# Package 'paws.security.identity'

July 23, 2025

**Title** 'Amazon Web Services' Security, Identity, & Compliance Services

**Version** 0.9.0

**Description** Interface to 'Amazon Web Services' security, identity, and compliance services, including the 'Identity & Access Management' ('IAM') service for managing access to services and resources, 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.security.identity

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

**Imports** paws.common (>= 0.8.0)

**Suggests** testthat

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Collate** 'accessanalyzer_service.R' 'accessanalyzer_interfaces.R' 'accessanalyzer_operations.R' 'account_service.R' 'account_interfaces.R' 'account_operations.R' 'acm_service.R' 'acm_interfaces.R' 'acm_operations.R' 'acmpca_service.R' 'acmpca_interfaces.R' 'acmpca_operations.R' 'cleanroomsml_service.R' 'cleanroomsml_interfaces.R' 'cleanroomsml_operations.R' 'clouddirectory_service.R' 'clouddirectory_interfaces.R' 'clouddirectory_operations.R' 'cloudhsm_service.R' 'cloudhsm_interfaces.R' 'cloudhsm_operations.R' 'cloudhsmv2_service.R' 'cloudhsmv2_interfaces.R' 'cloudhsmv2_operations.R' 'cognitoidentity_service.R' 'cognitoidentity_interfaces.R' 'cognitoidentity_operations.R' 'cognitoidentityprovider_service.R' 'cognitoidentityprovider_interfaces.R' 'cognitoidentityprovider_operations.R' 'cognitosync_service.R' 'cognitosync_interfaces.R' 'cognitosync_operations.R' 'detective_service.R' 'detective_interfaces.R'

'detective_operations.R' 'directoryservice_service.R'
'directoryservice_interfaces.R' 'directoryservice_operations.R'
'fms_service.R' 'fms_interfaces.R' 'fms_operations.R'
'guardduty_service.R' 'guardduty_interfaces.R'
'guardduty_operations.R' 'iam_service.R' 'iam_interfaces.R'
'iam_operations.R' 'iamrolesanywhere_service.R'
'iamrolesanywhere_interfaces.R' 'iamrolesanywhere_operations.R'
'identitystore_service.R' 'identitystore_interfaces.R'
'identitystore_operations.R' 'inspector2_service.R'
'inspector2_interfaces.R' 'inspector2_operations.R'
'inspector_service.R' 'inspector_interfaces.R'
'inspector_operations.R' 'kms_service.R' 'kms_interfaces.R'
'kms_operations.R' 'macie2_service.R' 'macie2_interfaces.R'
'macie2_operations.R' 'pcaconnectorad_service.R'
'pcaconnectorad_interfaces.R' 'pcaconnectorad_operations.R'
'ram_service.R' 'ram_interfaces.R' 'ram_operations.R'
'reexports_paws.common.R' 'secretsmanager_service.R'
'secretsmanager_interfaces.R' 'secretsmanager_operations.R'
'securityhub_service.R' 'securityhub_interfaces.R'
'securityhub_operations.R' 'securitylake_service.R'
'securitylake_interfaces.R' 'securitylake_operations.R'
'shield_service.R' 'shield_interfaces.R' 'shield_operations.R'
'sso_service.R' 'sso_interfaces.R' 'sso_operations.R'
'ssoadmin_service.R' 'ssoadmin_interfaces.R'
'ssoadmin_operations.R' 'ssooidc_service.R'
'ssooidc_interfaces.R' 'ssooidc_operations.R' 'sts_service.R'
'sts_interfaces.R' 'sts_operations.R'
'verifiedpermissions_service.R'
'verifiedpermissions_interfaces.R'
'verifiedpermissions_operations.R' 'waf_service.R'
'waf_interfaces.R' 'waf_operations.R' 'wafregional_service.R'
'wafregional_interfaces.R' 'wafregional_operations.R'
'wafv2_service.R' 'wafv2_interfaces.R' 'wafv2_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 16:50:02 UTC

# Contents

---

`accessanalyzer` *Access Analyzer*

---

## Description

Identity and Access Management Access Analyzer helps you to set, verify, and refine your IAM policies by providing a suite of capabilities. Its features include findings for external and unused access, basic and custom policy checks for validating policies, and policy generation to generate fine-grained policies. To start using IAM Access Analyzer to identify external or unused access, you first need to create an analyzer.

**External access analyzers** help identify potential risks of accessing resources by enabling you to identify any resource policies that grant access to an external principal. It does this by using logic-based reasoning to analyze resource-based policies in your Amazon Web Services environment. An external principal can be another Amazon Web Services account, a root user, an IAM user or role, a federated user, an Amazon Web Services service, or an anonymous user. You can also use IAM Access Analyzer to preview public and cross-account access to your resources before deploying permissions changes.

**Unused access analyzers** help identify potential identity access risks by enabling you to identify unused IAM roles, unused access keys, unused console passwords, and IAM principals with unused service and action-level permissions.

Beyond findings, IAM Access Analyzer provides basic and custom policy checks to validate IAM policies before deploying permissions changes. You can use policy generation to refine permissions by attaching a policy generated using access activity logged in CloudTrail logs.

This guide describes the IAM Access Analyzer operations that you can call programmatically. For general information about IAM Access Analyzer, see Identity and Access Management Access Analyzer in the **IAM User Guide**.

## Usage

```
accessanalyzer(
  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 <- accessanalyzer(
  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

| | |
|---|---|
| apply_archive_rule | Retroactively applies the archive rule to existing findings that meet the archive rule criter |
| cancel_policy_generation | Cancels the requested policy generation |
| check_access_not_granted | Checks whether the specified access isn't allowed by a policy |
| check_no_new_access | Checks whether new access is allowed for an updated policy when compared to the exist |
| check_no_public_access | Checks whether a resource policy can grant public access to the specified resource type |
| create_access_preview | Creates an access preview that allows you to preview IAM Access Analyzer findings for |
| create_analyzer | Creates an analyzer for your account |
| create_archive_rule | Creates an archive rule for the specified analyzer |
| delete_analyzer | Deletes the specified analyzer |
| delete_archive_rule | Deletes the specified archive rule |
| generate_finding_recommendation | Creates a recommendation for an unused permissions finding |
| get_access_preview | Retrieves information about an access preview for the specified analyzer |
| get_analyzed_resource | Retrieves information about a resource that was analyzed |
| get_analyzer | Retrieves information about the specified analyzer |
| get_archive_rule | Retrieves information about an archive rule |
| get_finding | Retrieves information about the specified finding |
| get_finding_recommendation | Retrieves information about a finding recommendation for the specified analyzer |
| get_findings_statistics | Retrieves a list of aggregated finding statistics for an external access or unused access an |
| get_finding_v2 | Retrieves information about the specified finding |
| get_generated_policy | Retrieves the policy that was generated using StartPolicyGeneration |
| list_access_preview_findings | Retrieves a list of access preview findings generated by the specified access preview |
| list_access_previews | Retrieves a list of access previews for the specified analyzer |
| list_analyzed_resources | Retrieves a list of resources of the specified type that have been analyzed by the specifie |
| list_analyzers | Retrieves a list of analyzers |
| list_archive_rules | Retrieves a list of archive rules created for the specified analyzer |
| list_findings | Retrieves a list of findings generated by the specified analyzer |
| list_findings_v2 | Retrieves a list of findings generated by the specified analyzer |
| list_policy_generations | Lists all of the policy generations requested in the last seven days |
| list_tags_for_resource | Retrieves a list of tags applied to the specified resource |
| start_policy_generation | Starts the policy generation request |
| start_resource_scan | Immediately starts a scan of the policies applied to the specified resource |
| tag_resource | Adds a tag to the specified resource |
| untag_resource | Removes a tag from the specified resource |
| update_analyzer | Modifies the configuration of an existing analyzer |
| update_archive_rule | Updates the criteria and values for the specified archive rule |
| update_findings | Updates the status for the specified findings |
| validate_policy | Requests the validation of a policy and returns a list of findings |

## Examples

```
## Not run:
```

```
svc <- accessanalyzer()
svc$apply_archive_rule(
  Foo = 123
)

## End(Not run)
```

---

`account` *AWS Account*

---

## Description

Operations for Amazon Web Services Account Management

## Usage

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

## Operations

| | |
|---|---|
| accept_primary_email_update | Accepts the request that originated from StartPrimaryEmailUpdate to update the primary ema |
| delete_alternate_contact | Deletes the specified alternate contact from an Amazon Web Services account |
| disable_region | Disables (opts-out) a particular Region for an account |
| enable_region | Enables (opts-in) a particular Region for an account |
| get_alternate_contact | Retrieves the specified alternate contact attached to an Amazon Web Services account |
| get_contact_information | Retrieves the primary contact information of an Amazon Web Services account |
| get_primary_email | Retrieves the primary email address for the specified account |
| get_region_opt_status | Retrieves the opt-in status of a particular Region |
| list_regions | Lists all the Regions for a given account and their respective opt-in statuses |
| put_alternate_contact | Modifies the specified alternate contact attached to an Amazon Web Services account |
| put_contact_information | Updates the primary contact information of an Amazon Web Services account |
| start_primary_email_update | Starts the process to update the primary email address for the specified account |

## Examples

```
## Not run:
svc <- account()
svc$accept_primary_email_update(
  Foo = 123
)

## End(Not run)
```

---

`acm` *AWS Certificate Manager*

---

## Description

Certificate Manager

You can use Certificate Manager (ACM) to manage SSL/TLS certificates for your Amazon Web Services-based websites and applications. For more information about using ACM, see the Certificate Manager User Guide.

## Usage

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

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

## Operations

| | |
|---|---|
| add_tags_to_certificate | Adds one or more tags to an ACM certificate |
| delete_certificate | Deletes a certificate and its associated private key |
| describe_certificate | Returns detailed metadata about the specified ACM certificate |
| export_certificate | Exports a private certificate issued by a private certificate authority (CA) for use anywhere |
| get_account_configuration | Returns the account configuration options associated with an Amazon Web Services account |
| get_certificate | Retrieves a certificate and its certificate chain |
| import_certificate | Imports a certificate into Certificate Manager (ACM) to use with services that are integrated w |
| list_certificates | Retrieves a list of certificate ARNs and domain names |
| list_tags_for_certificate | Lists the tags that have been applied to the ACM certificate |
| put_account_configuration | Adds or modifies account-level configurations in ACM |
| remove_tags_from_certificate | Remove one or more tags from an ACM certificate |
| renew_certificate | Renews an eligible ACM certificate |
| request_certificate | Requests an ACM certificate for use with other Amazon Web Services services |
| resend_validation_email | Resends the email that requests domain ownership validation |
| update_certificate_options | Updates a certificate |

## Examples

```
## Not run:
svc <- acm()
svc$add_tags_to_certificate(
  Foo = 123
)

## End(Not run)
```

acmpca *AWS Certificate Manager Private Certificate Authority*

## Description

This is the *Amazon Web Services Private Certificate Authority API Reference*. It provides descriptions, syntax, and usage examples for each of the actions and data types involved in creating and managing a private certificate authority (CA) for your organization.

The documentation for each action shows the API request parameters and the JSON response. Alternatively, you can use one of the Amazon Web Services SDKs to access an API that is tailored to the programming language or platform that you prefer. For more information, see Amazon Web Services SDKs.

Each Amazon Web Services Private CA API operation has a quota that determines the number of times the operation can be called per second. Amazon Web Services Private CA throttles API requests at different rates depending on the operation. Throttling means that Amazon Web Services Private CA rejects an otherwise valid request because the request exceeds the operation's quota for the number of requests per second. When a request is throttled, Amazon Web Services Private CA returns a ThrottlingException error. Amazon Web Services Private CA does not guarantee a minimum request rate for APIs.

To see an up-to-date list of your Amazon Web Services Private CA quotas, or to request a quota increase, log into your Amazon Web Services account and visit the Service Quotas console.

## Usage

```
acmpca(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 <- acmpca(
  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_certificate_authority | Creates a root or subordinate private certificate authority (CA) |
| create_certificate_authority_audit_report | Creates an audit report that lists every time that your CA private key is used to is |
| create_permission | Grants one or more permissions on a private CA to the Certificate Manager (AC |
| delete_certificate_authority | Deletes a private certificate authority (CA) |
| delete_permission | Revokes permissions on a private CA granted to the Certificate Manager (ACM) |
| delete_policy | Deletes the resource-based policy attached to a private CA |
| describe_certificate_authority | Lists information about your private certificate authority (CA) or one that has be |
| describe_certificate_authority_audit_report | Lists information about a specific audit report created by calling the CreateCerti |
| get_certificate | Retrieves a certificate from your private CA or one that has been shared with you |
| get_certificate_authority_certificate | Retrieves the certificate and certificate chain for your private certificate authority |
| get_certificate_authority_csr | Retrieves the certificate signing request (CSR) for your private certificate author |
| get_policy | Retrieves the resource-based policy attached to a private CA |
| import_certificate_authority_certificate | Imports a signed private CA certificate into Amazon Web Services Private CA |
| issue_certificate | Uses your private certificate authority (CA), or one that has been shared with yo |
| list_certificate_authorities | Lists the private certificate authorities that you created by using the CreateCertif |
| list_permissions | List all permissions on a private CA, if any, granted to the Certificate Manager ( |
| list_tags | Lists the tags, if any, that are associated with your private CA or one that has bee |
| put_policy | Attaches a resource-based policy to a private CA |
| restore_certificate_authority | Restores a certificate authority (CA) that is in the DELETED state |
| revoke_certificate | Revokes a certificate that was issued inside Amazon Web Services Private CA |
| tag_certificate_authority | Adds one or more tags to your private CA |
| untag_certificate_authority | Remove one or more tags from your private CA |
| update_certificate_authority | Updates the status or configuration of a private certificate authority (CA) |

## Examples

```
## Not run:
svc <- acmpca()
svc$create_certificate_authority(
  Foo = 123
)

## End(Not run)
```

`cleanroomsml` *AWS Clean Rooms ML*

## Description

Welcome to the *Amazon Web Services Clean Rooms ML API Reference*.

Amazon Web Services Clean Rooms ML provides a privacy-enhancing method for two parties to identify similar users in their data without the need to share their data with each other. The first party brings the training data to Clean Rooms so that they can create and configure an audience model (lookalike model) and associate it with a collaboration. The second party then brings their seed data to Clean Rooms and generates an audience (lookalike segment) that resembles the training data.

To learn more about Amazon Web Services Clean Rooms ML concepts, procedures, and best practices, see the Clean Rooms User Guide.

To learn more about SQL commands, functions, and conditions supported in Clean Rooms, see the Clean Rooms SQL Reference.

## Usage

```
cleanroomsml(
  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 <- cleanroomsml(
  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_trained_model | Submits a request to cancel the trained model job |
| cancel_trained_model_inference_job | Submits a request to cancel a trained model inference job |
| create_audience_model | Defines the information necessary to create an audience mode |
| create_configured_audience_model | Defines the information necessary to create a configured audie |
| create_configured_model_algorithm | Creates a configured model algorithm using a container image |
| create_configured_model_algorithm_association | Associates a configured model algorithm to a collaboration fo |
| create_ml_input_channel | Provides the information to create an ML input channel |
| create_trained_model | Creates a trained model from an associated configured model |
| create_training_dataset | Defines the information necessary to create a training dataset |
| delete_audience_generation_job | Deletes the specified audience generation job, and removes all |
| delete_audience_model | Specifies an audience model that you want to delete |
| delete_configured_audience_model | Deletes the specified configured audience model |
| delete_configured_audience_model_policy | Deletes the specified configured audience model policy |
| delete_configured_model_algorithm | Deletes a configured model algorithm |
| delete_configured_model_algorithm_association | Deletes a configured model algorithm association |
| delete_ml_configuration | Deletes a ML modeling configuration |
| delete_ml_input_channel_data | Provides the information necessary to delete an ML input cha |
| delete_trained_model_output | Deletes the output of a trained model |
| delete_training_dataset | Specifies a training dataset that you want to delete |
| get_audience_generation_job | Returns information about an audience generation job |
| get_audience_model | Returns information about an audience model |
| get_collaboration_configured_model_algorithm_association | Returns information about the configured model algorithm ass |
| get_collaboration_ml_input_channel | Returns information about a specific ML input channel in a co |
| get_collaboration_trained_model | Returns information about a trained model in a collaboration |
| get_configured_audience_model | Returns information about a specified configured audience mo |
| get_configured_audience_model_policy | Returns information about a configured audience model policy |
| get_configured_model_algorithm | Returns information about a configured model algorithm |
| get_configured_model_algorithm_association | Returns information about a configured model algorithm asso |
| get_ml_configuration | Returns information about a specific ML configuration |
| get_ml_input_channel | Returns information about an ML input channel |
| get_trained_model | Returns information about a trained model |
| get_trained_model_inference_job | Returns information about a trained model inference job |
| get_training_dataset | Returns information about a training dataset |
| list_audience_export_jobs | Returns a list of the audience export jobs |
| list_audience_generation_jobs | Returns a list of audience generation jobs |
| list_audience_models | Returns a list of audience models |
| list_collaboration_configured_model_algorithm_associations | Returns a list of the configured model algorithm associations i |
| list_collaboration_ml_input_channels | Returns a list of the ML input channels in a collaboration |
| list_collaboration_trained_model_export_jobs | Returns a list of the export jobs for a trained model in a collab |
| list_collaboration_trained_model_inference_jobs | Returns a list of trained model inference jobs in a specified co |

| | |
|---|---|
| list_collaboration_trained_models | Returns a list of the trained models in a collaboration |
| list_configured_audience_models | Returns a list of the configured audience models |
| list_configured_model_algorithm_associations | Returns a list of configured model algorithm associations |
| list_configured_model_algorithms | Returns a list of configured model algorithms |
| list_ml_input_channels | Returns a list of ML input channels |
| list_tags_for_resource | Returns a list of tags for a provided resource |
| list_trained_model_inference_jobs | Returns a list of trained model inference jobs that match the re |
| list_trained_models | Returns a list of trained models |
| list_training_datasets | Returns a list of training datasets |
| put_configured_audience_model_policy | Create or update the resource policy for a configured audience |
| put_ml_configuration | Assigns information about an ML configuration |
| start_audience_export_job | Export an audience of a specified size after you have generate |
| start_audience_generation_job | Information necessary to start the audience generation job |
| start_trained_model_export_job | Provides the information necessary to start a trained model ex |
| start_trained_model_inference_job | Defines the information necessary to begin a trained model inf |
| tag_resource | Adds metadata tags to a specified resource |
| untag_resource | Removes metadata tags from a specified resource |
| update_configured_audience_model | Provides the information necessary to update a configured aud |

## Examples

```
## Not run:
svc <- cleanroomsml()
svc$cancel_trained_model(
  Foo = 123
)

## End(Not run)
```

---

`clouddirectory` *Amazon CloudDirectory*

---

## Description

Amazon Cloud Directory

Amazon Cloud Directory is a component of the AWS Directory Service that simplifies the development and management of cloud-scale web, mobile, and IoT applications. This guide describes the Cloud Directory operations that you can call programmatically and includes detailed information on data types and errors. For information about Cloud Directory features, see AWS Directory Service and the Amazon Cloud Directory Developer Guide.

## Usage

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

### Operations

| | |
|---|---|
| add_facet_to_object | Adds a new Facet to an object |
| apply_schema | Copies the input published schema, at the specified version, into the Directory with the sa |
| attach_object | Attaches an existing object to another object |
| attach_policy | Attaches a policy object to a regular object |
| attach_to_index | Attaches the specified object to the specified index |
| attach_typed_link | Attaches a typed link to a specified source and target object |
| batch_read | Performs all the read operations in a batch |
| batch_write | Performs all the write operations in a batch |
| create_directory | Creates a Directory by copying the published schema into the directory |
| create_facet | Creates a new Facet in a schema |
| create_index | Creates an index object |
| create_object | Creates an object in a Directory |
| create_schema | Creates a new schema in a development state |

| | |
|---|---|
| create_typed_link_facet | Creates a TypedLinkFacet |
| delete_directory | Deletes a directory |
| delete_facet | Deletes a given Facet |
| delete_object | Deletes an object and its associated attributes |
| delete_schema | Deletes a given schema |
| delete_typed_link_facet | Deletes a TypedLinkFacet |
| detach_from_index | Detaches the specified object from the specified index |
| detach_object | Detaches a given object from the parent object |
| detach_policy | Detaches a policy from an object |
| detach_typed_link | Detaches a typed link from a specified source and target object |
| disable_directory | Disables the specified directory |
| enable_directory | Enables the specified directory |
| get_applied_schema_version | Returns current applied schema version ARN, including the minor version in use |
| get_directory | Retrieves metadata about a directory |
| get_facet | Gets details of the Facet, such as facet name, attributes, Rules, or ObjectType |
| get_link_attributes | Retrieves attributes that are associated with a typed link |
| get_object_attributes | Retrieves attributes within a facet that are associated with an object |
| get_object_information | Retrieves metadata about an object |
| get_schema_as_json | Retrieves a JSON representation of the schema |
| get_typed_link_facet_information | Returns the identity attribute order for a specific TypedLinkFacet |
| list_applied_schema_arns | Lists schema major versions applied to a directory |
| list_attached_indices | Lists indices attached to the specified object |
| list_development_schema_arns | Retrieves each Amazon Resource Name (ARN) of schemas in the development state |
| list_directories | Lists directories created within an account |
| list_facet_attributes | Retrieves attributes attached to the facet |
| list_facet_names | Retrieves the names of facets that exist in a schema |
| list_incoming_typed_links | Returns a paginated list of all the incoming TypedLinkSpecifier information for an object |
| list_index | Lists objects attached to the specified index |
| list_managed_schema_arns | Lists the major version families of each managed schema |
| list_object_attributes | Lists all attributes that are associated with an object |
| list_object_children | Returns a paginated list of child objects that are associated with a given object |
| list_object_parent_paths | Retrieves all available parent paths for any object type such as node, leaf node, policy noc |
| list_object_parents | Lists parent objects that are associated with a given object in pagination fashion |
| list_object_policies | Returns policies attached to an object in pagination fashion |
| list_outgoing_typed_links | Returns a paginated list of all the outgoing TypedLinkSpecifier information for an object |
| list_policy_attachments | Returns all of the ObjectIdentifiers to which a given policy is attached |
| list_published_schema_arns | Lists the major version families of each published schema |
| list_tags_for_resource | Returns tags for a resource |
| list_typed_link_facet_attributes | Returns a paginated list of all attribute definitions for a particular TypedLinkFacet |
| list_typed_link_facet_names | Returns a paginated list of TypedLink facet names for a particular schema |
| lookup_policy | Lists all policies from the root of the Directory to the object specified |
| publish_schema | Publishes a development schema with a major version and a recommended minor version |
| put_schema_from_json | Allows a schema to be updated using JSON upload |
| remove_facet_from_object | Removes the specified facet from the specified object |
| tag_resource | An API operation for adding tags to a resource |
| untag_resource | An API operation for removing tags from a resource |
| update_facet | Does the following: |
| update_link_attributes | Updates a given typed link's attributes |

| | |
|---|---|
| update_object_attributes | Updates a given object's attributes |
| update_schema | Updates the schema name with a new name |
| update_typed_link_facet | Updates a TypedLinkFacet |
| upgrade_applied_schema | Upgrades a single directory in-place using the PublishedSchemaArn with schema updates |
| upgrade_published_schema | Upgrades a published schema under a new minor version revision using the current conte |

## Examples

```
## Not run:
svc <- clouddirectory()
svc$add_facet_to_object(
  Foo = 123
)

## End(Not run)
```

---

`cloudhsm` *Amazon CloudHSM*

---

## Description

AWS CloudHSM Service

This is documentation for **AWS CloudHSM Classic**. For more information, see AWS CloudHSM Classic FAQs, the AWS CloudHSM Classic User Guide, and the AWS CloudHSM Classic API Reference.

**For information about the current version of AWS CloudHSM**, see AWS CloudHSM, the AWS CloudHSM User Guide, and the AWS CloudHSM API Reference.

## Usage

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

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

## Operations

| | |
|---|---|
| add_tags_to_resource | This is documentation for AWS CloudHSM Classic |
| create_hapg | This is documentation for AWS CloudHSM Classic |
| create_hsm | This is documentation for AWS CloudHSM Classic |
| create_luna_client | This is documentation for AWS CloudHSM Classic |
| delete_hapg | This is documentation for AWS CloudHSM Classic |
| delete_hsm | This is documentation for AWS CloudHSM Classic |
| delete_luna_client | This is documentation for AWS CloudHSM Classic |
| describe_hapg | This is documentation for AWS CloudHSM Classic |
| describe_hsm | This is documentation for AWS CloudHSM Classic |
| describe_luna_client | This is documentation for AWS CloudHSM Classic |
| get_config | This is documentation for AWS CloudHSM Classic |
| list_available_zones | This is documentation for AWS CloudHSM Classic |
| list_hapgs | This is documentation for AWS CloudHSM Classic |
| list_hsms | This is documentation for AWS CloudHSM Classic |
| list_luna_clients | This is documentation for AWS CloudHSM Classic |
| list_tags_for_resource | This is documentation for AWS CloudHSM Classic |
| modify_hapg | This is documentation for AWS CloudHSM Classic |
| modify_hsm | This is documentation for AWS CloudHSM Classic |
| modify_luna_client | This is documentation for AWS CloudHSM Classic |
| remove_tags_from_resource | This is documentation for AWS CloudHSM Classic |

## Examples

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

## End(Not run)
```

`cloudhsmv2` *AWS CloudHSM V2*

## Description

For more information about CloudHSM, see CloudHSM and the CloudHSM User Guide.

## Usage

```
cloudhsmv2(
  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 <- cloudhsmv2(
  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

| | |
|---|---|
| copy_backup_to_region | Copy an CloudHSM cluster backup to a different region |
| create_cluster | Creates a new CloudHSM cluster |
| create_hsm | Creates a new hardware security module (HSM) in the specified CloudHSM cluster |

| | |
|---|---|
| delete_backup | Deletes a specified CloudHSM backup |
| delete_cluster | Deletes the specified CloudHSM cluster |
| delete_hsm | Deletes the specified HSM |
| delete_resource_policy | Deletes an CloudHSM resource policy |
| describe_backups | Gets information about backups of CloudHSM clusters |
| describe_clusters | Gets information about CloudHSM clusters |
| get_resource_policy | Retrieves the resource policy document attached to a given resource |
| initialize_cluster | Claims an CloudHSM cluster by submitting the cluster certificate issued by your issuing certifica |
| list_tags | Gets a list of tags for the specified CloudHSM cluster |
| modify_backup_attributes | Modifies attributes for CloudHSM backup |
| modify_cluster | Modifies CloudHSM cluster |
| put_resource_policy | Creates or updates an CloudHSM resource policy |
| restore_backup | Restores a specified CloudHSM backup that is in the PENDING_DELETION state |
| tag_resource | Adds or overwrites one or more tags for the specified CloudHSM cluster |
| untag_resource | Removes the specified tag or tags from the specified CloudHSM cluster |

## Examples

```
## Not run:
svc <- cloudhsmv2()
svc$copy_backup_to_region(
  Foo = 123
)

## End(Not run)
```

---

`cognitoidentity` *Amazon Cognito Identity*

---

## Description

Amazon Cognito Federated Identities

Amazon Cognito Federated Identities is a web service that delivers scoped temporary credentials to mobile devices and other untrusted environments. It uniquely identifies a device and supplies the user with a consistent identity over the lifetime of an application.

Using Amazon Cognito Federated Identities, you can enable authentication with one or more third-party identity providers (Facebook, Google, or Login with Amazon) or an Amazon Cognito user pool, and you can also choose to support unauthenticated access from your app. Cognito delivers a unique identifier for each user and acts as an OpenID token provider trusted by AWS Security Token Service (STS) to access temporary, limited-privilege AWS credentials.

For a description of the authentication flow from the Amazon Cognito Developer Guide see Authentication Flow.

For more information see Amazon Cognito Federated Identities.

## Usage

```
cognitoidentity(
  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 <- cognitoidentity(
  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_identity_pool | Creates a new identity pool |
| delete_identities | Deletes identities from an identity pool |
| delete_identity_pool | Deletes an identity pool |
| describe_identity | Returns metadata related to the given identity, including when the identity was c |
| describe_identity_pool | Gets details about a particular identity pool, including the pool name, ID descrip |
| get_credentials_for_identity | Returns credentials for the provided identity ID |
| get_id | Generates (or retrieves) a Cognito ID |
| get_identity_pool_roles | Gets the roles for an identity pool |
| get_open_id_token | Gets an OpenID token, using a known Cognito ID |
| get_open_id_token_for_developer_identity | Registers (or retrieves) a Cognito IdentityId and an OpenID Connect token for a |
| get_principal_tag_attribute_map | Use GetPrincipalTagAttributeMap to list all mappings between PrincipalTags an |
| list_identities | Lists the identities in an identity pool |
| list_identity_pools | Lists all of the Cognito identity pools registered for your account |

| | |
|---|---|
| list_tags_for_resource | Lists the tags that are assigned to an Amazon Cognito identity pool |
| lookup_developer_identity | Retrieves the IdentityID associated with a DeveloperUserIdentifier or the list of |
| merge_developer_identities | Merges two users having different IdentityIds, existing in the same identity pool |
| set_identity_pool_roles | Sets the roles for an identity pool |
| set_principal_tag_attribute_map | You can use this operation to use default (username and clientID) attribute or cu |
| tag_resource | Assigns a set of tags to the specified Amazon Cognito identity pool |
| unlink_developer_identity | Unlinks a DeveloperUserIdentifier from an existing identity |
| unlink_identity | Unlinks a federated identity from an existing account |
| untag_resource | Removes the specified tags from the specified Amazon Cognito identity pool |
| update_identity_pool | Updates an identity pool |

## Examples

```
## Not run:
svc <- cognitoidentity()
svc$create_identity_pool(
  Foo = 123
)

## End(Not run)
```

---

`cognitoidentityprovider`

*Amazon Cognito Identity Provider*

---

## Description

With the Amazon Cognito user pools API, you can configure user pools and authenticate users. To authenticate users from third-party identity providers (IdPs) in this API, you can link IdP users to native user profiles. Learn more about the authentication and authorization of federated users at Adding user pool sign-in through a third party and in the User pool federation endpoints and hosted UI reference.

This API reference provides detailed information about API operations and object types in Amazon Cognito.

Along with resource management operations, the Amazon Cognito user pools API includes classes of operations and authorization models for client-side and server-side authentication of users. You can interact with operations in the Amazon Cognito user pools API as any of the following subjects.

1. An administrator who wants to configure user pools, app clients, users, groups, or other user pool functions.
2. A server-side app, like a web application, that wants to use its Amazon Web Services privileges to manage, authenticate, or authorize a user.
3. A client-side app, like a mobile app, that wants to make unauthenticated requests to manage, authenticate, or authorize a user.

For more information, see Using the Amazon Cognito user pools API and user pool endpoints in the *Amazon Cognito Developer Guide*.

With your Amazon Web Services SDK, you can build the logic to support operational flows in every use case for this API. You can also make direct REST API requests to Amazon Cognito user pools service endpoints. The following links can get you started with the `CognitoIdentityProvider` client in other supported Amazon Web Services SDKs.

- Amazon Web Services Command Line Interface
- Amazon Web Services SDK for .NET
- Amazon Web Services SDK for C++
- Amazon Web Services SDK for Go
- Amazon Web Services SDK for Java V2
- Amazon Web Services SDK for JavaScript
- Amazon Web Services SDK for PHP V3
- Amazon Web Services SDK for Python
- Amazon Web Services SDK for Ruby V3
- Amazon Web Services SDK for Kotlin

To get started with an Amazon Web Services SDK, see Tools to Build on Amazon Web Services. For example actions and scenarios, see Code examples for Amazon Cognito Identity Provider using Amazon Web Services SDKs.

## Usage

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

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

## Operations

| | |
|---|---|
| add_custom_attributes | Adds additional user attributes to the user pool schema |
| admin_add_user_to_group | Adds a user to a group |
| admin_confirm_sign_up | Confirms user sign-up as an administrator |
| admin_create_user | Creates a new user in the specified user pool |
| admin_delete_user | Deletes a user profile in your user pool |
| admin_delete_user_attributes | Deletes attribute values from a user |
| admin_disable_provider_for_user | Prevents the user from signing in with the specified external (SAML or social |
| admin_disable_user | Deactivates a user profile and revokes all access tokens for the user |
| admin_enable_user | Activate sign-in for a user profile that previously had sign-in access disabled |
| admin_forget_device | Forgets, or deletes, a remembered device from a user's profile |
| admin_get_device | Given the device key, returns details for a user' device |
| admin_get_user | Given the username, returns details about a user profile in a user pool |
| admin_initiate_auth | Starts sign-in for applications with a server-side component, for example a tra |
| admin_link_provider_for_user | Links an existing user account in a user pool (DestinationUser) to an identity |
| admin_list_devices | Lists a user's registered devices |
| admin_list_groups_for_user | Lists the groups that a user belongs to |
| admin_list_user_auth_events | Requests a history of user activity and any risks detected as part of Amazon C |
| admin_remove_user_from_group | Given a username and a group name |
| admin_reset_user_password | Resets the specified user's password in a user pool |
| admin_respond_to_auth_challenge | Some API operations in a user pool generate a challenge, like a prompt for an |
| admin_set_user_mfa_preference | Sets the user's multi-factor authentication (MFA) preference, including which |
| admin_set_user_password | Sets the specified user's password in a user pool |
| admin_set_user_settings | This action is no longer supported |
| admin_update_auth_event_feedback | Provides feedback for an authentication event indicating if it was from a valid |
| admin_update_device_status | Updates the status of a user's device so that it is marked as remembered or no |
| admin_update_user_attributes | This action might generate an SMS text message |
| admin_user_global_sign_out | Invalidates the identity, access, and refresh tokens that Amazon Cognito issue |
| associate_software_token | Begins setup of time-based one-time password (TOTP) multi-factor authentic |
| change_password | Changes the password for a specified user in a user pool |
| complete_web_authn_registration | Completes registration of a passkey authenticator for the current user |
| confirm_device | Confirms a device that a user wants to remember |
| confirm_forgot_password | This public API operation accepts a confirmation code that Amazon Cognito s |
| confirm_sign_up | This public API operation submits a code that Amazon Cognito sent to your u |
| create_group | Creates a new group in the specified user pool |
| create_identity_provider | Adds a configuration and trust relationship between a third-party identity prov |
| create_managed_login_branding | Creates a new set of branding settings for a user pool style and associates it w |

| | |
|---|---|
| create_resource_server | Creates a new OAuth2 |
| create_user_import_job | Creates a user import job |
| create_user_pool | This action might generate an SMS text message |
| create_user_pool_client | Creates an app client in a user pool |
| create_user_pool_domain | A user pool domain hosts managed login, an authorization server and web ser |
| delete_group | Deletes a group from the specified user pool |
| delete_identity_provider | Deletes a user pool identity provider (IdP) |
| delete_managed_login_branding | Deletes a managed login branding style |
| delete_resource_server | Deletes a resource server |
| delete_user | Self-deletes a user profile |
| delete_user_attributes | Self-deletes attributes for a user |
| delete_user_pool | Deletes a user pool |
| delete_user_pool_client | Deletes a user pool app client |
| delete_user_pool_domain | Given a user pool ID and domain identifier, deletes a user pool domain |
| delete_web_authn_credential | Deletes a registered passkey, or webauthN, authenticator for the currently sign |
| describe_identity_provider | Given a user pool ID and identity provider (IdP) name, returns details about th |
| describe_managed_login_branding | Given the ID of a managed login branding style, returns detailed information |
| describe_managed_login_branding_by_client | Given the ID of a user pool app client, returns detailed information about the s |
| describe_resource_server | Describes a resource server |
| describe_risk_configuration | Given an app client or user pool ID where threat protection is configured, desc |
| describe_user_import_job | Describes a user import job |
| describe_user_pool | Given a user pool ID, returns configuration information |
| describe_user_pool_client | Given an app client ID, returns configuration information |
| describe_user_pool_domain | Given a user pool domain name, returns information about the domain configu |
| forget_device | Forgets the specified device |
| forgot_password | Calling this API causes a message to be sent to the end user with a confirmati |
| get_csv_header | Gets the header information for the comma-separated value (CSV) file to be u |
| get_device | Gets the device |
| get_group | Gets a group |
| get_identity_provider_by_identifier | Gets the specified IdP |
| get_log_delivery_configuration | Gets the logging configuration of a user pool |
| get_signing_certificate | This method takes a user pool ID, and returns the signing certificate |
| get_ui_customization | Gets the user interface (UI) Customization information for a particular app cli |
| get_user | Gets the user attributes and metadata for a user |
| get_user_attribute_verification_code | Generates a user attribute verification code for the specified attribute name |
| get_user_auth_factors | Lists the authentication options for the currently signed-in user |
| get_user_pool_mfa_config | Gets the user pool multi-factor authentication (MFA) configuration |
| global_sign_out | Invalidates the identity, access, and refresh tokens that Amazon Cognito issue |
| initiate_auth | Initiates sign-in for a user in the Amazon Cognito user directory |
| list_devices | Lists the sign-in devices that Amazon Cognito has registered to the current us |
| list_groups | Lists the groups associated with a user pool |
| list_identity_providers | Lists information about all IdPs for a user pool |
| list_resource_servers | Lists the resource servers for a user pool |
| list_tags_for_resource | Lists the tags that are assigned to an Amazon Cognito user pool |
| list_user_import_jobs | Lists user import jobs for a user pool |
| list_user_pool_clients | Lists the clients that have been created for the specified user pool |
| list_user_pools | Lists the user pools associated with an Amazon Web Services account |
| list_users | Lists users and their basic details in a user pool |

| | |
|---|---|
| list_users_in_group | Lists the users in the specified group |
| list_web_authn_credentials | Generates a list of the current user's registered passkey, or webauthN, credent |
| resend_confirmation_code | Resends the confirmation (for confirmation of registration) to a specific user i |
| respond_to_auth_challenge | Some API operations in a user pool generate a challenge, like a prompt for an |
| revoke_token | Revokes all of the access tokens generated by, and at the same time as, the spe |
| set_log_delivery_configuration | Sets up or modifies the logging configuration of a user pool |
| set_risk_configuration | Configures actions on detected risks |
| set_ui_customization | Sets the user interface (UI) customization information for a user pool's built-i |
| set_user_mfa_preference | Set the user's multi-factor authentication (MFA) method preference, including |
| set_user_pool_mfa_config | Sets the user pool multi-factor authentication (MFA) and passkey configuratio |
| set_user_settings | This action is no longer supported |
| sign_up | Registers the user in the specified user pool and creates a user name, passwor |
| start_user_import_job | Starts the user import |
| start_web_authn_registration | Requests credential creation options from your user pool for registration of a |
| stop_user_import_job | Stops the user import job |
| tag_resource | Assigns a set of tags to an Amazon Cognito user pool |
| untag_resource | Removes the specified tags from an Amazon Cognito user pool |
| update_auth_event_feedback | Provides the feedback for an authentication event, whether it was from a valid |
| update_device_status | Updates the device status |
| update_group | Updates the specified group with the specified attributes |
| update_identity_provider | Updates IdP information for a user pool |
| update_managed_login_branding | Configures the branding settings for a user pool style |
| update_resource_server | Updates the name and scopes of resource server |
| update_user_attributes | With this operation, your users can update one or more of their attributes with |
| update_user_pool | This action might generate an SMS text message |
| update_user_pool_client | Updates the specified user pool app client with the specified attributes |
| update_user_pool_domain | A user pool domain hosts managed login, an authorization server and web ser |
| verify_software_token | Use this API to register a user's entered time-based one-time password (TOT |
| verify_user_attribute | Verifies the specified user attributes in the user pool |

## Examples

```
## Not run:
svc <- cognitoidentityprovider()
svc$add_custom_attributes(
  Foo = 123
)

## End(Not run)
```

---

`cognitosync` *Amazon Cognito Sync*

---

## Description

Amazon Cognito Sync provides an AWS service and client library that enable cross-device syncing of application-related user data. High-level client libraries are available for both iOS and Android. You can use these libraries to persist data locally so that it's available even if the device is offline. Developer credentials don't need to be stored on the mobile device to access the service. You can use Amazon Cognito to obtain a normalized user ID and credentials. User data is persisted in a dataset that can store up to 1 MB of key-value pairs, and you can have up to 20 datasets per user identity.

With Amazon Cognito Sync, the data stored for each identity is accessible only to credentials assigned to that identity. In order to use the Cognito Sync service, you need to make API calls using credentials retrieved with Amazon Cognito Identity service.

If you want to use Cognito Sync in an Android or iOS application, you will probably want to make API calls via the AWS Mobile SDK. To learn more, see the Developer Guide for Android and the Developer Guide for iOS.

## Usage

```
cognitosync(
  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 <- cognitosync(
  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

| | |
|---|---|
| bulk_publish | Initiates a bulk publish of all existing datasets for an Identity Pool to the configured stream |
| delete_dataset | Deletes the specific dataset |
| describe_dataset | Gets meta data about a dataset by identity and dataset name |
| describe_identity_pool_usage | Gets usage details (for example, data storage) about a particular identity pool |
| describe_identity_usage | Gets usage information for an identity, including number of datasets and data usage |
| get_bulk_publish_details | Get the status of the last BulkPublish operation for an identity pool |
| get_cognito_events | Gets the events and the corresponding Lambda functions associated with an identity pool |
| get_identity_pool_configuration | Gets the configuration settings of an identity pool |
| list_datasets | Lists datasets for an identity |
| list_identity_pool_usage | Gets a list of identity pools registered with Cognito |
| list_records | Gets paginated records, optionally changed after a particular sync count for a dataset and id |
| register_device | Registers a device to receive push sync notifications |
| set_cognito_events | Sets the AWS Lambda function for a given event type for an identity pool |
| set_identity_pool_configuration | Sets the necessary configuration for push sync |
| subscribe_to_dataset | Subscribes to receive notifications when a dataset is modified by another device |
| unsubscribe_from_dataset | Unsubscribes from receiving notifications when a dataset is modified by another device |
| update_records | Posts updates to records and adds and deletes records for a dataset and user |

## Examples

```
## Not run:
svc <- cognitosync()
svc$bulk_publish(
  Foo = 123
)

## End(Not run)
```

---

# detective *Amazon Detective*

---

## Description

Detective uses machine learning and purpose-built visualizations to help you to analyze and investigate security issues across your Amazon Web Services (Amazon Web Services) workloads. Detective automatically extracts time-based events such as login attempts, API calls, and network traffic from CloudTrail and Amazon Virtual Private Cloud (Amazon VPC) flow logs. It also extracts findings detected by Amazon GuardDuty.

The Detective API primarily supports the creation and management of behavior graphs. A behavior graph contains the extracted data from a set of member accounts, and is created and managed by an administrator account.

To add a member account to the behavior graph, the administrator account sends an invitation to the account. When the account accepts the invitation, it becomes a member account in the behavior graph.

Detective is also integrated with Organizations. The organization management account designates the Detective administrator account for the organization. That account becomes the administrator account for the organization behavior graph. The Detective administrator account is also the delegated administrator account for Detective in Organizations.

The Detective administrator account can enable any organization account as a member account in the organization behavior graph. The organization accounts do not receive invitations. The Detective administrator account can also invite other accounts to the organization behavior graph.

Every behavior graph is specific to a Region. You can only use the API to manage behavior graphs that belong to the Region that is associated with the currently selected endpoint.

The administrator account for a behavior graph can use the Detective API to do the following:

- Enable and disable Detective. Enabling Detective creates a new behavior graph.
- View the list of member accounts in a behavior graph.
- Add member accounts to a behavior graph.
- Remove member accounts from a behavior graph.
- Apply tags to a behavior graph.

The organization management account can use the Detective API to select the delegated administrator for Detective.

The Detective administrator account for an organization can use the Detective API to do the following:

- Perform all of the functions of an administrator account.
- Determine whether to automatically enable new organization accounts as member accounts in the organization behavior graph.

An invited member account can use the Detective API to do the following:

- View the list of behavior graphs that they are invited to.
- Accept an invitation to contribute to a behavior graph.
- Decline an invitation to contribute to a behavior graph.
- Remove their account from a behavior graph.

All API actions are logged as CloudTrail events. See Logging Detective API Calls with CloudTrail.

We replaced the term "master account" with the term "administrator account". An administrator account is used to centrally manage multiple accounts. In the case of Detective, the administrator account manages the accounts in their behavior graph.

## Usage

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

### Operations

| | |
|---|---|
| accept_invitation | Accepts an invitation for the member account to contribute data to a behavior graph |
| batch_get_graph_member_datasources | Gets data source package information for the behavior graph |
| batch_get_membership_datasources | Gets information on the data source package history for an account |
| create_graph | Creates a new behavior graph for the calling account, and sets that account as the ad |
| create_members | CreateMembers is used to send invitations to accounts |
| delete_graph | Disables the specified behavior graph and queues it to be deleted |
| delete_members | Removes the specified member accounts from the behavior graph |
| describe_organization_configuration | Returns information about the configuration for the organization behavior graph |
| disable_organization_admin_account | Removes the Detective administrator account in the current Region |
| disassociate_membership | Removes the member account from the specified behavior graph |
| enable_organization_admin_account | Designates the Detective administrator account for the organization in the current Re |
| get_investigation | Detective investigations lets you investigate IAM users and IAM roles using indicato |
| get_members | Returns the membership details for specified member accounts for a behavior graph |

| | |
|---|---|
| list_datasource_packages | Lists data source packages in the behavior graph |
| list_graphs | Returns the list of behavior graphs that the calling account is an administrator accou |
| list_indicators | Gets the indicators from an investigation |
| list_investigations | Detective investigations lets you investigate IAM users and IAM roles using indicato |
| list_invitations | Retrieves the list of open and accepted behavior graph invitations for the member ac |
| list_members | Retrieves the list of member accounts for a behavior graph |
| list_organization_admin_accounts | Returns information about the Detective administrator account for an organization |
| list_tags_for_resource | Returns the tag values that are assigned to a behavior graph |
| reject_invitation | Rejects an invitation to contribute the account data to a behavior graph |
| start_investigation | Detective investigations lets you investigate IAM users and IAM roles using indicato |
| start_monitoring_member | Sends a request to enable data ingest for a member account that has a status of ACCI |
| tag_resource | Applies tag values to a behavior graph |
| untag_resource | Removes tags from a behavior graph |
| update_datasource_packages | Starts a data source package for the Detective behavior graph |
| update_investigation_state | Updates the state of an investigation |
| update_organization_configuration | Updates the configuration for the Organizations integration in the current Region |

## Examples

```
## Not run:
svc <- detective()
svc$accept_invitation(
  Foo = 123
)

## End(Not run)
```

---

`directoryservice` *AWS Directory Service*

---

## Description

Directory Service

Directory Service is a web service that makes it easy for you to setup and run directories in the Amazon Web Services cloud, or connect your Amazon Web Services resources with an existing self-managed Microsoft Active Directory. This guide provides detailed information about Directory Service operations, data types, parameters, and errors. For information about Directory Services features, see Directory Service and the Directory Service Administration Guide.

Amazon Web Services provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to Directory Service and other Amazon Web Services services. For more information about the Amazon Web Services SDKs, including how to download and install them, see Tools for Amazon Web Services.

## Usage

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

### Operations

| | |
|---|---|
| accept_shared_directory | Accepts a directory sharing request that was sent from the directory owner account |
| add_ip_routes | If the DNS server for your self-managed domain uses a publicly addressable IP add |
| add_region | Adds two domain controllers in the specified Region for the specified directory |
| add_tags_to_resource | Adds or overwrites one or more tags for the specified directory |
| cancel_schema_extension | Cancels an in-progress schema extension to a Microsoft AD directory |
| connect_directory | Creates an AD Connector to connect to a self-managed directory |
| create_alias | Creates an alias for a directory and assigns the alias to the directory |
| create_computer | Creates an Active Directory computer object in the specified directory |
| create_conditional_forwarder | Creates a conditional forwarder associated with your Amazon Web Services directo |
| create_directory | Creates a Simple AD directory |
| create_log_subscription | Creates a subscription to forward real-time Directory Service domain controller sec |
| create_microsoft_ad | Creates a Microsoft AD directory in the Amazon Web Services Cloud |
| create_snapshot | Creates a snapshot of a Simple AD or Microsoft AD directory in the Amazon Web S |

| | |
|---|---|
| create_trust | Directory Service for Microsoft Active Directory allows you to configure trust relati |
| delete_conditional_forwarder | Deletes a conditional forwarder that has been set up for your Amazon Web Services |
| delete_directory | Deletes an Directory Service directory |
| delete_log_subscription | Deletes the specified log subscription |
| delete_snapshot | Deletes a directory snapshot |
| delete_trust | Deletes an existing trust relationship between your Managed Microsoft AD director |
| deregister_certificate | Deletes from the system the certificate that was registered for secure LDAP or client |
| deregister_event_topic | Removes the specified directory as a publisher to the specified Amazon SNS topic |
| describe_certificate | Displays information about the certificate registered for secure LDAP or client certi |
| describe_client_authentication_settings | Retrieves information about the type of client authentication for the specified directo |
| describe_conditional_forwarders | Obtains information about the conditional forwarders for this account |
| describe_directories | Obtains information about the directories that belong to this account |
| describe_directory_data_access | Obtains status of directory data access enablement through the Directory Service Da |
| describe_domain_controllers | Provides information about any domain controllers in your directory |
| describe_event_topics | Obtains information about which Amazon SNS topics receive status messages from |
| describe_ldaps_settings | Describes the status of LDAP security for the specified directory |
| describe_regions | Provides information about the Regions that are configured for multi-Region replica |
| describe_settings | Retrieves information about the configurable settings for the specified directory |
| describe_shared_directories | Returns the shared directories in your account |
| describe_snapshots | Obtains information about the directory snapshots that belong to this account |
| describe_trusts | Obtains information about the trust relationships for this account |
| describe_update_directory | Describes the updates of a directory for a particular update type |
| disable_client_authentication | Disables alternative client authentication methods for the specified directory |
| disable_directory_data_access | Deactivates access to directory data via the Directory Service Data API for the speci |
| disable_ldaps | Deactivates LDAP secure calls for the specified directory |
| disable_radius | Disables multi-factor authentication (MFA) with the Remote Authentication Dial In |
| disable_sso | Disables single-sign on for a directory |
| enable_client_authentication | Enables alternative client authentication methods for the specified directory |
| enable_directory_data_access | Enables access to directory data via the Directory Service Data API for the specified |
| enable_ldaps | Activates the switch for the specific directory to always use LDAP secure calls |
| enable_radius | Enables multi-factor authentication (MFA) with the Remote Authentication Dial In |
| enable_sso | Enables single sign-on for a directory |
| get_directory_limits | Obtains directory limit information for the current Region |
| get_snapshot_limits | Obtains the manual snapshot limits for a directory |
| list_certificates | For the specified directory, lists all the certificates registered for a secure LDAP or c |
| list_ip_routes | Lists the address blocks that you have added to a directory |
| list_log_subscriptions | Lists the active log subscriptions for the Amazon Web Services account |
| list_schema_extensions | Lists all schema extensions applied to a Microsoft AD Directory |
| list_tags_for_resource | Lists all tags on a directory |
| register_certificate | Registers a certificate for a secure LDAP or client certificate authentication |
| register_event_topic | Associates a directory with an Amazon SNS topic |
| reject_shared_directory | Rejects a directory sharing request that was sent from the directory owner account |
| remove_ip_routes | Removes IP address blocks from a directory |
| remove_region | Stops all replication and removes the domain controllers from the specified Region |
| remove_tags_from_resource | Removes tags from a directory |
| reset_user_password | Resets the password for any user in your Managed Microsoft AD or Simple AD dire |
| restore_from_snapshot | Restores a directory using an existing directory snapshot |
| share_directory | Shares a specified directory (DirectoryId) in your Amazon Web Services account (d |

| | |
|---|---|
| start_schema_extension | Applies a schema extension to a Microsoft AD directory |
| unshare_directory | Stops the directory sharing between the directory owner and consumer accounts |
| update_conditional_forwarder | Updates a conditional forwarder that has been set up for your Amazon Web Service |
| update_directory_setup | Updates the directory for a particular update type |
| update_number_of_domain_controllers | Adds or removes domain controllers to or from the directory |
| update_radius | Updates the Remote Authentication Dial In User Service (RADIUS) server informa |
| update_settings | Updates the configurable settings for the specified directory |
| update_trust | Updates the trust that has been set up between your Managed Microsoft AD directo |
| verify_trust | Directory Service for Microsoft Active Directory allows you to configure and verify |

## Examples

```
## Not run:
svc <- directoryservice()
svc$accept_shared_directory(
  Foo = 123
)

## End(Not run)
```

---

# fms *Firewall Management Service*

---

## Description

This is the *Firewall Manager API Reference*. This guide is for developers who need detailed information about the Firewall Manager API actions, data types, and errors. For detailed information about Firewall Manager features, see the Firewall Manager Developer Guide.

Some API actions require explicit resource permissions. For information, see the developer guide topic Service roles for Firewall Manager.

## Usage

```
fms(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 <- fms(
  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_admin_account | Sets a Firewall Manager default administrator account |
| associate_third_party_firewall | Sets the Firewall Manager policy administrator as a tenant administrator of a thi |
| batch_associate_resource | Associate resources to a Firewall Manager resource set |
| batch_disassociate_resource | Disassociates resources from a Firewall Manager resource set |
| delete_apps_list | Permanently deletes an Firewall Manager applications list |
| delete_notification_channel | Deletes an Firewall Manager association with the IAM role and the Amazon Sir |
| delete_policy | Permanently deletes an Firewall Manager policy |
| delete_protocols_list | Permanently deletes an Firewall Manager protocols list |
| delete_resource_set | Deletes the specified ResourceSet |
| disassociate_admin_account | Disassociates an Firewall Manager administrator account |
| disassociate_third_party_firewall | Disassociates a Firewall Manager policy administrator from a third-party firewa |
| get_admin_account | Returns the Organizations account that is associated with Firewall Manager as t |
| get_admin_scope | Returns information about the specified account's administrative scope |
| get_apps_list | Returns information about the specified Firewall Manager applications list |
| get_compliance_detail | Returns detailed compliance information about the specified member account |
| get_notification_channel | Information about the Amazon Simple Notification Service (SNS) topic that is u |
| get_policy | Returns information about the specified Firewall Manager policy |
| get_protection_status | If you created a Shield Advanced policy, returns policy-level attack summary in |
| get_protocols_list | Returns information about the specified Firewall Manager protocols list |
| get_resource_set | Gets information about a specific resource set |
| get_third_party_firewall_association_status | The onboarding status of a Firewall Manager admin account to third-party firew |
| get_violation_details | Retrieves violations for a resource based on the specified Firewall Manager poli |
| list_admin_accounts_for_organization | Returns a AdminAccounts object that lists the Firewall Manager administrators |
| list_admins_managing_account | Lists the accounts that are managing the specified Organizations member accou |
| list_apps_lists | Returns an array of AppsListDataSummary objects |
| list_compliance_status | Returns an array of PolicyComplianceStatus objects |
| list_discovered_resources | Returns an array of resources in the organization's accounts that are available to |
| list_member_accounts | Returns a MemberAccounts object that lists the member accounts in the adminis |
| list_policies | Returns an array of PolicySummary objects |

| | |
|---|---|
| list_protocols_lists | Returns an array of ProtocolsListDataSummary objects |
| list_resource_set_resources | Returns an array of resources that are currently associated to a resource set |
| list_resource_sets | Returns an array of ResourceSetSummary objects |
| list_tags_for_resource | Retrieves the list of tags for the specified Amazon Web Services resource |
| list_third_party_firewall_firewall_policies | Retrieves a list of all of the third-party firewall policies that are associated with t |
| put_admin_account | Creates or updates an Firewall Manager administrator account |
| put_apps_list | Creates an Firewall Manager applications list |
| put_notification_channel | Designates the IAM role and Amazon Simple Notification Service (SNS) topic t |
| put_policy | Creates an Firewall Manager policy |
| put_protocols_list | Creates an Firewall Manager protocols list |
| put_resource_set | Creates the resource set |
| tag_resource | Adds one or more tags to an Amazon Web Services resource |
| untag_resource | Removes one or more tags from an Amazon Web Services resource |

## Examples

```
## Not run:
svc <- fms()
svc$associate_admin_account(
  Foo = 123
)

## End(Not run)
```

---

guardduty *Amazon GuardDuty*

---

## Description

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following foundational data sources - VPC flow logs, Amazon Web Services CloudTrail management event logs, CloudTrail S3 data event logs, EKS audit logs, DNS logs, Amazon EBS volume data, runtime activity belonging to container workloads, such as Amazon EKS, Amazon ECS (including Amazon Web Services Fargate), and Amazon EC2 instances. It uses threat intelligence feeds, such as lists of malicious IPs and domains, and machine learning to identify unexpected, potentially unauthorized, and malicious activity within your Amazon Web Services environment. This can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IPs, domains, or presence of malware on your Amazon EC2 instances and container workloads. For example, GuardDuty can detect compromised EC2 instances and container workloads serving malware, or mining bitcoin.

GuardDuty also monitors Amazon Web Services account access behavior for signs of compromise, such as unauthorized infrastructure deployments like EC2 instances deployed in a Region that has never been used, or unusual API calls like a password policy change to reduce password strength.

GuardDuty informs you about the status of your Amazon Web Services environment by producing security findings that you can view in the GuardDuty console or through Amazon EventBridge. For more information, see the *Amazon GuardDuty User Guide* .

## Usage

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

## Operations

| | |
|---|---|
| accept_administrator_invitation | Accepts the invitation to be a member account and get monitored by a GuardDuty |
| accept_invitation | Accepts the invitation to be monitored by a GuardDuty administrator account |
| archive_findings | Archives GuardDuty findings that are specified by the list of finding IDs |
| create_detector | Creates a single GuardDuty detector |
| create_filter | Creates a filter using the specified finding criteria |
| create_ip_set | Creates a new IPSet, which is called a trusted IP list in the console user interface |
| create_malware_protection_plan | Creates a new Malware Protection plan for the protected resource |
| create_members | Creates member accounts of the current Amazon Web Services account by specify |

| | |
|---|---|
| create_publishing_destination | Creates a publishing destination where you can export your GuardDuty findings |
| create_sample_findings | Generates sample findings of types specified by the list of finding types |
| create_threat_intel_set | Creates a new ThreatIntelSet |
| decline_invitations | Declines invitations sent to the current member account by Amazon Web Services |
| delete_detector | Deletes an Amazon GuardDuty detector that is specified by the detector ID |
| delete_filter | Deletes the filter specified by the filter name |
| delete_invitations | Deletes invitations sent to the current member account by Amazon Web Services |
| delete_ip_set | Deletes the IPSet specified by the ipSetId |
| delete_malware_protection_plan | Deletes the Malware Protection plan ID associated with the Malware Protection p |
| delete_members | Deletes GuardDuty member accounts (to the current GuardDuty administrator acc |
| delete_publishing_destination | Deletes the publishing definition with the specified destinationId |
| delete_threat_intel_set | Deletes the ThreatIntelSet specified by the ThreatIntelSet ID |
| describe_malware_scans | Returns a list of malware scans |
| describe_organization_configuration | Returns information about the account selected as the delegated administrator for |
| describe_publishing_destination | Returns information about the publishing destination specified by the provided de |
| disable_organization_admin_account | Removes the existing GuardDuty delegated administrator of the organization |
| disassociate_from_administrator_account | Disassociates the current GuardDuty member account from its administrator acco |
| disassociate_from_master_account | Disassociates the current GuardDuty member account from its administrator acco |
| disassociate_members | Disassociates GuardDuty member accounts (from the current administrator accou |
| enable_organization_admin_account | Designates an Amazon Web Services account within the organization as your Gua |
| get_administrator_account | Provides the details of the GuardDuty administrator account associated with the c |
| get_coverage_statistics | Retrieves aggregated statistics for your account |
| get_detector | Retrieves a GuardDuty detector specified by the detectorId |
| get_filter | Returns the details of the filter specified by the filter name |
| get_findings | Describes Amazon GuardDuty findings specified by finding IDs |
| get_findings_statistics | Lists GuardDuty findings statistics for the specified detector ID |
| get_invitations_count | Returns the count of all GuardDuty membership invitations that were sent to the c |
| get_ip_set | Retrieves the IPSet specified by the ipSetId |
| get_malware_protection_plan | Retrieves the Malware Protection plan details associated with a Malware Protectio |
| get_malware_scan_settings | Returns the details of the malware scan settings |
| get_master_account | Provides the details for the GuardDuty administrator account associated with the c |
| get_member_detectors | Describes which data sources are enabled for the member account's detector |
| get_members | Retrieves GuardDuty member accounts (of the current GuardDuty administrator a |
| get_organization_statistics | Retrieves how many active member accounts have each feature enabled within Gu |
| get_remaining_free_trial_days | Provides the number of days left for each data source used in the free trial period |
| get_threat_intel_set | Retrieves the ThreatIntelSet that is specified by the ThreatIntelSet ID |
| get_usage_statistics | Lists Amazon GuardDuty usage statistics over the last 30 days for the specified de |
| invite_members | Invites Amazon Web Services accounts to become members of an organization ad |
| list_coverage | Lists coverage details for your GuardDuty account |
| list_detectors | Lists detectorIds of all the existing Amazon GuardDuty detector resources |
| list_filters | Returns a paginated list of the current filters |
| list_findings | Lists GuardDuty findings for the specified detector ID |
| list_invitations | Lists all GuardDuty membership invitations that were sent to the current Amazon |
| list_ip_sets | Lists the IPSets of the GuardDuty service specified by the detector ID |
| list_malware_protection_plans | Lists the Malware Protection plan IDs associated with the protected resources in y |
| list_members | Lists details about all member accounts for the current GuardDuty administrator a |
| list_organization_admin_accounts | Lists the accounts designated as GuardDuty delegated administrators |
| list_publishing_destinations | Returns a list of publishing destinations associated with the specified detectorId |

| | |
|---|---|
| list_tags_for_resource | Lists tags for a resource |
| list_threat_intel_sets | Lists the ThreatIntelSets of the GuardDuty service specified by the detector ID |
| start_malware_scan | Initiates the malware scan |
| start_monitoring_members | Turns on GuardDuty monitoring of the specified member accounts |
| stop_monitoring_members | Stops GuardDuty monitoring for the specified member accounts |
| tag_resource | Adds tags to a resource |
| unarchive_findings | Unarchives GuardDuty findings specified by the findingIds |
| untag_resource | Removes tags from a resource |
| update_detector | Updates the GuardDuty detector specified by the detector ID |
| update_filter | Updates the filter specified by the filter name |
| update_findings_feedback | Marks the specified GuardDuty findings as useful or not useful |
| update_ip_set | Updates the IPSet specified by the IPSet ID |
| update_malware_protection_plan | Updates an existing Malware Protection plan resource |
| update_malware_scan_settings | Updates the malware scan settings |
| update_member_detectors | Contains information on member accounts to be updated |
| update_organization_configuration | Configures the delegated administrator account with the provided values |
| update_publishing_destination | Updates information about the publishing destination specified by the destinationI |
| update_threat_intel_set | Updates the ThreatIntelSet specified by the ThreatIntelSet ID |

## Examples

```
## Not run:
svc <- guardduty()
svc$accept_administrator_invitation(
  Foo = 123
)

## End(Not run)
```

---

`iam` *AWS Identity and Access Management*

---

## Description

Identity and Access Management

Identity and Access Management (IAM) is a web service for securely controlling access to Amazon Web Services services. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which Amazon Web Services resources users and applications can access. For more information about IAM, see Identity and Access Management (IAM) and the Identity and Access Management User Guide.

## Usage

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

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

## Operations

| | |
|---|---|
| add_client_id_to_open_id_connect_provider | Adds a new client ID (also known as audience) to the list of client IDs |
| add_role_to_instance_profile | Adds the specified IAM role to the specified instance profile |
| add_user_to_group | Adds the specified user to the specified group |
| attach_group_policy | Attaches the specified managed policy to the specified IAM group |
| attach_role_policy | Attaches the specified managed policy to the specified IAM role |
| attach_user_policy | Attaches the specified managed policy to the specified user |
| change_password | Changes the password of the IAM user who is calling this operation |
| create_access_key | Creates a new Amazon Web Services secret access key and correspond |
| create_account_alias | Creates an alias for your Amazon Web Services account |
| create_group | Creates a new group |
| create_instance_profile | Creates a new instance profile |
| create_login_profile | Creates a password for the specified IAM user |
| create_open_id_connect_provider | Creates an IAM entity to describe an identity provider (IdP) that suppo |
| create_policy | Creates a new managed policy for your Amazon Web Services accoun |
| create_policy_version | Creates a new version of the specified managed policy |
| create_role | Creates a new role for your Amazon Web Services account |
| create_saml_provider | Creates an IAM resource that describes an identity provider (IdP) that |
| create_service_linked_role | Creates an IAM role that is linked to a specific Amazon Web Services |
| create_service_specific_credential | Generates a set of credentials consisting of a user name and password |
| create_user | Creates a new IAM user for your Amazon Web Services account |

| | |
|---|---|
| create_virtual_mfa_device | Creates a new virtual MFA device for the Amazon Web Services accou |
| deactivate_mfa_device | Deactivates the specified MFA device and removes it from association |
| delete_access_key | Deletes the access key pair associated with the specified IAM user |
| delete_account_alias | Deletes the specified Amazon Web Services account alias |
| delete_account_password_policy | Deletes the password policy for the Amazon Web Services account |
| delete_group | Deletes the specified IAM group |
| delete_group_policy | Deletes the specified inline policy that is embedded in the specified IA |
| delete_instance_profile | Deletes the specified instance profile |
| delete_login_profile | Deletes the password for the specified IAM user, For more information |
| delete_open_id_connect_provider | Deletes an OpenID Connect identity provider (IdP) resource object in |
| delete_policy | Deletes the specified managed policy |
| delete_policy_version | Deletes the specified version from the specified managed policy |
| delete_role | Deletes the specified role |
| delete_role_permissions_boundary | Deletes the permissions boundary for the specified IAM role |
| delete_role_policy | Deletes the specified inline policy that is embedded in the specified IA |
| delete_saml_provider | Deletes a SAML provider resource in IAM |
| delete_server_certificate | Deletes the specified server certificate |
| delete_service_linked_role | Submits a service-linked role deletion request and returns a DeletionTa |
| delete_service_specific_credential | Deletes the specified service-specific credential |
| delete_signing_certificate | Deletes a signing certificate associated with the specified IAM user |
| delete_ssh_public_key | Deletes the specified SSH public key |
| delete_user | Deletes the specified IAM user |
| delete_user_permissions_boundary | Deletes the permissions boundary for the specified IAM user |
| delete_user_policy | Deletes the specified inline policy that is embedded in the specified IA |
| delete_virtual_mfa_device | Deletes a virtual MFA device |
| detach_group_policy | Removes the specified managed policy from the specified IAM group |
| detach_role_policy | Removes the specified managed policy from the specified role |
| detach_user_policy | Removes the specified managed policy from the specified user |
| disable_organizations_root_credentials_management | Disables the management of privileged root user credentials across me |
| disable_organizations_root_sessions | Disables root user sessions for privileged tasks across member accoun |
| enable_mfa_device | Enables the specified MFA device and associates it with the specified |
| enable_organizations_root_credentials_management | Enables the management of privileged root user credentials across me |
| enable_organizations_root_sessions | Allows the management account or delegated administrator to perform |
| generate_credential_report | Generates a credential report for the Amazon Web Services account |
| generate_organizations_access_report | Generates a report for service last accessed data for Organizations |
| generate_service_last_accessed_details | Generates a report that includes details about when an IAM resource ( |
| get_access_key_last_used | Retrieves information about when the specified access key was last use |
| get_account_authorization_details | Retrieves information about all IAM users, groups, roles, and policies |
| get_account_password_policy | Retrieves the password policy for the Amazon Web Services account |
| get_account_summary | Retrieves information about IAM entity usage and IAM quotas in the A |
| get_context_keys_for_custom_policy | Gets a list of all of the context keys referenced in the input policies |
| get_context_keys_for_principal_policy | Gets a list of all of the context keys referenced in all the IAM policies |
| get_credential_report | Retrieves a credential report for the Amazon Web Services account |
| get_group | Returns a list of IAM users that are in the specified IAM group |
| get_group_policy | Retrieves the specified inline policy document that is embedded in the |
| get_instance_profile | Retrieves information about the specified instance profile, including th |
| get_login_profile | Retrieves the user name for the specified IAM user |
| get_mfa_device | Retrieves information about an MFA device for a specified user |

| | |
|---|---|
| get_open_id_connect_provider | Returns information about the specified OpenID Connect (OIDC) prov |
| get_organizations_access_report | Retrieves the service last accessed data report for Organizations that w |
| get_policy | Retrieves information about the specified managed policy, including th |
| get_policy_version | Retrieves information about the specified version of the specified mana |
| get_role | Retrieves information about the specified role, including the role's path |
| get_role_policy | Retrieves the specified inline policy document that is embedded with t |
| get_saml_provider | Returns the SAML provider metadocument that was uploaded when th |
| get_server_certificate | Retrieves information about the specified server certificate stored in IA |
| get_service_last_accessed_details | Retrieves a service last accessed report that was created using the Gen |
| get_service_last_accessed_details_with_entities | After you generate a group or policy report using the GenerateServiceL |
| get_service_linked_role_deletion_status | Retrieves the status of your service-linked role deletion |
| get_ssh_public_key | Retrieves the specified SSH public key, including metadata about the k |
| get_user | Retrieves information about the specified IAM user, including the user |
| get_user_policy | Retrieves the specified inline policy document that is embedded in the |
| list_access_keys | Returns information about the access key IDs associated with the spec |
| list_account_aliases | Lists the account alias associated with the Amazon Web Services acco |
| list_attached_group_policies | Lists all managed policies that are attached to the specified IAM group |
| list_attached_role_policies | Lists all managed policies that are attached to the specified IAM role |
| list_attached_user_policies | Lists all managed policies that are attached to the specified IAM user |
| list_entities_for_policy | Lists all IAM users, groups, and roles that the specified managed polic |
| list_group_policies | Lists the names of the inline policies that are embedded in the specifie |
| list_groups | Lists the IAM groups that have the specified path prefix |
| list_groups_for_user | Lists the IAM groups that the specified IAM user belongs to |
| list_instance_profiles | Lists the instance profiles that have the specified path prefix |
| list_instance_profiles_for_role | Lists the instance profiles that have the specified associated IAM role |
| list_instance_profile_tags | Lists the tags that are attached to the specified IAM instance profile |
| list_mfa_devices | Lists the MFA devices for an IAM user |
| list_mfa_device_tags | Lists the tags that are attached to the specified IAM virtual multi-facto |
| list_open_id_connect_providers | Lists information about the IAM OpenID Connect (OIDC) provider re |
| list_open_id_connect_provider_tags | Lists the tags that are attached to the specified OpenID Connect (OIDC |
| list_organizations_features | Lists the centralized root access features enabled for your organization |
| list_policies | Lists all the managed policies that are available in your Amazon Web S |
| list_policies_granting_service_access | Retrieves a list of policies that the IAM identity (user, group, or role) c |
| list_policy_tags | Lists the tags that are attached to the specified IAM customer managed |
| list_policy_versions | Lists information about the versions of the specified managed policy, i |
| list_role_policies | Lists the names of the inline policies that are embedded in the specifie |
| list_roles | Lists the IAM roles that have the specified path prefix |
| list_role_tags | Lists the tags that are attached to the specified role |
| list_saml_providers | Lists the SAML provider resource objects defined in IAM in the accou |
| list_saml_provider_tags | Lists the tags that are attached to the specified Security Assertion Mark |
| list_server_certificates | Lists the server certificates stored in IAM that have the specified path p |
| list_server_certificate_tags | Lists the tags that are attached to the specified IAM server certificate |
| list_service_specific_credentials | Returns information about the service-specific credentials associated w |
| list_signing_certificates | Returns information about the signing certificates associated with the s |
| list_ssh_public_keys | Returns information about the SSH public keys associated with the spe |
| list_user_policies | Lists the names of the inline policies embedded in the specified IAM u |
| list_users | Lists the IAM users that have the specified path prefix |
| list_user_tags | Lists the tags that are attached to the specified IAM user |

| | |
|---|---|
| list_virtual_mfa_devices | Lists the virtual MFA devices defined in the Amazon Web Services ac |
| put_group_policy | Adds or updates an inline policy document that is embedded in the spe |
| put_role_permissions_boundary | Adds or updates the policy that is specified as the IAM role's permissi |
| put_role_policy | Adds or updates an inline policy document that is embedded in the spe |
| put_user_permissions_boundary | Adds or updates the policy that is specified as the IAM user's permissi |
| put_user_policy | Adds or updates an inline policy document that is embedded in the spe |
| remove_client_id_from_open_id_connect_provider | Removes the specified client ID (also known as audience) from the list |
| remove_role_from_instance_profile | Removes the specified IAM role from the specified Amazon EC2 insta |
| remove_user_from_group | Removes the specified user from the specified group |
| reset_service_specific_credential | Resets the password for a service-specific credential |
| resync_mfa_device | Synchronizes the specified MFA device with its IAM resource object o |
| set_default_policy_version | Sets the specified version of the specified policy as the policy's default |
| set_security_token_service_preferences | Sets the specified version of the global endpoint token as the token ver |
| simulate_custom_policy | Simulate how a set of IAM policies and optionally a resource-based po |
| simulate_principal_policy | Simulate how a set of IAM policies attached to an IAM entity works w |
| tag_instance_profile | Adds one or more tags to an IAM instance profile |
| tag_mfa_device | Adds one or more tags to an IAM virtual multi-factor authentication (M |
| tag_open_id_connect_provider | Adds one or more tags to an OpenID Connect (OIDC)-compatible iden |
| tag_policy | Adds one or more tags to an IAM customer managed policy |
| tag_role | Adds one or more tags to an IAM role |
| tag_saml_provider | Adds one or more tags to a Security Assertion Markup Language (SAM |
| tag_server_certificate | Adds one or more tags to an IAM server certificate |
| tag_user | Adds one or more tags to an IAM user |
| untag_instance_profile | Removes the specified tags from the IAM instance profile |
| untag_mfa_device | Removes the specified tags from the IAM virtual multi-factor authenti |
| untag_open_id_connect_provider | Removes the specified tags from the specified OpenID Connect (OIDC |
| untag_policy | Removes the specified tags from the customer managed policy |
| untag_role | Removes the specified tags from the role |
| untag_saml_provider | Removes the specified tags from the specified Security Assertion Marl |
| untag_server_certificate | Removes the specified tags from the IAM server certificate |
| untag_user | Removes the specified tags from the user |
| update_access_key | Changes the status of the specified access key from Active to Inactive, |
| update_account_password_policy | Updates the password policy settings for the Amazon Web Services ac |
| update_assume_role_policy | Updates the policy that grants an IAM entity permission to assume a r |
| update_group | Updates the name and/or the path of the specified IAM group |
| update_login_profile | Changes the password for the specified IAM user |
| update_open_id_connect_provider_thumbprint | Replaces the existing list of server certificate thumbprints associated w |
| update_role | Updates the description or maximum session duration setting of a role |
| update_role_description | Use UpdateRole instead |
| update_saml_provider | Updates the metadata document, SAML encryption settings, and priva |
| update_server_certificate | Updates the name and/or the path of the specified server certificate sto |
| update_service_specific_credential | Sets the status of a service-specific credential to Active or Inactive |
| update_signing_certificate | Changes the status of the specified user signing certificate from active |
| update_ssh_public_key | Sets the status of an IAM user's SSH public key to active or inactive |
| update_user | Updates the name and/or the path of the specified IAM user |
| upload_server_certificate | Uploads a server certificate entity for the Amazon Web Services accou |
| upload_signing_certificate | Uploads an X |
| upload_ssh_public_key | Uploads an SSH public key and associates it with the specified IAM u |

## Examples

```
## Not run:
svc <- iam()
# The following add-client-id-to-open-id-connect-provider command adds the
# client ID my-application-ID to the OIDC provider named
# server.example.com:
svc$add_client_id_to_open_id_connect_provider(
  ClientID = "my-application-ID",
 OpenIDConnectProviderArn = "arn:aws:iam::123456789012:oidc-provider/server.example.com"
)

## End(Not run)
```

---

# iamrolesanywhere *IAM Roles Anywhere*

---

## Description

Identity and Access Management Roles Anywhere provides a secure way for your workloads such as servers, containers, and applications that run outside of Amazon Web Services to obtain temporary Amazon Web Services credentials. Your workloads can use the same IAM policies and roles you have for native Amazon Web Services applications to access Amazon Web Services resources. Using IAM Roles Anywhere eliminates the need to manage long-term credentials for workloads running outside of Amazon Web Services.

To use IAM Roles Anywhere, your workloads must use X.509 certificates issued by their certificate authority (CA). You register the CA with IAM Roles Anywhere as a trust anchor to establish trust between your public key infrastructure (PKI) and IAM Roles Anywhere. If you don't manage your own PKI system, you can use Private Certificate Authority to create a CA and then use that to establish trust with IAM Roles Anywhere.

This guide describes the IAM Roles Anywhere operations that you can call programmatically. For more information about IAM Roles Anywhere, see the IAM Roles Anywhere User Guide.

## Usage

```
iamrolesanywhere(
  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 <- iamrolesanywhere(
  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_profile | Creates a profile, a list of the roles that Roles Anywhere service is trusted to assume |
| create_trust_anchor | Creates a trust anchor to establish trust between IAM Roles Anywhere and your certificate autho |
| delete_attribute_mapping | Delete an entry from the attribute mapping rules enforced by a given profile |
| delete_crl | Deletes a certificate revocation list (CRL) |
| delete_profile | Deletes a profile |
| delete_trust_anchor | Deletes a trust anchor |
| disable_crl | Disables a certificate revocation list (CRL) |
| disable_profile | Disables a profile |
| disable_trust_anchor | Disables a trust anchor |
| enable_crl | Enables a certificate revocation list (CRL) |
| enable_profile | Enables temporary credential requests for a profile |
| enable_trust_anchor | Enables a trust anchor |
| get_crl | Gets a certificate revocation list (CRL) |
| get_profile | Gets a profile |
| get_subject | Gets a subject, which associates a certificate identity with authentication attempts |
| get_trust_anchor | Gets a trust anchor |
| import_crl | Imports the certificate revocation list (CRL) |
| list_crls | Lists all certificate revocation lists (CRL) in the authenticated account and Amazon Web Services |
| list_profiles | Lists all profiles in the authenticated account and Amazon Web Services Region |
| list_subjects | Lists the subjects in the authenticated account and Amazon Web Services Region |

| | |
|---|---|
| list_tags_for_resource | Lists the tags attached to the resource |
| list_trust_anchors | Lists the trust anchors in the authenticated account and Amazon Web Services Region |
| put_attribute_mapping | Put an entry in the attribute mapping rules that will be enforced by a given profile |
| put_notification_settings | Attaches a list of notification settings to a trust anchor |
| reset_notification_settings | Resets the custom notification setting to IAM Roles Anywhere default setting |
| tag_resource | Attaches tags to a resource |
| untag_resource | Removes tags from the resource |
| update_crl | Updates the certificate revocation list (CRL) |
| update_profile | Updates a profile, a list of the roles that IAM Roles Anywhere service is trusted to assume |
| update_trust_anchor | Updates a trust anchor |

## Examples

```
## Not run:
svc <- iamrolesanywhere()
svc$create_profile(
  Foo = 123
)

## End(Not run)
```

---

`identitystore` *AWS SSO Identity Store*

---

## Description

The Identity Store service used by IAM Identity Center provides a single place to retrieve all of your identities (users and groups). For more information, see the IAM Identity Center User Guide.

This reference guide describes the identity store operations that you can call programmatically and includes detailed information about data types and errors.

IAM Identity Center uses the `sso` and `identitystore` API namespaces.

## Usage

```
identitystore(
  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 <- identitystore(
  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_group | Creates a group within the specified identity store |
| create_group_membership | Creates a relationship between a member and a group |
| create_user | Creates a user within the specified identity store |
| delete_group | Delete a group within an identity store given GroupId |
| delete_group_membership | Delete a membership within a group given MembershipId |
| delete_user | Deletes a user within an identity store given UserId |
| describe_group | Retrieves the group metadata and attributes from GroupId in an identity store |
| describe_group_membership | Retrieves membership metadata and attributes from MembershipId in an identity stor |
| describe_user | Retrieves the user metadata and attributes from the UserId in an identity store |
| get_group_id | Retrieves GroupId in an identity store |
| get_group_membership_id | Retrieves the MembershipId in an identity store |
| get_user_id | Retrieves the UserId in an identity store |
| is_member_in_groups | Checks the user's membership in all requested groups and returns if the member exis |
| list_group_memberships | For the specified group in the specified identity store, returns the list of all GroupMer |
| list_group_memberships_for_member | For the specified member in the specified identity store, returns the list of all GroupM |
| list_groups | Lists all groups in the identity store |
| list_users | Lists all users in the identity store |
| update_group | For the specified group in the specified identity store, updates the group metadata and |
| update_user | For the specified user in the specified identity store, updates the user metadata and att |

## Examples

```
## Not run:
svc <- identitystore()
svc$create_group(
  Foo = 123
)

## End(Not run)
```

---

`inspector` *Amazon Inspector*

---

## Description

Amazon Inspector enables you to analyze the behavior of your AWS resources and to identify potential security issues. For more information, see Amazon Inspector User Guide.

## Usage

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

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

## Operations

| | |
|---|---|
| add_attributes_to_findings | Assigns attributes (key and value pairs) to the findings that are specified by the ARNs o |
| create_assessment_target | Creates a new assessment target using the ARN of the resource group that is generated |
| create_assessment_template | Creates an assessment template for the assessment target that is specified by the ARN o |
| create_exclusions_preview | Starts the generation of an exclusions preview for the specified assessment template |
| create_resource_group | Creates a resource group using the specified set of tags (key and value pairs) that are us |
| delete_assessment_run | Deletes the assessment run that is specified by the ARN of the assessment run |
| delete_assessment_target | Deletes the assessment target that is specified by the ARN of the assessment target |
| delete_assessment_template | Deletes the assessment template that is specified by the ARN of the assessment templat |
| describe_assessment_runs | Describes the assessment runs that are specified by the ARNs of the assessment runs |
| describe_assessment_targets | Describes the assessment targets that are specified by the ARNs of the assessment targe |
| describe_assessment_templates | Describes the assessment templates that are specified by the ARNs of the assessment te |
| describe_cross_account_access_role | Describes the IAM role that enables Amazon Inspector to access your AWS account |
| describe_exclusions | Describes the exclusions that are specified by the exclusions' ARNs |
| describe_findings | Describes the findings that are specified by the ARNs of the findings |
| describe_resource_groups | Describes the resource groups that are specified by the ARNs of the resource groups |
| describe_rules_packages | Describes the rules packages that are specified by the ARNs of the rules packages |
| get_assessment_report | Produces an assessment report that includes detailed and comprehensive results of a sp |
| get_exclusions_preview | Retrieves the exclusions preview (a list of ExclusionPreview objects) specified by the p |
| get_telemetry_metadata | Information about the data that is collected for the specified assessment run |
| list_assessment_run_agents | Lists the agents of the assessment runs that are specified by the ARNs of the assessmen |
| list_assessment_runs | Lists the assessment runs that correspond to the assessment templates that are specified |
| list_assessment_targets | Lists the ARNs of the assessment targets within this AWS account |
| list_assessment_templates | Lists the assessment templates that correspond to the assessment targets that are specifi |
| list_event_subscriptions | Lists all the event subscriptions for the assessment template that is specified by the AR |
| list_exclusions | List exclusions that are generated by the assessment run |
| list_findings | Lists findings that are generated by the assessment runs that are specified by the ARNs |
| list_rules_packages | Lists all available Amazon Inspector rules packages |
| list_tags_for_resource | Lists all tags associated with an assessment template |
| preview_agents | Previews the agents installed on the EC2 instances that are part of the specified assessm |
| register_cross_account_access_role | Registers the IAM role that grants Amazon Inspector access to AWS Services needed t |
| remove_attributes_from_findings | Removes entire attributes (key and value pairs) from the findings that are specified by t |
| set_tags_for_resource | Sets tags (key and value pairs) to the assessment template that is specified by the ARN |
| start_assessment_run | Starts the assessment run specified by the ARN of the assessment template |
| stop_assessment_run | Stops the assessment run that is specified by the ARN of the assessment run |
| subscribe_to_event | Enables the process of sending Amazon Simple Notification Service (SNS) notification |
| unsubscribe_from_event | Disables the process of sending Amazon Simple Notification Service (SNS) notificatio |
| update_assessment_target | Updates the assessment target that is specified by the ARN of the assessment target |

## Examples

```
## Not run:
svc <- inspector()
# Assigns attributes (key and value pairs) to the findings that are
# specified by the ARNs of the findings.
svc$add_attributes_to_findings(
  attributes = list(
    list(
      key = "Example",
      value = "example"
    )
  ),
  findingArns = list(
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-..."
  )
)

## End(Not run)
```

---

`inspector2` *Inspector2*

---

## Description

Amazon Inspector is a vulnerability discovery service that automates continuous scanning for security vulnerabilities within your Amazon EC2, Amazon ECR, and Amazon Web Services Lambda environments.

## Usage

```
inspector2(
  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 <- inspector2(
  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_member | Associates an Amazon Web Services account with an Amazon Inspect |
| batch_get_account_status | Retrieves the Amazon Inspector status of multiple Amazon Web Servi |
| batch_get_code_snippet | Retrieves code snippets from findings that Amazon Inspector detected |
| batch_get_finding_details | Gets vulnerability details for findings |
| batch_get_free_trial_info | Gets free trial status for multiple Amazon Web Services accounts |
| batch_get_member_ec_2_deep_inspection_status | Retrieves Amazon Inspector deep inspection activation status of multi |
| batch_update_member_ec_2_deep_inspection_status | Activates or deactivates Amazon Inspector deep inspection for the pro |
| cancel_findings_report | Cancels the given findings report |
| cancel_sbom_export | Cancels a software bill of materials (SBOM) report |
| create_cis_scan_configuration | Creates a CIS scan configuration |
| create_filter | Creates a filter resource using specified filter criteria |
| create_findings_report | Creates a finding report |
| create_sbom_export | Creates a software bill of materials (SBOM) report |
| delete_cis_scan_configuration | Deletes a CIS scan configuration |
| delete_filter | Deletes a filter resource |
| describe_organization_configuration | Describe Amazon Inspector configuration settings for an Amazon Wel |
| disable | Disables Amazon Inspector scans for one or more Amazon Web Servi |
| disable_delegated_admin_account | Disables the Amazon Inspector delegated administrator for your organ |
| disassociate_member | Disassociates a member account from an Amazon Inspector delegated |
| enable | Enables Amazon Inspector scans for one or more Amazon Web Servic |
| enable_delegated_admin_account | Enables the Amazon Inspector delegated administrator for your Organ |
| get_cis_scan_report | Retrieves a CIS scan report |
| get_cis_scan_result_details | Retrieves CIS scan result details |
| get_configuration | Retrieves setting configurations for Inspector scans |
| get_delegated_admin_account | Retrieves information about the Amazon Inspector delegated administ |
| get_ec_2_deep_inspection_configuration | Retrieves the activation status of Amazon Inspector deep inspection ar |
| get_encryption_key | Gets an encryption key |
| get_findings_report_status | Gets the status of a findings report |

| | |
|---|---|
| get_member | Gets member information for your organization |
| get_sbom_export | Gets details of a software bill of materials (SBOM) report |
| list_account_permissions | Lists the permissions an account has to configure Amazon Inspector |
| list_cis_scan_configurations | Lists CIS scan configurations |
| list_cis_scan_results_aggregated_by_checks | Lists scan results aggregated by checks |
| list_cis_scan_results_aggregated_by_target_resource | Lists scan results aggregated by a target resource |
| list_cis_scans | Returns a CIS scan list |
| list_coverage | Lists coverage details for your environment |
| list_coverage_statistics | Lists Amazon Inspector coverage statistics for your environment |
| list_delegated_admin_accounts | Lists information about the Amazon Inspector delegated administrator |
| list_filters | Lists the filters associated with your account |
| list_finding_aggregations | Lists aggregated finding data for your environment based on specific c |
| list_findings | Lists findings for your environment |
| list_members | List members associated with the Amazon Inspector delegated admini |
| list_tags_for_resource | Lists all tags attached to a given resource |
| list_usage_totals | Lists the Amazon Inspector usage totals over the last 30 days |
| reset_encryption_key | Resets an encryption key |
| search_vulnerabilities | Lists Amazon Inspector coverage details for a specific vulnerability |
| send_cis_session_health | Sends a CIS session health |
| send_cis_session_telemetry | Sends a CIS session telemetry |
| start_cis_session | Starts a CIS session |
| stop_cis_session | Stops a CIS session |
| tag_resource | Adds tags to a resource |
| untag_resource | Removes tags from a resource |
| update_cis_scan_configuration | Updates a CIS scan configuration |
| update_configuration | Updates setting configurations for your Amazon Inspector account |
| update_ec_2_deep_inspection_configuration | Activates, deactivates Amazon Inspector deep inspection, or updates c |
| update_encryption_key | Updates an encryption key |
| update_filter | Specifies the action that is to be applied to the findings that match the |
| update_organization_configuration | Updates the configurations for your Amazon Inspector organization |
| update_org_ec_2_deep_inspection_configuration | Updates the Amazon Inspector deep inspection custom paths for your |

## Examples

```
## Not run:
svc <- inspector2()
svc$associate_member(
  Foo = 123
)

## End(Not run)
```

kms *AWS Key Management Service*

## Description

Key Management Service

Key Management Service (KMS) is an encryption and key management web service. This guide describes the KMS operations that you can call programmatically. For general information about KMS, see the *Key Management Service Developer Guide* .

KMS has replaced the term *customer master key (CMK)* with *KMS key* and *KMS key*. The concept has not changed. To prevent breaking changes, KMS is keeping some variations of this term.

Amazon Web Services provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, macOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to KMS and other Amazon Web Services services. For example, the SDKs take care of tasks such as signing requests (see below), managing errors, and retrying requests automatically. For more information about the Amazon Web Services SDKs, including how to download and install them, see Tools for Amazon Web Services.

We recommend that you use the Amazon Web Services SDKs to make programmatic API calls to KMS.

If you need to use FIPS 140-2 validated cryptographic modules when communicating with Amazon Web Services, use the FIPS endpoint in your preferred Amazon Web Services Region. For more information about the available FIPS endpoints, see Service endpoints in the Key Management Service topic of the *Amazon Web Services General Reference*.

All KMS API calls must be signed and be transmitted using Transport Layer Security (TLS). KMS recommends you always use the latest supported TLS version. Clients must also support cipher suites with Perfect Forward Secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

### Signing Requests

Requests must be signed using an access key ID and a secret access key. We strongly recommend that you do not use your Amazon Web Services account root access key ID and secret access key for everyday work. You can use the access key ID and secret access key for an IAM user or you can use the Security Token Service (STS) to generate temporary security credentials and use those to sign requests.

All KMS requests must be signed with Signature Version 4.

### Logging API Requests

KMS supports CloudTrail, a service that logs Amazon Web Services API calls and related events for your Amazon Web Services account and delivers them to an Amazon S3 bucket that you specify. By using the information collected by CloudTrail, you can determine what requests were made to KMS, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the CloudTrail User Guide.

### Additional Resources

For more information about credentials and request signing, see the following:

- Amazon Web Services Security Credentials - This topic provides general information about the types of credentials used to access Amazon Web Services.
- Temporary Security Credentials - This section of the *IAM User Guide* describes how to create and use temporary security credentials.
- Signature Version 4 Signing Process - This set of topics walks you through the process of signing a request using an access key ID and a secret access key.

**Commonly Used API Operations**

Of the API operations discussed in this guide, the following will prove the most useful for most applications. You will likely perform operations other than these, such as creating keys and assigning policies, by using the console.

- `encrypt`
- `decrypt`
- `generate_data_key`
- `generate_data_key_without_plaintext`

## Usage

```
kms(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 <- kms(
  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_key_deletion | Cancels the deletion of a KMS key |
| connect_custom_key_store | Connects or reconnects a custom key store to its backing key store |
| create_alias | Creates a friendly name for a KMS key |
| create_custom_key_store | Creates a custom key store backed by a key store that you own and manage |
| create_grant | Adds a grant to a KMS key |
| create_key | Creates a unique customer managed KMS key in your Amazon Web Services ac |
| decrypt | Decrypts ciphertext that was encrypted by a KMS key using any of the following |
| delete_alias | Deletes the specified alias |
| delete_custom_key_store | Deletes a custom key store |
| delete_imported_key_material | Deletes key material that was previously imported |
| derive_shared_secret | Derives a shared secret using a key agreement algorithm |
| describe_custom_key_stores | Gets information about custom key stores in the account and Region |
| describe_key | Provides detailed information about a KMS key |
| disable_key | Sets the state of a KMS key to disabled |
| disable_key_rotation | Disables automatic rotation of the key material of the specified symmetric encryp |
| disconnect_custom_key_store | Disconnects the custom key store from its backing key store |
| enable_key | Sets the key state of a KMS key to enabled |
| enable_key_rotation | Enables automatic rotation of the key material of the specified symmetric encryp |
| encrypt | Encrypts plaintext of up to 4,096 bytes using a KMS key |
| generate_data_key | Returns a unique symmetric data key for use outside of KMS |
| generate_data_key_pair | Returns a unique asymmetric data key pair for use outside of KMS |
| generate_data_key_pair_without_plaintext | Returns a unique asymmetric data key pair for use outside of KMS |
| generate_data_key_without_plaintext | Returns a unique symmetric data key for use outside of KMS |
| generate_mac | Generates a hash-based message authentication code (HMAC) for a message usi |
| generate_random | Returns a random byte string that is cryptographically secure |
| get_key_policy | Gets a key policy attached to the specified KMS key |
| get_key_rotation_status | Provides detailed information about the rotation status for a KMS key, including |
| get_parameters_for_import | Returns the public key and an import token you need to import or reimport key m |
| get_public_key | Returns the public key of an asymmetric KMS key |
| import_key_material | Imports or reimports key material into an existing KMS key that was created wit |
| list_aliases | Gets a list of aliases in the caller's Amazon Web Services account and region |
| list_grants | Gets a list of all grants for the specified KMS key |
| list_key_policies | Gets the names of the key policies that are attached to a KMS key |
| list_key_rotations | Returns information about all completed key material rotations for the specified |
| list_keys | Gets a list of all KMS keys in the caller's Amazon Web Services account and Re |
| list_resource_tags | Returns all tags on the specified KMS key |
| list_retirable_grants | Returns information about all grants in the Amazon Web Services account and R |
| put_key_policy | Attaches a key policy to the specified KMS key |
| re_encrypt | Decrypts ciphertext and then reencrypts it entirely within KMS |
| replicate_key | Replicates a multi-Region key into the specified Region |
| retire_grant | Deletes a grant |
| revoke_grant | Deletes the specified grant |
| rotate_key_on_demand | Immediately initiates rotation of the key material of the specified symmetric encr |
| schedule_key_deletion | Schedules the deletion of a KMS key |
| sign | Creates a digital signature for a message or message digest by using the private k |
| tag_resource | Adds or edits tags on a customer managed key |

| | |
|---|---|
| untag_resource | Deletes tags from a customer managed key |
| update_alias | Associates an existing KMS alias with a different KMS key |
| update_custom_key_store | Changes the properties of a custom key store |
| update_key_description | Updates the description of a KMS key |
| update_primary_region | Changes the primary key of a multi-Region key |
| verify | Verifies a digital signature that was generated by the Sign operation |
| verify_mac | Verifies the hash-based message authentication code (HMAC) for a specified me |

## Examples

```
## Not run:
svc <- kms()
# The following example cancels deletion of the specified KMS key.
svc$cancel_key_deletion(
  KeyId = "1234abcd-12ab-34cd-56ef-1234567890ab"
)

## End(Not run)
```

---

`macie2` *Amazon Macie 2*

---

## Description

Amazon Macie

## Usage

```
macie2(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 <- macie2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
```

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

## Operations

| | |
|---|---|
| accept_invitation | Accepts an Amazon Macie membership invitation that was received from a sp |
| batch_get_custom_data_identifiers | Retrieves information about one or more custom data identifiers |
| batch_update_automated_discovery_accounts | Changes the status of automated sensitive data discovery for one or more acc |
| create_allow_list | Creates and defines the settings for an allow list |
| create_classification_job | Creates and defines the settings for a classification job |
| create_custom_data_identifier | Creates and defines the criteria and other settings for a custom data identifier |
| create_findings_filter | Creates and defines the criteria and other settings for a findings filter |
| create_invitations | Sends an Amazon Macie membership invitation to one or more accounts |
| create_member | Associates an account with an Amazon Macie administrator account |
| create_sample_findings | Creates sample findings |
| decline_invitations | Declines Amazon Macie membership invitations that were received from spec |
| delete_allow_list | Deletes an allow list |
| delete_custom_data_identifier | Soft deletes a custom data identifier |
| delete_findings_filter | Deletes a findings filter |
| delete_invitations | Deletes Amazon Macie membership invitations that were received from speci |
| delete_member | Deletes the association between an Amazon Macie administrator account and |
| describe_buckets | Retrieves (queries) statistical data and other information about one or more S |
| describe_classification_job | Retrieves the status and settings for a classification job |
| describe_organization_configuration | Retrieves the Amazon Macie configuration settings for an organization in Org |
| disable_macie | Disables Amazon Macie and deletes all settings and resources for a Macie ac |
| disable_organization_admin_account | Disables an account as the delegated Amazon Macie administrator account fo |
| disassociate_from_administrator_account | Disassociates a member account from its Amazon Macie administrator accou |
| disassociate_from_master_account | (Deprecated) Disassociates a member account from its Amazon Macie admin |
| disassociate_member | Disassociates an Amazon Macie administrator account from a member accou |
| enable_macie | Enables Amazon Macie and specifies the configuration settings for a Macie a |
| enable_organization_admin_account | Designates an account as the delegated Amazon Macie administrator account |
| get_administrator_account | Retrieves information about the Amazon Macie administrator account for an a |
| get_allow_list | Retrieves the settings and status of an allow list |
| get_automated_discovery_configuration | Retrieves the configuration settings and status of automated sensitive data dis |
| get_bucket_statistics | Retrieves (queries) aggregated statistical data about all the S3 buckets that An |
| get_classification_export_configuration | Retrieves the configuration settings for storing data classification results |
| get_classification_scope | Retrieves the classification scope settings for an account |
| get_custom_data_identifier | Retrieves the criteria and other settings for a custom data identifier |
| get_findings | Retrieves the details of one or more findings |

| | |
|---|---|
| get_findings_filter | Retrieves the criteria and other settings for a findings filter |
| get_findings_publication_configuration | Retrieves the configuration settings for publishing findings to Security Hub |
| get_finding_statistics | Retrieves (queries) aggregated statistical data about findings |
| get_invitations_count | Retrieves the count of Amazon Macie membership invitations that were recei |
| get_macie_session | Retrieves the status and configuration settings for an Amazon Macie account |
| get_master_account | (Deprecated) Retrieves information about the Amazon Macie administrator ac |
| get_member | Retrieves information about an account that's associated with an Amazon Ma |
| get_resource_profile | Retrieves (queries) sensitive data discovery statistics and the sensitivity score |
| get_reveal_configuration | Retrieves the status and configuration settings for retrieving occurrences of se |
| get_sensitive_data_occurrences | Retrieves occurrences of sensitive data reported by a finding |
| get_sensitive_data_occurrences_availability | Checks whether occurrences of sensitive data can be retrieved for a finding |
| get_sensitivity_inspection_template | Retrieves the settings for the sensitivity inspection template for an account |
| get_usage_statistics | Retrieves (queries) quotas and aggregated usage data for one or more account |
| get_usage_totals | Retrieves (queries) aggregated usage data for an account |
| list_allow_lists | Retrieves a subset of information about all the allow lists for an account |
| list_automated_discovery_accounts | Retrieves the status of automated sensitive data discovery for one or more acc |
| list_classification_jobs | Retrieves a subset of information about one or more classification jobs |
| list_classification_scopes | Retrieves a subset of information about the classification scope for an account |
| list_custom_data_identifiers | Retrieves a subset of information about the custom data identifiers for an acc |
| list_findings | Retrieves a subset of information about one or more findings |
| list_findings_filters | Retrieves a subset of information about all the findings filters for an account |
| list_invitations | Retrieves information about Amazon Macie membership invitations that were |
| list_managed_data_identifiers | Retrieves information about all the managed data identifiers that Amazon Ma |
| list_members | Retrieves information about the accounts that are associated with an Amazon |
| list_organization_admin_accounts | Retrieves information about the delegated Amazon Macie administrator accou |
| list_resource_profile_artifacts | Retrieves information about objects that Amazon Macie selected from an S3 b |
| list_resource_profile_detections | Retrieves information about the types and amount of sensitive data that Amaz |
| list_sensitivity_inspection_templates | Retrieves a subset of information about the sensitivity inspection template for |
| list_tags_for_resource | Retrieves the tags (keys and values) that are associated with an Amazon Maci |
| put_classification_export_configuration | Adds or updates the configuration settings for storing data classification result |
| put_findings_publication_configuration | Updates the configuration settings for publishing findings to Security Hub |
| search_resources | Retrieves (queries) statistical data and other information about Amazon Web S |
| tag_resource | Adds or updates one or more tags (keys and values) that are associated with a |
| test_custom_data_identifier | Tests criteria for a custom data identifier |
| untag_resource | Removes one or more tags (keys and values) from an Amazon Macie resource |
| update_allow_list | Updates the settings for an allow list |
| update_automated_discovery_configuration | Changes the configuration settings and status of automated sensitive data disc |
| update_classification_job | Changes the status of a classification job |
| update_classification_scope | Updates the classification scope settings for an account |
| update_findings_filter | Updates the criteria and other settings for a findings filter |
| update_macie_session | Suspends or re-enables Amazon Macie, or updates the configuration settings f |
| update_member_session | Enables an Amazon Macie administrator to suspend or re-enable Macie for a |
| update_organization_configuration | Updates the Amazon Macie configuration settings for an organization in Orga |
| update_resource_profile | Updates the sensitivity score for an S3 bucket |
| update_resource_profile_detections | Updates the sensitivity scoring settings for an S3 bucket |
| update_reveal_configuration | Updates the status and configuration settings for retrieving occurrences of sen |
| update_sensitivity_inspection_template | Updates the settings for the sensitivity inspection template for an account |

## Examples

```
## Not run:
svc <- macie2()
svc$accept_invitation(
  Foo = 123
)

## End(Not run)
```

---

`pcaconnectorad` *PcaConnectorAd*

---

## Description

Amazon Web Services Private CA Connector for Active Directory creates a connector between Amazon Web Services Private CA and Active Directory (AD) that enables you to provision security certificates for AD signed by a private CA that you own. For more information, see Amazon Web Services Private CA Connector for Active Directory.

## Usage

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

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

## Operations

| | |
|---|---|
| create_connector | Creates a connector between Amazon Web Services Private CA and an Activ |
| create_directory_registration | Creates a directory registration that authorizes communication between Amaz |
| create_service_principal_name | Creates a service principal name (SPN) for the service account in Active Dire |
| create_template | Creates an Active Directory compatible certificate template |
| create_template_group_access_control_entry | Create a group access control entry |
| delete_connector | Deletes a connector for Active Directory |
| delete_directory_registration | Deletes a directory registration |
| delete_service_principal_name | Deletes the service principal name (SPN) used by a connector to authenticate |
| delete_template | Deletes a template |
| delete_template_group_access_control_entry | Deletes a group access control entry |
| get_connector | Lists information about your connector |
| get_directory_registration | A structure that contains information about your directory registration |
| get_service_principal_name | Lists the service principal name that the connector uses to authenticate with A |
| get_template | Retrieves a certificate template that the connector uses to issue certificates fro |
| get_template_group_access_control_entry | Retrieves the group access control entries for a template |
| list_connectors | Lists the connectors that you created by using the https://docs |
| list_directory_registrations | Lists the directory registrations that you created by using the https://docs |
| list_service_principal_names | Lists the service principal names that the connector uses to authenticate with |
| list_tags_for_resource | Lists the tags, if any, that are associated with your resource |
| list_template_group_access_control_entries | Lists group access control entries you created |
| list_templates | Lists the templates, if any, that are associated with a connector |
| tag_resource | Adds one or more tags to your resource |
| untag_resource | Removes one or more tags from your resource |
| update_template | Update template configuration to define the information included in certificat |
| update_template_group_access_control_entry | Update a group access control entry you created using CreateTemplateGroup |

## Examples

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

## End(Not run)
```

`ram` *AWS Resource Access Manager*

## Description

This is the *Resource Access Manager API Reference*. This documentation provides descriptions and syntax for each of the actions and data types in RAM. RAM is a service that helps you securely share your Amazon Web Services resources to other Amazon Web Services accounts. If you use Organizations to manage your accounts, then you can share your resources with your entire organization or to organizational units (OUs). For supported resource types, you can also share resources with individual Identity and Access Management (IAM) roles and users.

To learn more about RAM, see the following resources:

- Resource Access Manager product page
- Resource Access Manager User Guide

## Usage

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

## Operations

| | |
|---|---|
| accept_resource_share_invitation | Accepts an invitation to a resource share from another Amazon Web Service |
| associate_resource_share | Adds the specified list of principals and list of resources to a resource share |
| associate_resource_share_permission | Adds or replaces the RAM permission for a resource type included in a reso |
| create_permission | Creates a customer managed permission for a specified resource type that yo |
| create_permission_version | Creates a new version of the specified customer managed permission |
| create_resource_share | Creates a resource share |
| delete_permission | Deletes the specified customer managed permission in the Amazon Web Serv |
| delete_permission_version | Deletes one version of a customer managed permission |
| delete_resource_share | Deletes the specified resource share |
| disassociate_resource_share | Removes the specified principals or resources from participating in the speci |
| disassociate_resource_share_permission | Removes a managed permission from a resource share |
| enable_sharing_with_aws_organization | Enables resource sharing within your organization in Organizations |
| get_permission | Retrieves the contents of a managed permission in JSON format |
| get_resource_policies | Retrieves the resource policies for the specified resources that you own and h |
| get_resource_share_associations | Retrieves the lists of resources and principals that associated for resource sha |
| get_resource_share_invitations | Retrieves details about invitations that you have received for resource shares |
| get_resource_shares | Retrieves details about the resource shares that you own or that are shared wi |
| list_pending_invitation_resources | Lists the resources in a resource share that is shared with you but for which t |
| list_permission_associations | Lists information about the managed permission and its associations to any r |
| list_permissions | Retrieves a list of available RAM permissions that you can use for the suppo |
| list_permission_versions | Lists the available versions of the specified RAM permission |
| list_principals | Lists the principals that you are sharing resources with or that are sharing res |
| list_replace_permission_associations_work | Retrieves the current status of the asynchronous tasks performed by RAM wh |
| list_resources | Lists the resources that you added to a resource share or the resources that ar |
| list_resource_share_permissions | Lists the RAM permissions that are associated with a resource share |
| list_resource_types | Lists the resource types that can be shared by RAM |
| promote_permission_created_from_policy | When you attach a resource-based policy to a resource, RAM automatically c |
| promote_resource_share_created_from_policy | When you attach a resource-based policy to a resource, RAM automatically c |
| reject_resource_share_invitation | Rejects an invitation to a resource share from another Amazon Web Services |
| replace_permission_associations | Updates all resource shares that use a managed permission to a different man |
| set_default_permission_version | Designates the specified version number as the default version for the specifi |
| tag_resource | Adds the specified tag keys and values to a resource share or managed permi |
| untag_resource | Removes the specified tag key and value pairs from the specified resource sh |
| update_resource_share | Modifies some of the properties of the specified resource share |

## Examples

```
## Not run:
svc <- ram()
svc$accept_resource_share_invitation(
  Foo = 123
)

## End(Not run)
```

`secretsmanager` *AWS Secrets Manager*

## Description

Amazon Web Services Secrets Manager

Amazon Web Services Secrets Manager provides a service to enable you to store, manage, and retrieve, secrets.

This guide provides descriptions of the Secrets Manager API. For more information about using this service, see the Amazon Web Services Secrets Manager User Guide.

**API Version**

This version of the Secrets Manager API Reference documents the Secrets Manager API version 2017-10-17.

For a list of endpoints, see Amazon Web Services Secrets Manager endpoints.

**Support and Feedback for Amazon Web Services Secrets Manager**

We welcome your feedback. Send your comments to awssecretsmanager-feedback@amazon.com, or post your feedback and questions in the Amazon Web Services Secrets Manager Discussion Forum. For more information about the Amazon Web Services Discussion Forums, see Forums Help.

**Logging API Requests**

Amazon Web Services Secrets Manager supports Amazon Web Services CloudTrail, a service that records Amazon Web Services API calls for your Amazon Web Services account and delivers log files to an Amazon S3 bucket. By using information that's collected by Amazon Web Services CloudTrail, you can determine the requests successfully made to Secrets Manager, who made the request, when it was made, and so on. For more about Amazon Web Services Secrets Manager and support for Amazon Web Services CloudTrail, see Logging Amazon Web Services Secrets Manager Events with Amazon Web Services CloudTrail in the *Amazon Web Services Secrets Manager User Guide*. To learn more about CloudTrail, including enabling it and find your log files, see the Amazon Web Services CloudTrail User Guide.

## Usage

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

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

## Operations

| | |
|---|---|
| batch_get_secret_value | Retrieves the contents of the encrypted fields SecretString or SecretBinary for up to 20 se |
| cancel_rotate_secret | Turns off automatic rotation, and if a rotation is currently in progress, cancels the rotation |
| create_secret | Creates a new secret |
| delete_resource_policy | Deletes the resource-based permission policy attached to the secret |
| delete_secret | Deletes a secret and all of its versions |
| describe_secret | Retrieves the details of a secret |
| get_random_password | Generates a random password |
| get_resource_policy | Retrieves the JSON text of the resource-based policy document attached to the secret |
| get_secret_value | Retrieves the contents of the encrypted fields SecretString or SecretBinary from the speci |
| list_secrets | Lists the secrets that are stored by Secrets Manager in the Amazon Web Services account |
| list_secret_version_ids | Lists the versions of a secret |
| put_resource_policy | Attaches a resource-based permission policy to a secret |
| put_secret_value | Creates a new version with a new encrypted secret value and attaches it to the secret |
| remove_regions_from_replication | For a secret that is replicated to other Regions, deletes the secret replicas from the Region |
| replicate_secret_to_regions | Replicates the secret to a new Regions |
| restore_secret | Cancels the scheduled deletion of a secret by removing the DeletedDate time stamp |
| rotate_secret | Configures and starts the asynchronous process of rotating the secret |
| stop_replication_to_replica | Removes the link between the replica secret and the primary secret and promotes the repl |
| tag_resource | Attaches tags to a secret |
| untag_resource | Removes specific tags from a secret |
| update_secret | Modifies the details of a secret, including metadata and the secret value |
| update_secret_version_stage | Modifies the staging labels attached to a version of a secret |
| validate_resource_policy | Validates that a resource policy does not grant a wide range of principals access to your s |

## Examples

```
## Not run:
svc <- secretsmanager()
# The following example shows how to cancel rotation for a secret. The
# operation sets the RotationEnabled field to false and cancels all
# scheduled rotations. To resume scheduled rotations, you must re-enable
# rotation by calling the rotate-secret operation.
svc$cancel_rotate_secret(
  SecretId = "MyTestDatabaseSecret"
)

## End(Not run)
```

---

securityhub *AWS SecurityHub*

---

## Description

Security Hub provides you with a comprehensive view of your security state in Amazon Web Services and helps you assess your Amazon Web Services environment against security industry standards and best practices.

Security Hub collects security data across Amazon Web Services accounts, Amazon Web Services services, and supported third-party products and helps you analyze your security trends and identify the highest priority security issues.

To help you manage the security state of your organization, Security Hub supports multiple security standards. These include the Amazon Web Services Foundational Security Best Practices (FSBP) standard developed by Amazon Web Services, and external compliance frameworks such as the Center for Internet Security (CIS), the Payment Card Industry Data Security Standard (PCI DSS), and the National Institute of Standards and Technology (NIST). Each standard includes several security controls, each of which represents a security best practice. Security Hub runs checks against security controls and generates control findings to help you assess your compliance against security best practices.

In addition to generating control findings, Security Hub also receives findings from other Amazon Web Services services, such as Amazon GuardDuty and Amazon Inspector, and supported third-party products. This gives you a single pane of glass into a variety of security-related issues. You can also send Security Hub findings to other Amazon Web Services services and supported third-party products.

Security Hub offers automation features that help you triage and remediate security issues. For example, you can use automation rules to automatically update critical findings when a security check fails. You can also leverage the integration with Amazon EventBridge to trigger automatic responses to specific findings.

This guide, the *Security Hub API Reference*, provides information about the Security Hub API. This includes supported resources, HTTP methods, parameters, and schemas. If you're new to Security Hub, you might find it helpful to also review the *Security Hub User Guide* . The user guide explains key concepts and provides procedures that demonstrate how to use Security Hub features.

It also provides information about topics such as integrating Security Hub with other Amazon Web Services services.

In addition to interacting with Security Hub by making calls to the Security Hub API, you can use a current version of an Amazon Web Services command line tool or SDK. Amazon Web Services provides tools and SDKs that consist of libraries and sample code for various languages and platforms, such as PowerShell, Java, Go, Python, C++, and .NET. These tools and SDKs provide convenient, programmatic access to Security Hub and other Amazon Web Services services . They also handle tasks such as signing requests, managing errors, and retrying requests automatically. For information about installing and using the Amazon Web Services tools and SDKs, see Tools to Build on Amazon Web Services.

With the exception of operations that are related to central configuration, Security Hub API requests are executed only in the Amazon Web Services Region that is currently active or in the specific Amazon Web Services Region that you specify in your request. Any configuration or settings change that results from the operation is applied only to that Region. To make the same change in other Regions, call the same API operation in each Region in which you want to apply the change. When you use central configuration, API requests for enabling Security Hub, standards, and controls are executed in the home Region and all linked Regions. For a list of central configuration operations, see the Central configuration terms and concepts section of the *Security Hub User Guide*.

The following throttling limits apply to Security Hub API operations.

- `batch_enable_standards` - `RateLimit` of 1 request per second. `BurstLimit` of 1 request per second.
- `get_findings` - `RateLimit` of 3 requests per second. `BurstLimit` of 6 requests per second.
- `batch_import_findings` - `RateLimit` of 10 requests per second. `BurstLimit` of 30 requests per second.
- `batch_update_findings` - `RateLimit` of 10 requests per second. `BurstLimit` of 30 requests per second.
- `update_standards_control` - `RateLimit` of 1 request per second. `BurstLimit` of 5 requests per second.
- All other operations - `RateLimit` of 10 requests per second. `BurstLimit` of 30 requests per second.

## Usage

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

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

## Operations

| | |
|---|---|
| accept_administrator_invitation | We recommend using Organizations instead of Security Hub invitations to ma |
| accept_invitation | This method is deprecated |
| batch_delete_automation_rules | Deletes one or more automation rules |
| batch_disable_standards | Disables the standards specified by the provided StandardsSubscriptionArns |
| batch_enable_standards | Enables the standards specified by the provided StandardsArn |
| batch_get_automation_rules | Retrieves a list of details for automation rules based on rule Amazon Resourc |
| batch_get_configuration_policy_associations | Returns associations between an Security Hub configuration and a batch of ta |
| batch_get_security_controls | Provides details about a batch of security controls for the current Amazon We |
| batch_get_standards_control_associations | For a batch of security controls and standards, identifies whether each control |
| batch_import_findings | Imports security findings generated by a finding provider into Security Hub |
| batch_update_automation_rules | Updates one or more automation rules based on rule Amazon Resource Name |
| batch_update_findings | Used by Security Hub customers to update information about their investigati |
| batch_update_standards_control_associations | For a batch of security controls and standards, this operation updates the enab |
| create_action_target | Creates a custom action target in Security Hub |
| create_automation_rule | Creates an automation rule based on input parameters |
| create_configuration_policy | Creates a configuration policy with the defined configuration |
| create_finding_aggregator | The aggregation Region is now called the home Region |
| create_insight | Creates a custom insight in Security Hub |
| create_members | Creates a member association in Security Hub between the specified accounts |
| decline_invitations | We recommend using Organizations instead of Security Hub invitations to ma |
| delete_action_target | Deletes a custom action target from Security Hub |
| delete_configuration_policy | Deletes a configuration policy |
| delete_finding_aggregator | The aggregation Region is now called the home Region |
| delete_insight | Deletes the insight specified by the InsightArn |
| delete_invitations | We recommend using Organizations instead of Security Hub invitations to ma |

| | |
|---|---|
| delete_members | Deletes the specified member accounts from Security Hub |
| describe_action_targets | Returns a list of the custom action targets in Security Hub in your account |
| describe_hub | Returns details about the Hub resource in your account, including the HubArn |
| describe_organization_configuration | Returns information about the way your organization is configured in Security |
| describe_products | Returns information about product integrations in Security Hub |
| describe_standards | Returns a list of the available standards in Security Hub |
| describe_standards_controls | Returns a list of security standards controls |
| disable_import_findings_for_product | Disables the integration of the specified product with Security Hub |
| disable_organization_admin_account | Disables a Security Hub administrator account |
| disable_security_hub | Disables Security Hub in your account only in the current Amazon Web Servi |
| disassociate_from_administrator_account | Disassociates the current Security Hub member account from the associated a |
| disassociate_from_master_account | This method is deprecated |
| disassociate_members | Disassociates the specified member accounts from the associated administrato |
| enable_import_findings_for_product | Enables the integration of a partner product with Security Hub |
| enable_organization_admin_account | Designates the Security Hub administrator account for an organization |
| enable_security_hub | Enables Security Hub for your account in the current Region or the Region yo |
| get_administrator_account | Provides the details for the Security Hub administrator account for the current |
| get_configuration_policy | Provides information about a configuration policy |
| get_configuration_policy_association | Returns the association between a configuration and a target account, organiza |
| get_enabled_standards | Returns a list of the standards that are currently enabled |
| get_finding_aggregator | The aggregation Region is now called the home Region |
| get_finding_history | Returns history for a Security Hub finding in the last 90 days |
| get_findings | Returns a list of findings that match the specified criteria |
| get_insight_results | Lists the results of the Security Hub insight specified by the insight ARN |
| get_insights | Lists and describes insights for the specified insight ARNs |
| get_invitations_count | We recommend using Organizations instead of Security Hub invitations to ma |
| get_master_account | This method is deprecated |
| get_members | Returns the details for the Security Hub member accounts for the specified ac |
| get_security_control_definition | Retrieves the definition of a security control |
| invite_members | We recommend using Organizations instead of Security Hub invitations to ma |
| list_automation_rules | A list of automation rules and their metadata for the calling account |
| list_configuration_policies | Lists the configuration policies that the Security Hub delegated administrator |
| list_configuration_policy_associations | Provides information about the associations for your configuration policies an |
| list_enabled_products_for_import | Lists all findings-generating solutions (products) that you are subscribed to re |
| list_finding_aggregators | If cross-Region aggregation is enabled, then ListFindingAggregators returns t |
| list_invitations | We recommend using Organizations instead of Security Hub invitations to ma |
| list_members | Lists details about all member accounts for the current Security Hub administ |
| list_organization_admin_accounts | Lists the Security Hub administrator accounts |
| list_security_control_definitions | Lists all of the security controls that apply to a specified standard |
| list_standards_control_associations | Specifies whether a control is currently enabled or disabled in each enabled st |
| list_tags_for_resource | Returns a list of tags associated with a resource |
| start_configuration_policy_association | Associates a target account, organizational unit, or the root with a specified co |
| start_configuration_policy_disassociation | Disassociates a target account, organizational unit, or the root from a specifie |
| tag_resource | Adds one or more tags to a resource |
| untag_resource | Removes one or more tags from a resource |
| update_action_target | Updates the name and description of a custom action target in Security Hub |
| update_configuration_policy | Updates a configuration policy |
| update_finding_aggregator | The aggregation Region is now called the home Region |

| | |
|---|---|
| update_findings | UpdateFindings is a deprecated operation |
| update_insight | Updates the Security Hub insight identified by the specified insight ARN |
| update_organization_configuration | Updates the configuration of your organization in Security Hub |
| update_security_control | Updates the properties of a security control |
| update_security_hub_configuration | Updates configuration options for Security Hub |
| update_standards_control | Used to control whether an individual security standard control is enabled or d |

## Examples

```
## Not run:
svc <- securityhub()
svc$accept_administrator_invitation(
  Foo = 123
)

## End(Not run)
```

---

`securitylake` *Amazon Security Lake*

---

## Description

Amazon Security Lake is a fully managed security data lake service. You can use Security Lake to automatically centralize security data from cloud, on-premises, and custom sources into a data lake that's stored in your Amazon Web Services account. Amazon Web Services Organizations is an account management service that lets you consolidate multiple Amazon Web Services accounts into an organization that you create and centrally manage. With Organizations, you can create member accounts and invite existing accounts to join your organization. Security Lake helps you analyze security data for a more complete understanding of your security posture across the entire organization. It can also help you improve the protection of your workloads, applications, and data.

The data lake is backed by Amazon Simple Storage Service (Amazon S3) buckets, and you retain ownership over your data.

Amazon Security Lake integrates with CloudTrail, a service that provides a record of actions taken by a user, role, or an Amazon Web Services service. In Security Lake, CloudTrail captures API calls for Security Lake as events. The calls captured include calls from the Security Lake console and code calls to the Security Lake API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Security Lake. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail you can determine the request that was made to Security Lake, the IP address from which the request was made, who made the request, when it was made, and additional details. To learn more about Security Lake information in CloudTrail, see the Amazon Security Lake User Guide.

Security Lake automates the collection of security-related log and event data from integrated Amazon Web Services services and third-party services. It also helps you manage the lifecycle of data

with customizable retention and replication settings. Security Lake converts ingested data into Apache Parquet format and a standard open-source schema called the Open Cybersecurity Schema Framework (OCSF).

Other Amazon Web Services services and third-party services can subscribe to the data that's stored in Security Lake for incident response and security data analytics.

## Usage

```
securitylake(
  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 <- securitylake(
  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_aws_log_source | Adds a natively supported Amazon Web Services service as an Amazon Secu |
| create_custom_log_source | Adds a third-party custom source in Amazon Security Lake, from the Amazo |
| create_data_lake | Initializes an Amazon Security Lake instance with the provided (or default) c |
| create_data_lake_exception_subscription | Creates the specified notification subscription in Amazon Security Lake for th |
| create_data_lake_organization_configuration | Automatically enables Amazon Security Lake for new member accounts in yo |
| create_subscriber | Creates a subscriber for accounts that are already enabled in Amazon Security |
| create_subscriber_notification | Notifies the subscriber when new data is written to the data lake for the source |
| delete_aws_log_source | Removes a natively supported Amazon Web Services service as an Amazon S |

| | |
|---|---|
| delete_custom_log_source | Removes a custom log source from Amazon Security Lake, to stop sending d |
| delete_data_lake | When you disable Amazon Security Lake from your account, Security Lake i |
| delete_data_lake_exception_subscription | Deletes the specified notification subscription in Amazon Security Lake for th |
| delete_data_lake_organization_configuration | Turns off automatic enablement of Amazon Security Lake for member accou |
| delete_subscriber | Deletes the subscription permission and all notification settings for accounts t |
| delete_subscriber_notification | Deletes the specified subscription notification in Amazon Security Lake for th |
| deregister_data_lake_delegated_administrator | Deletes the Amazon Security Lake delegated administrator account for the or |
| get_data_lake_exception_subscription | Retrieves the protocol and endpoint that were provided when subscribing to A |
| get_data_lake_organization_configuration | Retrieves the configuration that will be automatically set up for accounts adde |
| get_data_lake_sources | Retrieves a snapshot of the current Region, including whether Amazon Secur |
| get_subscriber | Retrieves the subscription information for the specified subscription ID |
| list_data_lake_exceptions | Lists the Amazon Security Lake exceptions that you can use to find the source |
| list_data_lakes | Retrieves the Amazon Security Lake configuration object for the specified Ar |
| list_log_sources | Retrieves the log sources |
| list_subscribers | Lists all subscribers for the specific Amazon Security Lake account ID |
| list_tags_for_resource | Retrieves the tags (keys and values) that are associated with an Amazon Secu |
| register_data_lake_delegated_administrator | Designates the Amazon Security Lake delegated administrator account for the |
| tag_resource | Adds or updates one or more tags that are associated with an Amazon Securit |
| untag_resource | Removes one or more tags (keys and values) from an Amazon Security Lake |
| update_data_lake | You can use UpdateDataLake to specify where to store your security data, ho |
| update_data_lake_exception_subscription | Updates the specified notification subscription in Amazon Security Lake for t |
| update_subscriber | Updates an existing subscription for the given Amazon Security Lake accoun |
| update_subscriber_notification | Updates an existing notification method for the subscription (SQS or HTTPs |

## Examples

```
## Not run:
svc <- securitylake()
svc$create_aws_log_source(
  Foo = 123
)

## End(Not run)
```

---

`shield` *AWS Shield*

---

## Description

Shield Advanced

This is the *Shield Advanced API Reference*. This guide is for developers who need detailed information about the Shield Advanced API actions, data types, and errors. For detailed information about WAF and Shield Advanced features and an overview of how to use the WAF and Shield Advanced APIs, see the WAF and Shield Developer Guide.

## Usage

```
shield(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 <- shield(
  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_drt_log_bucket | Authorizes the Shield Response Team (SRT) to access the specified Amazon |
| associate_drt_role | Authorizes the Shield Response Team (SRT) using the specified role, to acce |
| associate_health_check | Adds health-based detection to the Shield Advanced protection for a resourc |
| associate_proactive_engagement_details | Initializes proactive engagement and sets the list of contacts for the Shield R |
| create_protection | Enables Shield Advanced for a specific Amazon Web Services resource |
| create_protection_group | Creates a grouping of protected resources so they can be handled as a collect |
| create_subscription | Activates Shield Advanced for an account |
| delete_protection | Deletes an Shield Advanced Protection |
| delete_protection_group | Removes the specified protection group |
| delete_subscription | Removes Shield Advanced from an account |
| describe_attack | Describes the details of a DDoS attack |
| describe_attack_statistics | Provides information about the number and type of attacks Shield has detect |
| describe_drt_access | Returns the current role and list of Amazon S3 log buckets used by the Shiel |

| | |
|---|---|
| describe_emergency_contact_settings | A list of email addresses and phone numbers that the Shield Response Team |
| describe_protection | Lists the details of a Protection object |
| describe_protection_group | Returns the specification for the specified protection group |
| describe_subscription | Provides details about the Shield Advanced subscription for an account |
| disable_application_layer_automatic_response | Disable the Shield Advanced automatic application layer DDoS mitigation fe |
| disable_proactive_engagement | Removes authorization from the Shield Response Team (SRT) to notify cont |
| disassociate_drt_log_bucket | Removes the Shield Response Team's (SRT) access to the specified Amazon |
| disassociate_drt_role | Removes the Shield Response Team's (SRT) access to your Amazon Web Se |
| disassociate_health_check | Removes health-based detection from the Shield Advanced protection for a r |
| enable_application_layer_automatic_response | Enable the Shield Advanced automatic application layer DDoS mitigation fo |
| enable_proactive_engagement | Authorizes the Shield Response Team (SRT) to use email and phone to notif |
| get_subscription_state | Returns the SubscriptionState, either Active or Inactive |
| list_attacks | Returns all ongoing DDoS attacks or all DDoS attacks during a specified tim |
| list_protection_groups | Retrieves ProtectionGroup objects for the account |
| list_protections | Retrieves Protection objects for the account |
| list_resources_in_protection_group | Retrieves the resources that are included in the protection group |
| list_tags_for_resource | Gets information about Amazon Web Services tags for a specified Amazon R |
| tag_resource | Adds or updates tags for a resource in Shield |
| untag_resource | Removes tags from a resource in Shield |
| update_application_layer_automatic_response | Updates an existing Shield Advanced automatic application layer DDoS miti |
| update_emergency_contact_settings | Updates the details of the list of email addresses and phone numbers that the |
| update_protection_group | Updates an existing protection group |
| update_subscription | Updates the details of an existing subscription |

## Examples

```
## Not run:
svc <- shield()
svc$associate_drt_log_bucket(
  Foo = 123
)

## End(Not run)
```

---

`sso` *AWS Single Sign-On*

---

## Description

AWS IAM Identity Center (successor to AWS Single Sign-On) Portal is a web service that makes it easy for you to assign user access to IAM Identity Center resources such as the AWS access portal. Users can get AWS account applications and roles assigned to them and get federated into the application.

Although AWS Single Sign-On was renamed, the `sso` and `identitystore` API namespaces will continue to retain their original name for backward compatibility purposes. For more information, see IAM Identity Center rename.

This reference guide describes the IAM Identity Center Portal operations that you can call programatically and includes detailed information on data types and errors.

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms, such as Java, Ruby, .Net, iOS, or Android. The SDKs provide a convenient way to create programmatic access to IAM Identity Center and other AWS services. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

## Usage

```
sso(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 <- sso(
  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

| | |
|---|---|
| get_role_credentials | Returns the STS short-term credentials for a given role name that is assigned to the user |
| list_account_roles | Lists all roles that are assigned to the user for a given AWS account |
| list_accounts | Lists all AWS accounts assigned to the user |
| logout | Removes the locally stored SSO tokens from the client-side cache and sends an API call to the IAM Ide |

## Examples

```
## Not run:
svc <- sso()
svc$get_role_credentials(
  Foo = 123
)

## End(Not run)
```

---

`ssoadmin` *AWS Single Sign-On Admin*

---

## Description

IAM Identity Center (successor to Single Sign-On) helps you securely create, or connect, your workforce identities and manage their access centrally across Amazon Web Services accounts and applications. IAM Identity Center is the recommended approach for workforce authentication and authorization in Amazon Web Services, for organizations of any size and type.

IAM Identity Center uses the `sso` and `identitystore` API namespaces.

This reference guide provides information on single sign-on operations which could be used for access management of Amazon Web Services accounts. For information about IAM Identity Center features, see the IAM Identity Center User Guide.

Many operations in the IAM Identity Center APIs rely on identifiers for users and groups, known as principals. For more information about how to work with principals and principal IDs in IAM Identity Center, see the Identity Store API Reference.

Amazon Web Services provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, and more). The SDKs provide a convenient way to create programmatic access to IAM Identity Center and other Amazon Web Services services. For more information about the Amazon Web Services SDKs, including how to download and install them, see Tools for Amazon Web Services.

## Usage

```
ssoadmin(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 <- ssoadmin(
  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

| | |
|---|---|
| attach_customer_managed_policy_reference_to_permission_set | Attaches the specified customer managed policy to the s |
| attach_managed_policy_to_permission_set | Attaches an Amazon Web Services managed policy AR |
| create_account_assignment | Assigns access to a principal for a specified Amazon We |
| create_application | Creates an application in IAM Identity Center for the gi |
| create_application_assignment | Grant application access to a user or group |
| create_instance | Creates an instance of IAM Identity Center for a standal |
| create_instance_access_control_attribute_configuration | Enables the attributes-based access control (ABAC) feat |
| create_permission_set | Creates a permission set within a specified IAM Identity |
| create_trusted_token_issuer | Creates a connection to a trusted token issuer in an insta |
| delete_account_assignment | Deletes a principal's access from a specified Amazon W |
| delete_application | Deletes the association with the application |
| delete_application_access_scope | Deletes an IAM Identity Center access scope from an ap |
| delete_application_assignment | Revoke application access to an application by deleting |
| delete_application_authentication_method | Deletes an authentication method from an application |
| delete_application_grant | Deletes a grant from an application |
| delete_inline_policy_from_permission_set | Deletes the inline policy from a specified permission set |
| delete_instance | Deletes the instance of IAM Identity Center |
| delete_instance_access_control_attribute_configuration | Disables the attributes-based access control (ABAC) fea |
| delete_permissions_boundary_from_permission_set | Deletes the permissions boundary from a specified Perm |
| delete_permission_set | Deletes the specified permission set |
| delete_trusted_token_issuer | Deletes a trusted token issuer configuration from an inst |
| describe_account_assignment_creation_status | Describes the status of the assignment creation request |
| describe_account_assignment_deletion_status | Describes the status of the assignment deletion request |
| describe_application | Retrieves the details of an application associated with a |
| describe_application_assignment | Retrieves a direct assignment of a user or group to an ap |
| describe_application_provider | Retrieves details about a provider that can be used to co |
| describe_instance | Returns the details of an instance of IAM Identity Cente |
| describe_instance_access_control_attribute_configuration | Returns the list of IAM Identity Center identity store att |
| describe_permission_set | Gets the details of the permission set |

| | |
|---|---|
| describe_permission_set_provisioning_status | Describes the status for the given permission set provisi |
| describe_trusted_token_issuer | Retrieves details about a trusted token issuer configurati |
| detach_customer_managed_policy_reference_from_permission_set | Detaches the specified customer managed policy from th |
| detach_managed_policy_from_permission_set | Detaches the attached Amazon Web Services managed p |
| get_application_access_scope | Retrieves the authorized targets for an IAM Identity Cen |
| get_application_assignment_configuration | Retrieves the configuration of PutApplicationAssignmen |
| get_application_authentication_method | Retrieves details about an authentication method used b |
| get_application_grant | Retrieves details about an application grant |
| get_inline_policy_for_permission_set | Obtains the inline policy assigned to the permission set |
| get_permissions_boundary_for_permission_set | Obtains the permissions boundary for a specified Permis |
| list_account_assignment_creation_status | Lists the status of the Amazon Web Services account as |
| list_account_assignment_deletion_status | Lists the status of the Amazon Web Services account as |
| list_account_assignments | Lists the assignee of the specified Amazon Web Service |
| list_account_assignments_for_principal | Retrieves a list of the IAM Identity Center associated A |
| list_accounts_for_provisioned_permission_set | Lists all the Amazon Web Services accounts where the s |
| list_application_access_scopes | Lists the access scopes and authorized targets associated |
| list_application_assignments | Lists Amazon Web Services account users that are assig |
| list_application_assignments_for_principal | Lists the applications to which a specified principal is as |
| list_application_authentication_methods | Lists all of the authentication methods supported by the |
| list_application_grants | List the grants associated with an application |
| list_application_providers | Lists the application providers configured in the IAM Id |
| list_applications | Lists all applications associated with the instance of IAM |
| list_customer_managed_policy_references_in_permission_set | Lists all customer managed policies attached to a specif |
| list_instances | Lists the details of the organization and account instanc |
| list_managed_policies_in_permission_set | Lists the Amazon Web Services managed policy that is |
| list_permission_set_provisioning_status | Lists the status of the permission set provisioning reque |
| list_permission_sets | Lists the PermissionSets in an IAM Identity Center inst |
| list_permission_sets_provisioned_to_account | Lists all the permission sets that are provisioned to a spe |
| list_tags_for_resource | Lists the tags that are attached to a specified resource |
| list_trusted_token_issuers | Lists all the trusted token issuers configured in an instan |
| provision_permission_set | The process by which a specified permission set is prov |
| put_application_access_scope | Adds or updates the list of authorized targets for an IAM |
| put_application_assignment_configuration | Configure how users gain access to an application |
| put_application_authentication_method | Adds or updates an authentication method for an applica |
| put_application_grant | Adds a grant to an application |
| put_inline_policy_to_permission_set | Attaches an inline policy to a permission set |
| put_permissions_boundary_to_permission_set | Attaches an Amazon Web Services managed or custome |
| tag_resource | Associates a set of tags with a specified resource |
| untag_resource | Disassociates a set of tags from a specified resource |
| update_application | Updates application properties |
| update_instance | Update the details for the instance of IAM Identity Cent |
| update_instance_access_control_attribute_configuration | Updates the IAM Identity Center identity store attribute |
| update_permission_set | Updates an existing permission set |
| update_trusted_token_issuer | Updates the name of the trusted token issuer, or the path |

## Examples

```
## Not run:
svc <- ssoadmin()
svc$attach_customer_managed_policy_reference_to_permission_set(
  Foo = 123
)

## End(Not run)
```

---

ssooidc *AWS SSO OIDC*

---

## Description

IAM Identity Center OpenID Connect (OIDC) is a web service that enables a client (such as CLI or a native application) to register with IAM Identity Center. The service also enables the client to fetch the user's access token upon successful authentication and authorization with IAM Identity Center.

### API namespaces

IAM Identity Center uses the `sso` and `identitystore` API namespaces. IAM Identity Center OpenID Connect uses the `sso-oidc` namespace.

### Considerations for using this guide

Before you begin using this guide, we recommend that you first review the following important information about how the IAM Identity Center OIDC service works.

- The IAM Identity Center OIDC service currently implements only the portions of the OAuth 2.0 Device Authorization Grant standard (https://tools.ietf.org/html/rfc8628) that are necessary to enable single sign-on authentication with the CLI.
- With older versions of the CLI, the service only emits OIDC access tokens, so to obtain a new token, users must explicitly re-authenticate. To access the OIDC flow that supports token refresh and doesn't require re-authentication, update to the latest CLI version (1.27.10 for CLI V1 and 2.9.0 for CLI V2) with support for OIDC token refresh and configurable IAM Identity Center session durations. For more information, see Configure Amazon Web Services access portal session duration .
- The access tokens provided by this service grant access to all Amazon Web Services account entitlements assigned to an IAM Identity Center user, not just a particular application.
- The documentation in this guide does not describe the mechanism to convert the access token into Amazon Web Services Auth ("sigv4") credentials for use with IAM-protected Amazon Web Services service endpoints. For more information, see GetRoleCredentials in the *IAM Identity Center Portal API Reference Guide*.

For general information about IAM Identity Center, see What is IAM Identity Center? in the *IAM Identity Center User Guide*.

## Usage

```
ssooidc(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 <- ssooidc(
  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_token | Creates and returns access and refresh tokens for clients that are authenticated using client secrets |
| create_token_with_iam | Creates and returns access and refresh tokens for clients and applications that are authenticated u |
| register_client | Registers a public client with IAM Identity Center |
| start_device_authorization | Initiates device authorization by requesting a pair of verification codes from the authorization ser |

## Examples

```
## Not run:
svc <- ssooidc()
svc$create_token(
  Foo = 123
)
```

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

---

`sts` *AWS Security Token Service*

---

## Description

Security Token Service

Security Token Service (STS) enables you to request temporary, limited-privilege credentials for users. This guide provides descriptions of the STS API. For more information about using this service, see Temporary Security Credentials.

## Usage

```
sts(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 <- sts(
  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

| | |
|---|---|
| assume_role | Returns a set of temporary security credentials that you can use to access Amazon Web Ser |
| assume_role_with_saml | Returns a set of temporary security credentials for users who have been authenticated via a |
| assume_role_with_web_identity | Returns a set of temporary security credentials for users who have been authenticated in a r |
| assume_root | Returns a set of short term credentials you can use to perform privileged tasks on a membe |
| decode_authorization_message | Decodes additional information about the authorization status of a request from an encodec |
| get_access_key_info | Returns the account identifier for the specified access key ID |
| get_caller_identity | Returns details about the IAM user or role whose credentials are used to call the operation |
| get_federation_token | Returns a set of temporary security credentials (consisting of an access key ID, a secret acc |
| get_session_token | Returns a set of temporary credentials for an Amazon Web Services account or IAM user |

## Examples

```
## Not run:
svc <- sts()
#
svc$assume_role(
  ExternalId = "123ABC",
  Policy = "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"Stmt1\",\"Effect\":\"A...",
  RoleArn = "arn:aws:iam::123456789012:role/demo",
  RoleSessionName = "testAssumeRoleSession",
  Tags = list(
    list(
      Key = "Project",
      Value = "Unicorn"
    ),
    list(
      Key = "Team",
      Value = "Automation"
    ),
    list(
      Key = "Cost-Center",
      Value = "12345"
    )
  ),
  TransitiveTagKeys = list(
    "Project",
    "Cost-Center"
  )
)

## End(Not run)
```

---

# verifiedpermissions *Amazon Verified Permissions*

---

## Description

Amazon Verified Permissions is a permissions management service from Amazon Web Services. You can use Verified Permissions to manage permissions for your application, and authorize user access based on those permissions. Using Verified Permissions, application developers can grant access based on information about the users, resources, and requested actions. You can also evaluate additional information like group membership, attributes of the resources, and session context, such as time of request and IP addresses. Verified Permissions manages these permissions by letting you create and store authorization policies for your applications, such as consumer-facing web sites and enterprise business systems.

Verified Permissions uses Cedar as the policy language to express your permission requirements. Cedar supports both role-based access control (RBAC) and attribute-based access control (ABAC) authorization models.

For more information about configuring, administering, and using Amazon Verified Permissions in your applications, see the Amazon Verified Permissions User Guide.

For more information about the Cedar policy language, see the Cedar Policy Language Guide.

When you write Cedar policies that reference principals, resources and actions, you can define the unique identifiers used for each of those elements. We strongly recommend that you follow these best practices:

- **Use values like universally unique identifiers (UUIDs) for all principal and resource identifiers.**

  For example, if user `jane` leaves the company, and you later let someone else use the name `jane`, then that new user automatically gets access to everything granted by policies that still reference `User::"jane"`. Cedar can't distinguish between the new user and the old. This applies to both principal and resource identifiers. Always use identifiers that are guaranteed unique and never reused to ensure that you don't unintentionally grant access because of the presence of an old identifier in a policy.

  Where you use a UUID for an entity, we recommend that you follow it with the // comment specifier and the 'friendly' name of your entity. This helps to make your policies easier to understand. For example: principal == User::"a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111", // alice

- **Do not include personally identifying, confidential, or sensitive information as part of the unique identifier for your principals or resources.** These identifiers are included in log entries shared in CloudTrail trails.

Several operations return structures that appear similar, but have different purposes. As new functionality is added to the product, the structure used in a parameter of one operation might need to change in a way that wouldn't make sense for the same parameter in a different operation. To help you understand the purpose of each, the following naming convention is used for the structures:

- Parameter type structures that end in `Detail` are used in `Get` operations.
- Parameter type structures that end in `Item` are used in `List` operations.
- Parameter type structures that use neither suffix are used in the mutating (create and update) operations.

## Usage

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

### Operations

| | |
|---|---|
| batch_get_policy | Retrieves information about a group (batch) of policies |
| batch_is_authorized | Makes a series of decisions about multiple authorization requests for one principal or resou |
| batch_is_authorized_with_token | Makes a series of decisions about multiple authorization requests for one token |
| create_identity_source | Adds an identity source to a policy store–an Amazon Cognito user pool or OpenID Connec |
| create_policy | Creates a Cedar policy and saves it in the specified policy store |
| create_policy_store | Creates a policy store |
| create_policy_template | Creates a policy template |
| delete_identity_source | Deletes an identity source that references an identity provider (IdP) such as Amazon Cogni |
| delete_policy | Deletes the specified policy from the policy store |
| delete_policy_store | Deletes the specified policy store |
| delete_policy_template | Deletes the specified policy template from the policy store |
| get_identity_source | Retrieves the details about the specified identity source |
| get_policy | Retrieves information about the specified policy |

| | |
|---|---|
| get_policy_store | Retrieves details about a policy store |
| get_policy_template | Retrieve the details for the specified policy template in the specified policy store |
| get_schema | Retrieve the details for the specified schema in the specified policy store |
| is_authorized | Makes an authorization decision about a service request described in the parameters |
| is_authorized_with_token | Makes an authorization decision about a service request described in the parameters |
| list_identity_sources | Returns a paginated list of all of the identity sources defined in the specified policy store |
| list_policies | Returns a paginated list of all policies stored in the specified policy store |
| list_policy_stores | Returns a paginated list of all policy stores in the calling Amazon Web Services account |
| list_policy_templates | Returns a paginated list of all policy templates in the specified policy store |
| put_schema | Creates or updates the policy schema in the specified policy store |
| update_identity_source | Updates the specified identity source to use a new identity provider (IdP), or to change the |
| update_policy | Modifies a Cedar static policy in the specified policy store |
| update_policy_store | Modifies the validation setting for a policy store |
| update_policy_template | Updates the specified policy template |

## Examples

```
## Not run:
svc <- verifiedpermissions()
svc$batch_get_policy(
  Foo = 123
)

## End(Not run)
```

---

`waf` *AWS WAF*

---

## Description

This is **AWS WAF Classic** documentation. For more information, see AWS WAF Classic in the developer guide.

**For the latest version of AWS WAF**, use the AWS WAFV2 API and see the AWS WAF Developer Guide. With the latest version, AWS WAF has a single set of endpoints for regional and global use.

This is the *AWS WAF Classic API Reference* for using AWS WAF Classic with Amazon CloudFront. The AWS WAF Classic actions and data types listed in the reference are available for protecting Amazon CloudFront distributions. You can use these actions and data types via the endpoint *waf.amazonaws.com*. This guide is for developers who need detailed information about the AWS WAF Classic API actions, data types, and errors. For detailed information about AWS WAF Classic features and an overview of how to use the AWS WAF Classic API, see the AWS WAF Classic in the developer guide.

## Usage

```
waf(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 <- waf(
  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_byte_match_set | This is AWS WAF Classic documentation |
| create_geo_match_set | This is AWS WAF Classic documentation |
| create_ip_set | This is AWS WAF Classic documentation |
| create_rate_based_rule | This is AWS WAF Classic documentation |
| create_regex_match_set | This is AWS WAF Classic documentation |
| create_regex_pattern_set | This is AWS WAF Classic documentation |
| create_rule | This is AWS WAF Classic documentation |
| create_rule_group | This is AWS WAF Classic documentation |
| create_size_constraint_set | This is AWS WAF Classic documentation |
| create_sql_injection_match_set | This is AWS WAF Classic documentation |
| create_web_acl | This is AWS WAF Classic documentation |
| create_web_acl_migration_stack | Creates an AWS CloudFormation WAFV2 template for the specified web ACL in the sp |
| create_xss_match_set | This is AWS WAF Classic documentation |
| delete_byte_match_set | This is AWS WAF Classic documentation |
| delete_geo_match_set | This is AWS WAF Classic documentation |
| delete_ip_set | This is AWS WAF Classic documentation |
| delete_logging_configuration | This is AWS WAF Classic documentation |
| delete_permission_policy | This is AWS WAF Classic documentation |
| delete_rate_based_rule | This is AWS WAF Classic documentation |
| delete_regex_match_set | This is AWS WAF Classic documentation |

| | |
|---|---|
| delete_regex_pattern_set | This is AWS WAF Classic documentation |
| delete_rule | This is AWS WAF Classic documentation |
| delete_rule_group | This is AWS WAF Classic documentation |
| delete_size_constraint_set | This is AWS WAF Classic documentation |
| delete_sql_injection_match_set | This is AWS WAF Classic documentation |
| delete_web_acl | This is AWS WAF Classic documentation |
| delete_xss_match_set | This is AWS WAF Classic documentation |
| get_byte_match_set | This is AWS WAF Classic documentation |
| get_change_token | This is AWS WAF Classic documentation |
| get_change_token_status | This is AWS WAF Classic documentation |
| get_geo_match_set | This is AWS WAF Classic documentation |
| get_ip_set | This is AWS WAF Classic documentation |
| get_logging_configuration | This is AWS WAF Classic documentation |
| get_permission_policy | This is AWS WAF Classic documentation |
| get_rate_based_rule | This is AWS WAF Classic documentation |
| get_rate_based_rule_managed_keys | This is AWS WAF Classic documentation |
| get_regex_match_set | This is AWS WAF Classic documentation |
| get_regex_pattern_set | This is AWS WAF Classic documentation |
| get_rule | This is AWS WAF Classic documentation |
| get_rule_group | This is AWS WAF Classic documentation |
| get_sampled_requests | This is AWS WAF Classic documentation |
| get_size_constraint_set | This is AWS WAF Classic documentation |
| get_sql_injection_match_set | This is AWS WAF Classic documentation |
| get_web_acl | This is AWS WAF Classic documentation |
| get_xss_match_set | This is AWS WAF Classic documentation |
| list_activated_rules_in_rule_group | This is AWS WAF Classic documentation |
| list_byte_match_sets | This is AWS WAF Classic documentation |
| list_geo_match_sets | This is AWS WAF Classic documentation |
| list_ip_sets | This is AWS WAF Classic documentation |
| list_logging_configurations | This is AWS WAF Classic documentation |
| list_rate_based_rules | This is AWS WAF Classic documentation |
| list_regex_match_sets | This is AWS WAF Classic documentation |
| list_regex_pattern_sets | This is AWS WAF Classic documentation |
| list_rule_groups | This is AWS WAF Classic documentation |
| list_rules | This is AWS WAF Classic documentation |
| list_size_constraint_sets | This is AWS WAF Classic documentation |
| list_sql_injection_match_sets | This is AWS WAF Classic documentation |
| list_subscribed_rule_groups | This is AWS WAF Classic documentation |
| list_tags_for_resource | This is AWS WAF Classic documentation |
| list_web_ac_ls | This is AWS WAF Classic documentation |
| list_xss_match_sets | This is AWS WAF Classic documentation |
| put_logging_configuration | This is AWS WAF Classic documentation |
| put_permission_policy | This is AWS WAF Classic documentation |
| tag_resource | This is AWS WAF Classic documentation |
| untag_resource | This is AWS WAF Classic documentation |
| update_byte_match_set | This is AWS WAF Classic documentation |
| update_geo_match_set | This is AWS WAF Classic documentation |
| update_ip_set | This is AWS WAF Classic documentation |

| | |
|---|---|
| update_rate_based_rule | This is AWS WAF Classic documentation |
| update_regex_match_set | This is AWS WAF Classic documentation |
| update_regex_pattern_set | This is AWS WAF Classic documentation |
| update_rule | This is AWS WAF Classic documentation |
| update_rule_group | This is AWS WAF Classic documentation |
| update_size_constraint_set | This is AWS WAF Classic documentation |
| update_sql_injection_match_set | This is AWS WAF Classic documentation |
| update_web_acl | This is AWS WAF Classic documentation |
| update_xss_match_set | This is AWS WAF Classic documentation |

## Examples

```
## Not run:
svc <- waf()
# The following example creates an IP match set named MyIPSetFriendlyName.
svc$create_ip_set(
  ChangeToken = "abcd12f2-46da-4fdb-b8d5-fbd4c466928f",
  Name = "MyIPSetFriendlyName"
)

## End(Not run)
```

---

`wafregional` *AWS WAF Regional*

---

## Description

This is **AWS WAF Classic Regional** documentation. For more information, see AWS WAF Classic in the developer guide.

**For the latest version of AWS WAF**, use the AWS WAFV2 API and see the AWS WAF Developer Guide. With the latest version, AWS WAF has a single set of endpoints for regional and global use.

This is the *AWS WAF Regional Classic API Reference* for using AWS WAF Classic with the AWS resources, Elastic Load Balancing (ELB) Application Load Balancers and API Gateway APIs. The AWS WAF Classic actions and data types listed in the reference are available for protecting Elastic Load Balancing (ELB) Application Load Balancers and API Gateway APIs. You can use these actions and data types by means of the endpoints listed in AWS Regions and Endpoints. This guide is for developers who need detailed information about the AWS WAF Classic API actions, data types, and errors. For detailed information about AWS WAF Classic features and an overview of how to use the AWS WAF Classic API, see the AWS WAF Classic in the developer guide.

## Usage

```
wafregional(
  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 <- wafregional(
  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_web_acl | This is AWS WAF Classic Regional documentation |
| create_byte_match_set | This is AWS WAF Classic documentation |
| create_geo_match_set | This is AWS WAF Classic documentation |
| create_ip_set | This is AWS WAF Classic documentation |
| create_rate_based_rule | This is AWS WAF Classic documentation |
| create_regex_match_set | This is AWS WAF Classic documentation |
| create_regex_pattern_set | This is AWS WAF Classic documentation |
| create_rule | This is AWS WAF Classic documentation |
| create_rule_group | This is AWS WAF Classic documentation |
| create_size_constraint_set | This is AWS WAF Classic documentation |
| create_sql_injection_match_set | This is AWS WAF Classic documentation |
| create_web_acl | This is AWS WAF Classic documentation |
| create_web_acl_migration_stack | Creates an AWS CloudFormation WAFV2 template for the specified web ACL in the sp |

| | |
|---|---|
| create_xss_match_set | This is AWS WAF Classic documentation |
| delete_byte_match_set | This is AWS WAF Classic documentation |
| delete_geo_match_set | This is AWS WAF Classic documentation |
| delete_ip_set | This is AWS WAF Classic documentation |
| delete_logging_configuration | This is AWS WAF Classic documentation |
| delete_permission_policy | This is AWS WAF Classic documentation |
| delete_rate_based_rule | This is AWS WAF Classic documentation |
| delete_regex_match_set | This is AWS WAF Classic documentation |
| delete_regex_pattern_set | This is AWS WAF Classic documentation |
| delete_rule | This is AWS WAF Classic documentation |
| delete_rule_group | This is AWS WAF Classic documentation |
| delete_size_constraint_set | This is AWS WAF Classic documentation |
| delete_sql_injection_match_set | This is AWS WAF Classic documentation |
| delete_web_acl | This is AWS WAF Classic documentation |
| delete_xss_match_set | This is AWS WAF Classic documentation |
| disassociate_web_acl | This is AWS WAF Classic Regional documentation |
| get_byte_match_set | This is AWS WAF Classic documentation |
| get_change_token | This is AWS WAF Classic documentation |
| get_change_token_status | This is AWS WAF Classic documentation |
| get_geo_match_set | This is AWS WAF Classic documentation |
| get_ip_set | This is AWS WAF Classic documentation |
| get_logging_configuration | This is AWS WAF Classic documentation |
| get_permission_policy | This is AWS WAF Classic documentation |
| get_rate_based_rule | This is AWS WAF Classic documentation |
| get_rate_based_rule_managed_keys | This is AWS WAF Classic documentation |
| get_regex_match_set | This is AWS WAF Classic documentation |
| get_regex_pattern_set | This is AWS WAF Classic documentation |
| get_rule | This is AWS WAF Classic documentation |
| get_rule_group | This is AWS WAF Classic documentation |
| get_sampled_requests | This is AWS WAF Classic documentation |
| get_size_constraint_set | This is AWS WAF Classic documentation |
| get_sql_injection_match_set | This is AWS WAF Classic documentation |
| get_web_acl | This is AWS WAF Classic documentation |
| get_web_acl_for_resource | This is AWS WAF Classic Regional documentation |
| get_xss_match_set | This is AWS WAF Classic documentation |
| list_activated_rules_in_rule_group | This is AWS WAF Classic documentation |
| list_byte_match_sets | This is AWS WAF Classic documentation |
| list_geo_match_sets | This is AWS WAF Classic documentation |
| list_ip_sets | This is AWS WAF Classic documentation |
| list_logging_configurations | This is AWS WAF Classic documentation |
| list_rate_based_rules | This is AWS WAF Classic documentation |
| list_regex_match_sets | This is AWS WAF Classic documentation |
| list_regex_pattern_sets | This is AWS WAF Classic documentation |
| list_resources_for_web_acl | This is AWS WAF Classic Regional documentation |
| list_rule_groups | This is AWS WAF Classic documentation |
| list_rules | This is AWS WAF Classic documentation |
| list_size_constraint_sets | This is AWS WAF Classic documentation |
| list_sql_injection_match_sets | This is AWS WAF Classic documentation |

| | |
|---|---|
| list_subscribed_rule_groups | This is AWS WAF Classic documentation |
| list_tags_for_resource | This is AWS WAF Classic documentation |
| list_web_ac_ls | This is AWS WAF Classic documentation |
| list_xss_match_sets | This is AWS WAF Classic documentation |
| put_logging_configuration | This is AWS WAF Classic documentation |
| put_permission_policy | This is AWS WAF Classic documentation |
| tag_resource | This is AWS WAF Classic documentation |
| untag_resource | This is AWS WAF Classic documentation |
| update_byte_match_set | This is AWS WAF Classic documentation |
| update_geo_match_set | This is AWS WAF Classic documentation |
| update_ip_set | This is AWS WAF Classic documentation |
| update_rate_based_rule | This is AWS WAF Classic documentation |
| update_regex_match_set | This is AWS WAF Classic documentation |
| update_regex_pattern_set | This is AWS WAF Classic documentation |
| update_rule | This is AWS WAF Classic documentation |
| update_rule_group | This is AWS WAF Classic documentation |
| update_size_constraint_set | This is AWS WAF Classic documentation |
| update_sql_injection_match_set | This is AWS WAF Classic documentation |
| update_web_acl | This is AWS WAF Classic documentation |
| update_xss_match_set | This is AWS WAF Classic documentation |

## Examples

```
## Not run:
svc <- wafregional()
# The following example creates an IP match set named MyIPSetFriendlyName.
svc$create_ip_set(
  ChangeToken = "abcd12f2-46da-4fdb-b8d5-fbd4c466928f",
  Name = "MyIPSetFriendlyName"
)

## End(Not run)
```

---

# wafv2 *AWS WAFV2*

---

## Description

WAF

This is the latest version of the **WAF** API, released in November, 2019. The names of the entities that you use to access this API, like endpoints and namespaces, all have the versioning information added, like "V2" or "v2", to distinguish from the prior version. We recommend migrating your resources to this version, because it has a number of significant improvements.

If you used WAF prior to this release, you can't use this WAFV2 API to access any WAF resources that you created before. WAF Classic support will end on September 30, 2025.

For information about WAF, including how to migrate your WAF Classic resources to this version, see the WAF Developer Guide.

WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to a protected resource. Protected resource types include Amazon CloudFront distribution, Amazon API Gateway REST API, Application Load Balancer, AppSync GraphQL API, Amazon Cognito user pool, App Runner service, and Amazon Web Services Verified Access instance. WAF also lets you control access to your content, to protect the Amazon Web Services resource that WAF is monitoring. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, the protected resource responds to requests with either the requested content, an HTTP 403 status code (Forbidden), or with a custom response.

This API guide is for developers who need detailed information about WAF API actions, data types, and errors. For detailed information about WAF features and guidance for configuring and using WAF, see the WAF Developer Guide.

You can make calls using the endpoints listed in WAF endpoints and quotas.

- For regional resources, you can use any of the endpoints in the list. A regional application can be an Application Load Balancer (ALB), an Amazon API Gateway REST API, an AppSync GraphQL API, an Amazon Cognito user pool, an App Runner service, or an Amazon Web Services Verified Access instance.
- For Amazon CloudFront, you must use the API endpoint listed for US East (N. Virginia): us-east-1.

Alternatively, you can use one of the Amazon Web Services SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see Amazon Web Services SDKs.

## Usage

```
wafv2(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 <- wafv2(
  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_web_acl | Associates a web ACL with a resource, to protect the resource |
| check_capacity | Returns the web ACL capacity unit (WCU) requirements for a specified sco |
| create_api_key | Creates an API key that contains a set of token domains |
| create_ip_set | Creates an IPSet, which you use to identify web requests that originate from |
| create_regex_pattern_set | Creates a RegexPatternSet, which you reference in a RegexPatternSetRefere |
| create_rule_group | Creates a RuleGroup per the specifications provided |
| create_web_acl | Creates a WebACL per the specifications provided |
| delete_api_key | Deletes the specified API key |
| delete_firewall_manager_rule_groups | Deletes all rule groups that are managed by Firewall Manager from the spec |
| delete_ip_set | Deletes the specified IPSet |
| delete_logging_configuration | Deletes the LoggingConfiguration from the specified web ACL |
| delete_permission_policy | Permanently deletes an IAM policy from the specified rule group |
| delete_regex_pattern_set | Deletes the specified RegexPatternSet |
| delete_rule_group | Deletes the specified RuleGroup |
| delete_web_acl | Deletes the specified WebACL |
| describe_all_managed_products | Provides high-level information for the Amazon Web Services Managed Ru |
| describe_managed_products_by_vendor | Provides high-level information for the managed rule groups owned by a sp |
| describe_managed_rule_group | Provides high-level information for a managed rule group, including descrip |
| disassociate_web_acl | Disassociates the specified resource from its web ACL association, if it has |
| generate_mobile_sdk_release_url | Generates a presigned download URL for the specified release of the mobil |
| get_decrypted_api_key | Returns your API key in decrypted form |
| get_ip_set | Retrieves the specified IPSet |
| get_logging_configuration | Returns the LoggingConfiguration for the specified web ACL |
| get_managed_rule_set | Retrieves the specified managed rule set |
| get_mobile_sdk_release | Retrieves information for the specified mobile SDK release, including relea |
| get_permission_policy | Returns the IAM policy that is attached to the specified rule group |
| get_rate_based_statement_managed_keys | Retrieves the IP addresses that are currently blocked by a rate-based rule ins |
| get_regex_pattern_set | Retrieves the specified RegexPatternSet |
| get_rule_group | Retrieves the specified RuleGroup |
| get_sampled_requests | Gets detailed information about a specified number of requests–a sample–th |
| get_web_acl | Retrieves the specified WebACL |
| get_web_acl_for_resource | Retrieves the WebACL for the specified resource |
| list_api_keys | Retrieves a list of the API keys that you've defined for the specified scope |
| list_available_managed_rule_groups | Retrieves an array of managed rule groups that are available for you to use |
| list_available_managed_rule_group_versions | Returns a list of the available versions for the specified managed rule group |
| list_ip_sets | Retrieves an array of IPSetSummary objects for the IP sets that you manage |

| | |
|---|---|
| list_logging_configurations | Retrieves an array of your LoggingConfiguration objects |
| list_managed_rule_sets | Retrieves the managed rule sets that you own |
| list_mobile_sdk_releases | Retrieves a list of the available releases for the mobile SDK and the specifie |
| list_regex_pattern_sets | Retrieves an array of RegexPatternSetSummary objects for the regex patter |
| list_resources_for_web_acl | Retrieves an array of the Amazon Resource Names (ARNs) for the resource |
| list_rule_groups | Retrieves an array of RuleGroupSummary objects for the rule groups that y |
| list_tags_for_resource | Retrieves the TagInfoForResource for the specified resource |
| list_web_ac_ls | Retrieves an array of WebACLSummary objects for the web ACLs that you |
| put_logging_configuration | Enables the specified LoggingConfiguration, to start logging from a web AC |
| put_managed_rule_set_versions | Defines the versions of your managed rule set that you are offering to the cu |
| put_permission_policy | Use this to share a rule group with other accounts |
| tag_resource | Associates tags with the specified Amazon Web Services resource |
| untag_resource | Disassociates tags from an Amazon Web Services resource |
| update_ip_set | Updates the specified IPSet |
| update_managed_rule_set_version_expiry_date | Updates the expiration information for your managed rule set |
| update_regex_pattern_set | Updates the specified RegexPatternSet |
| update_rule_group | Updates the specified RuleGroup |
| update_web_acl | Updates the specified WebACL |

## Examples

```
## Not run:
svc <- wafv2()
svc$associate_web_acl(
  Foo = 123
)

## End(Not run)
```

# Index