Package 'paws.database'

August 23, 2021

Title 'Amazon Web Services' Database Services

Version 0.1.12

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

License Apache License (>= 2.0)

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

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

Imports paws.common (>= 0.3.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.1.1

Collate 'dax_service.R' 'dax_interfaces.R' 'dax_operations.R' 'docdb_service.R' 'docdb_interfaces.R' 'docdb_operations.R' 'dynamodb_service.R' 'dynamodb_interfaces.R' 'dynamodb_operations.R' 'dynamodbstreams_service.R' 'dynamodbstreams_interfaces.R' 'dynamodbstreams_operations.R' 'elasticache_service.R' 'elasticache_interfaces.R' 'elasticache_operations.R' 'neptune_service.R' 'neptune_interfaces.R' 'neptune_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' 'simpledb_service.R' 'simpledb_interfaces.R' 'simpledb_operations.R'

NeedsCompilation no

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

Maintainer David Kretch <david.kretch@gmail.com>

Repository CRAN

Date/Publication 2021-08-23 07:10:46 UTC

R topics documented:

dax *Amazon DynamoDB Accelerator (DAX)*

Description

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

Usage

```
dax(config = list())
```

Arguments

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

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

Operations

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

Examples

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

## End(Not run)
```

docdb *Amazon DocumentDB with MongoDB compatibility*

Description

Amazon DocumentDB API documentation

Usage

```
docdb(config = list())
```

Arguments

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

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

Operations

add_tags_to_resource	Adds metadata tags to an Amazon DocumentDB resource
apply_pending_maintenance_action	Applies a pending maintenance action to a resource (for example, to an Amazo
copy_db_cluster_parameter_group	Copies the specified cluster parameter group
copy_db_cluster_snapshot	Copies a snapshot of a cluster
create_db_cluster	Creates a new Amazon DocumentDB cluster
create_db_cluster_parameter_group	Creates a new cluster parameter group
create_db_cluster_snapshot	Creates a snapshot of a cluster
create_db_instance	Creates a new instance
create_db_subnet_group	Creates a new subnet group
delete_db_cluster	Deletes a previously provisioned cluster
delete_db_cluster_parameter_group	Deletes a specified cluster parameter group
delete_db_cluster_snapshot	Deletes a cluster snapshot
delete_db_instance	Deletes a previously provisioned instance
delete_db_subnet_group	Deletes a subnet group
describe_certificates	Returns a list of certificate authority (CA) certificates provided by Amazon Do
describe_db_cluster_parameter_groups	Returns a list of DBClusterParameterGroup descriptions
describe_db_cluster_parameters	Returns the detailed parameter list for a particular cluster parameter group
describe_db_clusters	Returns information about provisioned Amazon DocumentDB clusters
describe_db_cluster_snapshot_attributes	Returns a list of cluster snapshot attribute names and values for a manual DB c
describe_db_cluster_snapshots	Returns information about cluster snapshots
describe_db_engine_versions	Returns a list of the available engines
describe_db_instances	Returns information about provisioned Amazon DocumentDB instances
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 da
describe_event_categories	Displays a list of categories for all event source types, or, if specified, for a spe
describe_events	Returns events related to instances, security groups, snapshots, and DB parame
describe_orderable_db_instance_options	Returns a list of orderable instance options for the specified engine
describe_pending_maintenance_actions	Returns a list of resources (for example, instances) that have at least one pendi
failover_db_cluster	Forces a failover for a cluster
list_tags_for_resource	Lists all tags on an Amazon DocumentDB resource
modify_db_cluster	Modifies a setting for an Amazon DocumentDB cluster
modify_db_cluster_parameter_group	Modifies the parameters of a cluster parameter group
modify_db_cluster_snapshot_attribute	Adds an attribute and values to, or removes an attribute and values from, a man
modify_db_instance	Modifies settings for an instance
modify_db_subnet_group	Modifies an existing subnet group
reboot_db_instance	You might need to reboot your instance, usually for maintenance reasons
remove_tags_from_resource	Removes metadata tags from an Amazon DocumentDB resource
reset_db_cluster_parameter_group	Modifies the parameters of a cluster parameter group to the default value
restore_db_cluster_from_snapshot	Creates a new cluster from a snapshot or cluster snapshot
restore_db_cluster_to_point_in_time	Restores a cluster to an arbitrary point in time
start_db_cluster	Restarts the stopped cluster that is specified by DBClusterIdentifier
stop_db_cluster	Stops the running cluster that is specified by DBClusterIdentifier

Examples

```
## Not run:
svc <- docdb()
svc$add_tags_to_resource(
  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 AWS 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 AWS region, providing built-in high availability and data durability.

Usage

```
dynamodb(config = list())
```

Arguments

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

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

Operations

batch_execute_statement	This operation allows you to perform batch reads and writes on data stored in Dyna
batch_get_item	The BatchGetItem operation returns the attributes of one or more items from one or
batch_write_item	The BatchWriteItem operation puts or deletes multiple items in one or more tables
create_backup	Creates a backup for an existing table
create_global_table	Creates a global table from an existing table
create_table	The CreateTable operation adds a new table to your account
delete_backup	Deletes an existing backup of a table
delete_item	Deletes a single item in a table by primary key
delete_table	The DeleteTable operation deletes a table and all of its items
describe_backup	Describes an existing backup of a table
describe_continuous_backups	Checks the status of continuous backups and point in time recovery on the specified
describe_contributor_insights	Returns information about contributor insights, for a given table or global secondary
describe_endpoints	Returns the regional endpoint information
describe_export	Describes an existing table export
describe_global_table	Returns information about the specified global table
describe_global_table_settings	Describes Region-specific settings for a global table
describe_kinesis_streaming_destination	Returns information about the status of Kinesis streaming
describe_limits	Returns the current provisioned-capacity quotas for your AWS account in a Region,
describe_table	Returns information about the table, including the current status of the table, when i
describe_table_replica_auto_scaling	Describes auto scaling settings across replicas of the global table at once
describe_time_to_live	Gives a description of the Time to Live (TTL) status on the specified table
disable_kinesis_streaming_destination	Stops replication from the DynamoDB table to the Kinesis data stream
enable_kinesis_streaming_destination	Starts table data replication to the specified Kinesis data stream at a timestamp chos
execute_statement	This operation allows you to perform reads and singleton writes on data stored in D
execute_transaction	This operation allows you to perform transactional reads or writes on data stored in
export_table_to_point_in_time	Exports table data to an S3 bucket
get_item	The GetItem operation returns a set of attributes for the item with the given primary
list_backups	List backups associated with an AWS account
list_contributor_insights	Returns a list of ContributorInsightsSummary for a table and all its global secondar
list_exports	Lists completed exports within the past 90 days

list_global_tables	Lists all global tables that have a replica in the specified Region
list_tables	Returns an array of table names associated with the current account and endpoint
list_tags_of_resource	List all tags on an Amazon DynamoDB resource
put_item	Creates a new item, or replaces an old item with a new item
query	The Query operation finds items based on primary key values
restore_table_from_backup	Creates a new table from an existing backup
restore_table_to_point_in_time	Restores the specified table to the specified point in time within EarliestRestorableD
scan	The Scan operation returns one or more items and item attributes by accessing every
tag_resource	Associate a set of tags with an Amazon DynamoDB resource
transact_get_items	TransactGetItems is a synchronous operation that atomically retrieves multiple item
transact_write_items	TransactWriteItems is a synchronous write operation that groups up to 25 action rec
untag_resource	Removes the association of tags from an Amazon DynamoDB resource
update_continuous_backups	UpdateContinuousBackups enables or disables point in time recovery for the specif
update_contributor_insights	Updates the status for contributor insights for a specific table or index
update_global_table	Adds or removes replicas in the specified global table
update_global_table_settings	Updates settings for a global table
update_item	Edits an existing item's attributes, or adds a new item to the table if it does not alrea
update_table	Modifies the provisioned throughput settings, global secondary indexes, or Dynamo
update_table_replica_auto_scaling	Updates auto scaling settings on your global tables at once
update_time_to_live	The UpdateTimeToLive method enables or disables Time to Live (TTL) for the 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())
```

Arguments

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

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

Operations

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

Examples

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

## End(Not run)
```

elasticache *Amazon ElastiCache*

Description

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

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

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

Usage

```
elasticache(config = list())
```

Arguments

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

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

Operations

add_tags_to_resource	Adds up to 50 cost allocation tags to the named resource
authorize_cache_security_group_ingress	Allows network ingress to a cache security group
batch_apply_update_action	Apply the service update
batch_stop_update_action	Stop the service update
complete_migration	Complete the migration of data
copy_snapshot	Makes a copy of an existing snapshot
create_cache_cluster	Creates a cluster
create_cache_parameter_group	Creates a new Amazon ElastiCache cache parameter group
create_cache_security_group	Creates a new cache security group
create_cache_subnet_group	Creates a new cache subnet group
create_global_replication_group	Global Datastore for Redis offers fully managed, fast, reliable and secu
create_replication_group	Creates a Redis (cluster mode disabled) or a Redis (cluster mode enabl
create_snapshot	Creates a copy of an entire cluster or replication group at a specific mo
create_user	For Redis engine version 6
create_user_group	For Redis engine version 6
decrease_node_groups_in_global_replication_group	Decreases the number of node groups in a Global Datastore
decrease_replica_count	Dynamically decreases the number of replicas in a Redis (cluster mode
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_snapshot	Deletes an existing snapshot
delete_user	For Redis engine version 6
delete_user_group	For Redis engine version 6
describe_cache_clusters	Returns information about all provisioned clusters if no cluster identifi
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 gro
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 sp
describe_events	Returns events related to clusters, cache security groups, and cache par
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 ab
describe_reserved_cache_nodes_offerings	Lists available reserved cache node offerings
describe_service_updates	Returns details of the service updates
describe_snapshots	Returns information about cluster or replication group snapshots
describe_update_actions	Returns details of the update actions
describe_user_groups	Returns a list of user groups
describe_users	Returns a list of users
disassociate_global_replication_group	Remove a secondary cluster from the Global Datastore using the Globa
failover_global_replication_group	Used to failover the primary region to a selected secondary region
increase_node_groups_in_global_replication_group	Increase the number of node groups in the Global Datastore
increase_replica_count	Dynamically increases the number of replics in a Redis (cluster mode d
list_allowed_node_type_modifications	Lists all available node types that you can scale your Redis cluster's or
list_tags_for_resource	Lists all cost allocation tags currently on the named resource
modify_cache_cluster	Modifies the settings for a cluster
modify_cache_parameter_group	Modifies the parameters of a cache parameter group
modify_cache_subnet_group	Modifies an existing cache subnet group
modify_global_replication_group	Modifies the settings for a Global Datastore
modify_replication_group	Modifies the settings for a replication group
modify_replication_group_shard_configuration	Modifies a replication group's shards (node groups) by allowing you to
modify_user	Changes user password(s) and/or access string
modify_user_group	Changes the list of users that belong to the user group
purchase_reserved_cache_nodes_offering	Allows you to purchase a reserved cache node offering
rebalance_slots_in_global_replication_group	Redistribute slots to ensure uniform distribution across existing shards
reboot_cache_cluster	Reboots some, or all, of the cache nodes within a provisioned cluster
remove_tags_from_resource	Removes the tags identified by the TagKeys list from the named resour
reset_cache_parameter_group	Modifies the parameters of a cache parameter group to the engine or sy
revoke_cache_security_group_ingress	Revokes ingress from a cache security group
start_migration	Start the migration of data
test_failover	Represents the input of a TestFailover operation which test automatic f

Examples

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

Arguments

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

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

Operations

add_role_to_db_cluster	Associates an Identity and Access Management (IAM) role from an Neptune I
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 inst
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
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
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 cluste
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
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 I
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
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_subnet_groups	Returns a list of DBSubnetGroup descriptions
describe_engine_default_cluster_parameters	Returns the default engine and system parameter information for the cluster da
describe_engine_default_parameters	Returns the default engine and system parameter information for the specified
describe_event_categories	Displays a list of categories for all event source types, or, if specified, for a spe
describe_events	Returns events related to DB instances, DB security groups, DB snapshots, and
describe_event_subscriptions	Lists all the subscription descriptions for a customer account
describe_orderable_db_instance_options	Returns a list of orderable DB instance options for the specified engine
describe_pending_maintenance_actions	Returns a list of resources (for example, DB instances) that have at least one p
describe_valid_db_instance_modifications	You can call DescribeValidDBInstanceModifications to learn what modificatio
failover_db_cluster	Forces a failover for a DB cluster
list_tags_for_resource	Lists all tags on an Amazon Neptune resource
modify_db_cluster	Modify a setting for a DB cluster
modify_db_cluster_endpoint	Modifies the properties of an endpoint in an Amazon Neptune DB cluster
modify_db_cluster_parameter_group	Modifies the parameters of a DB cluster parameter group
modify_db_cluster_snapshot_attribute	Adds an attribute and values to, or removes an attribute and values from, a man
modify_db_instance	Modifies settings for a DB instance
modify_db_parameter_group	Modifies the parameters of a DB parameter group
modify_db_subnet_group	Modifies an existing DB subnet group
modify_event_subscription	Modifies an existing event notification subscription
promote_read_replica_db_cluster	Not supported
reboot_db_instance	You might need to reboot your DB instance, usually for maintenance reasons
remove_role_from_db_cluster	Disassociates an Identity and Access Management (IAM) role from a DB clus
remove_source_identifier_from_subscription	Removes a source identifier from an existing event notification subscription
remove_tags_from_resource	Removes metadata tags from an Amazon Neptune resource
reset_db_cluster_parameter_group	Modifies the parameters of a DB cluster parameter group to the default value
reset_db_parameter_group	Modifies the parameters of a DB parameter group to the engine/system default
restore_db_cluster_from_snapshot	Creates a new DB cluster from a DB snapshot or DB cluster snapshot
restore_db_cluster_to_point_in_time	Restores a DB cluster to an arbitrary point in time
start_db_cluster	Starts an Amazon Neptune DB cluster that was stopped using the AWS consol
stop_db_cluster	Stops an Amazon Neptune DB cluster

Examples

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

## End(Not run)
```

rds *Amazon Relational Database Service*

Description

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

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

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

Amazon RDS API Reference

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

Amazon RDS User Guide

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

Usage

```
rds(config = list())
```

Arguments

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

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

Operations

add_role_to_db_cluster	Associates an Identity and Access Management (IAM) role from an Ama
add_role_to_db_instance	Associates an AWS Identity and Access Management (IAM) role with a
add_source_identifier_to_subscription	Adds a source identifier to an existing RDS event notification subscriptio
add_tags_to_resource	Adds metadata tags to an Amazon RDS resource
apply_pending_maintenance_action	Applies a pending maintenance action to a resource (for example, to a D
authorize_db_security_group_ingress	Enables ingress to a DBSecurityGroup using one of two forms of authori
backtrack_db_cluster	Backtracks a DB cluster to a specific time, without creating a new DB clu
build_auth_token	Return an authentication token for a database connection
cancel_export_task	Cancels an export task in progress that is exporting a snapshot to Amazo
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
copy_db_snapshot	Copies the specified DB snapshot
copy_option_group	Copies the specified option group
create_custom_availability_zone	Creates a custom Availability Zone (AZ)
create_db_cluster	Creates a new Amazon Aurora DB cluster
create_db_cluster_endpoint	Creates a new custom endpoint and associates it with an Amazon Aurora
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_instance_read_replica	Creates a new DB instance that acts as a read replica for an existing sourc
create_db_parameter_group	Creates a new DB parameter group
create_db_proxy	Creates a new DB proxy
create_db_security_group	Creates a new DB security group
create_db_snapshot	Creates a snapshot of a DB instance
create_db_subnet_group	Creates a new DB subnet group
create_event_subscription	Creates an RDS event notification subscription
create_global_cluster	Creates an Aurora global database spread across multiple AWS Regions
create_option_group	Creates a new option group
delete_custom_availability_zone	Deletes a custom Availability Zone (AZ)

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 Aurora DB c
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 instan
delete_db_instance_automated_backup	Deletes automated backups using the DbiResourceId value of the source
delete_db_parameter_group	Deletes a specified DB parameter group
delete_db_proxy	Deletes an existing proxy
delete_db_security_group	Deletes a DB security group
delete_db_snapshot	Deletes a DB snapshot
delete_db_subnet_group	Deletes a DB subnet group
delete_event_subscription	Deletes an RDS event notification subscription
delete_global_cluster	Deletes a global database cluster
delete_installation_media	Deletes the installation medium for a DB engine that requires an on-pren
delete_option_group	Deletes an existing option group
deregister_db_proxy_targets	Remove the association between one or more DBProxyTarget data struct
describe_account_attributes	Lists all of the attributes for a customer account
describe_certificates	Lists the set of CA certificates provided by Amazon RDS for this AWS a
describe_custom_availability_zones	Returns information about custom Availability Zones (AZs)
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 g
describe_db_clusters	Returns information about provisioned Aurora DB clusters
describe_db_cluster_snapshot_attributes	Returns a list of DB cluster snapshot attribute names and values for a ma
describe_db_cluster_snapshots	Returns information about DB cluster snapshots
describe_db_engine_versions	Returns a list of the available DB engines
describe_db_instance_automated_backups	Displays backups for both current and deleted instances
describe_db_instances	Returns information about provisioned RDS instances
describe_db_log_files	Returns a list of DB log files for the DB instance
describe_db_parameter_groups	Returns a list of DBParameterGroup descriptions
describe_db_parameters	Returns the detailed parameter list for a particular DB parameter group
describe_db_proxies	Returns information about DB proxies
describe_db_proxy_target_groups	Returns information about DB proxy target groups, represented by DBPr
describe_db_proxy_targets	Returns information about DBProxyTarget objects
describe_db_security_groups	Returns a list of DBSecurityGroup descriptions
describe_db_snapshot_attributes	Returns a list of DB snapshot attribute names and values for a manual DI
describe_db_snapshots	Returns information about DB snapshots
describe_db_subnet_groups	Returns a list of DBSubnetGroup descriptions
describe_engine_default_cluster_parameters	Returns the default engine and system parameter information for the clus
describe_engine_default_parameters	Returns the default engine and system parameter information for the spec
describe_event_categories	Displays a list of categories for all event source types, or, if specified, for
describe_events	Returns events related to DB instances, DB clusters, DB parameter group
describe_event_subscriptions	Lists all the subscription descriptions for a customer account
describe_export_tasks	Returns information about a snapshot export to Amazon S3
describe_global_clusters	Returns information about Aurora global database clusters
describe_installation_media	Describes the available installation media for a DB engine that requires a
describe_option_group_options	Describes all available options

describe_option_groups	Describes the available option groups
describe_orderable_db_instance_options	Returns a list of orderable DB instance options for the specified engine
describe_pending_maintenance_actions	Returns a list of resources (for example, DB instances) that have at least
describe_reserved_db_instances	Returns information about reserved DB instances for this account, or abo
describe_reserved_db_instances_offerings	Lists available reserved DB instance offerings
describe_source_regions	Returns a list of the source AWS Regions where the current AWS Regio
describe_valid_db_instance_modifications	You can call DescribeValidDBInstanceModifications to learn what modif
download_db_log_file_portion	Downloads all or a portion of the specified log file, up to 1 MB in size
failover_db_cluster	Forces a failover for a DB cluster
import_installation_media	Imports the installation media for a DB engine that requires an on-premis
list_tags_for_resource	Lists all tags on an Amazon RDS resource
modify_certificates	Override the system-default Secure Sockets Layer/Transport Layer Secu
modify_current_db_cluster_capacity	Set the capacity of an Aurora Serverless DB cluster to a specific value
modify_db_cluster	Modify a setting for an Amazon Aurora DB cluster
modify_db_cluster_endpoint	Modifies the properties of an endpoint in an Amazon Aurora DB cluster
modify_db_cluster_parameter_group	Modifies the parameters of a DB cluster parameter group
modify_db_cluster_snapshot_attribute	Adds an attribute and values to, or removes an attribute and values from,
modify_db_instance	Modifies settings for a DB instance
modify_db_parameter_group	Modifies the parameters of a DB parameter group
modify_db_proxy	Changes the settings for an existing DB proxy
modify_db_proxy_target_group	Modifies the properties of a DBProxyTargetGroup
modify_db_snapshot	Updates a manual DB snapshot with a new engine version
modify_db_snapshot_attribute	Adds an attribute and values to, or removes an attribute and values from,
modify_db_subnet_group	Modifies an existing DB subnet group
modify_event_subscription	Modifies an existing RDS event notification subscription
modify_global_cluster	Modify a setting for an Amazon Aurora global cluster
modify_option_group	Modifies an existing option group
promote_read_replica	Promotes a read replica DB instance to a standalone DB instance
promote_read_replica_db_cluster	Promotes a read replica DB cluster to a standalone DB cluster
purchase_reserved_db_instances_offering	Purchases a reserved DB instance offering
reboot_db_instance	You might need to reboot your DB instance, usually for maintenance rea
register_db_proxy_targets	Associate one or more DBProxyTarget data structures with a DBProxyTa
remove_from_global_cluster	Detaches an Aurora secondary cluster from an Aurora global database cl
remove_role_from_db_cluster	Disassociates an AWS Identity and Access Management (IAM) role fron
remove_role_from_db_instance	Disassociates an AWS Identity and Access Management (IAM) role fron
remove_source_identifier_from_subscription	Removes a source identifier from an existing RDS event notification subs
remove_tags_from_resource	Removes metadata tags from an Amazon RDS resource
reset_db_cluster_parameter_group	Modifies the parameters of a DB cluster parameter group to the default v
reset_db_parameter_group	Modifies the parameters of a DB parameter group to the engine/system d
restore_db_cluster_from_s3	Creates an Amazon Aurora DB cluster from MySQL data stored in an A
restore_db_cluster_from_snapshot	Creates a new DB cluster from a DB snapshot or DB cluster snapshot
restore_db_cluster_to_point_in_time	Restores a DB cluster to an arbitrary point in time
restore_db_instance_from_db_snapshot	Creates a new DB instance from a DB snapshot
restore_db_instance_from_s3	Amazon Relational Database Service (Amazon RDS) supports importing
restore_db_instance_to_point_in_time	Restores a DB instance to an arbitrary point in time
revoke_db_security_group_ingress	Revokes ingress from a DBSecurityGroup for previously authorized IP ra
start_activity_stream	Starts a database activity stream to monitor activity on the database
start_db_cluster	Starts an Amazon Aurora DB cluster that was stopped using the AWS co

start_db_instance	Starts an Amazon RDS DB instance that was stopped using the AWS cor
start_db_instance_automated_backups_replication	Enables replication of automated backups to a different AWS Region
start_export_task	Starts an export of a snapshot to Amazon S3
stop_activity_stream	Stops a database activity stream that was started using the AWS console,
stop_db_cluster	Stops an Amazon Aurora DB cluster
stop_db_instance	Stops an Amazon RDS DB instance
stop_db_instance_automated_backups_replication	Stops automated backup replication for a DB instance

Examples

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

## End(Not run)
```

rdsdataservice *AWS RDS DataService*

Description

Amazon RDS Data Service

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

For more information about the Data Service API, see Using the Data API for Aurora Serverless in the *Amazon Aurora User Guide*.

If you have questions or comments related to the Data API, send email to Rds-data-api-feedback@amazon.com.

Usage

```
rdsdataservice(config = list())
```

Arguments

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

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"
    ),
    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())
```

Arguments

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

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

Operations

accept_reserved_node_exchange	Exchanges a DC1 Reserved Node for a DC2 Reserved Node with no changes to t

authorize_cluster_security_group_ingress	Adds an inbound (ingress) rule to an Amazon Redshift security group
authorize_snapshot_access	Authorizes the specified AWS customer account to restore the specified snapshot
batch_delete_cluster_snapshots	Deletes a set of cluster snapshots
batch_modify_cluster_snapshots	Modifies the settings for a set of cluster snapshots
cancel_resize	Cancels a resize operation for a cluster
copy_cluster_snapshot	Copies the specified automated cluster snapshot to a new manual cluster snapshot
create_cluster	Creates a new cluster with the specified parameters
create_cluster_parameter_group	Creates an Amazon Redshift parameter group
create_cluster_security_group	Creates a new Amazon Redshift security group
create_cluster_snapshot	Creates a manual snapshot of the specified cluster
create_cluster_subnet_group	Creates a new Amazon Redshift subnet group
create_event_subscription	Creates an Amazon Redshift event notification subscription
create_hsm_client_certificate	Creates an HSM client certificate that an Amazon Redshift cluster will use to con
create_hsm_configuration	Creates an HSM configuration that contains the information required by an Amaz
create_scheduled_action	Creates a scheduled action
create_snapshot_copy_grant	Creates a snapshot copy grant that permits Amazon Redshift to use a customer ma
create_snapshot_schedule	Create a snapshot schedule that can be associated to a cluster and which overrides
create_tags	Adds tags to a cluster
create_usage_limit	Creates a usage limit for a specified Amazon Redshift feature on a cluster
delete_cluster	Deletes a previously provisioned cluster without its final snapshot being created
delete_cluster_parameter_group	Deletes a specified Amazon Redshift parameter group
delete_cluster_security_group	Deletes an Amazon Redshift security group
delete_cluster_snapshot	Deletes the specified manual snapshot
delete_cluster_subnet_group	Deletes the specified cluster subnet group
delete_event_subscription	Deletes an Amazon Redshift event notification subscription
delete_hsm_client_certificate	Deletes the specified HSM client certificate
delete_hsm_configuration	Deletes the specified Amazon Redshift HSM configuration
delete_scheduled_action	Deletes a scheduled action
delete_snapshot_copy_grant	Deletes the specified snapshot copy grant
delete_snapshot_schedule	Deletes a snapshot schedule
delete_tags	Deletes tags from a resource
delete_usage_limit	Deletes a usage limit from a cluster
describe_account_attributes	Returns a list of attributes attached to an account
describe_cluster_db_revisions	Returns an array of ClusterDbRevision objects
describe_cluster_parameter_groups	Returns a list of Amazon Redshift parameter groups, including parameter groups
describe_cluster_parameters	Returns a detailed list of parameters contained within the specified Amazon Reds
describe_clusters	Returns properties of provisioned clusters including general cluster properties, clu
describe_cluster_security_groups	Returns information about Amazon Redshift security groups
describe_cluster_snapshots	Returns one or more snapshot objects, which contain metadata about your cluster
describe_cluster_subnet_groups	Returns one or more cluster subnet group objects, which contain metadata about y
describe_cluster_tracks	Returns a list of all the available maintenance tracks
describe_cluster_versions	Returns descriptions of the available Amazon Redshift cluster versions
describe_default_cluster_parameters	Returns a list of parameter settings for the specified parameter group family
describe_event_categories	Displays a list of event categories for all event source types, or for a specified sou
describe_events	Returns events related to clusters, security groups, snapshots, and parameter grou
describe_event_subscriptions	Lists descriptions of all the Amazon Redshift event notification subscriptions for a
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_logging_status	Describes whether information, such as queries and connection attempts, is being
describe_node_configuration_options	Returns properties of possible node configurations such as node type, number of r
describe_orderable_cluster_options	Returns a list of orderable cluster options
describe_reserved_node_offerings	Returns a list of the available reserved node offerings by Amazon Redshift with th
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 AWS account in the destinatio
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 RestoreTable
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 specif
disable_snapshot_copy	Disables the automatic copying of snapshots from one region to another region fo
enable_logging	Starts logging information, such as queries and connection attempts, for the speci
enable_snapshot_copy	Enables the automatic copy of snapshots from one region to another region for a s
get_cluster_credentials	Returns a database user name and temporary password with temporary authorizat
get_reserved_node_exchange_offerings	Returns an array of DC2 ReservedNodeOfferings that matches the payment type,
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 AWS Identity and Access Management (IAM) roles that can b
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_event_subscription	Modifies an existing Amazon Redshift event notification subscription
modify_scheduled_action	Modifies a scheduled action
modify_snapshot_copy_retention_period	Modifies the number of days to retain snapshots in the destination AWS Region a
modify_snapshot_schedule	Modifies a snapshot schedule
modify_usage_limit	Modifies a usage limit in a cluster
pause_cluster	Pauses a cluster
purchase_reserved_node_offering	Allows you to purchase reserved nodes
reboot_cluster	Reboots a cluster
reset_cluster_parameter_group	Sets one or more parameters of the specified parameter group to their default valu
resize_cluster	Changes the size of the cluster
restore_from_cluster_snapshot	Creates a new cluster from a snapshot
restore_table_from_cluster_snapshot	Creates a new table from a table in an Amazon Redshift cluster snapshot
resume_cluster	Resumes a paused cluster
revoke_cluster_security_group_ingress	Revokes an ingress rule in an Amazon Redshift security group for a previously au
revoke_snapshot_access	Removes the ability of the specified AWS customer account to restore the specifie
rotate_encryption_key	Rotates the encryption keys for a cluster

Examples

```
## Not run:
```

```
svc <- redshift()
svc$accept_reserved_node_exchange(
  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())
```

Arguments

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

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 <- simpledb(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

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

Examples

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

## End(Not run)
```

Index