Package 'paws.analytics'

May 30, 2026

Title 'Amazon Web Services' Analytics Services

Version 0.10.0

Description Interface to 'Amazon Web Services' 'analytics' services, including 'Elastic MapReduce' 'Hadoop' and 'Spark' big data service, 'Elasticsearch' search engine, 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.analytics, https://www.paws-r-sdk.com

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

Imports paws.common (>= 0.8.0)

Suggests testthat

Encoding UTF-8

Config/roxygen2/version 8.0.0

Collate 'athena_service.R' 'athena_interfaces.R' 'athena_operations.R' 'cloudsearch_service.R' 'cloudsearch_interfaces.R' 'cloudsearch_operations.R' 'cloudsearchdomain_service.R' 'cloudsearchdomain_interfaces.R' 'cloudsearchdomain_operations.R' 'datapipeline_service.R' 'datapipeline_interfaces.R' 'datapipeline_operations.R' 'datazone_service.R' 'datazone_interfaces.R' 'datazone_operations.R' 'elasticsearchservice_service.R' 'elasticsearchservice_interfaces.R' 'elasticsearchservice_operations.R' 'emr_service.R' 'emr_interfaces.R' 'emr_operations.R' 'entityresolution_service.R' 'entityresolution_interfaces.R' 'entityresolution_operations.R' 'firehose_service.R' 'firehose_interfaces.R' 'firehose_operations.R' 'glue_service.R' 'glue_interfaces.R' 'glue_operations.R' 'gluedatabrew_service.R' 'gluedatabrew_interfaces.R' 'gluedatabrew_operations.R' 'healthlake_service.R' 'healthlake_interfaces.R' 'healthlake_operations.R'

'ivs_service.R' 'ivs_interfaces.R' 'ivs_operations.R'
'ivsrealtime_service.R' 'ivsrealtime_interfaces.R'
'ivsrealtime_operations.R' 'kafka_service.R'
'kafka_interfaces.R' 'kafka_operations.R'
'kafkaconnect_service.R' 'kafkaconnect_interfaces.R'
'kafkaconnect_operations.R' 'kendra_service.R'
'kendra_interfaces.R' 'kendra_operations.R'
'kendraranking_service.R' 'kendraranking_interfaces.R'
'kendraranking_operations.R' 'kinesis_service.R'
'kinesis_interfaces.R' 'kinesis_operations.R'
'kinesisanalytics_service.R' 'kinesisanalytics_interfaces.R'
'kinesisanalytics_operations.R' 'kinesisanalyticsv2_service.R'
'kinesisanalyticsv2_interfaces.R'
'kinesisanalyticsv2_operations.R' 'mturk_service.R'
'mturk_interfaces.R' 'mturk_operations.R'
'opensearchingestion_service.R'
'opensearchingestion_interfaces.R'
'opensearchingestion_operations.R'
'opensearchservice_service.R' 'opensearchservice_interfaces.R'
'opensearchservice_operations.R'
'opensearchserviceserverless_service.R'
'opensearchserviceserverless_interfaces.R'
'opensearchserviceserverless_operations.R'
'quicksight_service.R' 'quicksight_interfaces.R'
'quicksight_operations.R' 'reexports_paws.common.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 2026-05-30 05:12:10 UTC

Contents

`athena` *Amazon Athena*

Description

Amazon Athena is an interactive query service that lets you use standard SQL to analyze data directly in Amazon S3. You can point Athena at your data in Amazon S3 and run ad-hoc queries and get results in seconds. Athena is serverless, so there is no infrastructure to set up or manage. You pay only for the queries you run. Athena scales automatically—executing queries in parallel—so results are fast, even with large datasets and complex queries. For more information, see What is Amazon Athena in the *Amazon Athena User Guide*.

If you connect to Athena using the JDBC driver, use version 1.1.0 of the driver or later with the Amazon Athena API. Earlier version drivers do not support the API. For more information and to download the driver, see Accessing Amazon Athena with JDBC.

Usage

```
athena(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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

batch_get_named_query	Returns the details of a single named query or a list of up to 50 queries, which you
batch_get_prepared_statement	Returns the details of a single prepared statement or a list of up to 256 prepared stat
batch_get_query_execution	Returns the details of a single query execution or a list of up to 50 query executions
cancel_capacity_reservation	Cancels the capacity reservation with the specified name
create_capacity_reservation	Creates a capacity reservation with the specified name and number of requested dat
create_data_catalog	Creates (registers) a data catalog with the specified name and properties
create_named_query	Creates a named query in the specified workgroup
create_notebook	Creates an empty ipynb file in the specified Apache Spark enabled workgroup
create_prepared_statement	Creates a prepared statement for use with SQL queries in Athena
create_presigned_notebook_url	Gets an authentication token and the URL at which the notebook can be accessed
create_work_group	Creates a workgroup with the specified name
delete_capacity_reservation	Deletes a cancelled capacity reservation
delete_data_catalog	Deletes a data catalog
delete_named_query	Deletes the named query if you have access to the workgroup in which the query wa
delete_notebook	Deletes the specified notebook
delete_prepared_statement	Deletes the prepared statement with the specified name from the specified workgrou
delete_work_group	Deletes the workgroup with the specified name
export_notebook	Exports the specified notebook and its metadata
get_calculation_execution	Describes a previously submitted calculation execution
get_calculation_execution_code	Retrieves the unencrypted code that was executed for the calculation
get_calculation_execution_status	Gets the status of a current calculation
get_capacity_assignment_configuration	Gets the capacity assignment configuration for a capacity reservation, if one exists
get_capacity_reservation	Returns information about the capacity reservation with the specified name
get_database	Returns a database object for the specified database and data catalog
get_data_catalog	Returns the specified data catalog
get_named_query	Returns information about a single query
get_notebook_metadata	Retrieves notebook metadata for the specified notebook ID
get_prepared_statement	Retrieves the prepared statement with the specified name from the specified workgr
get_query_execution	Returns information about a single execution of a query if you have access to the w

get_query_results	Streams the results of a single query execution specified by QueryExecutionId from
get_query_runtime_statistics	Returns query execution runtime statistics related to a single execution of a query if
get_resource_dashboard	Gets the Live UI/Persistence UI for a session
get_session	Gets the full details of a previously created session, including the session status and
get_session_endpoint	Gets a connection endpoint and authentication token for a given session Id
get_session_status	Gets the current status of a session
get_table_metadata	Returns table metadata for the specified catalog, database, and table
get_work_group	Returns information about the workgroup with the specified name
import_notebook	Imports a single ipynb file to a Spark enabled workgroup
list_application_dpu_sizes	Returns the supported DPU sizes for the supported application runtimes (for examp
list_calculation_executions	Lists the calculations that have been submitted to a session in descending order
list_capacity_reservations	Lists the capacity reservations for the current account
list_databases	Lists the databases in the specified data catalog
list_data_catalogs	Lists the data catalogs in the current Amazon Web Services account
list_engine_versions	Returns a list of engine versions that are available to choose from, including the Au
list_executors	Lists, in descending order, the executors that joined a session
list_named_queries	Provides a list of available query IDs only for queries saved in the specified workgr
list_notebook_metadata	Displays the notebook files for the specified workgroup in paginated format
list_notebook_sessions	Lists, in descending order, the sessions that have been created in a notebook that are
list_prepared_statements	Lists the prepared statements in the specified workgroup
list_query_executions	Provides a list of available query execution IDs for the queries in the specified work
list_sessions	Lists the sessions in a workgroup that are in an active state like CREATING, CREA
list_table_metadata	Lists the metadata for the tables in the specified data catalog database
list_tags_for_resource	Lists the tags associated with an Athena resource
list_work_groups	Lists available workgroups for the account
put_capacity_assignment_configuration	Puts a new capacity assignment configuration for a specified capacity reservation
start_calculation_execution	Submits calculations for execution within a session
start_query_execution	Runs the SQL query statements contained in the Query
start_session	Creates a session for running calculations within a workgroup
stop_calculation_execution	Requests the cancellation of a calculation
stop_query_execution	Stops a query execution
tag_resource	Adds one or more tags to an Athena resource
terminate_session	Terminates an active session
untag_resource	Removes one or more tags from an Athena resource
update_capacity_reservation	Updates the number of requested data processing units for the capacity reservation
update_data_catalog	Updates the data catalog that has the specified name
update_named_query	Updates a NamedQuery object
update_notebook	Updates the contents of a Spark notebook
update_notebook_metadata	Updates the metadata for a notebook
update_prepared_statement	Updates a prepared statement
update_work_group	Updates the workgroup with the specified name

Examples

```
## Not run:
svc <- athena()
svc$batch_get_named_query(
```

```
  Foo = 123
)

## End(Not run)
```

`cloudsearch` *Amazon CloudSearch*

Description

Amazon CloudSearch Configuration Service

You use the Amazon CloudSearch configuration service to create, configure, and manage search domains. Configuration service requests are submitted using the AWS Query protocol. AWS Query requests are HTTP or HTTPS requests submitted via HTTP GET or POST with a query parameter named Action.

The endpoint for configuration service requests is region-specific: cloudsearch.*region*.amazonaws.com. For example, cloudsearch.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

Usage

```
cloudsearch(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

build_suggesters	Indexes the search suggestions
create_domain	Creates a new search domain
define_analysis_scheme	Configures an analysis scheme that can be applied to a text or text-array field to define l
define_expression	Configures an Expression for the search domain
define_index_field	Configures an IndexField for the search domain
define_suggester	Configures a suggester for a domain
delete_analysis_scheme	Deletes an analysis scheme
delete_domain	Permanently deletes a search domain and all of its data
delete_expression	Removes an Expression from the search domain
delete_index_field	Removes an IndexField from the search domain
delete_suggester	Deletes a suggester
describe_analysis_schemes	Gets the analysis schemes configured for a domain
describe_availability_options	Gets the availability options configured for a domain
describe_domain_endpoint_options	Returns the domain's endpoint options, specifically whether all requests to the domain r
describe_domains	Gets information about the search domains owned by this account
describe_expressions	Gets the expressions configured for the search domain
describe_index_fields	Gets information about the index fields configured for the search domain
describe_scaling_parameters	Gets the scaling parameters configured for a domain
describe_service_access_policies	Gets information about the access policies that control access to the domain's document
describe_suggesters	Gets the suggesters configured for a domain
index_documents	Tells the search domain to start indexing its documents using the latest indexing options
list_domain_names	Lists all search domains owned by an account
update_availability_options	Configures the availability options for a domain
update_domain_endpoint_options	Updates the domain's endpoint options, specifically whether all requests to the domain r
update_scaling_parameters	Configures scaling parameters for a domain
update_service_access_policies	Configures the access rules that control access to the domain's document and search end

Examples

```
## Not run:
svc <- cloudsearch()
svc$build_suggesters(
  Foo = 123
)

## End(Not run)
```

`cloudsearchdomain` *Amazon CloudSearch Domain*

Description

You use the AmazonCloudSearch2013 API to upload documents to a search domain and search those documents.

The endpoints for submitting `upload_documents`, `search`, and `suggest` requests are domain-specific. To get the endpoints for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. The domain endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console. You submit suggest requests to the search endpoint.

For more information, see the Amazon CloudSearch Developer Guide.

Usage

```
cloudsearchdomain(
  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-e` `html`

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

search	Retrieves a list of documents that match the specified search criteria
suggest	Retrieves autocomplete suggestions for a partial query string
upload_documents	Posts a batch of documents to a search domain for indexing

Examples

```
## Not run:
svc <- cloudsearchdomain()
svc$search(
  Foo = 123
)

## End(Not run)
```

`datapipeline` *AWS Data Pipeline*

Description

AWS Data Pipeline configures and manages a data-driven workflow called a pipeline. AWS Data Pipeline handles the details of scheduling and ensuring that data dependencies are met so that your application can focus on processing the data.

AWS Data Pipeline provides a JAR implementation of a task runner called AWS Data Pipeline Task Runner. AWS Data Pipeline Task Runner provides logic for common data management scenarios, such as performing database queries and running data analysis using Amazon Elastic MapReduce (Amazon EMR). You can use AWS Data Pipeline Task Runner as your task runner, or you can write your own task runner to provide custom data management.

AWS Data Pipeline implements two main sets of functionality. Use the first set to create a pipeline and define data sources, schedules, dependencies, and the transforms to be performed on the data. Use the second set in your task runner application to receive the next task ready for processing. The logic for performing the task, such as querying the data, running data analysis, or converting the data from one format to another, is contained within the task runner. The task runner performs the task assigned to it by the web service, reporting progress to the web service as it does so. When the task is done, the task runner reports the final success or failure of the task to the web service.

Usage

```
datapipeline(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

activate_pipeline	Validates the specified pipeline and starts processing pipeline tasks
add_tags	Adds or modifies tags for the specified pipeline
create_pipeline	Creates a new, empty pipeline
deactivate_pipeline	Deactivates the specified running pipeline
delete_pipeline	Deletes a pipeline, its pipeline definition, and its run history
describe_objects	Gets the object definitions for a set of objects associated with the pipeline
describe_pipelines	Retrieves metadata about one or more pipelines
evaluate_expression	Task runners call EvaluateExpression to evaluate a string in the context of the specified object
get_pipeline_definition	Gets the definition of the specified pipeline
list_pipelines	Lists the pipeline identifiers for all active pipelines that you have permission to access
poll_for_task	Task runners call PollForTask to receive a task to perform from AWS Data Pipeline
put_pipeline_definition	Adds tasks, schedules, and preconditions to the specified pipeline
query_objects	Queries the specified pipeline for the names of objects that match the specified set of condition
remove_tags	Removes existing tags from the specified pipeline
report_task_progress	Task runners call ReportTaskProgress when assigned a task to acknowledge that it has the task
report_task_runner_heartbeat	Task runners call ReportTaskRunnerHeartbeat every 15 minutes to indicate that they are opera
set_status	Requests that the status of the specified physical or logical pipeline objects be updated in the s
set_task_status	Task runners call SetTaskStatus to notify AWS Data Pipeline that a task is completed and prov
validate_pipeline_definition	Validates the specified pipeline definition to ensure that it is well formed and can be run witho

Examples

```
## Not run:
svc <- datapipeline()
svc$activate_pipeline(
  Foo = 123
)

## End(Not run)
```

`datazone` *Amazon DataZone*

Description

Amazon DataZone is a data management service that enables you to catalog, discover, govern, share, and analyze your data. With Amazon DataZone, you can share and access your data across accounts and supported regions. Amazon DataZone simplifies your experience across Amazon Web Services services, including, but not limited to, Amazon Redshift, Amazon Athena, Amazon Web Services Glue, and Amazon Web Services Lake Formation.

Usage

```
datazone(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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

accept_predictions	Accepts automatically generated business-friendly metadata for your Amazon
accept_subscription_request	Accepts a subscription request to a specific asset
add_entity_owner	Adds the owner of an entity (a domain unit)
add_policy_grant	Adds a policy grant (an authorization policy) to a specified entity, including do
associate_environment_role	Associates the environment role in Amazon DataZone
associate_governed_terms	Associates governed terms with an asset
batch_get_attributes_metadata	Gets the attribute metadata
batch_put_attributes_metadata	Writes the attribute metadata
cancel_metadata_generation_run	Cancels the metadata generation run
cancel_subscription	Cancels the subscription to the specified asset
create_account_pool	Creates an account pool
create_asset	Creates an asset in Amazon DataZone catalog
create_asset_filter	Creates a data asset filter
create_asset_revision	Creates a revision of the asset
create_asset_type	Creates a custom asset type
create_connection	Creates a new connection
create_data_product	Creates a data product
create_data_product_revision	Creates a data product revision
create_data_source	Creates an Amazon DataZone data source
create_domain	Creates an Amazon DataZone domain
create_domain_unit	Creates a domain unit in Amazon DataZone
create_environment	Create an Amazon DataZone environment
create_environment_action	Creates an action for the environment, for example, creates a console link for a
create_environment_blueprint	Creates a Amazon DataZone blueprint
create_environment_profile	Creates an Amazon DataZone environment profile
create_form_type	Creates a metadata form type
create_glossary	Creates an Amazon DataZone business glossary
create_glossary_term	Creates a business glossary term
create_group_profile	Creates a group profile in Amazon DataZone
create_listing_change_set	Publishes a listing (a record of an asset at a given time) or removes a listing fro
create_notebook	Creates a notebook in Amazon SageMaker Unified Studio
create_project	Creates an Amazon DataZone project
create_project_membership	Creates a project membership in Amazon DataZone
create_project_profile	Creates a project profile
create_rule	Creates a rule in Amazon DataZone
create_subscription_grant	Creates a subsscription grant in Amazon DataZone
create_subscription_request	Creates a subscription request in Amazon DataZone
create_subscription_target	Creates a subscription target in Amazon DataZone
create_user_profile	Creates a user profile in Amazon DataZone
delete_account_pool	Deletes an account pool

delete_asset	Deletes an asset in Amazon DataZone
delete_asset_filter	Deletes an asset filter
delete_asset_type	Deletes an asset type in Amazon DataZone
delete_connection	Deletes and connection
delete_data_export_configuration	Deletes data export configuration for a domain
delete_data_product	Deletes a data product in Amazon DataZone
delete_data_source	Deletes a data source in Amazon DataZone
delete_domain	Deletes a Amazon DataZone domain
delete_domain_unit	Deletes a domain unit
delete_environment	Deletes an environment in Amazon DataZone
delete_environment_action	Deletes an action for the environment, for example, deletes a console link for a
delete_environment_blueprint	Deletes a blueprint in Amazon DataZone
delete_environment_blueprint_configuration	Deletes the blueprint configuration in Amazon DataZone
delete_environment_profile	Deletes an environment profile in Amazon DataZone
delete_form_type	Deletes and metadata form type in Amazon DataZone
delete_glossary	Deletes a business glossary in Amazon DataZone
delete_glossary_term	Deletes a business glossary term in Amazon DataZone
delete_listing	Deletes a listing (a record of an asset at a given time)
delete_notebook	Deletes a notebook in Amazon SageMaker Unified Studio
delete_project	Deletes a project in Amazon DataZone
delete_project_membership	Deletes project membership in Amazon DataZone
delete_project_profile	Deletes a project profile
delete_rule	Deletes a rule in Amazon DataZone
delete_subscription_grant	Deletes and subscription grant in Amazon DataZone
delete_subscription_request	Deletes a subscription request in Amazon DataZone
delete_subscription_target	Deletes a subscription target in Amazon DataZone
delete_time_series_data_points	Deletes the specified time series form for the specified asset
disassociate_environment_role	Disassociates the environment role in Amazon DataZone
disassociate_governed_terms	Disassociates restricted terms from an asset
get_account_pool	Gets the details of the account pool
get_asset	Gets an Amazon DataZone asset
get_asset_filter	Gets an asset filter
get_asset_type	Gets an Amazon DataZone asset type
get_connection	Gets a connection
get_data_export_configuration	Gets data export configuration details
get_data_product	Gets the data product
get_data_source	Gets an Amazon DataZone data source
get_data_source_run	Gets an Amazon DataZone data source run
get_domain	Gets an Amazon DataZone domain
get_domain_unit	Gets the details of the specified domain unit
get_environment	Gets an Amazon DataZone environment
get_environment_action	Gets the specified environment action
get_environment_blueprint	Gets an Amazon DataZone blueprint
get_environment_blueprint_configuration	Gets the blueprint configuration in Amazon DataZone
get_environment_credentials	Gets the credentials of an environment in Amazon DataZone
get_environment_profile	Gets an evinronment profile in Amazon DataZone
get_form_type	Gets a metadata form type in Amazon DataZone
get_glossary	Gets a business glossary in Amazon DataZone

get_glossary_term	Gets a business glossary term in Amazon DataZone
get_group_profile	Gets a group profile in Amazon DataZone
get_iam_portal_login_url	Gets the data portal URL for the specified Amazon DataZone domain
get_job_run	The details of the job run
get_lineage_event	Describes the lineage event
get_lineage_node	Gets the data lineage node
get_listing	Gets a listing (a record of an asset at a given time)
get_metadata_generation_run	Gets a metadata generation run in Amazon DataZone
get_notebook	Gets the details of a notebook in Amazon SageMaker Unified Studio
get_notebook_export	Gets the details of a notebook export in Amazon SageMaker Unified Studio
get_notebook_run	Gets the details of a notebook run in Amazon SageMaker Unified Studio
get_project	Gets a project in Amazon DataZone
get_project_profile	The details of the project profile
get_rule	Gets the details of a rule in Amazon DataZone
get_subscription	Gets a subscription in Amazon DataZone
get_subscription_grant	Gets the subscription grant in Amazon DataZone
get_subscription_request_details	Gets the details of the specified subscription request
get_subscription_target	Gets the subscription target in Amazon DataZone
get_time_series_data_point	Gets the existing data point for the asset
get_user_profile	Gets a user profile in Amazon DataZone
list_account_pools	Lists existing account pools
list_accounts_in_account_pool	Lists the accounts in the specified account pool
list_asset_filters	Lists asset filters
list_asset_revisions	Lists the revisions for the asset
list_connections	Lists connections
list_data_product_revisions	Lists data product revisions
list_data_source_run_activities	Lists data source run activities
list_data_source_runs	Lists data source runs in Amazon DataZone
list_data_sources	Lists data sources in Amazon DataZone
list_domains	Lists Amazon DataZone domains
list_domain_units_for_parent	Lists child domain units for the specified parent domain unit
list_entity_owners	Lists the entity (domain units) owners
list_environment_actions	Lists existing environment actions
list_environment_blueprint_configurations	Lists blueprint configurations for a Amazon DataZone environment
list_environment_blueprints	Lists blueprints in an Amazon DataZone environment
list_environment_profiles	Lists Amazon DataZone environment profiles
list_environments	Lists Amazon DataZone environments
list_job_runs	Lists job runs
list_lineage_events	Lists lineage events
list_lineage_node_history	Lists the history of the specified data lineage node
list_metadata_generation_runs	Lists all metadata generation runs
list_notebook_runs	Lists notebook runs in Amazon SageMaker Unified Studio
list_notebooks	Lists notebooks in Amazon SageMaker Unified Studio
list_notifications	Lists all Amazon DataZone notifications
list_policy_grants	Lists policy grants
list_project_memberships	Lists all members of the specified project
list_project_profiles	Lists project profiles
list_projects	Lists Amazon DataZone projects

list_rules	Lists existing rules
list_subscription_grants	Lists subscription grants
list_subscription_requests	Lists Amazon DataZone subscription requests
list_subscriptions	Lists subscriptions in Amazon DataZone
list_subscription_targets	Lists subscription targets in Amazon DataZone
list_tags_for_resource	Lists tags for the specified resource in Amazon DataZone
list_time_series_data_points	Lists time series data points
post_lineage_event	Posts a data lineage event
post_time_series_data_points	Posts time series data points to Amazon DataZone for the specified asset
put_data_export_configuration	Creates data export configuration details
put_environment_blueprint_configuration	Writes the configuration for the specified environment blueprint in Amazon Da
query_graph	Queries entities in the graph store
reject_predictions	Rejects automatically generated business-friendly metadata for your Amazon I
reject_subscription_request	Rejects the specified subscription request
remove_entity_owner	Removes an owner from an entity
remove_policy_grant	Removes a policy grant
revoke_subscription	Revokes a specified subscription in Amazon DataZone
search	Searches for assets in Amazon DataZone
search_group_profiles	Searches group profiles in Amazon DataZone
search_listings	Searches listings in Amazon DataZone
search_types	Searches for types in Amazon DataZone
search_user_profiles	Searches user profiles in Amazon DataZone
start_data_source_run	Start the run of the specified data source in Amazon DataZone
start_metadata_generation_run	Starts the metadata generation run
start_notebook_export	Starts a notebook export in Amazon SageMaker Unified Studio
start_notebook_import	Starts a notebook import in Amazon SageMaker Unified Studio
start_notebook_run	Starts a notebook run in Amazon SageMaker Unified Studio
stop_notebook_run	Stops a running notebook run in Amazon SageMaker Unified Studio
tag_resource	Tags a resource in Amazon DataZone
untag_resource	Untags a resource in Amazon DataZone
update_account_pool	Updates the account pool
update_asset_filter	Updates an asset filter
update_connection	Updates a connection
update_data_source	Updates the specified data source in Amazon DataZone
update_domain	Updates a Amazon DataZone domain
update_domain_unit	Updates the domain unit
update_environment	Updates the specified environment in Amazon DataZone
update_environment_action	Updates an environment action
update_environment_blueprint	Updates an environment blueprint in Amazon DataZone
update_environment_profile	Updates the specified environment profile in Amazon DataZone
update_glossary	Updates the business glossary in Amazon DataZone
update_glossary_term	Updates a business glossary term in Amazon DataZone
update_group_profile	Updates the specified group profile in Amazon DataZone
update_notebook	Updates a notebook in Amazon SageMaker Unified Studio
update_project	Updates the specified project in Amazon DataZone
update_project_profile	Updates a project profile
update_root_domain_unit_owner	Updates the owner of the root domain unit
update_rule	Updates a rule

update_subscription_grant_status	Updates the status of the specified subscription grant status in Amazon DataZo
update_subscription_request	Updates a specified subscription request in Amazon DataZone
update_subscription_target	Updates the specified subscription target in Amazon DataZone
update_user_profile	Updates the specified user profile in Amazon DataZone

Examples

```
## Not run:
svc <- datazone()
svc$accept_predictions(
  Foo = 123
)

## End(Not run)
```

`elasticsearchservice` *Amazon Elasticsearch Service*

Description

Amazon Elasticsearch Configuration Service

Use the Amazon Elasticsearch Configuration API to create, configure, and manage Elasticsearch domains.

For sample code that uses the Configuration API, see the Amazon Elasticsearch Service Developer Guide. The guide also contains sample code for sending signed HTTP requests to the Elasticsearch APIs.

The endpoint for configuration service requests is region-specific: es.*region*.amazonaws.com. For example, es.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

Usage

```
elasticsearchservice(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

accept_inbound_cross_cluster_search_connection	Allows the destination domain owner to accept an inbound cross-clus
add_tags	Attaches tags to an existing Elasticsearch domain
associate_package	Associates a package with an Amazon ES domain
authorize_vpc_endpoint_access	Provides access to an Amazon OpenSearch Service domain through t
cancel_domain_config_change	Cancels a pending configuration change on an Amazon OpenSearch S
cancel_elasticsearch_service_software_update	Cancels a scheduled service software update for an Amazon ES doma
create_elasticsearch_domain	Creates a new Elasticsearch domain
create_outbound_cross_cluster_search_connection	Creates a new cross-cluster search connection from a source domain t
create_package	Create a package for use with Amazon ES domains
create_vpc_endpoint	Creates an Amazon OpenSearch Service-managed VPC endpoint
delete_elasticsearch_domain	Permanently deletes the specified Elasticsearch domain and all of its
delete_elasticsearch_service_role	Deletes the service-linked role that Elasticsearch Service uses to man
delete_inbound_cross_cluster_search_connection	Allows the destination domain owner to delete an existing inbound cr
delete_outbound_cross_cluster_search_connection	Allows the source domain owner to delete an existing outbound cross
delete_package	Delete the package
delete_vpc_endpoint	Deletes an Amazon OpenSearch Service-managed interface VPC end
describe_domain_auto_tunes	Provides scheduled Auto-Tune action details for the Elasticsearch do
describe_domain_change_progress	Returns information about the current blue/green deployment happen
describe_elasticsearch_domain	Returns domain configuration information about the specified Elastic
describe_elasticsearch_domain_config	Provides cluster configuration information about the specified Elastic

describe_elasticsearch_domains	Returns domain configuration information about the specified Elastic
describe_elasticsearch_instance_type_limits	Describe Elasticsearch Limits for a given InstanceType and Elasticsea
describe_inbound_cross_cluster_search_connections	Lists all the inbound cross-cluster search connections for a destinatio
describe_outbound_cross_cluster_search_connections	Lists all the outbound cross-cluster search connections for a source d
describe_packages	Describes all packages available to Amazon ES
describe_reserved_elasticsearch_instance_offerings	Lists available reserved Elasticsearch instance offerings
describe_reserved_elasticsearch_instances	Returns information about reserved Elasticsearch instances for this ac
describe_vpc_endpoints	Describes one or more Amazon OpenSearch Service-managed VPC e
dissociate_package	Dissociates a package from the Amazon ES domain
get_compatible_elasticsearch_versions	Returns a list of upgrade compatible Elastisearch versions
get_package_version_history	Returns a list of versions of the package, along with their creation tim
get_upgrade_history	Retrieves the complete history of the last 10 upgrades that were perfo
get_upgrade_status	Retrieves the latest status of the last upgrade or upgrade eligibility ch
list_domain_names	Returns the name of all Elasticsearch domains owned by the current u
list_domains_for_package	Lists all Amazon ES domains associated with the package
list_elasticsearch_instance_types	List all Elasticsearch instance types that are supported for given Elast
list_elasticsearch_versions	List all supported Elasticsearch versions
list_packages_for_domain	Lists all packages associated with the Amazon ES domain
list_tags	Returns all tags for the given Elasticsearch domain
list_vpc_endpoint_access	Retrieves information about each principal that is allowed to access a
list_vpc_endpoints	Retrieves all Amazon OpenSearch Service-managed VPC endpoints i
list_vpc_endpoints_for_domain	Retrieves all Amazon OpenSearch Service-managed VPC endpoints a
purchase_reserved_elasticsearch_instance_offering	Allows you to purchase reserved Elasticsearch instances
reject_inbound_cross_cluster_search_connection	Allows the destination domain owner to reject an inbound cross-clust
remove_tags	Removes the specified set of tags from the specified Elasticsearch do
revoke_vpc_endpoint_access	Revokes access to an Amazon OpenSearch Service domain that was p
start_elasticsearch_service_software_update	Schedules a service software update for an Amazon ES domain
update_elasticsearch_domain_config	Modifies the cluster configuration of the specified Elasticsearch doma
update_package	Updates a package for use with Amazon ES domains
update_vpc_endpoint	Modifies an Amazon OpenSearch Service-managed interface VPC en
upgrade_elasticsearch_domain	Allows you to either upgrade your domain or perform an Upgrade eli

Examples

```
## Not run:
svc <- elasticsearchservice()
svc$accept_inbound_cross_cluster_search_connection(
  Foo = 123
)

## End(Not run)
```

emr *Amazon EMR*

Description

Amazon EMR is a web service that makes it easier to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several Amazon Web Services services to do tasks such as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehouse management.

Usage

```
emr(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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

add_instance_fleet	Adds an instance fleet to a running cluster
add_instance_groups	Adds one or more instance groups to a running cluster
add_job_flow_steps	AddJobFlowSteps adds new steps to a running cluster
add_tags	Adds tags to an Amazon EMR resource, such as a cluster or an Amazon EMR Stu
cancel_steps	Cancels a pending step or steps in a running cluster
create_persistent_app_ui	Creates a persistent application user interface
create_security_configuration	Creates a security configuration, which is stored in the service and can be specifie
create_studio	Creates a new Amazon EMR Studio

create_studio_session_mapping	Maps a user or group to the Amazon EMR Studio specified by StudioId, and appli
delete_security_configuration	Deletes a security configuration
delete_studio	Removes an Amazon EMR Studio from the Studio metadata store
delete_studio_session_mapping	Removes a user or group from an Amazon EMR Studio
describe_cluster	Provides cluster-level details including status, hardware and software configuratio
describe_job_flows	This API is no longer supported and will eventually be removed
describe_notebook_execution	Provides details of a notebook execution
describe_persistent_app_ui	Describes a persistent application user interface
describe_release_label	Provides Amazon EMR release label details, such as the releases available the Reg
describe_security_configuration	Provides the details of a security configuration by returning the configuration JSO
describe_step	Provides more detail about the cluster step
describe_studio	Returns details for the specified Amazon EMR Studio including ID, Name, VPC,
get_auto_termination_policy	Returns the auto-termination policy for an Amazon EMR cluster
get_block_public_access_configuration	Returns the Amazon EMR block public access configuration for your Amazon We
get_cluster_session_credentials	Provides temporary, HTTP basic credentials that are associated with a given runti
get_managed_scaling_policy	Fetches the attached managed scaling policy for an Amazon EMR cluster
get_on_cluster_app_ui_presigned_url	The presigned URL properties for the cluster's application user interface
get_persistent_app_ui_presigned_url	The presigned URL properties for the cluster's application user interface
get_studio_session_mapping	Fetches mapping details for the specified Amazon EMR Studio and identity (user
list_bootstrap_actions	Provides information about the bootstrap actions associated with a cluster
list_clusters	Provides the status of all clusters visible to this Amazon Web Services account
list_instance_fleets	Lists all available details about the instance fleets in a cluster
list_instance_groups	Provides all available details about the instance groups in a cluster
list_instances	Provides information for all active Amazon EC2 instances and Amazon EC2 insta
list_notebook_executions	Provides summaries of all notebook executions
list_release_labels	Retrieves release labels of Amazon EMR services in the Region where the API is
list_security_configurations	Lists all the security configurations visible to this account, providing their creatio
list_steps	Provides a list of steps for the cluster in reverse order unless you specify stepIds w
list_studios	Returns a list of all Amazon EMR Studios associated with the Amazon Web Servi
list_studio_session_mappings	Returns a list of all user or group session mappings for the Amazon EMR Studio s
list_supported_instance_types	A list of the instance types that Amazon EMR supports
modify_cluster	Modifies the number of steps that can be executed concurrently for the cluster spe
modify_instance_fleet	Modifies the target On-Demand and target Spot capacities for the instance fleet wi
modify_instance_groups	ModifyInstanceGroups modifies the number of nodes and configuration settings o
put_auto_scaling_policy	Creates or updates an automatic scaling policy for a core instance group or task in
put_auto_termination_policy	Auto-termination is supported in Amazon EMR releases 5
put_block_public_access_configuration	Creates or updates an Amazon EMR block public access configuration for your A
put_managed_scaling_policy	Creates or updates a managed scaling policy for an Amazon EMR cluster
remove_auto_scaling_policy	Removes an automatic scaling policy from a specified instance group within an A
remove_auto_termination_policy	Removes an auto-termination policy from an Amazon EMR cluster
remove_managed_scaling_policy	Removes a managed scaling policy from a specified Amazon EMR cluster
remove_tags	Removes tags from an Amazon EMR resource, such as a cluster or Amazon EMR
run_job_flow	RunJobFlow creates and starts running a new cluster (job flow)
set_keep_job_flow_alive_when_no_steps	You can use the SetKeepJobFlowAliveWhenNoSteps to configure a cluster (job fl
set_termination_protection	SetTerminationProtection locks a cluster (job flow) so the Amazon EC2 instances
set_unhealthy_node_replacement	Specify whether to enable unhealthy node replacement, which lets Amazon EMR
set_visible_to_all_users	The SetVisibleToAllUsers parameter is no longer supported
start_notebook_execution	Starts a notebook execution

stop_notebook_execution	Stops a notebook execution
terminate_job_flows	TerminateJobFlows shuts a list of clusters (job flows) down
update_studio	Updates an Amazon EMR Studio configuration, including attributes such as name
update_studio_session_mapping	Updates the session policy attached to the user or group for the specified Amazon

Examples

```
## Not run:
svc <- emr()
svc$add_instance_fleet(
  Foo = 123
)

## End(Not run)
```

`entityresolution` *AWS EntityResolution*

Description

Welcome to the *Entity Resolution API Reference*.

Entity Resolution is an Amazon Web Services service that provides pre-configured entity resolution capabilities that enable developers and analysts at advertising and marketing companies to build an accurate and complete view of their consumers.

With Entity Resolution, you can match source records containing consumer identifiers, such as name, email address, and phone number. This is true even when these records have incomplete or conflicting identifiers. For example, Entity Resolution can effectively match a source record from a customer relationship management (CRM) system with a source record from a marketing system containing campaign information.

To learn more about Entity Resolution concepts, procedures, and best practices, see the Entity Resolution User Guide.

Usage

```
entityresolution(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

add_policy_statement	Adds a policy statement object
batch_delete_unique_id	Deletes multiple unique IDs in a matching workflow
create_id_mapping_workflow	Creates an IdMappingWorkflow object which stores the configuration of the data processing
create_id_namespace	Creates an ID namespace object which will help customers provide metadata explaining thei
create_matching_workflow	Creates a matching workflow that defines the configuration for a data processing job
create_schema_mapping	Creates a schema mapping, which defines the schema of the input customer records table
delete_id_mapping_workflow	Deletes the IdMappingWorkflow with a given name
delete_id_namespace	Deletes the IdNamespace with a given name
delete_matching_workflow	Deletes the MatchingWorkflow with a given name
delete_policy_statement	Deletes the policy statement
delete_schema_mapping	Deletes the SchemaMapping with a given name
generate_match_id	Generates or retrieves Match IDs for records using a rule-based matching workflow
get_id_mapping_job	Returns the status, metrics, and errors (if there are any) that are associated with a job
get_id_mapping_workflow	Returns the IdMappingWorkflow with a given name, if it exists
get_id_namespace	Returns the IdNamespace with a given name, if it exists
get_match_id	Returns the corresponding Match ID of a customer record if the record has been processed in
get_matching_job	Returns the status, metrics, and errors (if there are any) that are associated with a job
get_matching_workflow	Returns the MatchingWorkflow with a given name, if it exists
get_policy	Returns the resource-based policy
get_provider_service	Returns the ProviderService of a given name

get_schema_mapping	Returns the SchemaMapping of a given name
list_id_mapping_jobs	Lists all ID mapping jobs for a given workflow
list_id_mapping_workflows	Returns a list of all the IdMappingWorkflows that have been created for an Amazon Web Ser
list_id_namespaces	Returns a list of all ID namespaces
list_matching_jobs	Lists all jobs for a given workflow
list_matching_workflows	Returns a list of all the MatchingWorkflows that have been created for an Amazon Web Serv
list_provider_services	Returns a list of all the ProviderServices that are available in this Amazon Web Services Reg
list_schema_mappings	Returns a list of all the SchemaMappings that have been created for an Amazon Web Service
list_tags_for_resource	Displays the tags associated with an Entity Resolution resource
put_policy	Updates the resource-based policy
start_id_mapping_job	Starts the IdMappingJob of a workflow
start_matching_job	Starts the MatchingJob of a workflow
tag_resource	Assigns one or more tags (key-value pairs) to the specified Entity Resolution resource
untag_resource	Removes one or more tags from the specified Entity Resolution resource
update_id_mapping_workflow	Updates an existing IdMappingWorkflow
update_id_namespace	Updates an existing ID namespace
update_matching_workflow	Updates an existing matching workflow
update_schema_mapping	Updates a schema mapping

Examples

```
## Not run:
svc <- entityresolution()
svc$add_policy_statement(
  Foo = 123
)

## End(Not run)
```

`firehose` *Amazon Kinesis Firehose*

Description

Amazon Data Firehose

Amazon Data Firehose was previously known as Amazon Kinesis Data Firehose.

Amazon Data Firehose is a fully managed service that delivers real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon OpenSearch Service, Amazon Redshift, Splunk, and various other supported destinations.

Usage

```
firehose(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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

create_delivery_stream	Creates a Firehose stream
delete_delivery_stream	Deletes a Firehose stream and its data
describe_delivery_stream	Describes the specified Firehose stream and its status
list_delivery_streams	Lists your Firehose streams in alphabetical order of their names
list_tags_for_delivery_stream	Lists the tags for the specified Firehose stream
put_record	Writes a single data record into an Firehose stream
put_record_batch	Writes multiple data records into a Firehose stream in a single call, which can achieve hig
start_delivery_stream_encryption	Enables server-side encryption (SSE) for the Firehose stream
stop_delivery_stream_encryption	Disables server-side encryption (SSE) for the Firehose stream
tag_delivery_stream	Adds or updates tags for the specified Firehose stream
untag_delivery_stream	Removes tags from the specified Firehose stream
update_destination	Updates the specified destination of the specified Firehose stream

Examples

```
## Not run:
svc <- firehose()
svc$create_delivery_stream(
```

```
  Foo = 123
)

## End(Not run)
```

---

glue *AWS Glue*

Description

Glue

Defines the public endpoint for the Glue service.

Usage

```
glue(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-e html
credentials	Optional credentials shorthand for the config parameter • **creds**: – **access_key_id**: AWS access key ID – **secret_access_key**: AWS secret access key – **session_token**: AWS temporary session token

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

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

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

Value

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

Service syntax

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

Operations

batch_create_partition	Creates one or more partitions in a batch operation
batch_delete_connection	Deletes a list of connection definitions from the Data Catalog
batch_delete_partition	Deletes one or more partitions in a batch operation
batch_delete_table	Deletes multiple tables at once
batch_delete_table_version	Deletes a specified batch of versions of a table
batch_get_blueprints	Retrieves information about a list of blueprints
batch_get_crawlers	Returns a list of resource metadata for a given list of crawler names
batch_get_custom_entity_types	Retrieves the details for the custom patterns specified by a list of names
batch_get_data_quality_result	Retrieves a list of data quality results for the specified result IDs
batch_get_dev_endpoints	Returns a list of resource metadata for a given list of development endpoint
batch_get_jobs	Returns a list of resource metadata for a given list of job names
batch_get_partition	Retrieves partitions in a batch request
batch_get_table_optimizer	Returns the configuration for the specified table optimizers
batch_get_triggers	Returns a list of resource metadata for a given list of trigger names
batch_get_workflows	Returns a list of resource metadata for a given list of workflow names
batch_put_data_quality_statistic_annotation	Annotate datapoints over time for a specific data quality statistic
batch_stop_job_run	Stops one or more job runs for a specified job definition
batch_update_partition	Updates one or more partitions in a batch operation
cancel_data_quality_rule_recommendation_run	Cancels the specified recommendation run that was being used to generate r
cancel_data_quality_ruleset_evaluation_run	Cancels a run where a ruleset is being evaluated against a data source
cancel_ml_task_run	Cancels (stops) a task run
cancel_statement	Cancels the statement
check_schema_version_validity	Validates the supplied schema
create_blueprint	Registers a blueprint with Glue
create_catalog	Creates a new catalog in the Glue Data Catalog
create_classifier	Creates a classifier in the user's account
create_column_statistics_task_settings	Creates settings for a column statistics task
create_connection	Creates a connection definition in the Data Catalog
create_crawler	Creates a new crawler with specified targets, role, configuration, and option
create_custom_entity_type	Creates a custom pattern that is used to detect sensitive data across the colu
create_database	Creates a new database in a Data Catalog
create_data_quality_ruleset	Creates a data quality ruleset with DQDL rules applied to a specified Glue t
create_dev_endpoint	Creates a new development endpoint
create_glue_identity_center_configuration	Creates a new Glue Identity Center configuration to enable integration betw
create_integration	Creates a Zero-ETL integration in the caller's account between two resourc
create_integration_resource_property	This API can be used for setting up the ResourceProperty of the Glue conne
create_integration_table_properties	This API is used to provide optional override properties for the the tables th
create_job	Creates a new job definition
create_ml_transform	Creates an Glue machine learning transform
create_partition	Creates a new partition
create_partition_index	Creates a specified partition index in an existing table
create_registry	Creates a new registry which may be used to hold a collection of schemas
create_schema	Creates a new schema set and registers the schema definition
create_script	Transforms a directed acyclic graph (DAG) into code
create_security_configuration	Creates a new security configuration
create_session	Creates a new session
create_table	Creates a new table definition in the Data Catalog
create_table_optimizer	Creates a new table optimizer for a specific function

create_trigger	Creates a new trigger
create_usage_profile	Creates an Glue usage profile
create_user_defined_function	Creates a new function definition in the Data Catalog
create_workflow	Creates a new workflow
delete_blueprint	Deletes an existing blueprint
delete_catalog	Removes the specified catalog from the Glue Data Catalog
delete_classifier	Removes a classifier from the Data Catalog
delete_column_statistics_for_partition	Delete the partition column statistics of a column
delete_column_statistics_for_table	Retrieves table statistics of columns
delete_column_statistics_task_settings	Deletes settings for a column statistics task
delete_connection	Deletes a connection from the Data Catalog
delete_connection_type	Deletes a custom connection type in Glue
delete_crawler	Removes a specified crawler from the Glue Data Catalog, unless the crawle
delete_custom_entity_type	Deletes a custom pattern by specifying its name
delete_database	Removes a specified database from a Data Catalog
delete_data_quality_ruleset	Deletes a data quality ruleset
delete_dev_endpoint	Deletes a specified development endpoint
delete_glue_identity_center_configuration	Deletes the existing Glue Identity Center configuration, removing the integr
delete_integration	Deletes the specified Zero-ETL integration
delete_integration_resource_property	This API is used for deleting the ResourceProperty of the Glue connection (
delete_integration_table_properties	Deletes the table properties that have been created for the tables that need to
delete_job	Deletes a specified job definition
delete_ml_transform	Deletes an Glue machine learning transform
delete_partition	Deletes a specified partition
delete_partition_index	Deletes a specified partition index from an existing table
delete_registry	Delete the entire registry including schema and all of its versions
delete_resource_policy	Deletes a specified policy
delete_schema	Deletes the entire schema set, including the schema set and all of its version
delete_schema_versions	Remove versions from the specified schema
delete_security_configuration	Deletes a specified security configuration
delete_session	Deletes the session
delete_table	Removes a table definition from the Data Catalog
delete_table_optimizer	Deletes an optimizer and all associated metadata for a table
delete_table_version	Deletes a specified version of a table
delete_trigger	Deletes a specified trigger
delete_usage_profile	Deletes the Glue specified usage profile
delete_user_defined_function	Deletes an existing function definition from the Data Catalog
delete_workflow	Deletes a workflow
describe_connection_type	The DescribeConnectionType API provides full details of the supported opt
describe_entity	Provides details regarding the entity used with the connection type, with a d
describe_inbound_integrations	Returns a list of inbound integrations for the specified integration
describe_integrations	The API is used to retrieve a list of integrations
get_blueprint	Retrieves the details of a blueprint
get_blueprint_run	Retrieves the details of a blueprint run
get_blueprint_runs	Retrieves the details of blueprint runs for a specified blueprint
get_catalog	The name of the Catalog to retrieve
get_catalog_import_status	Retrieves the status of a migration operation
get_catalogs	Retrieves all catalogs defined in a catalog in the Glue Data Catalog

get_classifier	Retrieve a classifier by name
get_classifiers	Lists all classifier objects in the Data Catalog
get_column_statistics_for_partition	Retrieves partition statistics of columns
get_column_statistics_for_table	Retrieves table statistics of columns
get_column_statistics_task_run	Get the associated metadata/information for a task run, given a task run ID
get_column_statistics_task_runs	Retrieves information about all runs associated with the specified table
get_column_statistics_task_settings	Gets settings for a column statistics task
get_connection	Retrieves a connection definition from the Data Catalog
get_connections	Retrieves a list of connection definitions from the Data Catalog
get_crawler	Retrieves metadata for a specified crawler
get_crawler_metrics	Retrieves metrics about specified crawlers
get_crawlers	Retrieves metadata for all crawlers defined in the customer account
get_custom_entity_type	Retrieves the details of a custom pattern by specifying its name
get_database	Retrieves the definition of a specified database
get_databases	Retrieves all databases defined in a given Data Catalog
get_data_catalog_encryption_settings	Retrieves the security configuration for a specified catalog
get_dataflow_graph	Transforms a Python script into a directed acyclic graph (DAG)
get_data_quality_model	Retrieve the training status of the model along with more information (Com
get_data_quality_model_result	Retrieve a statistic's predictions for a given Profile ID
get_data_quality_result	Retrieves the result of a data quality rule evaluation
get_data_quality_rule_recommendation_run	Gets the specified recommendation run that was used to generate rules
get_data_quality_ruleset	Returns an existing ruleset by identifier or name
get_data_quality_ruleset_evaluation_run	Retrieves a specific run where a ruleset is evaluated against a data source
get_dev_endpoint	Retrieves information about a specified development endpoint
get_dev_endpoints	Retrieves all the development endpoints in this Amazon Web Services acco
get_entity_records	This API is used to query preview data from a given connection type or fro
get_glue_identity_center_configuration	Retrieves the current Glue Identity Center configuration details, including t
get_integration_resource_property	This API is used for fetching the ResourceProperty of the Glue connection
get_integration_table_properties	This API is used to retrieve optional override properties for the tables that n
get_job	Retrieves an existing job definition
get_job_bookmark	Returns information on a job bookmark entry
get_job_run	Retrieves the metadata for a given job run
get_job_runs	Retrieves metadata for all runs of a given job definition
get_jobs	Retrieves all current job definitions
get_mapping	Creates mappings
get_materialized_view_refresh_task_run	Get the associated metadata/information for a task run, given a task run ID
get_ml_task_run	Gets details for a specific task run on a machine learning transform
get_ml_task_runs	Gets a list of runs for a machine learning transform
get_ml_transform	Gets an Glue machine learning transform artifact and all its corresponding
get_ml_transforms	Gets a sortable, filterable list of existing Glue machine learning transforms
get_partition	Retrieves information about a specified partition
get_partition_indexes	Retrieves the partition indexes associated with a table
get_partitions	Retrieves information about the partitions in a table
get_plan	Gets code to perform a specified mapping
get_registry	Describes the specified registry in detail
get_resource_policies	Retrieves the resource policies set on individual resources by Resource Acc
get_resource_policy	Retrieves a specified resource policy
get_schema	Describes the specified schema in detail

get_schema_by_definition	Retrieves a schema by the SchemaDefinition
get_schema_version	Get the specified schema by its unique ID assigned when a version of the sc
get_schema_versions_diff	Fetches the schema version difference in the specified difference type betwe
get_security_configuration	Retrieves a specified security configuration
get_security_configurations	Retrieves a list of all security configurations
get_session	Retrieves the session
get_statement	Retrieves the statement
get_table	Retrieves the Table definition in a Data Catalog for a specified table
get_table_optimizer	Returns the configuration of all optimizers associated with a specified table
get_tables	Retrieves the definitions of some or all of the tables in a given Database
get_table_version	Retrieves a specified version of a table
get_table_versions	Retrieves a list of strings that identify available versions of a specified table
get_tags	Retrieves a list of tags associated with a resource
get_trigger	Retrieves the definition of a trigger
get_triggers	Gets all the triggers associated with a job
get_unfiltered_partition_metadata	Retrieves partition metadata from the Data Catalog that contains unfiltered m
get_unfiltered_partitions_metadata	Retrieves partition metadata from the Data Catalog that contains unfiltered m
get_unfiltered_table_metadata	Allows a third-party analytical engine to retrieve unfiltered table metadata f
get_usage_profile	Retrieves information about the specified Glue usage profile
get_user_defined_function	Retrieves a specified function definition from the Data Catalog
get_user_defined_functions	Retrieves multiple function definitions from the Data Catalog
get_workflow	Retrieves resource metadata for a workflow
get_workflow_run	Retrieves the metadata for a given workflow run
get_workflow_run_properties	Retrieves the workflow run properties which were set during the run
get_workflow_runs	Retrieves metadata for all runs of a given workflow
import_catalog_to_glue	Imports an existing Amazon Athena Data Catalog to Glue
list_blueprints	Lists all the blueprint names in an account
list_column_statistics_task_runs	List all task runs for a particular account
list_connection_types	The ListConnectionTypes API provides a discovery mechanism to learn ava
list_crawlers	Retrieves the names of all crawler resources in this Amazon Web Services a
list_crawls	Returns all the crawls of a specified crawler
list_custom_entity_types	Lists all the custom patterns that have been created
list_data_quality_results	Returns all data quality execution results for your account
list_data_quality_rule_recommendation_runs	Lists the recommendation runs meeting the filter criteria
list_data_quality_ruleset_evaluation_runs	Lists all the runs meeting the filter criteria, where a ruleset is evaluated agai
list_data_quality_rulesets	Returns a paginated list of rulesets for the specified list of Glue tables
list_data_quality_statistic_annotations	Retrieve annotations for a data quality statistic
list_data_quality_statistics	Retrieves a list of data quality statistics
list_dev_endpoints	Retrieves the names of all DevEndpoint resources in this Amazon Web Serv
list_entities	Returns the available entities supported by the connection type
list_integration_resource_properties	List integration resource properties for a single customer
list_jobs	Retrieves the names of all job resources in this Amazon Web Services accou
list_materialized_view_refresh_task_runs	List all task runs for a particular account
list_ml_transforms	Retrieves a sortable, filterable list of existing Glue machine learning transfo
list_registries	Returns a list of registries that you have created, with minimal registry infor
list_schemas	Returns a list of schemas with minimal details
list_schema_versions	Returns a list of schema versions that you have created, with minimal inform
list_sessions	Retrieve a list of sessions

list_statements	Lists statements for the session
list_table_optimizer_runs	Lists the history of previous optimizer runs for a specific table
list_triggers	Retrieves the names of all trigger resources in this Amazon Web Services ac
list_usage_profiles	List all the Glue usage profiles
list_workflows	Lists names of workflows created in the account
modify_integration	Modifies a Zero-ETL integration in the caller's account
put_data_catalog_encryption_settings	Sets the security configuration for a specified catalog
put_data_quality_profile_annotation	Annotate all datapoints for a Profile
put_resource_policy	Sets the Data Catalog resource policy for access control
put_schema_version_metadata	Puts the metadata key value pair for a specified schema version ID
put_workflow_run_properties	Puts the specified workflow run properties for the given workflow run
query_schema_version_metadata	Queries for the schema version metadata information
register_connection_type	Registers a custom connection type in Glue based on the configuration prov
register_schema_version	Adds a new version to the existing schema
remove_schema_version_metadata	Removes a key value pair from the schema version metadata for the specifie
reset_job_bookmark	Resets a bookmark entry
resume_workflow_run	Restarts selected nodes of a previous partially completed workflow run and
run_statement	Executes the statement
search_tables	Searches a set of tables based on properties in the table metadata as well as
start_blueprint_run	Starts a new run of the specified blueprint
start_column_statistics_task_run	Starts a column statistics task run, for a specified table and columns
start_column_statistics_task_run_schedule	Starts a column statistics task run schedule
start_crawler	Starts a crawl using the specified crawler, regardless of what is scheduled
start_crawler_schedule	Changes the schedule state of the specified crawler to SCHEDULED, unles
start_data_quality_rule_recommendation_run	Starts a recommendation run that is used to generate rules when you don't k
start_data_quality_ruleset_evaluation_run	Once you have a ruleset definition (either recommended or your own), you
start_export_labels_task_run	Begins an asynchronous task to export all labeled data for a particular transf
start_import_labels_task_run	Enables you to provide additional labels (examples of truth) to be used to te
start_job_run	Starts a job run using a job definition
start_materialized_view_refresh_task_run	Starts a materialized view refresh task run, for a specified table and columns
start_ml_evaluation_task_run	Starts a task to estimate the quality of the transform
start_ml_labeling_set_generation_task_run	Starts the active learning workflow for your machine learning transform to i
start_trigger	Starts an existing trigger
start_workflow_run	Starts a new run of the specified workflow
stop_column_statistics_task_run	Stops a task run for the specified table
stop_column_statistics_task_run_schedule	Stops a column statistics task run schedule
stop_crawler	If the specified crawler is running, stops the crawl
stop_crawler_schedule	Sets the schedule state of the specified crawler to NOT_SCHEDULED, but
stop_materialized_view_refresh_task_run	Stops a materialized view refresh task run, for a specified table and columns
stop_session	Stops the session
stop_trigger	Stops a specified trigger
stop_workflow_run	Stops the execution of the specified workflow run
tag_resource	Adds tags to a resource
test_connection	Tests a connection to a service to validate the service credentials that you pr
untag_resource	Removes tags from a resource
update_blueprint	Updates a registered blueprint
update_catalog	Updates an existing catalog's properties in the Glue Data Catalog
update_classifier	Modifies an existing classifier (a GrokClassifier, an XMLClassifier, a JsonC

update_column_statistics_for_partition	Creates or updates partition statistics of columns
update_column_statistics_for_table	Creates or updates table statistics of columns
update_column_statistics_task_settings	Updates settings for a column statistics task
update_connection	Updates a connection definition in the Data Catalog
update_crawler	Updates a crawler
update_crawler_schedule	Updates the schedule of a crawler using a cron expression
update_database	Updates an existing database definition in a Data Catalog
update_data_quality_ruleset	Updates the specified data quality ruleset
update_dev_endpoint	Updates a specified development endpoint
update_glue_identity_center_configuration	Updates the existing Glue Identity Center configuration, allowing modificat
update_integration_resource_property	This API can be used for updating the ResourceProperty of the Glue connec
update_integration_table_properties	This API is used to provide optional override properties for the tables that n
update_job	Updates an existing job definition
update_job_from_source_control	Synchronizes a job from the source control repository
update_ml_transform	Updates an existing machine learning transform
update_partition	Updates a partition
update_registry	Updates an existing registry which is used to hold a collection of schemas
update_schema	Updates the description, compatibility setting, or version checkpoint for a s
update_source_control_from_job	Synchronizes a job to the source control repository
update_table	Updates a metadata table in the Data Catalog
update_table_optimizer	Updates the configuration for an existing table optimizer
update_trigger	Updates a trigger definition
update_usage_profile	Update an Glue usage profile
update_user_defined_function	Updates an existing function definition in the Data Catalog
update_workflow	Updates an existing workflow

Examples

```
## Not run:
svc <- glue()
svc$batch_create_partition(
  Foo = 123
)

## End(Not run)
```

`gluedatabrew` *AWS Glue DataBrew*

Description

Glue DataBrew is a visual, cloud-scale data-preparation service. DataBrew simplifies data preparation tasks, targeting data issues that are hard to spot and time-consuming to fix. DataBrew empowers users of all technical levels to visualize the data and perform one-click data transformations, with no coding required.

Usage

```
gluedatabrew(
  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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

batch_delete_recipe_version	Deletes one or more versions of a recipe at a time
create_dataset	Creates a new DataBrew dataset
create_profile_job	Creates a new job to analyze a dataset and create its data profile
create_project	Creates a new DataBrew project
create_recipe	Creates a new DataBrew recipe
create_recipe_job	Creates a new job to transform input data, using steps defined in an existing Glue DataBrew rec
create_ruleset	Creates a new ruleset that can be used in a profile job to validate the data quality of a dataset
create_schedule	Creates a new schedule for one or more DataBrew jobs
delete_dataset	Deletes a dataset from DataBrew
delete_job	Deletes the specified DataBrew job
delete_project	Deletes an existing DataBrew project
delete_recipe_version	Deletes a single version of a DataBrew recipe
delete_ruleset	Deletes a ruleset

delete_schedule	Deletes the specified DataBrew schedule
describe_dataset	Returns the definition of a specific DataBrew dataset
describe_job	Returns the definition of a specific DataBrew job
describe_job_run	Represents one run of a DataBrew job
describe_project	Returns the definition of a specific DataBrew project
describe_recipe	Returns the definition of a specific DataBrew recipe corresponding to a particular version
describe_ruleset	Retrieves detailed information about the ruleset
describe_schedule	Returns the definition of a specific DataBrew schedule
list_datasets	Lists all of the DataBrew datasets
list_job_runs	Lists all of the previous runs of a particular DataBrew job
list_jobs	Lists all of the DataBrew jobs that are defined
list_projects	Lists all of the DataBrew projects that are defined
list_recipes	Lists all of the DataBrew recipes that are defined
list_recipe_versions	Lists the versions of a particular DataBrew recipe, except for LATEST_WORKING
list_rulesets	List all rulesets available in the current account or rulesets associated with a specific resource (
list_schedules	Lists the DataBrew schedules that are defined
list_tags_for_resource	Lists all the tags for a DataBrew resource
publish_recipe	Publishes a new version of a DataBrew recipe
send_project_session_action	Performs a recipe step within an interactive DataBrew session that's currently open
start_job_run	Runs a DataBrew job
start_project_session	Creates an interactive session, enabling you to manipulate data in a DataBrew project
stop_job_run	Stops a particular run of a job
tag_resource	Adds metadata tags to a DataBrew resource, such as a dataset, project, recipe, job, or schedule
untag_resource	Removes metadata tags from a DataBrew resource
update_dataset	Modifies the definition of an existing DataBrew dataset
update_profile_job	Modifies the definition of an existing profile job
update_project	Modifies the definition of an existing DataBrew project
update_recipe	Modifies the definition of the LATEST_WORKING version of a DataBrew recipe
update_recipe_job	Modifies the definition of an existing DataBrew recipe job
update_ruleset	Updates specified ruleset
update_schedule	Modifies the definition of an existing DataBrew schedule

Examples

```
## Not run:
svc <- gluedatabrew()
svc$batch_delete_recipe_version(
  Foo = 123
)

## End(Not run)
```

`healthlake` *Amazon HealthLake*

Description

This is the *AWS HealthLake API Reference*. For an introduction to the service, see What is AWS HealthLake? in the *AWS HealthLake Developer Guide*.

Usage

```
healthlake(
  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-e html`

`credentials` Optional credentials shorthand for the config parameter

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

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

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

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

Value

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

Service syntax

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

Operations

create_fhir_datastore	Create a FHIR-enabled data store
delete_fhir_datastore	Delete a FHIR-enabled data store
describe_fhir_datastore	Get properties for a FHIR-enabled data store
describe_fhir_export_job	Get FHIR export job properties
describe_fhir_import_job	Get the import job properties to learn more about the job or job progress
list_fhir_datastores	List all FHIR-enabled data stores in a user's account, regardless of data store status
list_fhir_export_jobs	Lists all FHIR export jobs associated with an account and their statuses
list_fhir_import_jobs	List all FHIR import jobs associated with an account and their statuses
list_tags_for_resource	Returns a list of all existing tags associated with a data store
start_fhir_export_job	Start a FHIR export job
start_fhir_import_job	Start importing bulk FHIR data into an ACTIVE data store
tag_resource	Add a user-specifed key and value tag to a data store
untag_resource	Remove a user-specifed key and value tag from a data store

Examples

```
## Not run:
svc <- healthlake()
svc$create_fhir_datastore(
  Foo = 123
)

## End(Not run)
```

`ivs` *Amazon Interactive Video Service*

Description

Introduction

The Amazon Interactive Video Service (IVS) API is REST compatible, using a standard HTTP API and an Amazon Web Services EventBridge event stream for responses. JSON is used for both requests and responses, including errors.

The API is an Amazon Web Services regional service. For a list of supported regions and Amazon IVS HTTPS service endpoints, see the Amazon IVS page in the *Amazon Web Services General Reference*.

*All API request parameters and URLs are case sensitive. *

For a summary of notable documentation changes in each release, see Document History.

Allowed Header Values

- `Accept:` application/json
- `Accept-Encoding:` gzip, deflate
- `Content-Type:`application/json

Key Concepts

- **Channel** — Stores configuration data related to your live stream. You first create a channel and then use the channel's stream key to start your live stream.
- **Stream key** — An identifier assigned by Amazon IVS when you create a channel, which is then used to authorize streaming. *Treat the stream key like a secret, since it allows anyone to stream to the channel.*
- **Playback key pair** — Video playback may be restricted using playback-authorization tokens, which use public-key encryption. A playback key pair is the public-private pair of keys used to sign and validate the playback-authorization token.
- **Recording configuration** — Stores configuration related to recording a live stream and where to store the recorded content. Multiple channels can reference the same recording configuration.
- **Playback restriction policy** — Restricts playback by countries and/or origin sites.

For more information about your IVS live stream, also see Getting Started with IVS Low-Latency Streaming.

Tagging

A *tag* is a metadata label that you assign to an Amazon Web Services resource. A tag comprises a *key* and a *value*, both set by you. For example, you might set a tag as `topic:nature` to label a particular video category. See Best practices and strategies in *Tagging Amazon Web Services Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS has no service-specific constraints beyond what is documented there.

Tags can help you identify and organize your Amazon Web Services resources. For example, you can use the same tag for different resources to indicate that they are related. You can also use tags to manage access (see Access Tags).

The Amazon IVS API has these tag-related operations: `tag_resource`, `untag_resource`, and `list_tags_for_resource`. The following resources support tagging: Channels, Stream Keys, Playback Key Pairs, and Recording Configurations.

At most 50 tags can be applied to a resource.

Authentication versus Authorization

Note the differences between these concepts:

- *Authentication* is about verifying identity. You need to be authenticated to sign Amazon IVS API requests.
- *Authorization* is about granting permissions. Your IAM roles need to have permissions for Amazon IVS API requests. In addition, authorization is needed to view Amazon IVS private channels. (Private channels are channels that are enabled for "playback authorization.")

Authentication

All Amazon IVS API requests must be authenticated with a signature. The Amazon Web Services Command-Line Interface (CLI) and Amazon IVS Player SDKs take care of signing the underlying API calls for you. However, if your application calls the Amazon IVS API directly, it's your responsibility to sign the requests.

You generate a signature using valid Amazon Web Services credentials that have permission to perform the requested action. For example, you must sign PutMetadata requests with a signature generated from a user account that has the `ivs:PutMetadata` permission.

For more information:

- Authentication and generating signatures — See Authenticating Requests (Amazon Web Services Signature Version 4) in the *Amazon Web Services General Reference*.
- Managing Amazon IVS permissions — See Identity and Access Management on the Security page of the *Amazon IVS User Guide*.

Amazon Resource Names (ARNs)

ARNs uniquely identify AWS resources. An ARN is required when you need to specify a resource unambiguously across all of AWS, such as in IAM policies and API calls. For more information, see Amazon Resource Names in the *AWS General Reference*.

Usage

```
ivs(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-e html`

`credentials` Optional credentials shorthand for the config parameter

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

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

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

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

Value

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

Service syntax

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

Operations

batch_get_channel	Performs GetChannel on multiple ARNs simultaneously
batch_get_stream_key	Performs GetStreamKey on multiple ARNs simultaneously
batch_start_viewer_session_revocation	Performs StartViewerSessionRevocation on multiple channel ARN and viewer ID pa
create_ad_configuration	Creates a new ad configuration to be used for server-side ad insertion
create_channel	Creates a new channel and an associated stream key to start streaming
create_playback_restriction_policy	Creates a new playback restriction policy, for constraining playback by countries and
create_recording_configuration	Creates a new recording configuration, used to enable recording to Amazon S3
create_stream_key	Creates a stream key, used to initiate a stream, for the specified channel ARN
delete_ad_configuration	Deletes the specified ad configuration
delete_channel	Deletes the specified channel and its associated stream keys
delete_playback_key_pair	Deletes a specified authorization key pair
delete_playback_restriction_policy	Deletes the specified playback restriction policy
delete_recording_configuration	Deletes the recording configuration for the specified ARN
delete_stream_key	Deletes the stream key for the specified ARN, so it can no longer be used to stream
get_ad_configuration	Gets the ad configuration represented by the specified ARN
get_channel	Gets the channel configuration for the specified channel ARN
get_playback_key_pair	Gets a specified playback authorization key pair and returns the arn and fingerprint
get_playback_restriction_policy	Gets the specified playback restriction policy
get_recording_configuration	Gets the recording configuration for the specified ARN
get_stream	Gets information about the active (live) stream on a specified channel
get_stream_key	Gets stream-key information for a specified ARN
get_stream_session	Gets metadata on a specified stream
import_playback_key_pair	Imports the public portion of a new key pair and returns its arn and fingerprint
insert_ad_break	Inserts an ad marker in the playlist for the specified channel and duration using the a
list_ad_configurations	Gets summary information about all ad configurations in your account, in the AWS r
list_channels	Gets summary information about all channels in your account, in the Amazon Web S
list_playback_key_pairs	Gets summary information about playback key pairs
list_playback_restriction_policies	Gets summary information about playback restriction policies
list_recording_configurations	Gets summary information about all recording configurations in your account, in the
list_stream_keys	Gets summary information about stream keys for the specified channel
list_streams	Gets summary information about live streams in your account, in the Amazon Web S
list_stream_sessions	Gets a summary of current and previous streams for a specified channel in your acco
list_tags_for_resource	Gets information about Amazon Web Services tags for the specified ARN
put_metadata	Inserts metadata into the active stream of the specified channel
start_viewer_session_revocation	Starts the process of revoking the viewer session associated with a specified channel
stop_stream	Disconnects the incoming RTMPS stream for the specified channel
tag_resource	Adds or updates tags for the Amazon Web Services resource with the specified ARN
untag_resource	Removes tags from the resource with the specified ARN
update_channel	Updates a channel's configuration
update_playback_restriction_policy	Updates a specified playback restriction policy

Examples

```
## Not run:
svc <- ivs()
svc$batch_get_channel(
  Foo = 123
```

```
)

## End(Not run)
```

`ivsrealtime` *Amazon Interactive Video Service RealTime*

Description

The Amazon Interactive Video Service (IVS) real-time API is REST compatible, using a standard HTTP API and an AWS EventBridge event stream for responses. JSON is used for both requests and responses, including errors.

Key Concepts

- **Stage** — A virtual space where participants can exchange video in real time.
- **Participant token** — A token that authenticates a participant when they join a stage.
- **Participant object** — Represents participants (people) in the stage and contains information about them. When a token is created, it includes a participant ID; when a participant uses that token to join a stage, the participant is associated with that participant ID. There is a 1:1 mapping between participant tokens and participants.

For server-side composition:

- **Composition process** — Composites participants of a stage into a single video and forwards it to a set of outputs (e.g., IVS channels). Composition operations support this process.
- **Composition** — Controls the look of the outputs, including how participants are positioned in the video.

For participant replication:

- **Source stage** — The stage where the participant originally joined, which is used as the source for replication.
- **Destination stage** — The stage to which the participant is replicated.
- **Replicated participant** — A participant in a stage that is replicated to one or more destination stages.
- **Replica participant** — A participant in a destination stage that is replicated from another stage (the source stage).

For more information about your IVS live stream, also see Getting Started with Amazon IVS Real-Time Streaming.

Tagging

A *tag* is a metadata label that you assign to an AWS resource. A tag comprises a *key* and a *value*, both set by you. For example, you might set a tag as `topic:nature` to label a particular video category. See Best practices and strategies in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS stages has no service-specific constraints beyond what is documented there.

Tags can help you identify and organize your AWS resources. For example, you can use the same tag for different resources to indicate that they are related. You can also use tags to manage access (see Access Tags).

The Amazon IVS real-time API has these tag-related operations: tag_resource, untag_resource, and list_tags_for_resource. The following resource supports tagging: Stage.

At most 50 tags can be applied to a resource.

Usage

```
ivsrealtime(
  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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

Operations

create_encoder_configuration	Creates an EncoderConfiguration object
create_ingest_configuration	Creates a new IngestConfiguration resource, used to specify the ingest protocol for a stage
create_participant_token	Creates an additional token for a specified stage
create_stage	Creates a new stage (and optionally participant tokens)

create_storage_configuration	Creates a new storage configuration, used to enable recording to Amazon S3
delete_encoder_configuration	Deletes an EncoderConfiguration resource
delete_ingest_configuration	Deletes a specified IngestConfiguration, so it can no longer be used to broadcast
delete_public_key	Deletes the specified public key used to sign stage participant tokens
delete_stage	Shuts down and deletes the specified stage (disconnecting all participants)
delete_storage_configuration	Deletes the storage configuration for the specified ARN
disconnect_participant	Disconnects a specified participant from a specified stage
get_composition	Get information about the specified Composition resource
get_encoder_configuration	Gets information about the specified EncoderConfiguration resource
get_ingest_configuration	Gets information about the specified IngestConfiguration
get_participant	Gets information about the specified participant token
get_public_key	Gets information for the specified public key
get_stage	Gets information for the specified stage
get_stage_session	Gets information for the specified stage session
get_storage_configuration	Gets the storage configuration for the specified ARN
import_public_key	Import a public key to be used for signing stage participant tokens
list_compositions	Gets summary information about all Compositions in your account, in the AWS region where
list_encoder_configurations	Gets summary information about all EncoderConfigurations in your account, in the AWS regi
list_ingest_configurations	Lists all IngestConfigurations in your account, in the AWS region where the API request is pr
list_participant_events	Lists events for a specified participant that occurred during a specified stage session
list_participant_replicas	Lists all the replicas for a participant from a source stage
list_participants	Lists all participants in a specified stage session
list_public_keys	Gets summary information about all public keys in your account, in the AWS region where th
list_stages	Gets summary information about all stages in your account, in the AWS region where the API
list_stage_sessions	Gets all sessions for a specified stage
list_storage_configurations	Gets summary information about all storage configurations in your account, in the AWS regio
list_tags_for_resource	Gets information about AWS tags for the specified ARN
start_composition	Starts a Composition from a stage based on the configuration provided in the request
start_participant_replication	Starts replicating a publishing participant from a source stage to a destination stage
stop_composition	Stops and deletes a Composition resource
stop_participant_replication	Stops a replicated participant session
tag_resource	Adds or updates tags for the AWS resource with the specified ARN
untag_resource	Removes tags from the resource with the specified ARN
update_ingest_configuration	Updates a specified IngestConfiguration
update_stage	Updates a stage's configuration

Examples

```
## Not run:
svc <- ivsrealtime()
svc$create_encoder_configuration(
  Foo = 123
)

## End(Not run)
```

kafka *Managed Streaming for Kafka*

Description

The operations for managing an Amazon MSK cluster.

Usage

```
kafka(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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

batch_associate_scram_secret	Associates one or more Scram Secrets with an Amazon MSK cluster
batch_disassociate_scram_secret	Disassociates one or more Scram Secrets from an Amazon MSK cluster
create_cluster	Creates a new MSK cluster
create_cluster_v2	Creates a new MSK cluster
create_configuration	Creates a new MSK configuration
create_replicator	Creates the replicator
create_topic	Creates a topic in the specified MSK cluster
create_vpc_connection	Creates a new MSK VPC connection

delete_cluster	Deletes the MSK cluster specified by the Amazon Resource Name (ARN) in the request
delete_cluster_policy	Deletes the MSK cluster policy specified by the Amazon Resource Name (ARN) in the req
delete_configuration	Deletes an MSK Configuration
delete_replicator	Deletes a replicator
delete_topic	Deletes a topic in the specified MSK cluster
delete_vpc_connection	Deletes a MSK VPC connection
describe_cluster	Returns a description of the MSK cluster whose Amazon Resource Name (ARN) is specifi
describe_cluster_operation	Returns a description of the cluster operation specified by the ARN
describe_cluster_operation_v2	Returns a description of the cluster operation specified by the ARN
describe_cluster_v2	Returns a description of the MSK cluster whose Amazon Resource Name (ARN) is specifi
describe_configuration	Returns a description of this MSK configuration
describe_configuration_revision	Returns a description of this revision of the configuration
describe_replicator	Describes a replicator
describe_topic	Returns topic details of this topic on a MSK cluster
describe_topic_partitions	Returns partition details of this topic on a MSK cluster
describe_vpc_connection	Returns a description of this MSK VPC connection
get_bootstrap_brokers	A list of brokers that a client application can use to bootstrap
get_cluster_policy	Get the MSK cluster policy specified by the Amazon Resource Name (ARN) in the reques
get_compatible_kafka_versions	Gets the Apache Kafka versions to which you can update the MSK cluster
list_client_vpc_connections	Returns a list of all the VPC connections in this Region
list_cluster_operations	Returns a list of all the operations that have been performed on the specified MSK cluster
list_cluster_operations_v2	Returns a list of all the operations that have been performed on the specified MSK cluster
list_clusters	Returns a list of all the MSK clusters in the current Region
list_clusters_v2	Returns a list of all the MSK clusters in the current Region
list_configuration_revisions	Returns a list of all the MSK configurations in this Region
list_configurations	Returns a list of all the MSK configurations in this Region
list_kafka_versions	Returns a list of Apache Kafka versions
list_nodes	Returns a list of the broker nodes in the cluster
list_replicators	Lists the replicators
list_scram_secrets	Returns a list of the Scram Secrets associated with an Amazon MSK cluster
list_tags_for_resource	Returns a list of the tags associated with the specified resource
list_topics	List topics in a MSK cluster
list_vpc_connections	Returns a list of all the VPC connections in this Region
put_cluster_policy	Creates or updates the MSK cluster policy specified by the cluster Amazon Resource Nam
reboot_broker	Reboots brokers
reject_client_vpc_connection	Returns empty response
tag_resource	Adds tags to the specified MSK resource
untag_resource	Removes the tags associated with the keys that are provided in the query
update_broker_count	Updates the number of broker nodes in the cluster
update_broker_storage	Updates the EBS storage associated with MSK brokers
update_broker_type	Updates EC2 instance type
update_cluster_configuration	Updates the cluster with the configuration that is specified in the request body
update_cluster_kafka_version	Updates the Apache Kafka version for the cluster
update_configuration	Updates an MSK configuration
update_connectivity	Updates the cluster's connectivity configuration
update_monitoring	Updates the monitoring settings for the cluster
update_rebalancing	Use this resource to update the intelligent rebalancing status of an Amazon MSK Provisio
update_replication_info	Updates replication info of a replicator

update_security	Updates the security settings for the cluster
update_storage	Updates cluster broker volume size (or) sets cluster storage mode to TIERED
update_topic	Updates the configuration of the specified topic

Examples

```
## Not run:
svc <- kafka()
svc$batch_associate_scram_secret(
  Foo = 123
)

## End(Not run)
```

`kafkaconnect` *Managed Streaming for Kafka Connect*

Description

Managed Streaming for Kafka Connect

Usage

```
kafkaconnect(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

create_connector	Creates a connector using the specified properties
create_custom_plugin	Creates a custom plugin using the specified properties
create_worker_configuration	Creates a worker configuration using the specified properties
delete_connector	Deletes the specified connector
delete_custom_plugin	Deletes a custom plugin
delete_worker_configuration	Deletes the specified worker configuration
describe_connector	Returns summary information about the connector
describe_connector_operation	Returns information about the specified connector's operations
describe_custom_plugin	A summary description of the custom plugin
describe_worker_configuration	Returns information about a worker configuration
list_connector_operations	Lists information about a connector's operation(s)
list_connectors	Returns a list of all the connectors in this account and Region
list_custom_plugins	Returns a list of all of the custom plugins in this account and Region
list_tags_for_resource	Lists all the tags attached to the specified resource
list_worker_configurations	Returns a list of all of the worker configurations in this account and Region
tag_resource	Attaches tags to the specified resource
untag_resource	Removes tags from the specified resource
update_connector	Updates the specified connector

Examples

```
## Not run:
svc <- kafkaconnect()
svc$create_connector(
  Foo = 123
)

## End(Not run)
```

kendra *AWSKendraFrontendService*

Description

Amazon Kendra is a service for indexing large document sets.

Usage

```
kendra(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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

associate_entities_to_experience	Grants users or groups in your IAM Identity Center identity source access to your A
associate_personas_to_entities	Defines the specific permissions of users or groups in your IAM Identity Center iden
batch_delete_document	Removes one or more documents from an index
batch_delete_featured_results_set	Removes one or more sets of featured results
batch_get_document_status	Returns the indexing status for one or more documents submitted with the BatchPut
batch_put_document	Adds one or more documents to an index
clear_query_suggestions	Clears existing query suggestions from an index
create_access_control_configuration	Creates an access configuration for your documents

create_data_source	Creates a data source connector that you want to use with an Amazon Kendra index
create_experience	Creates an Amazon Kendra experience such as a search application
create_faq	Creates a set of frequently ask questions (FAQs) using a specified FAQ file stored in
create_featured_results_set	Creates a set of featured results to display at the top of the search results page
create_index	Creates an Amazon Kendra index
create_query_suggestions_block_list	Creates a block list to exlcude certain queries from suggestions
create_thesaurus	Creates a thesaurus for an index
delete_access_control_configuration	Deletes an access control configuration that you created for your documents in an in
delete_data_source	Deletes an Amazon Kendra data source connector
delete_experience	Deletes your Amazon Kendra experience such as a search application
delete_faq	Removes a FAQ from an index
delete_index	Deletes an Amazon Kendra index
delete_principal_mapping	Deletes a group so that all users that belong to the group can no longer access docun
delete_query_suggestions_block_list	Deletes a block list used for query suggestions for an index
delete_thesaurus	Deletes an Amazon Kendra thesaurus
describe_access_control_configuration	Gets information about an access control configuration that you created for your doc
describe_data_source	Gets information about an Amazon Kendra data source connector
describe_experience	Gets information about your Amazon Kendra experience such as a search applicatio
describe_faq	Gets information about a FAQ
describe_featured_results_set	Gets information about a set of featured results
describe_index	Gets information about an Amazon Kendra index
describe_principal_mapping	Describes the processing of PUT and DELETE actions for mapping users to their gr
describe_query_suggestions_block_list	Gets information about a block list used for query suggestions for an index
describe_query_suggestions_config	Gets information on the settings of query suggestions for an index
describe_thesaurus	Gets information about an Amazon Kendra thesaurus
disassociate_entities_from_experience	Prevents users or groups in your IAM Identity Center identity source from accessing
disassociate_personas_from_entities	Removes the specific permissions of users or groups in your IAM Identity Center id
get_query_suggestions	Fetches the queries that are suggested to your users
get_snapshots	Retrieves search metrics data
list_access_control_configurations	Lists one or more access control configurations for an index
list_data_sources	Lists the data source connectors that you have created
list_data_source_sync_jobs	Gets statistics about synchronizing a data source connector
list_entity_personas	Lists specific permissions of users and groups with access to your Amazon Kendra e
list_experience_entities	Lists users or groups in your IAM Identity Center identity source that are granted ac
list_experiences	Lists one or more Amazon Kendra experiences
list_faqs	Gets a list of FAQs associated with an index
list_featured_results_sets	Lists all your sets of featured results for a given index
list_groups_older_than_ordering_id	Provides a list of groups that are mapped to users before a given ordering or timesta
list_indices	Lists the Amazon Kendra indexes that you created
list_query_suggestions_block_lists	Lists the block lists used for query suggestions for an index
list_tags_for_resource	Gets a list of tags associated with a resource
list_thesauri	Lists the thesauri for an index
put_principal_mapping	Maps users to their groups so that you only need to provide the user ID when you is
query	Searches an index given an input query
retrieve	Retrieves relevant passages or text excerpts given an input query
start_data_source_sync_job	Starts a synchronization job for a data source connector
stop_data_source_sync_job	Stops a synchronization job that is currently running
submit_feedback	Enables you to provide feedback to Amazon Kendra to improve the performance of

tag_resource	Adds the specified tag to the specified index, FAQ, data source, or other resource
untag_resource	Removes a tag from an index, FAQ, data source, or other resource
update_access_control_configuration	Updates an access control configuration for your documents in an index
update_data_source	Updates an Amazon Kendra data source connector
update_experience	Updates your Amazon Kendra experience such as a search application
update_featured_results_set	Updates a set of featured results
update_index	Updates an Amazon Kendra index
update_query_suggestions_block_list	Updates a block list used for query suggestions for an index
update_query_suggestions_config	Updates the settings of query suggestions for an index
update_thesaurus	Updates a thesaurus for an index

Examples

```
## Not run:
svc <- kendra()
svc$associate_entities_to_experience(
  Foo = 123
)

## End(Not run)
```

`kendraranking` *Amazon Kendra Intelligent Ranking*

Description

Amazon Kendra Intelligent Ranking uses Amazon Kendra semantic search capabilities to intelligently re-rank a search service's results.

Usage

```
kendraranking(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

create_rescore_execution_plan	Creates a rescore execution plan
delete_rescore_execution_plan	Deletes a rescore execution plan
describe_rescore_execution_plan	Gets information about a rescore execution plan
list_rescore_execution_plans	Lists your rescore execution plans
list_tags_for_resource	Gets a list of tags associated with a specified resource
rescore	Rescores or re-ranks search results from a search service such as OpenSearch (self manag
tag_resource	Adds a specified tag to a specified rescore execution plan
untag_resource	Removes a tag from a rescore execution plan
update_rescore_execution_plan	Updates a rescore execution plan

Examples

```
## Not run:
svc <- kendraranking()
svc$create_rescore_execution_plan(
  Foo = 123
)

## End(Not run)
```

`kinesis` *Amazon Kinesis*

Description

Amazon Kinesis Data Streams Service API Reference

Amazon Kinesis Data Streams is a managed service that scales elastically for real-time processing of streaming big data.

Usage

```
kinesis(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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

add_tags_to_stream	Adds or updates tags for the specified Kinesis data stream
create_stream	Creates a Kinesis data stream
decrease_stream_retention_period	Decreases the Kinesis data stream's retention period, which is the length of time data rec
delete_resource_policy	Delete a policy for the specified data stream or consumer
delete_stream	Deletes a Kinesis data stream and all its shards and data
deregister_stream_consumer	To deregister a consumer, provide its ARN
describe_account_settings	Describes the account-level settings for Amazon Kinesis Data Streams
describe_limits	Describes the shard limits and usage for the account

describe_stream	Describes the specified Kinesis data stream
describe_stream_consumer	To get the description of a registered consumer, provide the ARN of the consumer
describe_stream_summary	Provides a summarized description of the specified Kinesis data stream without the shard
disable_enhanced_monitoring	Disables enhanced monitoring
enable_enhanced_monitoring	Enables enhanced Kinesis data stream monitoring for shard-level metrics
get_records	Gets data records from a Kinesis data stream's shard
get_resource_policy	Returns a policy attached to the specified data stream or consumer
get_shard_iterator	Gets an Amazon Kinesis shard iterator
increase_stream_retention_period	Increases the Kinesis data stream's retention period, which is the length of time data reco
list_shards	Lists the shards in a stream and provides information about each shard
list_stream_consumers	Lists the consumers registered to receive data from a stream using enhanced fan-out, and
list_streams	Lists your Kinesis data streams
list_tags_for_resource	List all tags added to the specified Kinesis resource
list_tags_for_stream	Lists the tags for the specified Kinesis data stream
merge_shards	Merges two adjacent shards in a Kinesis data stream and combines them into a single sha
put_record	Writes a single data record into an Amazon Kinesis data stream
put_records	Writes multiple data records into a Kinesis data stream in a single call (also referred to as
put_resource_policy	Attaches a resource-based policy to a data stream or registered consumer
register_stream_consumer	Registers a consumer with a Kinesis data stream
remove_tags_from_stream	Removes tags from the specified Kinesis data stream
split_shard	Splits a shard into two new shards in the Kinesis data stream, to increase the stream's cap
start_stream_encryption	Enables or updates server-side encryption using an Amazon Web Services KMS key for a
stop_stream_encryption	Disables server-side encryption for a specified stream
subscribe_to_shard	This operation establishes an HTTP/2 connection between the consumer you specify in th
tag_resource	Adds or updates tags for the specified Kinesis resource
untag_resource	Removes tags from the specified Kinesis resource
update_account_settings	Updates the account-level settings for Amazon Kinesis Data Streams
update_max_record_size	This allows you to update the MaxRecordSize of a single record that you can write to, an
update_shard_count	Updates the shard count of the specified stream to the specified number of shards
update_stream_mode	Updates the capacity mode of the data stream
update_stream_warm_throughput	Updates the warm throughput configuration for the specified Amazon Kinesis Data Strea

Examples

```
## Not run:
svc <- kinesis()
svc$add_tags_to_stream(
  Foo = 123
)

## End(Not run)
```

`kinesisanalytics` *Amazon Kinesis Analytics*

Description

Overview

This documentation is for version 1 of the Amazon Kinesis Data Analytics API, which only supports SQL applications. Version 2 of the API supports SQL and Java applications. For more information about version 2, see Amazon Kinesis Data Analytics API V2 Documentation.

This is the *Amazon Kinesis Analytics v1 API Reference*. The Amazon Kinesis Analytics Developer Guide provides additional information.

Usage

```
kinesisanalytics(
  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-e
  html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:

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

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

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

Value

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

Service syntax

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

Operations

add_application_cloud_watch_logging_option	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_input	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_input_processing_configuration	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_output	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_reference_data_source	This documentation is for version 1 of the Amazon Kinesis Data Analyt
create_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_cloud_watch_logging_option	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_input_processing_configuration	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_output	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_reference_data_source	This documentation is for version 1 of the Amazon Kinesis Data Analyt
describe_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
discover_input_schema	This documentation is for version 1 of the Amazon Kinesis Data Analyt
list_applications	This documentation is for version 1 of the Amazon Kinesis Data Analyt
list_tags_for_resource	Retrieves the list of key-value tags assigned to the application
start_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
stop_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
tag_resource	Adds one or more key-value tags to a Kinesis Analytics application
untag_resource	Removes one or more tags from a Kinesis Analytics application
update_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt

Examples

```
## Not run:
svc <- kinesisanalytics()
svc$add_application_cloud_watch_logging_option(
  Foo = 123
)

## End(Not run)
```

`kinesisanalyticsv2` *Amazon Kinesis Analytics*

Description

Amazon Managed Service for Apache Flink was previously known as Amazon Kinesis Data Analytics for Apache Flink.

Amazon Managed Service for Apache Flink is a fully managed service that you can use to process and analyze streaming data using Java, Python, SQL, or Scala. The service enables you to quickly author and run Java, SQL, or Scala code against streaming sources to perform time series analytics, feed real-time dashboards, and create real-time metrics.

Usage

```
kinesisanalyticsv2(
  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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

```
svc <- kinesisanalyticsv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

add_application_cloud_watch_logging_option	Adds an Amazon CloudWatch log stream to monitor application configu
add_application_input	Adds a streaming source to your SQL-based Kinesis Data Analytics app
add_application_input_processing_configuration	Adds an InputProcessingConfiguration to a SQL-based Kinesis Data An
add_application_output	Adds an external destination to your SQL-based Kinesis Data Analytics
add_application_reference_data_source	Adds a reference data source to an existing SQL-based Kinesis Data An
add_application_vpc_configuration	Adds a Virtual Private Cloud (VPC) configuration to the application
create_application	Creates a Managed Service for Apache Flink application
create_application_presigned_url	Creates and returns a URL that you can use to connect to an application'
create_application_snapshot	Creates a snapshot of the application's state data
delete_application	Deletes the specified application
delete_application_cloud_watch_logging_option	Deletes an Amazon CloudWatch log stream from an SQL-based Kinesis
delete_application_input_processing_configuration	Deletes an InputProcessingConfiguration from an input
delete_application_output	Deletes the output destination configuration from your SQL-based Kines

delete_application_reference_data_source	Deletes a reference data source configuration from the specified SQL-ba
delete_application_snapshot	Deletes a snapshot of application state
delete_application_vpc_configuration	Removes a VPC configuration from a Managed Service for Apache Flin
describe_application	Returns information about a specific Managed Service for Apache Flink
describe_application_operation	Provides a detailed description of a specified application operation
describe_application_snapshot	Returns information about a snapshot of application state data
describe_application_version	Provides a detailed description of a specified version of the application
discover_input_schema	Infers a schema for a SQL-based Kinesis Data Analytics application by
list_application_operations	Lists all the operations performed for the specified application such as U
list_applications	Returns a list of Managed Service for Apache Flink applications in your
list_application_snapshots	Lists information about the current application snapshots
list_application_versions	Lists all the versions for the specified application, including versions tha
list_tags_for_resource	Retrieves the list of key-value tags assigned to the application
rollback_application	Reverts the application to the previous running version
start_application	Starts the specified Managed Service for Apache Flink application
stop_application	Stops the application from processing data
tag_resource	Adds one or more key-value tags to a Managed Service for Apache Flink
untag_resource	Removes one or more tags from a Managed Service for Apache Flink ap
update_application	Updates an existing Managed Service for Apache Flink application
update_application_maintenance_configuration	Updates the maintenance configuration of the Managed Service for Apa

Examples

```
## Not run:
svc <- kinesisanalyticsv2()
svc$add_application_cloud_watch_logging_option(
  Foo = 123
)

## End(Not run)
```

---

mturk *Amazon Mechanical Turk*

Description

Amazon Mechanical Turk API Reference

Usage

```
mturk(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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

accept_qualification_request	The AcceptQualificationRequest operation approves a Worker's request for a Quali
approve_assignment	The ApproveAssignment operation approves the results of a completed assignment
associate_qualification_with_worker	The AssociateQualificationWithWorker operation gives a Worker a Qualification
create_additional_assignments_for_hit	The CreateAdditionalAssignmentsForHIT operation increases the maximum numbe
create_hit	The CreateHIT operation creates a new Human Intelligence Task (HIT)
create_hit_type	The CreateHITType operation creates a new HIT type
create_hit_with_hit_type	The CreateHITWithHITType operation creates a new Human Intelligence Task (HI
create_qualification_type	The CreateQualificationType operation creates a new Qualification type, which is r
create_worker_block	The CreateWorkerBlock operation allows you to prevent a Worker from working or
delete_hit	The DeleteHIT operation is used to delete HIT that is no longer needed
delete_qualification_type	The DeleteQualificationType deletes a Qualification type and deletes any HIT types
delete_worker_block	The DeleteWorkerBlock operation allows you to reinstate a blocked Worker to worl
disassociate_qualification_from_worker	The DisassociateQualificationFromWorker revokes a previously granted Qualificati
get_account_balance	The GetAccountBalance operation retrieves the Prepaid HITs balance in your Ama
get_assignment	The GetAssignment operation retrieves the details of the specified Assignment
get_file_upload_url	The GetFileUploadURL operation generates and returns a temporary URL
get_hit	The GetHIT operation retrieves the details of the specified HIT
get_qualification_score	The GetQualificationScore operation returns the value of a Worker's Qualification f
get_qualification_type	The GetQualificationTypeoperation retrieves information about a Qualification type
list_assignments_for_hit	The ListAssignmentsForHIT operation retrieves completed assignments for a HIT

list_bonus_payments	The ListBonusPayments operation retrieves the amounts of bonuses you have paid
list_hi_ts	The ListHITs operation returns all of a Requester's HITs
list_hi_ts_for_qualification_type	The ListHITsForQualificationType operation returns the HITs that use the given Qu
list_qualification_requests	The ListQualificationRequests operation retrieves requests for Qualifications of a p
list_qualification_types	The ListQualificationTypes operation returns a list of Qualification types, filtered b
list_reviewable_hi_ts	The ListReviewableHITs operation retrieves the HITs with Status equal to Reviewa
list_review_policy_results_for_hit	The ListReviewPolicyResultsForHIT operation retrieves the computed results and t
list_worker_blocks	The ListWorkersBlocks operation retrieves a list of Workers who are blocked from
list_workers_with_qualification_type	The ListWorkersWithQualificationType operation returns all of the Workers that ha
notify_workers	The NotifyWorkers operation sends an email to one or more Workers that you speci
reject_assignment	The RejectAssignment operation rejects the results of a completed assignment
reject_qualification_request	The RejectQualificationRequest operation rejects a user's request for a Qualificatio
send_bonus	The SendBonus operation issues a payment of money from your account to a Work
send_test_event_notification	The SendTestEventNotification operation causes Amazon Mechanical Turk to send
update_expiration_for_hit	The UpdateExpirationForHIT operation allows you update the expiration time of a
update_hit_review_status	The UpdateHITReviewStatus operation updates the status of a HIT
update_hit_type_of_hit	The UpdateHITTypeOfHIT operation allows you to change the HITType properties
update_notification_settings	The UpdateNotificationSettings operation creates, updates, disables or re-enables n
update_qualification_type	The UpdateQualificationType operation modifies the attributes of an existing Quali

Examples

```
## Not run:
svc <- mturk()
svc$accept_qualification_request(
  Foo = 123
)

## End(Not run)
```

`opensearchingestion` *Amazon OpenSearch Ingestion*

Description

Use the Amazon OpenSearch Ingestion API to create and manage ingestion pipelines. OpenSearch Ingestion is a fully managed data collector that delivers real-time log and trace data to OpenSearch Service domains. For more information, see Getting data into your cluster using OpenSearch Ingestion.

Usage

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

Value

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

Service syntax

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

Operations

create_pipeline	Creates an OpenSearch Ingestion pipeline
create_pipeline_endpoint	Creates a VPC endpoint for an OpenSearch Ingestion pipeline
delete_pipeline	Deletes an OpenSearch Ingestion pipeline
delete_pipeline_endpoint	Deletes a VPC endpoint for an OpenSearch Ingestion pipeline
delete_resource_policy	Deletes a resource-based policy from an OpenSearch Ingestion resource
get_pipeline	Retrieves information about an OpenSearch Ingestion pipeline
get_pipeline_blueprint	Retrieves information about a specific blueprint for OpenSearch Ingestion
get_pipeline_change_progress	Returns progress information for the current change happening on an OpenSearch Ir
get_resource_policy	Retrieves the resource-based policy attached to an OpenSearch Ingestion resource
list_pipeline_blueprints	Retrieves a list of all available blueprints for Data Prepper
list_pipeline_endpoint_connections	Lists the pipeline endpoints connected to pipelines in your account
list_pipeline_endpoints	Lists all pipeline endpoints in your account
list_pipelines	Lists all OpenSearch Ingestion pipelines in the current Amazon Web Services accou

list_tags_for_resource	Lists all resource tags associated with an OpenSearch Ingestion pipeline
put_resource_policy	Attaches a resource-based policy to an OpenSearch Ingestion resource
revoke_pipeline_endpoint_connections	Revokes pipeline endpoints from specified endpoint IDs
start_pipeline	Starts an OpenSearch Ingestion pipeline
stop_pipeline	Stops an OpenSearch Ingestion pipeline
tag_resource	Tags an OpenSearch Ingestion pipeline
untag_resource	Removes one or more tags from an OpenSearch Ingestion pipeline
update_pipeline	Updates an OpenSearch Ingestion pipeline
validate_pipeline	Checks whether an OpenSearch Ingestion pipeline configuration is valid prior to cre

Examples

```
## Not run:
svc <- opensearchingestion()
svc$create_pipeline(
  Foo = 123
)

## End(Not run)
```

`opensearchservice` *Amazon OpenSearch Service*

Description

Use the Amazon OpenSearch Service configuration API to create, configure, and manage OpenSearch Service domains. The endpoint for configuration service requests is Region specific: es.*region*.amazonaws.com. For example, es.us-east-1.amazonaws.com. For a current list of supported Regions and endpoints, see Amazon Web Services service endpoints.

Usage

```
opensearchservice(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

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

Operations

accept_inbound_connection	Allows the destination Amazon OpenSearch Service domain owner to accept an inbou
add_data_source	Creates a new direct-query data source to the specified domain
add_direct_query_data_source	Adds a new data source in Amazon OpenSearch Service so that you can perform direc
add_tags	Attaches tags to an existing Amazon OpenSearch Service domain, data source, or app
associate_package	Associates a package with an Amazon OpenSearch Service domain
associate_packages	Operation in the Amazon OpenSearch Service API for associating multiple packages
authorize_vpc_endpoint_access	Provides access to an Amazon OpenSearch Service domain through the use of an inte
cancel_domain_config_change	Cancels a pending configuration change on an Amazon OpenSearch Service domain
cancel_service_software_update	Cancels a scheduled service software update for an Amazon OpenSearch Service dom
create_application	Creates an OpenSearch UI application
create_domain	Creates an Amazon OpenSearch Service domain
create_index	Creates an OpenSearch index with optional automatic semantic enrichment for specifi
create_outbound_connection	Creates a new cross-cluster search connection from a source Amazon OpenSearch Ser
create_package	Creates a package for use with Amazon OpenSearch Service domains
create_vpc_endpoint	Creates an Amazon OpenSearch Service-managed VPC endpoint
delete_application	Deletes a specified OpenSearch application
delete_data_source	Deletes a direct-query data source
delete_direct_query_data_source	Deletes a previously configured direct query data source from Amazon OpenSearch S
delete_domain	Deletes an Amazon OpenSearch Service domain and all of its data
delete_inbound_connection	Allows the destination Amazon OpenSearch Service domain owner to delete an existi
delete_index	Deletes an OpenSearch index
delete_outbound_connection	Allows the source Amazon OpenSearch Service domain owner to delete an existing o
delete_package	Deletes an Amazon OpenSearch Service package
delete_vpc_endpoint	Deletes an Amazon OpenSearch Service-managed interface VPC endpoint
deregister_capability	Deregisters a capability from an OpenSearch UI application

describe_domain	Describes the domain configuration for the specified Amazon OpenSearch Service do
describe_domain_auto_tunes	Returns the list of optimizations that Auto-Tune has made to an Amazon OpenSearch
describe_domain_change_progress	Returns information about the current blue/green deployment happening on an Amazo
describe_domain_config	Returns the configuration of an Amazon OpenSearch Service domain
describe_domain_health	Returns information about domain and node health, the standby Availability Zone, nu
describe_domain_nodes	Returns information about domain and nodes, including data nodes, master nodes, ulti
describe_domains	Returns domain configuration information about the specified Amazon OpenSearch S
describe_dry_run_progress	Describes the progress of a pre-update dry run analysis on an Amazon OpenSearch Se
describe_inbound_connections	Lists all the inbound cross-cluster search connections for a destination (remote) Amaz
describe_insight_details	Describes the details of an existing insight for an Amazon OpenSearch Service domai
describe_instance_type_limits	Describes the instance count, storage, and master node limits for a given OpenSearch
describe_outbound_connections	Lists all the outbound cross-cluster connections for a local (source) Amazon OpenSea
describe_packages	Describes all packages available to OpenSearch Service
describe_reserved_instance_offerings	Describes the available Amazon OpenSearch Service Reserved Instance offerings for
describe_reserved_instances	Describes the Amazon OpenSearch Service instances that you have reserved in a give
describe_vpc_endpoints	Describes one or more Amazon OpenSearch Service-managed VPC endpoints
dissociate_package	Removes a package from the specified Amazon OpenSearch Service domain
dissociate_packages	Dissociates multiple packages from a domain simultaneously
get_application	Retrieves the configuration and status of an existing OpenSearch application
get_capability	Retrieves information about a registered capability for an OpenSearch UI application,
get_compatible_versions	Returns a map of OpenSearch or Elasticsearch versions and the versions you can upgr
get_data_source	Retrieves information about a direct query data source
get_default_application_setting	Gets the ARN of the current default application
get_direct_query_data_source	Returns detailed configuration information for a specific direct query data source in A
get_domain_maintenance_status	The status of the maintenance action
get_index	Retrieves information about an OpenSearch index including its schema and semantic
get_package_version_history	Returns a list of Amazon OpenSearch Service package versions, along with their crea
get_upgrade_history	Retrieves the complete history of the last 10 upgrades performed on an Amazon Open
get_upgrade_status	Returns the most recent status of the last upgrade or upgrade eligibility check perform
list_applications	Lists all OpenSearch applications under your account
list_data_sources	Lists direct-query data sources for a specific domain
list_direct_query_data_sources	Lists an inventory of all the direct query data sources that you have configured within
list_domain_maintenances	A list of maintenance actions for the domain
list_domain_names	Returns the names of all Amazon OpenSearch Service domains owned by the current
list_domains_for_package	Lists all Amazon OpenSearch Service domains associated with a given package
list_insights	Lists insights for an Amazon OpenSearch Service domain or Amazon Web Services a
list_instance_type_details	Lists all instance types and available features for a given OpenSearch or Elasticsearch
list_packages_for_domain	Lists all packages associated with an Amazon OpenSearch Service domain
list_scheduled_actions	Retrieves a list of configuration changes that are scheduled for a domain
list_tags	Returns all resource tags for an Amazon OpenSearch Service domain, data source, or
list_versions	Lists all versions of OpenSearch and Elasticsearch that Amazon OpenSearch Service
list_vpc_endpoint_access	Retrieves information about each Amazon Web Services principal that is allowed to a
list_vpc_endpoints	Retrieves all Amazon OpenSearch Service-managed VPC endpoints in the current An
list_vpc_endpoints_for_domain	Retrieves all Amazon OpenSearch Service-managed VPC endpoints associated with a
purchase_reserved_instance_offering	Allows you to purchase Amazon OpenSearch Service Reserved Instances
put_default_application_setting	Sets the default application to the application with the specified ARN
register_capability	Registers a capability for an OpenSearch UI application
reject_inbound_connection	Allows the remote Amazon OpenSearch Service domain owner to reject an inbound c

remove_tags	Removes the specified set of tags from an Amazon OpenSearch Service domain, data
revoke_vpc_endpoint_access	Revokes access to an Amazon OpenSearch Service domain that was provided through
rollback_service_software_update	Rolls back a service software update for a domain to the previous version
start_domain_maintenance	Starts the node maintenance process on the data node
start_service_software_update	Schedules a service software update for an Amazon OpenSearch Service domain
update_application	Updates the configuration and settings of an existing OpenSearch application
update_data_source	Updates a direct-query data source
update_direct_query_data_source	Updates the configuration or properties of an existing direct query data source in Ama
update_domain_config	Modifies the cluster configuration of the specified Amazon OpenSearch Service doma
update_index	Updates an existing OpenSearch index schema and semantic enrichment configuratio
update_package	Updates a package for use with Amazon OpenSearch Service domains
update_package_scope	Updates the scope of a package
update_scheduled_action	Reschedules a planned domain configuration change for a later time
update_vpc_endpoint	Modifies an Amazon OpenSearch Service-managed interface VPC endpoint
upgrade_domain	Allows you to either upgrade your Amazon OpenSearch Service domain or perform a

Examples

```
## Not run:
svc <- opensearchservice()
svc$accept_inbound_connection(
  Foo = 123
)

## End(Not run)
```

`opensearchserviceserverless`

OpenSearch Service Serverless

Description

Use the Amazon OpenSearch Serverless API to create, configure, and manage OpenSearch Serverless collections and security policies.

OpenSearch Serverless is an on-demand, pre-provisioned serverless configuration for Amazon OpenSearch Service. OpenSearch Serverless removes the operational complexities of provisioning, configuring, and tuning your OpenSearch clusters. It enables you to easily search and analyze petabytes of data without having to worry about the underlying infrastructure and data management.

To learn more about OpenSearch Serverless, see What is Amazon OpenSearch Serverless?

Usage

```
opensearchserviceserverless(
  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-e
html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

Value

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

Service syntax

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

Operations

batch_get_collection	Returns attributes for one or more collections, including the collection endpoint, the O
batch_get_collection_group	Returns attributes for one or more collection groups, including capacity limits and the
batch_get_effective_lifecycle_policy	Returns a list of successful and failed retrievals for the OpenSearch Serverless indexes
batch_get_lifecycle_policy	Returns one or more configured OpenSearch Serverless lifecycle policies
batch_get_vpc_endpoint	Returns attributes for one or more VPC endpoints associated with the current account
create_access_policy	Creates a data access policy for OpenSearch Serverless
create_collection	Creates a new OpenSearch Serverless collection
create_collection_group	Creates a collection group within OpenSearch Serverless
create_index	Creates an index within an OpenSearch Serverless collection
create_lifecycle_policy	Creates a lifecyle policy to be applied to OpenSearch Serverless indexes
create_security_config	Specifies a security configuration for OpenSearch Serverless
create_security_policy	Creates a security policy to be used by one or more OpenSearch Serverless collections
create_vpc_endpoint	Creates an OpenSearch Serverless-managed interface VPC endpoint

delete_access_policy	Deletes an OpenSearch Serverless access policy
delete_collection	Deletes an OpenSearch Serverless collection
delete_collection_group	Deletes a collection group
delete_index	Deletes an index from an OpenSearch Serverless collection
delete_lifecycle_policy	Deletes an OpenSearch Serverless lifecycle policy
delete_security_config	Deletes a security configuration for OpenSearch Serverless
delete_security_policy	Deletes an OpenSearch Serverless security policy
delete_vpc_endpoint	Deletes an OpenSearch Serverless-managed interface endpoint
get_access_policy	Returns an OpenSearch Serverless access policy
get_account_settings	Returns account-level settings related to OpenSearch Serverless
get_index	Retrieves information about an index in an OpenSearch Serverless collection, includin
get_policies_stats	Returns statistical information about your OpenSearch Serverless access policies, secu
get_security_config	Returns information about an OpenSearch Serverless security configuration
get_security_policy	Returns information about a configured OpenSearch Serverless security policy
list_access_policies	Returns information about a list of OpenSearch Serverless access policies
list_collection_groups	Returns a list of collection groups
list_collections	Lists all OpenSearch Serverless collections
list_lifecycle_policies	Returns a list of OpenSearch Serverless lifecycle policies
list_security_configs	Returns information about configured OpenSearch Serverless security configurations
list_security_policies	Returns information about configured OpenSearch Serverless security policies
list_tags_for_resource	Returns the tags for an OpenSearch Serverless resource
list_vpc_endpoints	Returns the OpenSearch Serverless-managed interface VPC endpoints associated with
tag_resource	Associates tags with an OpenSearch Serverless resource
untag_resource	Removes a tag or set of tags from an OpenSearch Serverless resource
update_access_policy	Updates an OpenSearch Serverless access policy
update_account_settings	Update the OpenSearch Serverless settings for the current Amazon Web Services acco
update_collection	Updates an OpenSearch Serverless collection
update_collection_group	Updates the description and capacity limits of a collection group
update_index	Updates an existing index in an OpenSearch Serverless collection
update_lifecycle_policy	Updates an OpenSearch Serverless access policy
update_security_config	Updates a security configuration for OpenSearch Serverless
update_security_policy	Updates an OpenSearch Serverless security policy
update_vpc_endpoint	Updates an OpenSearch Serverless-managed interface endpoint

Examples

```
## Not run:
svc <- opensearchserviceserverless()
svc$batch_get_collection(
  Foo = 123
)

## End(Not run)
```

quicksight *Amazon QuickSight*

Description

Amazon Quick API Reference

Amazon Quick Sight is a fully managed, serverless business intelligence service for the Amazon Web Services Cloud that makes it easy to extend data and insights to every user in your organization. This API reference contains documentation for a programming interface that you can use to manage Amazon Quick Sight.

Usage

```
quicksight(
  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-e html

`credentials` Optional credentials shorthand for the config parameter

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

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

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

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

Value

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

Service syntax

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

Operations

batch_create_topic_reviewed_answer	Creates new reviewed answers for a Q Topic
batch_delete_topic_reviewed_answer	Deletes reviewed answers for Q Topic
cancel_ingestion	Cancels an ongoing ingestion of data into SPICE
create_account_customization	Creates Amazon Quick Sight customizations
create_account_subscription	Creates an Amazon Quick Sight account, or subscribes to Amazon
create_action_connector	Creates an action connector that enables Amazon Quick Sight to co
create_analysis	Creates an analysis in Amazon Quick Sight
create_brand	Creates an Quick Sight brand
create_custom_permissions	Creates a custom permissions profile
create_dashboard	Creates a dashboard from either a template or directly with a Dashb
create_data_set	Creates a dataset
create_data_source	Creates a data source
create_folder	Creates an empty shared folder
create_folder_membership	Adds an asset, such as a dashboard, analysis, or dataset into a folde
create_group	Use the CreateGroup operation to create a group in Quick Sight
create_group_membership	Adds an Amazon Quick Sight user to an Amazon Quick Sight grou
create_iam_policy_assignment	Creates an assignment with one specified IAM policy, identified by
create_ingestion	Creates and starts a new SPICE ingestion for a dataset
create_namespace	(Enterprise edition only) Creates a new namespace for you to use w
create_refresh_schedule	Creates a refresh schedule for a dataset
create_role_membership	Use CreateRoleMembership to add an existing Quick Sight group to
create_template	Creates a template either from a TemplateDefinition or from an exis
create_template_alias	Creates a template alias for a template
create_theme	Creates a theme
create_theme_alias	Creates a theme alias for a theme
create_topic	Creates a new Q topic
create_topic_refresh_schedule	Creates a topic refresh schedule
create_vpc_connection	Creates a new VPC connection
delete_account_customization	This API permanently deletes all Quick Sight customizations for the
delete_account_custom_permission	Unapplies a custom permissions profile from an account
delete_account_subscription	Deleting your Quick Sight account subscription has permanent, irre
delete_action_connector	Hard deletes an action connector, making it unrecoverable
delete_analysis	Deletes an analysis from Amazon Quick Sight
delete_brand	This API permanently deletes the specified Quick Sight brand
delete_brand_assignment	Deletes a brand assignment
delete_custom_permissions	Deletes a custom permissions profile
delete_dashboard	Deletes a dashboard
delete_data_set	Deletes a dataset
delete_data_set_refresh_properties	Deletes the dataset refresh properties of the dataset
delete_data_source	Deletes the data source permanently
delete_default_q_business_application	Deletes a linked Amazon Q Business application from an Quick Sig
delete_folder	Deletes an empty folder
delete_folder_membership	Removes an asset, such as a dashboard, analysis, or dataset, from a
delete_group	Removes a user group from Amazon Quick Sight
delete_group_membership	Removes a user from a group so that the user is no longer a membe
delete_iam_policy_assignment	Deletes an existing IAM policy assignment

delete_identity_propagation_config	Deletes all access scopes and authorized targets that are associated v
delete_namespace	Deletes a namespace and the users and groups that are associated w
delete_refresh_schedule	Deletes a refresh schedule from a dataset
delete_role_custom_permission	Removes custom permissions from the role
delete_role_membership	Removes a group from a role
delete_template	Deletes a template
delete_template_alias	Deletes the item that the specified template alias points to
delete_theme	Deletes a theme
delete_theme_alias	Deletes the version of the theme that the specified theme alias point
delete_topic	Deletes a topic
delete_topic_refresh_schedule	Deletes a topic refresh schedule
delete_user	Deletes the Amazon Quick Sight user that is associated with the ide
delete_user_by_principal_id	Deletes a user identified by its principal ID
delete_user_custom_permission	Deletes a custom permissions profile from a user
delete_vpc_connection	Deletes a VPC connection
describe_account_customization	Describes the customizations associated with the provided Amazon
describe_account_custom_permission	Describes the custom permissions profile that is applied to an accou
describe_account_settings	Describes the settings that were used when your Quick Sight subscr
describe_account_subscription	Use the DescribeAccountSubscription operation to receive a descrip
describe_action_connector	Retrieves detailed information about an action connector, including
describe_action_connector_permissions	Retrieves the permissions configuration for an action connector, sho
describe_analysis	Provides a summary of the metadata for an analysis
describe_analysis_definition	Provides a detailed description of the definition of an analysis
describe_analysis_permissions	Provides the read and write permissions for an analysis
describe_asset_bundle_export_job	Describes an existing export job
describe_asset_bundle_import_job	Describes an existing import job
describe_automation_job	Retrieves the status and details of a specified automation job, includ
describe_brand	Describes a brand
describe_brand_assignment	Describes a brand assignment
describe_brand_published_version	Describes the published version of the brand
describe_custom_permissions	Describes a custom permissions profile
describe_dashboard	Provides a summary for a dashboard
describe_dashboard_definition	Provides a detailed description of the definition of a dashboard
describe_dashboard_permissions	Describes read and write permissions for a dashboard
describe_dashboard_snapshot_job	Describes an existing snapshot job
describe_dashboard_snapshot_job_result	Describes the result of an existing snapshot job that has finished run
describe_dashboards_qa_configuration	Describes an existing dashboard QA configuration
describe_data_set	Describes a dataset
describe_data_set_permissions	Describes the permissions on a dataset
describe_data_set_refresh_properties	Describes the refresh properties of a dataset
describe_data_source	Describes a data source
describe_data_source_permissions	Describes the resource permissions for a data source
describe_default_q_business_application	Describes a Amazon Q Business application that is linked to an Qui
describe_folder	Describes a folder
describe_folder_permissions	Describes permissions for a folder
describe_folder_resolved_permissions	Describes the folder resolved permissions
describe_group	Returns an Amazon Quick Sight group's description and Amazon R
describe_group_membership	Use the DescribeGroupMembership operation to determine if a user

describe_iam_policy_assignment	Describes an existing IAM policy assignment, as specified by the as
describe_ingestion	Describes a SPICE ingestion
describe_ip_restriction	Provides a summary and status of IP rules
describe_key_registration	Describes all customer managed key registrations in a Quick Sight a
describe_namespace	Describes the current namespace
describe_q_personalization_configuration	Describes a personalization configuration
describe_quick_sight_q_search_configuration	Describes the state of a Quick Sight Q Search configuration
describe_refresh_schedule	Provides a summary of a refresh schedule
describe_role_custom_permission	Describes all custom permissions that are mapped to a role
describe_self_upgrade_configuration	Describes the self-upgrade configuration for a Quick account
describe_template	Describes a template's metadata
describe_template_alias	Describes the template alias for a template
describe_template_definition	Provides a detailed description of the definition of a template
describe_template_permissions	Describes read and write permissions on a template
describe_theme	Describes a theme
describe_theme_alias	Describes the alias for a theme
describe_theme_permissions	Describes the read and write permissions for a theme
describe_topic	Describes a topic
describe_topic_permissions	Describes the permissions of a topic
describe_topic_refresh	Describes the status of a topic refresh
describe_topic_refresh_schedule	Deletes a topic refresh schedule
describe_user	Returns information about a user, given the user name
describe_vpc_connection	Describes a VPC connection
generate_embed_url_for_anonymous_user	Generates an embed URL that you can use to embed an Amazon Qu
generate_embed_url_for_registered_user	Generates an embed URL that you can use to embed an Amazon Qu
generate_embed_url_for_registered_user_with_identity	Generates an embed URL that you can use to embed an Amazon Qu
get_dashboard_embed_url	Generates a temporary session URL and authorization code(bearer t
get_flow_metadata	Retrieves the metadata of a flow, not including its definition specify
get_flow_permissions	Get permissions for a flow
get_identity_context	Retrieves the identity context for a Quick Sight user in a specified n
get_session_embed_url	Generates a session URL and authorization code that you can use to
list_action_connectors	Lists all action connectors in the specified Amazon Web Services ac
list_analyses	Lists Amazon Quick Sight analyses that exist in the specified Amaz
list_asset_bundle_export_jobs	Lists all asset bundle export jobs that have been taken place in the la
list_asset_bundle_import_jobs	Lists all asset bundle import jobs that have taken place in the last 14
list_brands	Lists all brands in an Quick Sight account
list_custom_permissions	Returns a list of all the custom permissions profiles
list_dashboards	Lists dashboards in an Amazon Web Services account
list_dashboard_versions	Lists all the versions of the dashboards in the Amazon Quick Sight s
list_data_sets	Lists all of the datasets belonging to the current Amazon Web Servi
list_data_sources	Lists data sources in current Amazon Web Services Region that belo
list_flows	Lists flows in an Amazon Web Services account
list_folder_members	List all assets (DASHBOARD, ANALYSIS, and DATASET) in a fo
list_folders	Lists all folders in an account
list_folders_for_resource	List all folders that a resource is a member of
list_group_memberships	Lists member users in a group
list_groups	Lists all user groups in Amazon Quick Sight
list_iam_policy_assignments	Lists the IAM policy assignments in the current Amazon Quick Sigh

list_iam_policy_assignments_for_user	Lists all of the IAM policy assignments, including the Amazon Res
list_identity_propagation_configs	Lists all services and authorized targets that the Quick Sight IAM Id
list_ingestions	Lists the history of SPICE ingestions for a dataset
list_namespaces	Lists the namespaces for the specified Amazon Web Services accou
list_refresh_schedules	Lists the refresh schedules of a dataset
list_role_memberships	Lists all groups that are associated with a role
list_self_upgrades	Lists all self-upgrade requests for a Quick account
list_tags_for_resource	Lists the tags assigned to a resource
list_template_aliases	Lists all the aliases of a template
list_templates	Lists all the templates in the current Amazon Quick Sight account
list_template_versions	Lists all the versions of the templates in the current Amazon Quick
list_theme_aliases	Lists all the aliases of a theme
list_themes	Lists all the themes in the current Amazon Web Services account
list_theme_versions	Lists all the versions of the themes in the current Amazon Web Serv
list_topic_refresh_schedules	Lists all of the refresh schedules for a topic
list_topic_reviewed_answers	Lists all reviewed answers for a Q Topic
list_topics	Lists all of the topics within an account
list_user_groups	Lists the Amazon Quick Sight groups that an Amazon Quick Sight u
list_users	Returns a list of all of the Amazon Quick Sight users belonging to th
list_vpc_connections	Lists all of the VPC connections in the current set Amazon Web Ser
predict_qa_results	Predicts existing visuals or generates new visuals to answer a given
put_data_set_refresh_properties	Creates or updates the dataset refresh properties for the dataset
register_user	Creates an Amazon Quick Sight user whose identity is associated w
restore_analysis	Restores an analysis
search_action_connectors	Searches for action connectors in the specified Amazon Web Servic
search_analyses	Searches for analyses that belong to the user specified in the filter
search_dashboards	Searches for dashboards that belong to a user
search_data_sets	Use the SearchDataSets operation to search for datasets that belong
search_data_sources	Use the SearchDataSources operation to search for data sources that
search_flows	Search for the flows in an Amazon Web Services account
search_folders	Searches the subfolders in a folder
search_groups	Use the SearchGroups operation to search groups in a specified Qui
search_topics	Searches for any Q topic that exists in an Quick account
start_asset_bundle_export_job	Starts an Asset Bundle export job
start_asset_bundle_import_job	Starts an Asset Bundle import job
start_automation_job	Starts a new job for a specified automation
start_dashboard_snapshot_job	Starts an asynchronous job that generates a snapshot of a dashboard
start_dashboard_snapshot_job_schedule	Starts an asynchronous job that runs an existing dashboard schedule
tag_resource	Assigns one or more tags (key-value pairs) to the specified Amazon
untag_resource	Removes a tag or tags from a resource
update_account_customization	Updates Amazon Quick Sight customizations
update_account_custom_permission	Applies a custom permissions profile to an account
update_account_settings	Updates the Amazon Quick Sight settings in your Amazon Web Ser
update_action_connector	Updates an existing action connector with new configuration details
update_action_connector_permissions	Updates the permissions for an action connector by granting or revo
update_analysis	Updates an analysis in Amazon Quick Sight
update_analysis_permissions	Updates the read and write permissions for an analysis
update_application_with_token_exchange_grant	Updates an Quick application with a token exchange grant

update_brand	Updates a brand
update_brand_assignment	Updates a brand assignment
update_brand_published_version	Updates the published version of a brand
update_custom_permissions	Updates a custom permissions profile
update_dashboard	Updates a dashboard in an Amazon Web Services account
update_dashboard_links	Updates the linked analyses on a dashboard
update_dashboard_permissions	Updates read and write permissions on a dashboard
update_dashboard_published_version	Updates the published version of a dashboard
update_dashboards_qa_configuration	Updates a Dashboard QA configuration
update_data_set	Updates a dataset
update_data_set_permissions	Updates the permissions on a dataset
update_data_source	Updates a data source
update_data_source_permissions	Updates the permissions to a data source
update_default_q_business_application	Updates a Amazon Q Business application that is linked to a Quick
update_flow_permissions	Updates permissions against principals on a flow
update_folder	Updates the name of a folder
update_folder_permissions	Updates permissions of a folder
update_group	Changes a group description
update_iam_policy_assignment	Updates an existing IAM policy assignment
update_identity_propagation_config	Adds or updates services and authorized targets to configure what th
update_ip_restriction	Updates the content and status of IP rules
update_key_registration	Updates a customer managed key in a Quick Sight account
update_public_sharing_settings	This API controls public sharing settings for your entire Quick Sigh
update_q_personalization_configuration	Updates a personalization configuration
update_quick_sight_q_search_configuration	Updates the state of a Quick Sight Q Search configuration
update_refresh_schedule	Updates a refresh schedule for a dataset
update_role_custom_permission	Updates the custom permissions that are associated with a role
update_self_upgrade	Updates a self-upgrade request for a Quick user by approving, deny
update_self_upgrade_configuration	Updates the self-upgrade configuration for a Quick account
update_spice_capacity_configuration	Updates the SPICE capacity configuration for a Quick Sight accoun
update_template	Updates a template from an existing Amazon Quick Sight analysis o
update_template_alias	Updates the template alias of a template
update_template_permissions	Updates the resource permissions for a template
update_theme	Updates a theme
update_theme_alias	Updates an alias of a theme
update_theme_permissions	Updates the resource permissions for a theme
update_topic	Updates a topic
update_topic_permissions	Updates the permissions of a topic
update_topic_refresh_schedule	Updates a topic refresh schedule
update_user	Updates an Amazon Quick Sight user
update_user_custom_permission	Updates a custom permissions profile for a user
update_vpc_connection	Updates a VPC connection

Examples

```
## Not run:
svc <- quicksight()
```

```
svc$batch_create_topic_reviewed_answer(
  Foo = 123
)

## End(Not run)
```

Index