [64](#)
- timestreamwrite, [66](#)
- transact_get_items, [15](#)
- transact_write_items, [15](#)
- untag_resource, [5](#), [11](#), [15](#), [25](#), [32](#), [60](#), [66](#), [69](#)
- update_account_settings, [66](#)
- update_acl, [32](#)
- update_cluster, [5](#), [11](#), [32](#)
- update_continuous_backups, [15](#)
- update_contributor_insights, [15](#)
- update_custom_domain_association, [60](#)
- update_data_cells_filter, [29](#)
- update_database, [69](#)
- update_endpoint_access, [60](#)
- update_global_table, [15](#)
- update_global_table_settings, [15](#)
- update_item, [15](#)
- update_keyspace, [25](#)
- update_kinesis_streaming_destination, [15](#)
- update_lake_formation_identity_center_configuration, [29](#)

- update_lakehouse_configuration, [60](#)
- update_lf_tag, [29](#)
- update_lf_tag_expression, [29](#)
- update_multi_region_cluster, [32](#)
- update_namespace, [60](#)
- update_parameter_group, [5](#), [32](#)
- update_partner_status, [54](#)
- update_resource, [29](#)
- update_scheduled_action, [61](#)
- update_scheduled_query, [66](#)
- update_snapshot, [61](#)
- update_snapshot_copy_configuration, [61](#)
- update_subnet_group, [5](#), [32](#)
- update_table, [15](#), [25](#), [69](#)
- update_table_objects, [29](#)
- update_table_replica_auto_scaling, [15](#)
- update_table_storage_optimizer, [29](#)
- update_time_to_live, [15](#)
- update_usage_limit, [61](#)
- update_user, [32](#)
- update_workgroup, [61](#)
- write_records, [69](#)