Package 'paws.storage'

March 14, 2025

Title 'Amazon Web Services' Storage Services

Version 0.9.0

Description Interface to 'Amazon Web Services' storage services, including 'Simple Storage Service' ('S3') 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.storage

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

Imports paws.common (>= 0.8.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.2

Collate 'backup_service.R' 'backup_interfaces.R' 'backup_operations.R' 'dlm_service.R' 'dlm_interfaces.R' 'dlm_operations.R' 'ebs_service.R' 'ebs_interfaces.R' 'ebs_operations.R' 'efs_service.R' 'efs_interfaces.R' 'efs_operations.R' 'finspacedata_service.R' 'finspacedata_interfaces.R' 'finspacedata_operations.R' 'fsx_service.R' 'fsx_interfaces.R' 'fsx_operations.R' 'glacier_service.R' 'glacier_interfaces.R' 'glacier_operations.R' 'omics_service.R' 'omics_interfaces.R' 'omics_operations.R' 'recyclebin_service.R' 'recyclebin_interfaces.R' 'recyclebin_operations.R' 'reexports_paws.common.R' 's3_service.R' 's3_operations.R' 's3_custom.R' 's3_interfaces.R' 's3control_service.R' 's3control_interfaces.R' 's3control_operations.R' 's3outposts_service.R' 's3outposts_interfaces.R' 's3outposts_operations.R' 's3tables_service.R' 's3tables_interfaces.R' 's3tables_operations.R' 'storagegateway_service.R' 'storagegateway_interfaces.R' 'storagegateway_operations.R'

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-03-14 17:10:02 UTC

Contents

---

backup *AWS Backup*

Description

Backup

Backup is a unified backup service designed to protect Amazon Web Services services and their associated data. Backup simplifies the creation, migration, restoration, and deletion of backups, while also providing reporting and auditing.

Usage

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

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

Operations

cancel_legal_hold	Removes the specified legal hold on a recovery point
create_backup_plan	Creates a backup plan using a backup plan name and backup rules
create_backup_selection	Creates a JSON document that specifies a set of resources to assign to a backup
create_backup_vault	Creates a logical container where backups are stored
create_framework	Creates a framework with one or more controls
create_legal_hold	Creates a legal hold on a recovery point (backup)
create_logically_air_gapped_backup_vault	Creates a logical container to where backups may be copied
create_report_plan	Creates a report plan
create_restore_testing_plan	Creates a restore testing plan
create_restore_testing_selection	This request can be sent after CreateRestoreTestingPlan request returns successf
delete_backup_plan	Deletes a backup plan
delete_backup_selection	Deletes the resource selection associated with a backup plan that is specified by
delete_backup_vault	Deletes the backup vault identified by its name
delete_backup_vault_access_policy	Deletes the policy document that manages permissions on a backup vault
delete_backup_vault_lock_configuration	Deletes Backup Vault Lock from a backup vault specified by a backup vault nam
delete_backup_vault_notifications	Deletes event notifications for the specified backup vault
delete_framework	Deletes the framework specified by a framework name
delete_recovery_point	Deletes the recovery point specified by a recovery point ID
delete_report_plan	Deletes the report plan specified by a report plan name
delete_restore_testing_plan	This request deletes the specified restore testing plan

delete_restore_testing_selection	Input the Restore Testing Plan name and Restore Testing Selection name
describe_backup_job	Returns backup job details for the specified BackupJobId
describe_backup_vault	Returns metadata about a backup vault specified by its name
describe_copy_job	Returns metadata associated with creating a copy of a resource
describe_framework	Returns the framework details for the specified FrameworkName
describe_global_settings	Describes whether the Amazon Web Services account is opted in to cross-accou
describe_protected_resource	Returns information about a saved resource, including the last time it was backe
describe_recovery_point	Returns metadata associated with a recovery point, including ID, status, encrypti
describe_region_settings	Returns the current service opt-in settings for the Region
describe_report_job	Returns the details associated with creating a report as specified by its ReportJob
describe_report_plan	Returns a list of all report plans for an Amazon Web Services account and Amaz
describe_restore_job	Returns metadata associated with a restore job that is specified by a job ID
disassociate_recovery_point	Deletes the specified continuous backup recovery point from Backup and release
disassociate_recovery_point_from_parent	This action to a specific child (nested) recovery point removes the relationship b
export_backup_plan_template	Returns the backup plan that is specified by the plan ID as a backup template
get_backup_plan	Returns BackupPlan details for the specified BackupPlanId
get_backup_plan_from_json	Returns a valid JSON document specifying a backup plan or an error
get_backup_plan_from_template	Returns the template specified by its templateId as a backup plan
get_backup_selection	Returns selection metadata and a document in JSON format that specifies a list o
get_backup_vault_access_policy	Returns the access policy document that is associated with the named backup va
get_backup_vault_notifications	Returns event notifications for the specified backup vault
get_legal_hold	This action returns details for a specified legal hold
get_recovery_point_index_details	This operation returns the metadata and details specific to the backup index asso
get_recovery_point_restore_metadata	Returns a set of metadata key-value pairs that were used to create the backup
get_restore_job_metadata	This request returns the metadata for the specified restore job
get_restore_testing_inferred_metadata	This request returns the minimal required set of metadata needed to start a restor
get_restore_testing_plan	Returns RestoreTestingPlan details for the specified RestoreTestingPlanName
get_restore_testing_selection	Returns RestoreTestingSelection, which displays resources and elements of the r
get_supported_resource_types	Returns the Amazon Web Services resource types supported by Backup
list_backup_jobs	Returns a list of existing backup jobs for an authenticated account for the last 30
list_backup_job_summaries	This is a request for a summary of backup jobs created or running within the mo
list_backup_plans	Lists the active backup plans for the account
list_backup_plan_templates	Lists the backup plan templates
list_backup_plan_versions	Returns version metadata of your backup plans, including Amazon Resource Na
list_backup_selections	Returns an array containing metadata of the resources associated with the target
list_backup_vaults	Returns a list of recovery point storage containers along with information about
list_copy_jobs	Returns metadata about your copy jobs
list_copy_job_summaries	This request obtains a list of copy jobs created or running within the the most re
list_frameworks	Returns a list of all frameworks for an Amazon Web Services account and Amaz
list_indexed_recovery_points	This operation returns a list of recovery points that have an associated index, bel
list_legal_holds	This action returns metadata about active and previous legal holds
list_protected_resources	Returns an array of resources successfully backed up by Backup, including the ti
list_protected_resources_by_backup_vault	This request lists the protected resources corresponding to each backup vault
list_recovery_points_by_backup_vault	Returns detailed information about the recovery points stored in a backup vault
list_recovery_points_by_legal_hold	This action returns recovery point ARNs (Amazon Resource Names) of the spec
list_recovery_points_by_resource	The information about the recovery points of the type specified by a resource Ar
list_report_jobs	Returns details about your report jobs
list_report_plans	Returns a list of your report plans

list_restore_jobs	Returns a list of jobs that Backup initiated to restore a saved resource, including
list_restore_jobs_by_protected_resource	This returns restore jobs that contain the specified protected resource
list_restore_job_summaries	This request obtains a summary of restore jobs created or running within the the
list_restore_testing_plans	Returns a list of restore testing plans
list_restore_testing_selections	Returns a list of restore testing selections
list_tags	Returns the tags assigned to the resource, such as a target recovery point, backup
put_backup_vault_access_policy	Sets a resource-based policy that is used to manage access permissions on the ta
put_backup_vault_lock_configuration	Applies Backup Vault Lock to a backup vault, preventing attempts to delete any
put_backup_vault_notifications	Turns on notifications on a backup vault for the specified topic and events
put_restore_validation_result	This request allows you to send your independent self-run restore test validation
start_backup_job	Starts an on-demand backup job for the specified resource
start_copy_job	Starts a job to create a one-time copy of the specified resource
start_report_job	Starts an on-demand report job for the specified report plan
start_restore_job	Recovers the saved resource identified by an Amazon Resource Name (ARN)
stop_backup_job	Attempts to cancel a job to create a one-time backup of a resource
tag_resource	Assigns a set of key-value pairs to a recovery point, backup plan, or backup vaul
untag_resource	Removes a set of key-value pairs from a recovery point, backup plan, or backup
update_backup_plan	Updates the specified backup plan
update_framework	Updates the specified framework
update_global_settings	Updates whether the Amazon Web Services account is opted in to cross-account
update_recovery_point_index_settings	This operation updates the settings of a recovery point index
update_recovery_point_lifecycle	Sets the transition lifecycle of a recovery point
update_region_settings	Updates the current service opt-in settings for the Region
update_report_plan	Updates the specified report plan
update_restore_testing_plan	This request will send changes to your specified restore testing plan
update_restore_testing_selection	Updates the specified restore testing selection

Examples

```
## Not run:
svc <- backup()
svc$cancel_legal_hold(
  Foo = 123
)

## End(Not run)
```

`dlm` *Amazon Data Lifecycle Manager*

Description

With Amazon Data Lifecycle Manager, you can manage the lifecycle of your Amazon Web Services resources. You create lifecycle policies, which are used to automate operations on the specified resources.

Amazon Data Lifecycle Manager supports Amazon EBS volumes and snapshots. For information about using Amazon Data Lifecycle Manager with Amazon EBS, see Amazon Data Lifecycle Manager in the *Amazon EC2 User Guide*.

Usage

```
dlm(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 <- dlm(
  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_lifecycle_policy	Creates an Amazon Data Lifecycle Manager lifecycle policy
delete_lifecycle_policy	Deletes the specified lifecycle policy and halts the automated operations that the policy specified
get_lifecycle_policies	Gets summary information about all or the specified data lifecycle policies
get_lifecycle_policy	Gets detailed information about the specified lifecycle policy
list_tags_for_resource	Lists the tags for the specified resource
tag_resource	Adds the specified tags to the specified resource
untag_resource	Removes the specified tags from the specified resource
update_lifecycle_policy	Updates the specified lifecycle policy

Examples

```
## Not run:
```

```
svc <- dlm()
svc$create_lifecycle_policy(
  Foo = 123
)

## End(Not run)
```

---

ebs *Amazon Elastic Block Store*

Description

You can use the Amazon Elastic Block Store (Amazon EBS) direct APIs to create Amazon EBS snapshots, write data directly to your snapshots, read data on your snapshots, and identify the differences or changes between two snapshots. If you're an independent software vendor (ISV) who offers backup services for Amazon EBS, the EBS direct APIs make it more efficient and cost-effective to track incremental changes on your Amazon EBS volumes through snapshots. This can be done without having to create new volumes from snapshots, and then use Amazon Elastic Compute Cloud (Amazon EC2) instances to compare the differences.

You can create incremental snapshots directly from data on-premises into volumes and the cloud to use for quick disaster recovery. With the ability to write and read snapshots, you can write your on-premises data to an snapshot during a disaster. Then after recovery, you can restore it back to Amazon Web Services or on-premises from the snapshot. You no longer need to build and maintain complex mechanisms to copy data to and from Amazon EBS.

This API reference provides detailed information about the actions, data types, parameters, and errors of the EBS direct APIs. For more information about the elements that make up the EBS direct APIs, and examples of how to use them effectively, see Accessing the Contents of an Amazon EBS Snapshot in the *Amazon Elastic Compute Cloud User Guide*. For more information about the supported Amazon Web Services Regions, endpoints, and service quotas for the EBS direct APIs, see Amazon Elastic Block Store Endpoints and Quotas in the *Amazon Web Services General Reference*.

Usage

```
ebs(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 <- ebs(
  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

complete_snapshot	Seals and completes the snapshot after all of the required blocks of data have been written to it
get_snapshot_block	Returns the data in a block in an Amazon Elastic Block Store snapshot
list_changed_blocks	Returns information about the blocks that are different between two Amazon Elastic Block Store snaps
list_snapshot_blocks	Returns information about the blocks in an Amazon Elastic Block Store snapshot
put_snapshot_block	Writes a block of data to a snapshot
start_snapshot	Creates a new Amazon EBS snapshot

Examples

```
## Not run:
svc <- ebs()
svc$complete_snapshot(
  Foo = 123
)

## End(Not run)
```

---

efs *Amazon Elastic File System*

Description

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 Linux and Mac instances in the Amazon Web Services Cloud. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so that your applications have the storage they need, when they need it. For more information, see the Amazon Elastic File System API Reference and the Amazon Elastic File System User Guide.

Usage

```
efs(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 <- efs(
  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_access_point	Creates an EFS access point
create_file_system	Creates a new, empty file system
create_mount_target	Creates a mount target for a file system
create_replication_configuration	Creates a replication configuration to either a new or existing EFS file system
create_tags	DEPRECATED - CreateTags is deprecated and not maintained
delete_access_point	Deletes the specified access point
delete_file_system	Deletes a file system, permanently severing access to its contents
delete_file_system_policy	Deletes the FileSystemPolicy for the specified file system
delete_mount_target	Deletes the specified mount target
delete_replication_configuration	Deletes a replication configuration
delete_tags	DEPRECATED - DeleteTags is deprecated and not maintained
describe_access_points	Returns the description of a specific Amazon EFS access point if the AccessPointId
describe_account_preferences	Returns the account preferences settings for the Amazon Web Services account ass

describe_backup_policy	Returns the backup policy for the specified EFS file system
describe_file_system_policy	Returns the FileSystemPolicy for the specified EFS file system
describe_file_systems	Returns the description of a specific Amazon EFS file system if either the file syste
describe_lifecycle_configuration	Returns the current LifecycleConfiguration object for the specified Amazon EFS fil
describe_mount_targets	Returns the descriptions of all the current mount targets, or a specific mount target,
describe_mount_target_security_groups	Returns the security groups currently in effect for a mount target
describe_replication_configurations	Retrieves the replication configuration for a specific file system
describe_tags	DEPRECATED - The DescribeTags action is deprecated and not maintained
list_tags_for_resource	Lists all tags for a top-level EFS resource
modify_mount_target_security_groups	Modifies the set of security groups in effect for a mount target
put_account_preferences	Use this operation to set the account preference in the current Amazon Web Service
put_backup_policy	Updates the file system's backup policy
put_file_system_policy	Applies an Amazon EFS FileSystemPolicy to an Amazon EFS file system
put_lifecycle_configuration	Use this action to manage storage for your file system
tag_resource	Creates a tag for an EFS resource
untag_resource	Removes tags from an EFS resource
update_file_system	Updates the throughput mode or the amount of provisioned throughput of an existi
update_file_system_protection	Updates protection on the file system

Examples

```
## Not run:
svc <- efs()
# This operation creates a new, encrypted file system with automatic
# backups enabled, and the default generalpurpose performance mode.
svc$create_file_system(
  Backup = TRUE,
  CreationToken = "tokenstring",
  Encrypted = TRUE,
  PerformanceMode = "generalPurpose",
  Tags = list(
    list(
      Key = "Name",
      Value = "MyFileSystem"
    )
  )
)

## End(Not run)
```

`finspacedata` *FinSpace Public API*

Description

The FinSpace APIs let you take actions inside the FinSpace.

Usage

```
finspacedata(
  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 <- finspacedata(
  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_user_to_permission_group	Adds a user to a permission group to grant permissions for actions a user can per
create_changeset	Creates a new Changeset in a FinSpace Dataset
create_dataset	Creates a new FinSpace Dataset
create_data_view	Creates a Dataview for a Dataset
create_permission_group	Creates a group of permissions for various actions that a user can perform in Fin
create_user	Creates a new user in FinSpace
delete_dataset	Deletes a FinSpace Dataset
delete_permission_group	Deletes a permission group
disable_user	Denies access to the FinSpace web application and API for the specified user
disassociate_user_from_permission_group	Removes a user from a permission group
enable_user	Allows the specified user to access the FinSpace web application and API
get_changeset	Get information about a Changeset
get_dataset	Returns information about a Dataset

get_data_view	Gets information about a Dataview
get_external_data_view_access_details	Returns the credentials to access the external Dataview from an S3 location
get_permission_group	Retrieves the details of a specific permission group
get_programmatic_access_credentials	Request programmatic credentials to use with FinSpace SDK
get_user	Retrieves details for a specific user
get_working_location	A temporary Amazon S3 location, where you can copy your files from a source l
list_changesets	Lists the FinSpace Changesets for a Dataset
list_datasets	Lists all of the active Datasets that a user has access to
list_data_views	Lists all available Dataviews for a Dataset
list_permission_groups	Lists all available permission groups in FinSpace
list_permission_groups_by_user	Lists all the permission groups that are associated with a specific user
list_users	Lists all available users in FinSpace
list_users_by_permission_group	Lists details of all the users in a specific permission group
reset_user_password	Resets the password for a specified user ID and generates a temporary one
update_changeset	Updates a FinSpace Changeset
update_dataset	Updates a FinSpace Dataset
update_permission_group	Modifies the details of a permission group
update_user	Modifies the details of the specified user

Examples

```
## Not run:
svc <- finspacedata()
svc$associate_user_to_permission_group(
  Foo = 123
)

## End(Not run)
```

`fsx` *Amazon FSx*

Description

Amazon FSx is a fully managed service that makes it easy for storage and application administrators to launch and use shared file storage.

Usage

```
fsx(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 <- fsx(
  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_file_system_aliases	Use this action to associate one or more Domain Name Server (DNS) aliases with an
cancel_data_repository_task	Cancels an existing Amazon FSx for Lustre data repository task if that task is in eithe
copy_backup	Copies an existing backup within the same Amazon Web Services account to another
copy_snapshot_and_update_volume	Updates an existing volume by using a snapshot from another Amazon FSx for Open
create_backup	Creates a backup of an existing Amazon FSx for Windows File Server file system, A
create_data_repository_association	Creates an Amazon FSx for Lustre data repository association (DRA)
create_data_repository_task	Creates an Amazon FSx for Lustre data repository task
create_file_cache	Creates a new Amazon File Cache resource
create_file_system	Creates a new, empty Amazon FSx file system
create_file_system_from_backup	Creates a new Amazon FSx for Lustre, Amazon FSx for Windows File Server, or Am
create_snapshot	Creates a snapshot of an existing Amazon FSx for OpenZFS volume
create_storage_virtual_machine	Creates a storage virtual machine (SVM) for an Amazon FSx for ONTAP file system
create_volume	Creates an FSx for ONTAP or Amazon FSx for OpenZFS storage volume
create_volume_from_backup	Creates a new Amazon FSx for NetApp ONTAP volume from an existing Amazon F
delete_backup	Deletes an Amazon FSx backup
delete_data_repository_association	Deletes a data repository association on an Amazon FSx for Lustre file system
delete_file_cache	Deletes an Amazon File Cache resource
delete_file_system	Deletes a file system
delete_snapshot	Deletes an Amazon FSx for OpenZFS snapshot
delete_storage_virtual_machine	Deletes an existing Amazon FSx for ONTAP storage virtual machine (SVM)

delete_volume	Deletes an Amazon FSx for NetApp ONTAP or Amazon FSx for OpenZFS volume
describe_backups	Returns the description of a specific Amazon FSx backup, if a BackupIds value is pro
describe_data_repository_associations	Returns the description of specific Amazon FSx for Lustre or Amazon File Cache da
describe_data_repository_tasks	Returns the description of specific Amazon FSx for Lustre or Amazon File Cache da
describe_file_caches	Returns the description of a specific Amazon File Cache resource, if a FileCacheIds
describe_file_system_aliases	Returns the DNS aliases that are associated with the specified Amazon FSx for Wind
describe_file_systems	Returns the description of specific Amazon FSx file systems, if a FileSystemIds valu
describe_shared_vpc_configuration	Indicates whether participant accounts in your organization can create Amazon FSx f
describe_snapshots	Returns the description of specific Amazon FSx for OpenZFS snapshots, if a Snapsh
describe_storage_virtual_machines	Describes one or more Amazon FSx for NetApp ONTAP storage virtual machines (S
describe_volumes	Describes one or more Amazon FSx for NetApp ONTAP or Amazon FSx for OpenZ
disassociate_file_system_aliases	Use this action to disassociate, or remove, one or more Domain Name Service (DNS
list_tags_for_resource	Lists tags for Amazon FSx resources
release_file_system_nfs_v3_locks	Releases the file system lock from an Amazon FSx for OpenZFS file system
restore_volume_from_snapshot	Returns an Amazon FSx for OpenZFS volume to the state saved by the specified snap
start_misconfigured_state_recovery	After performing steps to repair the Active Directory configuration of an FSx for Wi
tag_resource	Tags an Amazon FSx resource
untag_resource	This action removes a tag from an Amazon FSx resource
update_data_repository_association	Updates the configuration of an existing data repository association on an Amazon F
update_file_cache	Updates the configuration of an existing Amazon File Cache resource
update_file_system	Use this operation to update the configuration of an existing Amazon FSx file system
update_shared_vpc_configuration	Configures whether participant accounts in your organization can create Amazon FS
update_snapshot	Updates the name of an Amazon FSx for OpenZFS snapshot
update_storage_virtual_machine	Updates an FSx for ONTAP storage virtual machine (SVM)
update_volume	Updates the configuration of an Amazon FSx for NetApp ONTAP or Amazon FSx fo

Examples

```
## Not run:
svc <- fsx()
# This operation copies an Amazon FSx backup.
svc$copy_backup(
  SourceBackupId = "backup-03e3c82e0183b7b6b",
  SourceRegion = "us-east-2"
)

## End(Not run)
```

`glacier` *Amazon Glacier*

Description

Amazon S3 Glacier (Glacier) is a storage solution for "cold data."

Glacier is an extremely low-cost storage service that provides secure, durable, and easy-to-use storage for data backup and archival. With Glacier, customers can store their data cost effectively for months, years, or decades. Glacier also enables customers to offload the administrative burdens of operating and scaling storage to AWS, so they don't have to worry about capacity planning, hardware provisioning, data replication, hardware failure and recovery, or time-consuming hardware migrations.

Glacier is a great storage choice when low storage cost is paramount and your data is rarely retrieved. If your application requires fast or frequent access to your data, consider using Amazon S3. For more information, see Amazon Simple Storage Service (Amazon S3).

You can store any kind of data in any format. There is no maximum limit on the total amount of data you can store in Glacier.

If you are a first-time user of Glacier, we recommend that you begin by reading the following sections in the *Amazon S3 Glacier Developer Guide*:

- What is Amazon S3 Glacier - This section of the Developer Guide describes the underlying data model, the operations it supports, and the AWS SDKs that you can use to interact with the service.
- Getting Started with Amazon S3 Glacier - The Getting Started section walks you through the process of creating a vault, uploading archives, creating jobs to download archives, retrieving the job output, and deleting archives.

Usage

```
glacier(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 <- glacier(
  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

abort_multipart_upload	This operation aborts a multipart upload identified by the upload ID
abort_vault_lock	This operation aborts the vault locking process if the vault lock is not in the Locked state
add_tags_to_vault	This operation adds the specified tags to a vault
complete_multipart_upload	You call this operation to inform Amazon S3 Glacier (Glacier) that all the archive parts have
complete_vault_lock	This operation completes the vault locking process by transitioning the vault lock from the I
create_vault	This operation creates a new vault with the specified name
delete_archive	This operation deletes an archive from a vault
delete_vault	This operation deletes a vault
delete_vault_access_policy	This operation deletes the access policy associated with the specified vault
delete_vault_notifications	This operation deletes the notification configuration set for a vault
describe_job	This operation returns information about a job you previously initiated, including the job ini
describe_vault	This operation returns information about a vault, including the vault's Amazon Resource Na
get_data_retrieval_policy	This operation returns the current data retrieval policy for the account and region specified i
get_job_output	This operation downloads the output of the job you initiated using InitiateJob
get_vault_access_policy	This operation retrieves the access-policy subresource set on the vault; for more information
get_vault_lock	This operation retrieves the following attributes from the lock-policy subresource set on the
get_vault_notifications	This operation retrieves the notification-configuration subresource of the specified vault
initiate_job	This operation initiates a job of the specified type, which can be a select, an archival retrieva
initiate_multipart_upload	This operation initiates a multipart upload
initiate_vault_lock	This operation initiates the vault locking process by doing the following:
list_jobs	This operation lists jobs for a vault, including jobs that are in-progress and jobs that have rec
list_multipart_uploads	This operation lists in-progress multipart uploads for the specified vault
list_parts	This operation lists the parts of an archive that have been uploaded in a specific multipart up
list_provisioned_capacity	This operation lists the provisioned capacity units for the specified AWS account
list_tags_for_vault	This operation lists all the tags attached to a vault
list_vaults	This operation lists all vaults owned by the calling user's account
purchase_provisioned_capacity	This operation purchases a provisioned capacity unit for an AWS account
remove_tags_from_vault	This operation removes one or more tags from the set of tags attached to a vault
set_data_retrieval_policy	This operation sets and then enacts a data retrieval policy in the region specified in the PUT
set_vault_access_policy	This operation configures an access policy for a vault and will overwrite an existing policy
set_vault_notifications	This operation configures notifications that will be sent when specific events happen to a vau
upload_archive	This operation adds an archive to a vault
upload_multipart_part	This operation uploads a part of an archive

Examples

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

```
# The example deletes an in-progress multipart upload to a vault named
# my-vault:
svc$abort_multipart_upload(
  accountId = "-",
  uploadId = "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-OssZtLq...",
  vaultName = "my-vault"
)

## End(Not run)
```

`omics` *Amazon Omics*

Description

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

Usage

```
omics(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 <- omics(
  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

abort_multipart_read_set_upload	Stops a multipart upload
accept_share	Accept a resource share request
batch_delete_read_set	Deletes one or more read sets
cancel_annotation_import_job	Cancels an annotation import job
cancel_run	Cancels a run
cancel_variant_import_job	Cancels a variant import job
complete_multipart_read_set_upload	Concludes a multipart upload once you have uploaded all the components
create_annotation_store	Creates an annotation store
create_annotation_store_version	Creates a new version of an annotation store
create_multipart_read_set_upload	Begins a multipart read set upload
create_reference_store	Creates a reference store
create_run_cache	You can create a run cache to save the task outputs from completed tasks in a run for a
create_run_group	You can optionally create a run group to limit the compute resources for the runs that y
create_sequence_store	Creates a sequence store
create_share	Creates a cross-account shared resource
create_variant_store	Creates a variant store
create_workflow	Creates a workflow
delete_annotation_store	Deletes an annotation store
delete_annotation_store_versions	Deletes one or multiple versions of an annotation store
delete_reference	Deletes a genome reference
delete_reference_store	Deletes a genome reference store
delete_run	Deletes a workflow run
delete_run_cache	Delete a run cache
delete_run_group	Deletes a workflow run group
delete_s3_access_policy	Deletes an access policy for the specified store
delete_sequence_store	Deletes a sequence store
delete_share	Deletes a resource share
delete_variant_store	Deletes a variant store
delete_workflow	Deletes a workflow
get_annotation_import_job	Gets information about an annotation import job
get_annotation_store	Gets information about an annotation store
get_annotation_store_version	Retrieves the metadata for an annotation store version
get_read_set	Gets a file from a read set
get_read_set_activation_job	Gets information about a read set activation job
get_read_set_export_job	Gets information about a read set export job
get_read_set_import_job	Gets information about a read set import job
get_read_set_metadata	Gets details about a read set
get_reference	Gets a reference file
get_reference_import_job	Gets information about a reference import job
get_reference_metadata	Gets information about a genome reference's metadata
get_reference_store	Gets information about a reference store
get_run	Gets information about a workflow run
get_run_cache	Retrieve the details for the specified run cache
get_run_group	Gets information about a workflow run group
get_run_task	Gets information about a workflow run task
get_s3_access_policy	Retrieves details about an access policy on a given store

get_sequence_store	Gets information about a sequence store
get_share	Retrieves the metadata for the specified resource share
get_variant_import_job	Gets information about a variant import job
get_variant_store	Gets information about a variant store
get_workflow	Gets information about a workflow
list_annotation_import_jobs	Retrieves a list of annotation import jobs
list_annotation_stores	Retrieves a list of annotation stores
list_annotation_store_versions	Lists the versions of an annotation store
list_multipart_read_set_uploads	Lists multipart read set uploads and for in progress uploads
list_read_set_activation_jobs	Retrieves a list of read set activation jobs
list_read_set_export_jobs	Retrieves a list of read set export jobs
list_read_set_import_jobs	Retrieves a list of read set import jobs
list_read_sets	Retrieves a list of read sets
list_read_set_upload_parts	This operation will list all parts in a requested multipart upload for a sequence store
list_reference_import_jobs	Retrieves a list of reference import jobs
list_references	Retrieves a list of references
list_reference_stores	Retrieves a list of reference stores
list_run_caches	Retrieves a list of your run caches
list_run_groups	Retrieves a list of run groups
list_runs	Retrieves a list of runs
list_run_tasks	Retrieves a list of tasks for a run
list_sequence_stores	Retrieves a list of sequence stores
list_shares	Retrieves the resource shares associated with an account
list_tags_for_resource	Retrieves a list of tags for a resource
list_variant_import_jobs	Retrieves a list of variant import jobs
list_variant_stores	Retrieves a list of variant stores
list_workflows	Retrieves a list of workflows
put_s3_access_policy	Adds an access policy to the specified store
start_annotation_import_job	Starts an annotation import job
start_read_set_activation_job	Activates an archived read set
start_read_set_export_job	Exports a read set to Amazon S3
start_read_set_import_job	Starts a read set import job
start_reference_import_job	Starts a reference import job
start_run	Starts a workflow run
start_variant_import_job	Starts a variant import job
tag_resource	Tags a resource
untag_resource	Removes tags from a resource
update_annotation_store	Updates an annotation store
update_annotation_store_version	Updates the description of an annotation store version
update_run_cache	Update a run cache
update_run_group	Updates a run group
update_sequence_store	Update one or more parameters for the sequence store
update_variant_store	Updates a variant store
update_workflow	Updates a workflow
upload_read_set_part	This operation uploads a specific part of a read set

Examples

```
## Not run:
svc <- omics()
svc$abort_multipart_read_set_upload(
  Foo = 123
)

## End(Not run)
```

`recyclebin` *Amazon Recycle Bin*

Description

This is the *Recycle Bin API Reference*. This documentation provides descriptions and syntax for each of the actions and data types in Recycle Bin.

Recycle Bin is a resource recovery feature that enables you to restore accidentally deleted snapshots and EBS-backed AMIs. When using Recycle Bin, if your resources are deleted, they are retained in the Recycle Bin for a time period that you specify.

You can restore a resource from the Recycle Bin at any time before its retention period expires. After you restore a resource from the Recycle Bin, the resource is removed from the Recycle Bin, and you can then use it in the same way you use any other resource of that type in your account. If the retention period expires and the resource is not restored, the resource is permanently deleted from the Recycle Bin and is no longer available for recovery. For more information about Recycle Bin, see Recycle Bin in the *Amazon Elastic Compute Cloud User Guide*.

Usage

```
recyclebin(
  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 <- recyclebin(
  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_rule	Creates a Recycle Bin retention rule
delete_rule	Deletes a Recycle Bin retention rule
get_rule	Gets information about a Recycle Bin retention rule
list_rules	Lists the Recycle Bin retention rules in the Region
list_tags_for_resource	Lists the tags assigned to a retention rule
lock_rule	Locks a Region-level retention rule
tag_resource	Assigns tags to the specified retention rule
unlock_rule	Unlocks a retention rule
untag_resource	Unassigns a tag from a retention rule
update_rule	Updates an existing Recycle Bin retention rule

Examples

```
## Not run:
svc <- recyclebin()
svc$create_rule(
  Foo = 123
)

## End(Not run)
```

---

s3 *Amazon Simple Storage Service*

Description

Amazon Simple Storage Service

Usage

```
s3(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 <- s3(
  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

abort_multipart_upload	This operation aborts a multipart upload
complete_multipart_upload	Completes a multipart upload by assembling previously uploaded parts
copy_object	Creates a copy of an object that is already stored in Amazon S3
create_bucket	This action creates an Amazon S3 bucket
create_bucket_metadata_table_configuration	Creates a metadata table configuration for a general purpose bucket
create_multipart_upload	This action initiates a multipart upload and returns an upload ID
create_session	Creates a session that establishes temporary security credentials to support f
delete_bucket	Deletes the S3 bucket
delete_bucket_analytics_configuration	This operation is not supported for directory buckets
delete_bucket_cors	This operation is not supported for directory buckets
delete_bucket_encryption	This implementation of the DELETE action resets the default encryption fo
delete_bucket_intelligent_tiering_configuration	This operation is not supported for directory buckets
delete_bucket_inventory_configuration	This operation is not supported for directory buckets

delete_bucket_lifecycle	Deletes the lifecycle configuration from the specified bucket
delete_bucket_metadata_table_configuration	Deletes a metadata table configuration from a general purpose bucket
delete_bucket_metrics_configuration	This operation is not supported for directory buckets
delete_bucket_ownership_controls	This operation is not supported for directory buckets
delete_bucket_policy	Deletes the policy of a specified bucket
delete_bucket_replication	This operation is not supported for directory buckets
delete_bucket_tagging	This operation is not supported for directory buckets
delete_bucket_website	This operation is not supported for directory buckets
delete_object	Removes an object from a bucket
delete_objects	This operation enables you to delete multiple objects from a bucket using a
delete_object_tagging	This operation is not supported for directory buckets
delete_public_access_block	This operation is not supported for directory buckets
download_file	Download a file from S3 and store it at a specified file location
generate_presigned_url	@title Generate a presigned url given a client, its method, and arguments
get_bucket_accelerate_configuration	This operation is not supported for directory buckets
get_bucket_acl	This operation is not supported for directory buckets
get_bucket_analytics_configuration	This operation is not supported for directory buckets
get_bucket_cors	This operation is not supported for directory buckets
get_bucket_encryption	Returns the default encryption configuration for an Amazon S3 bucket
get_bucket_intelligent_tiering_configuration	This operation is not supported for directory buckets
get_bucket_inventory_configuration	This operation is not supported for directory buckets
get_bucket_lifecycle	For an updated version of this API, see GetBucketLifecycleConfiguration
get_bucket_lifecycle_configuration	Returns the lifecycle configuration information set on the bucket
get_bucket_location	This operation is not supported for directory buckets
get_bucket_logging	This operation is not supported for directory buckets
get_bucket_metadata_table_configuration	Retrieves the metadata table configuration for a general purpose bucket
get_bucket_metrics_configuration	This operation is not supported for directory buckets
get_bucket_notification	This operation is not supported for directory buckets
get_bucket_notification_configuration	This operation is not supported for directory buckets
get_bucket_ownership_controls	This operation is not supported for directory buckets
get_bucket_policy	Returns the policy of a specified bucket
get_bucket_policy_status	This operation is not supported for directory buckets
get_bucket_replication	This operation is not supported for directory buckets
get_bucket_request_payment	This operation is not supported for directory buckets
get_bucket_tagging	This operation is not supported for directory buckets
get_bucket_versioning	This operation is not supported for directory buckets
get_bucket_website	This operation is not supported for directory buckets
get_object	Retrieves an object from Amazon S3
get_object_acl	This operation is not supported for directory buckets
get_object_attributes	Retrieves all the metadata from an object without returning the object itself
get_object_legal_hold	This operation is not supported for directory buckets
get_object_lock_configuration	This operation is not supported for directory buckets
get_object_retention	This operation is not supported for directory buckets
get_object_tagging	This operation is not supported for directory buckets
get_object_torrent	This operation is not supported for directory buckets
get_public_access_block	This operation is not supported for directory buckets
head_bucket	You can use this operation to determine if a bucket exists and if you have pe
head_object	The HEAD operation retrieves metadata from an object without returning th

list_bucket_analytics_configurations	This operation is not supported for directory buckets
list_bucket_intelligent_tiering_configurations	This operation is not supported for directory buckets
list_bucket_inventory_configurations	This operation is not supported for directory buckets
list_bucket_metrics_configurations	This operation is not supported for directory buckets
list_buckets	This operation is not supported for directory buckets
list_directory_buckets	Returns a list of all Amazon S3 directory buckets owned by the authenticate
list_multipart_uploads	This operation lists in-progress multipart uploads in a bucket
list_objects	This operation is not supported for directory buckets
list_objects_v2	Returns some or all (up to 1,000) of the objects in a bucket with each reques
list_object_versions	This operation is not supported for directory buckets
list_parts	Lists the parts that have been uploaded for a specific multipart upload
put_bucket_accelerate_configuration	This operation is not supported for directory buckets
put_bucket_acl	This operation is not supported for directory buckets
put_bucket_analytics_configuration	This operation is not supported for directory buckets
put_bucket_cors	This operation is not supported for directory buckets
put_bucket_encryption	This operation configures default encryption and Amazon S3 Bucket Keys f
put_bucket_intelligent_tiering_configuration	This operation is not supported for directory buckets
put_bucket_inventory_configuration	This operation is not supported for directory buckets
put_bucket_lifecycle	This operation is not supported for directory buckets
put_bucket_lifecycle_configuration	Creates a new lifecycle configuration for the bucket or replaces an existing l
put_bucket_logging	This operation is not supported for directory buckets
put_bucket_metrics_configuration	This operation is not supported for directory buckets
put_bucket_notification	This operation is not supported for directory buckets
put_bucket_notification_configuration	This operation is not supported for directory buckets
put_bucket_ownership_controls	This operation is not supported for directory buckets
put_bucket_policy	Applies an Amazon S3 bucket policy to an Amazon S3 bucket
put_bucket_replication	This operation is not supported for directory buckets
put_bucket_request_payment	This operation is not supported for directory buckets
put_bucket_tagging	This operation is not supported for directory buckets
put_bucket_versioning	This operation is not supported for directory buckets
put_bucket_website	This operation is not supported for directory buckets
put_object	Adds an object to a bucket
put_object_acl	This operation is not supported for directory buckets
put_object_legal_hold	This operation is not supported for directory buckets
put_object_lock_configuration	This operation is not supported for directory buckets
put_object_retention	This operation is not supported for directory buckets
put_object_tagging	This operation is not supported for directory buckets
put_public_access_block	This operation is not supported for directory buckets
restore_object	This operation is not supported for directory buckets
select_object_content	This operation is not supported for directory buckets
upload_part	Uploads a part in a multipart upload
upload_part_copy	Uploads a part by copying data from an existing object as data source
write_get_object_response	This operation is not supported for directory buckets

Examples

```
## Not run:
```

```
svc <- s3()
# The following example aborts a multipart upload.
svc$abort_multipart_upload(
  Bucket = "examplebucket",
  Key = "bigobject",
  UploadId = "xadcOB_7YPBOJuoFiQ9cz4P3Pe6FIZwO4f7wN93uHsNBEw97pl5eNwzExg0LA..."
)

## End(Not run)
```

`s3control` *AWS S3 Control*

Description

Amazon Web Services S3 Control provides access to Amazon S3 control plane actions.

Usage

```
s3control(
  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 <- s3control(
  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_access_grants_identity_center	Associate your S3 Access Grants instance with an Amazon Web Servic
create_access_grant	Creates an access grant that gives a grantee access to your S3 data
create_access_grants_instance	Creates an S3 Access Grants instance, which serves as a logical groupi
create_access_grants_location	The S3 data location that you would like to register in your S3 Access G
create_access_point	This operation is not supported by directory buckets
create_access_point_for_object_lambda	This operation is not supported by directory buckets
create_bucket	This action creates an Amazon S3 on Outposts bucket
create_job	This operation creates an S3 Batch Operations job
create_multi_region_access_point	This operation is not supported by directory buckets
create_storage_lens_group	Creates a new S3 Storage Lens group and associates it with the specifie
delete_access_grant	Deletes the access grant from the S3 Access Grants instance
delete_access_grants_instance	Deletes your S3 Access Grants instance
delete_access_grants_instance_resource_policy	Deletes the resource policy of the S3 Access Grants instance
delete_access_grants_location	Deregisters a location from your S3 Access Grants instance
delete_access_point	This operation is not supported by directory buckets
delete_access_point_for_object_lambda	This operation is not supported by directory buckets
delete_access_point_policy	This operation is not supported by directory buckets
delete_access_point_policy_for_object_lambda	This operation is not supported by directory buckets
delete_bucket	This action deletes an Amazon S3 on Outposts bucket
delete_bucket_lifecycle_configuration	This action deletes an Amazon S3 on Outposts bucket's lifecycle config
delete_bucket_policy	This action deletes an Amazon S3 on Outposts bucket policy
delete_bucket_replication	This operation deletes an Amazon S3 on Outposts bucket's replication c
delete_bucket_tagging	This action deletes an Amazon S3 on Outposts bucket's tags
delete_job_tagging	Removes the entire tag set from the specified S3 Batch Operations job
delete_multi_region_access_point	This operation is not supported by directory buckets
delete_public_access_block	This operation is not supported by directory buckets
delete_storage_lens_configuration	This operation is not supported by directory buckets
delete_storage_lens_configuration_tagging	This operation is not supported by directory buckets
delete_storage_lens_group	Deletes an existing S3 Storage Lens group
describe_job	Retrieves the configuration parameters and status for a Batch Operation
describe_multi_region_access_point_operation	This operation is not supported by directory buckets
dissociate_access_grants_identity_center	Dissociates the Amazon Web Services IAM Identity Center instance fro
get_access_grant	Get the details of an access grant from your S3 Access Grants instance
get_access_grants_instance	Retrieves the S3 Access Grants instance for a Region in your account
get_access_grants_instance_for_prefix	Retrieve the S3 Access Grants instance that contains a particular prefix
get_access_grants_instance_resource_policy	Returns the resource policy of the S3 Access Grants instance
get_access_grants_location	Retrieves the details of a particular location registered in your S3 Acces
get_access_point	This operation is not supported by directory buckets
get_access_point_configuration_for_object_lambda	This operation is not supported by directory buckets
get_access_point_for_object_lambda	This operation is not supported by directory buckets

get_access_point_policy	This operation is not supported by directory buckets
get_access_point_policy_for_object_lambda	This operation is not supported by directory buckets
get_access_point_policy_status	This operation is not supported by directory buckets
get_access_point_policy_status_for_object_lambda	This operation is not supported by directory buckets
get_bucket	Gets an Amazon S3 on Outposts bucket
get_bucket_lifecycle_configuration	This action gets an Amazon S3 on Outposts bucket's lifecycle configura
get_bucket_policy	This action gets a bucket policy for an Amazon S3 on Outposts bucket
get_bucket_replication	This operation gets an Amazon S3 on Outposts bucket's replication con
get_bucket_tagging	This action gets an Amazon S3 on Outposts bucket's tags
get_bucket_versioning	This operation returns the versioning state for S3 on Outposts buckets o
get_data_access	Returns a temporary access credential from S3 Access Grants to the gra
get_job_tagging	Returns the tags on an S3 Batch Operations job
get_multi_region_access_point	This operation is not supported by directory buckets
get_multi_region_access_point_policy	This operation is not supported by directory buckets
get_multi_region_access_point_policy_status	This operation is not supported by directory buckets
get_multi_region_access_point_routes	This operation is not supported by directory buckets
get_public_access_block	This operation is not supported by directory buckets
get_storage_lens_configuration	This operation is not supported by directory buckets
get_storage_lens_configuration_tagging	This operation is not supported by directory buckets
get_storage_lens_group	Retrieves the Storage Lens group configuration details
list_access_grants	Returns the list of access grants in your S3 Access Grants instance
list_access_grants_instances	Returns a list of S3 Access Grants instances
list_access_grants_locations	Returns a list of the locations registered in your S3 Access Grants instan
list_access_points	This operation is not supported by directory buckets
list_access_points_for_object_lambda	This operation is not supported by directory buckets
list_caller_access_grants	Use this API to list the access grants that grant the caller access to Ama
list_jobs	Lists current S3 Batch Operations jobs as well as the jobs that have end
list_multi_region_access_points	This operation is not supported by directory buckets
list_regional_buckets	This operation is not supported by directory buckets
list_storage_lens_configurations	This operation is not supported by directory buckets
list_storage_lens_groups	Lists all the Storage Lens groups in the specified home Region
list_tags_for_resource	This operation allows you to list all the Amazon Web Services resource
put_access_grants_instance_resource_policy	Updates the resource policy of the S3 Access Grants instance
put_access_point_configuration_for_object_lambda	This operation is not supported by directory buckets
put_access_point_policy	This operation is not supported by directory buckets
put_access_point_policy_for_object_lambda	This operation is not supported by directory buckets
put_bucket_lifecycle_configuration	This action puts a lifecycle configuration to an Amazon S3 on Outposts
put_bucket_policy	This action puts a bucket policy to an Amazon S3 on Outposts bucket
put_bucket_replication	This action creates an Amazon S3 on Outposts bucket's replication con
put_bucket_tagging	This action puts tags on an Amazon S3 on Outposts bucket
put_bucket_versioning	This operation sets the versioning state for S3 on Outposts buckets only
put_job_tagging	Sets the supplied tag-set on an S3 Batch Operations job
put_multi_region_access_point_policy	This operation is not supported by directory buckets
put_public_access_block	This operation is not supported by directory buckets
put_storage_lens_configuration	This operation is not supported by directory buckets
put_storage_lens_configuration_tagging	This operation is not supported by directory buckets
submit_multi_region_access_point_routes	This operation is not supported by directory buckets
tag_resource	Creates a new Amazon Web Services resource tag or updates an existin

untag_resource	This operation removes the specified Amazon Web Services resource ta
update_access_grants_location	Updates the IAM role of a registered location in your S3 Access Grants
update_job_priority	Updates an existing S3 Batch Operations job's priority
update_job_status	Updates the status for the specified job
update_storage_lens_group	Updates the existing Storage Lens group

Examples

```
## Not run:
svc <- s3control()
svc$associate_access_grants_identity_center(
  Foo = 123
)

## End(Not run)
```

`s3outposts` *Amazon S3 on Outposts*

Description

Amazon S3 on Outposts provides access to S3 on Outposts operations.

Usage

```
s3outposts(
  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 <- s3outposts(
  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_endpoint	Creates an endpoint and associates it with the specified Outpost
delete_endpoint	Deletes an endpoint
list_endpoints	Lists endpoints associated with the specified Outpost
list_outposts_with_s3	Lists the Outposts with S3 on Outposts capacity for your Amazon Web Services account
list_shared_endpoints	Lists all endpoints associated with an Outpost that has been shared by Amazon Web Services Resourc

Examples

```
## Not run:
svc <- s3outposts()
svc$create_endpoint(
  Foo = 123
)

## End(Not run)
```

---

s3tables *Amazon S3 Tables*

Description

An Amazon S3 table represents a structured dataset consisting of tabular data in Apache Parquet format and related metadata. This data is stored inside an S3 table as a subresource. All tables in a table bucket are stored in the Apache Iceberg table format. Through integration with the AWS Glue Data Catalog you can interact with your tables using AWS analytics services, such as Amazon Athena and Amazon Redshift. Amazon S3 manages maintenance of your tables through automatic file compaction and snapshot management. For more information, see Amazon S3 table buckets.

Usage

```
s3tables(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 <- s3tables(
  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_namespace	Creates a namespace
create_table	Creates a new table associated with the given namespace in a table bucket
create_table_bucket	Creates a table bucket
delete_namespace	Deletes a namespace
delete_table	Deletes a table
delete_table_bucket	Deletes a table bucket
delete_table_bucket_policy	Deletes a table bucket policy
delete_table_policy	Deletes a table policy
get_namespace	Gets details about a namespace
get_table	Gets details about a table
get_table_bucket	Gets details on a table bucket
get_table_bucket_maintenance_configuration	Gets details about a maintenance configuration for a given table bucket
get_table_bucket_policy	Gets details about a table bucket policy

get_table_maintenance_configuration	Gets details about the maintenance configuration of a table
get_table_maintenance_job_status	Gets the status of a maintenance job for a table
get_table_metadata_location	Gets the location of the table metadata
get_table_policy	Gets details about a table policy
list_namespaces	Lists the namespaces within a table bucket
list_table_buckets	Lists table buckets for your account
list_tables	List tables in the given table bucket
put_table_bucket_maintenance_configuration	Creates a new maintenance configuration or replaces an existing maintenance
put_table_bucket_policy	Creates a new maintenance configuration or replaces an existing table bucket
put_table_maintenance_configuration	Creates a new maintenance configuration or replaces an existing maintenance
put_table_policy	Creates a new maintenance configuration or replaces an existing table policy f
rename_table	Renames a table or a namespace
update_table_metadata_location	Updates the metadata location for a table

Examples

```
## Not run:
svc <- s3tables()
svc$create_namespace(
  Foo = 123
)

## End(Not run)
```

`storagegateway` *AWS Storage Gateway*

Description

Storage Gateway Service

Amazon FSx File Gateway is no longer available to new customers. Existing customers of FSx File Gateway can continue to use the service normally. For capabilities similar to FSx File Gateway, visit this blog post.

Storage Gateway is the service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and the Amazon Web Services storage infrastructure. The service enables you to securely upload data to the Amazon Web Services Cloud for cost effective backup and rapid disaster recovery.

Use the following links to get started using the *Storage Gateway Service API Reference*:

- Storage Gateway required request headers: Describes the required headers that you must send with every POST request to Storage Gateway.
- Signing requests: Storage Gateway requires that you authenticate every request you send; this topic describes how sign such a request.

- Error responses: Provides reference information about Storage Gateway errors.
- Operations in Storage Gateway: Contains detailed descriptions of all Storage Gateway operations, their request parameters, response elements, possible errors, and examples of requests and responses.
- Storage Gateway endpoints and quotas: Provides a list of each Amazon Web Services Region and the endpoints available for use with Storage Gateway.

Storage Gateway resource IDs are in uppercase. When you use these resource IDs with the Amazon EC2 API, EC2 expects resource IDs in lowercase. You must change your resource ID to lowercase to use it with the EC2 API. For example, in Storage Gateway the ID for a volume might be `vol-AA22BB012345DAF670`. When you use this ID with the EC2 API, you must change it to `vol-aa22bb012345daf670`. Otherwise, the EC2 API might not behave as expected.

IDs for Storage Gateway volumes and Amazon EBS snapshots created from gateway volumes are changing to a longer format. Starting in December 2016, all new volumes and snapshots will be created with a 17-character string. Starting in April 2016, you will be able to use these longer IDs so you can test your systems with the new format. For more information, see Longer EC2 and EBS resource IDs.

For example, a volume Amazon Resource Name (ARN) with the longer volume ID format looks like the following:
`arn:aws:storagegateway:us-west-2:111122223333:gateway/sgw-12A3456B/volume/vol-1122AABBCCDDEEFFG`.

A snapshot ID with the longer ID format looks like the following: `snap-78e226633445566ee`.

For more information, see Announcement: Heads-up – Longer Storage Gateway volume and snapshot IDs coming in 2016.

Usage

```
storagegateway(
  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 <- storagegateway(
  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_gateway	Activates the gateway you previously deployed on your host
add_cache	Configures one or more gateway local disks as cache for a gateway
add_tags_to_resource	Adds one or more tags to the specified resource
add_upload_buffer	Configures one or more gateway local disks as upload buffer for a specified g
add_working_storage	Configures one or more gateway local disks as working storage for a gatewa
assign_tape_pool	Assigns a tape to a tape pool for archiving
associate_file_system	Associate an Amazon FSx file system with the FSx File Gateway
attach_volume	Connects a volume to an iSCSI connection and then attaches the volume to t
cancel_archival	Cancels archiving of a virtual tape to the virtual tape shelf (VTS) after the ar
cancel_cache_report	Cancels generation of a specified cache report
cancel_retrieval	Cancels retrieval of a virtual tape from the virtual tape shelf (VTS) to a gatev
create_cachedi_scsi_volume	Creates a cached volume on a specified cached volume gateway
create_nfs_file_share	Creates a Network File System (NFS) file share on an existing S3 File Gatev
create_smb_file_share	Creates a Server Message Block (SMB) file share on an existing S3 File Gat
create_snapshot	Initiates a snapshot of a volume
create_snapshot_from_volume_recovery_point	Initiates a snapshot of a gateway from a volume recovery point
create_storedi_scsi_volume	Creates a volume on a specified gateway
create_tape_pool	Creates a new custom tape pool
create_tapes	Creates one or more virtual tapes
create_tape_with_barcode	Creates a virtual tape by using your own barcode
delete_automatic_tape_creation_policy	Deletes the automatic tape creation policy of a gateway
delete_bandwidth_rate_limit	Deletes the bandwidth rate limits of a gateway
delete_cache_report	Deletes the specified cache report and any associated tags from the Storage G
delete_chap_credentials	Deletes Challenge-Handshake Authentication Protocol (CHAP) credentials f
delete_file_share	Deletes a file share from an S3 File Gateway
delete_gateway	Deletes a gateway
delete_snapshot_schedule	Deletes a snapshot of a volume
delete_tape	Deletes the specified virtual tape
delete_tape_archive	Deletes the specified virtual tape from the virtual tape shelf (VTS)
delete_tape_pool	Delete a custom tape pool
delete_volume	Deletes the specified storage volume that you previously created using the C
describe_availability_monitor_test	Returns information about the most recent high availability monitoring test t
describe_bandwidth_rate_limit	Returns the bandwidth rate limits of a gateway
describe_bandwidth_rate_limit_schedule	Returns information about the bandwidth rate limit schedule of a gateway

describe_cache	Returns information about the cache of a gateway
describe_cachedi_scsi_volumes	Returns a description of the gateway volumes specified in the request
describe_cache_report	Returns information about the specified cache report, including completion s
describe_chap_credentials	Returns an array of Challenge-Handshake Authentication Protocol (CHAP)
describe_file_system_associations	Gets the file system association information
describe_gateway_information	Returns metadata about a gateway such as its name, network interfaces, time
describe_maintenance_start_time	Returns your gateway's maintenance window schedule information, with val
describe_nfs_file_shares	Gets a description for one or more Network File System (NFS) file shares fr
describe_smb_file_shares	Gets a description for one or more Server Message Block (SMB) file shares
describe_smb_settings	Gets a description of a Server Message Block (SMB) file share settings from
describe_snapshot_schedule	Describes the snapshot schedule for the specified gateway volume
describe_storedi_scsi_volumes	Returns the description of the gateway volumes specified in the request
describe_tape_archives	Returns a description of specified virtual tapes in the virtual tape shelf (VTS
describe_tape_recovery_points	Returns a list of virtual tape recovery points that are available for the specifi
describe_tapes	Returns a description of virtual tapes that correspond to the specified Amazo
describe_upload_buffer	Returns information about the upload buffer of a gateway
describe_vtl_devices	Returns a description of virtual tape library (VTL) devices for the specified t
describe_working_storage	Returns information about the working storage of a gateway
detach_volume	Disconnects a volume from an iSCSI connection and then detaches the volu
disable_gateway	Disables a tape gateway when the gateway is no longer functioning
disassociate_file_system	Disassociates an Amazon FSx file system from the specified gateway
join_domain	Adds a file gateway to an Active Directory domain
list_automatic_tape_creation_policies	Lists the automatic tape creation policies for a gateway
list_cache_reports	Returns a list of existing cache reports for all file shares associated with you
list_file_shares	Gets a list of the file shares for a specific S3 File Gateway, or the list of file s
list_file_system_associations	Gets a list of FileSystemAssociationSummary objects
list_gateways	Lists gateways owned by an Amazon Web Services account in an Amazon W
list_local_disks	Returns a list of the gateway's local disks
list_tags_for_resource	Lists the tags that have been added to the specified resource
list_tape_pools	Lists custom tape pools
list_tapes	Lists virtual tapes in your virtual tape library (VTL) and your virtual tape sh
list_volume_initiators	Lists iSCSI initiators that are connected to a volume
list_volume_recovery_points	Lists the recovery points for a specified gateway
list_volumes	Lists the iSCSI stored volumes of a gateway
notify_when_uploaded	Sends you notification through Amazon EventBridge when all files written t
refresh_cache	Refreshes the cached inventory of objects for the specified file share
remove_tags_from_resource	Removes one or more tags from the specified resource
reset_cache	Resets all cache disks that have encountered an error and makes the disks av
retrieve_tape_archive	Retrieves an archived virtual tape from the virtual tape shelf (VTS) to a tape
retrieve_tape_recovery_point	Retrieves the recovery point for the specified virtual tape
set_local_console_password	Sets the password for your VM local console
set_smb_guest_password	Sets the password for the guest user smbguest
shutdown_gateway	Shuts down a Tape Gateway or Volume Gateway
start_availability_monitor_test	Start a test that verifies that the specified gateway is configured for High Ava
start_cache_report	Starts generating a report of the file metadata currently cached by an S3 File
start_gateway	Starts a gateway that you previously shut down (see ShutdownGateway)
update_automatic_tape_creation_policy	Updates the automatic tape creation policy of a gateway
update_bandwidth_rate_limit	Updates the bandwidth rate limits of a gateway

update_bandwidth_rate_limit_schedule	Updates the bandwidth rate limit schedule for a specified gateway
update_chap_credentials	Updates the Challenge-Handshake Authentication Protocol (CHAP) credent
update_file_system_association	Updates a file system association
update_gateway_information	Updates a gateway's metadata, which includes the gateway's name, time zor
update_gateway_software_now	Updates the gateway virtual machine (VM) software
update_maintenance_start_time	Updates a gateway's maintenance window schedule, with settings for month
update_nfs_file_share	Updates a Network File System (NFS) file share
update_smb_file_share	Updates a Server Message Block (SMB) file share
update_smb_file_share_visibility	Controls whether the shares on an S3 File Gateway are visible in a net view
update_smb_local_groups	Updates the list of Active Directory users and groups that have special perm
update_smb_security_strategy	Updates the SMB security strategy level for an Amazon S3 file gateway
update_snapshot_schedule	Updates a snapshot schedule configured for a gateway volume
update_vtl_device_type	Updates the type of medium changer in a tape gateway

Examples

```
## Not run:
svc <- storagegateway()
# Activates the gateway you previously deployed on your host.
svc$activate_gateway(
  ActivationKey = "29AV1-3OFV9-VVIUB-NKT0I-LRO6V",
  GatewayName = "My_Gateway",
  GatewayRegion = "us-east-1",
  GatewayTimezone = "GMT-12:00",
  GatewayType = "STORED",
  MediumChangerType = "AWS-Gateway-VTL",
  TapeDriveType = "IBM-ULT3580-TD5"
)

## End(Not run)
```

Index