Package 'paws.developer.tools'

March 14, 2025

Title 'Amazon Web Services' Developer Tools Services

Version 0.9.0

Description Interface to 'Amazon Web Services' developer tools services, including version control, continuous integration and deployment, and more <https://aws.amazon.com/products/developer-tools/>.

License Apache License (>= 2.0)

URL https://github.com/paws-r/paws, https://paws-r.r-universe.dev/paws.developer.tools

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

Imports paws.common (>= 0.8.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.2

Collate 'cloud9_service.R' 'cloud9_interfaces.R' 'cloud9_operations.R' 'cloudcontrolapi_service.R' 'cloudcontrolapi_interfaces.R' 'cloudcontrolapi_operations.R' 'codeartifact_service.R' 'codeartifact_interfaces.R' 'codeartifact_operations.R' 'codebuild_service.R' 'codebuild_interfaces.R' 'codebuild_operations.R' 'codecatalyst_service.R' 'codecatalyst_interfaces.R' 'codecatalyst_operations.R' 'codecommit_service.R' 'codecommit_interfaces.R' 'codecommit_operations.R' 'codeconnections_service.R' 'codeconnections_interfaces.R' 'codeconnections_operations.R' 'codedeploy_service.R' 'codedeploy_interfaces.R' 'codedeploy_operations.R' 'codeguruprofiler_service.R' 'codeguruprofiler_interfaces.R' 'codeguruprofiler_operations.R' 'codegurureviewer_service.R' 'codegurureviewer_interfaces.R' 'codegurureviewer_operations.R' 'codegurusecurity_service.R' 'codegurusecurity_interfaces.R' 'codegurusecurity_operations.R' 'codepipeline_service.R' 'codepipeline_interfaces.R' 'codepipeline_operations.R' 'codestarconnections_service.R' 'codestarconnections_interfaces.R'

'codestarconnections_operations.R'
'codestarnotifications_service.R'
'codestarnotifications_interfaces.R'
'codestarnotifications_operations.R' 'devopsguru_service.R'
'devopsguru_interfaces.R' 'devopsguru_operations.R'
'drs_service.R' 'drs_interfaces.R' 'drs_operations.R'
'fis_service.R' 'fis_interfaces.R' 'fis_operations.R'
'reexports_paws.common.R' 'wellarchitected_service.R'
'wellarchitected_interfaces.R' 'wellarchitected_operations.R'
'xray_service.R' 'xray_interfaces.R' 'xray_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 15:10:02 UTC

Contents

`cloud9` *AWS Cloud9*

Description

Cloud9

Cloud9 is a collection of tools that you can use to code, build, run, test, debug, and release software in the cloud.

For more information about Cloud9, see the Cloud9 User Guide.

Cloud9 is no longer available to new customers. Existing customers of Cloud9 can continue to use the service as normal. Learn more"

Cloud9 supports these operations:

- `create_environment_ec2`: Creates an Cloud9 development environment, launches an Amazon EC2 instance, and then connects from the instance to the environment.
- `create_environment_membership`: Adds an environment member to an environment.
- `delete_environment`: Deletes an environment. If an Amazon EC2 instance is connected to the environment, also terminates the instance.
- `delete_environment_membership`: Deletes an environment member from an environment.
- `describe_environment_memberships`: Gets information about environment members for an environment.
- `describe_environments`: Gets information about environments.
- `describe_environment_status`: Gets status information for an environment.
- `list_environments`: Gets a list of environment identifiers.
- `list_tags_for_resource`: Gets the tags for an environment.
- `tag_resource`: Adds tags to an environment.
- `untag_resource`: Removes tags from an environment.
- `update_environment`: Changes the settings of an existing environment.
- `update_environment_membership`: Changes the settings of an existing environment member for an environment.

Usage

```
cloud9(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 <- cloud9(
  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_environment_ec2	Creates an Cloud9 development environment, launches an Amazon Elastic Compute C
create_environment_membership	Adds an environment member to an Cloud9 development environment
delete_environment	Deletes an Cloud9 development environment
delete_environment_membership	Deletes an environment member from a development environment
describe_environment_memberships	Gets information about environment members for an Cloud9 development environmen
describe_environments	Gets information about Cloud9 development environments
describe_environment_status	Gets status information for an Cloud9 development environment
list_environments	Gets a list of Cloud9 development environment identifiers
list_tags_for_resource	Gets a list of the tags associated with an Cloud9 development environment
tag_resource	Adds tags to an Cloud9 development environment
untag_resource	Removes tags from an Cloud9 development environment
update_environment	Changes the settings of an existing Cloud9 development environment
update_environment_membership	Changes the settings of an existing environment member for an Cloud9 development e

Examples

```
## Not run:
svc <- cloud9()
#
svc$create_environment_ec2(
  name = "my-demo-environment",
  automaticStopTimeMinutes = 60L,
  description = "This is my demonstration environment.",
  instanceType = "t2.micro",
  ownerArn = "arn:aws:iam::123456789012:user/MyDemoUser",
  subnetId = "subnet-6300cd1b"
)
```

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

`cloudcontrolapi` *AWS Cloud Control API*

Description

For more information about Amazon Web Services Cloud Control API, see the Amazon Web Services Cloud Control API User Guide.

Usage

```
cloudcontrolapi(
  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 <- cloudcontrolapi(
  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_resource_request	Cancels the specified resource operation request
create_resource	Creates the specified resource
delete_resource	Deletes the specified resource
get_resource	Returns information about the current state of the specified resource
get_resource_request_status	Returns the current status of a resource operation request
list_resource_requests	Returns existing resource operation requests
list_resources	Returns information about the specified resources
update_resource	Updates the specified property values in the resource

Examples

```
## Not run:
svc <- cloudcontrolapi()
svc$cancel_resource_request(
  Foo = 123
)

## End(Not run)
```

`codeartifact` *CodeArtifact*

Description

CodeArtifact is a fully managed artifact repository compatible with language-native package managers and build tools such as npm, Apache Maven, pip, and dotnet. You can use CodeArtifact to share packages with development teams and pull packages. Packages can be pulled from both public and CodeArtifact repositories. You can also create an upstream relationship between a CodeArtifact repository and another repository, which effectively merges their contents from the point of view of a package manager client.

CodeArtifact concepts

- **Repository**: A CodeArtifact repository contains a set of package versions, each of which maps to a set of assets, or files. Repositories are polyglot, so a single repository can contain packages of any supported type. Each repository exposes endpoints for fetching and publishing packages using tools such as the `npm` CLI or the Maven CLI (`mvn`). For a list of supported package managers, see the CodeArtifact User Guide.
- **Domain**: Repositories are aggregated into a higher-level entity known as a *domain*. All package assets and metadata are stored in the domain, but are consumed through repositories. A given package asset, such as a Maven JAR file, is stored once per domain, no matter how many repositories it's present in. All of the assets and metadata in a domain are encrypted with the same customer master key (CMK) stored in Key Management Service (KMS).

Each repository is a member of a single domain and can't be moved to a different domain.

The domain allows organizational policy to be applied across multiple repositories, such as which accounts can access repositories in the domain, and which public repositories can be used as sources of packages.

Although an organization can have multiple domains, we recommend a single production domain that contains all published artifacts so that teams can find and share packages across their organization.

- **Package**: A *package* is a bundle of software and the metadata required to resolve dependencies and install the software. CodeArtifact supports npm, PyPI, Maven, NuGet, Swift, Ruby, Cargo, and generic package formats. For more information about the supported package formats and how to use CodeArtifact with them, see the CodeArtifact User Guide.

  In CodeArtifact, a package consists of:

  - A *name* (for example, `webpack` is the name of a popular npm package)
  - An optional namespace (for example, `@types` in `@types/node`)
  - A set of versions (for example, `1.0.0`, `1.0.1`, `1.0.2`, etc.)
  - Package-level metadata (for example, npm tags)

- **Package group**: A group of packages that match a specified definition. Package groups can be used to apply configuration to multiple packages that match a defined pattern using package format, package namespace, and package name. You can use package groups to more conveniently configure package origin controls for multiple packages. Package origin controls are used to block or allow ingestion or publishing of new package versions, which protects users from malicious actions known as dependency substitution attacks.

- **Package version**: A version of a package, such as `@types/node 12.6.9`. The version number format and semantics vary for different package formats. For example, npm package versions must conform to the Semantic Versioning specification. In CodeArtifact, a package version consists of the version identifier, metadata at the package version level, and a set of assets.

- **Upstream repository**: One repository is *upstream* of another when the package versions in it can be accessed from the repository endpoint of the downstream repository, effectively merging the contents of the two repositories from the point of view of a client. CodeArtifact allows creating an upstream relationship between two repositories.

- **Asset**: An individual file stored in CodeArtifact associated with a package version, such as an npm `.tgz` file or Maven POM and JAR files.

CodeArtifact supported API operations

- `associate_external_connection`: Adds an existing external connection to a repository.
- `copy_package_versions`: Copies package versions from one repository to another repository in the same domain.
- `create_domain`: Creates a domain.
- `create_package_group`: Creates a package group.
- `create_repository`: Creates a CodeArtifact repository in a domain.
- `delete_domain`: Deletes a domain. You cannot delete a domain that contains repositories.
- `delete_domain_permissions_policy`: Deletes the resource policy that is set on a domain.
- `delete_package`: Deletes a package and all associated package versions.

- delete_package_group: Deletes a package group. Does not delete packages or package versions that are associated with a package group.
- delete_package_versions: Deletes versions of a package. After a package has been deleted, it can be republished, but its assets and metadata cannot be restored because they have been permanently removed from storage.
- delete_repository: Deletes a repository.
- delete_repository_permissions_policy: Deletes the resource policy that is set on a repository.
- describe_domain: Returns a DomainDescription object that contains information about the requested domain.
- describe_package: Returns a PackageDescription object that contains details about a package.
- describe_package_group: Returns a PackageGroup object that contains details about a package group.
- describe_package_version: Returns a PackageVersionDescription object that contains details about a package version.
- describe_repository: Returns a RepositoryDescription object that contains detailed information about the requested repository.
- dispose_package_versions: Disposes versions of a package. A package version with the status Disposed cannot be restored because they have been permanently removed from storage.
- disassociate_external_connection: Removes an existing external connection from a repository.
- get_associated_package_group: Returns the most closely associated package group to the specified package.
- get_authorization_token: Generates a temporary authorization token for accessing repositories in the domain. The token expires the authorization period has passed. The default authorization period is 12 hours and can be customized to any length with a maximum of 12 hours.
- get_domain_permissions_policy: Returns the policy of a resource that is attached to the specified domain.
- get_package_version_asset: Returns the contents of an asset that is in a package version.
- get_package_version_readme: Gets the readme file or descriptive text for a package version.
- get_repository_endpoint: Returns the endpoint of a repository for a specific package format. A repository has one endpoint for each package format:
  - cargo
  - generic
  - maven
  - npm
  - nuget
  - pypi

  - ruby
  - swift
- get_repository_permissions_policy: Returns the resource policy that is set on a repository.
- list_allowed_repositories_for_group: Lists the allowed repositories for a package group that has origin configuration set to ALLOW_SPECIFIC_REPOSITORIES.
- list_associated_packages: Returns a list of packages associated with the requested package group.
- list_domains: Returns a list of DomainSummary objects. Each returned DomainSummary object contains information about a domain.
- list_packages: Lists the packages in a repository.
- list_package_groups: Returns a list of package groups in the requested domain.
- list_package_version_assets: Lists the assets for a given package version.
- list_package_version_dependencies: Returns a list of the direct dependencies for a package version.
- list_package_versions: Returns a list of package versions for a specified package in a repository.
- list_repositories: Returns a list of repositories owned by the Amazon Web Services account that called this method.
- list_repositories_in_domain: Returns a list of the repositories in a domain.
- list_sub_package_groups: Returns a list of direct children of the specified package group.
- publish_package_version: Creates a new package version containing one or more assets.
- put_domain_permissions_policy: Attaches a resource policy to a domain.
- put_package_origin_configuration: Sets the package origin configuration for a package, which determine how new versions of the package can be added to a specific repository.
- put_repository_permissions_policy: Sets the resource policy on a repository that specifies permissions to access it.
- update_package_group: Updates a package group. This API cannot be used to update a package group's origin configuration or pattern.
- update_package_group_origin_configuration: Updates the package origin configuration for a package group.
- update_package_versions_status: Updates the status of one or more versions of a package.
- update_repository: Updates the properties of a repository.

Usage

```
codeartifact(
  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 <- codeartifact(
  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_external_connection	Adds an existing external connection to a repository
copy_package_versions	Copies package versions from one repository to another repository in the same
create_domain	Creates a domain
create_package_group	Creates a package group
create_repository	Creates a repository
delete_domain	Deletes a domain
delete_domain_permissions_policy	Deletes the resource policy set on a domain
delete_package	Deletes a package and all associated package versions
delete_package_group	Deletes a package group
delete_package_versions	Deletes one or more versions of a package
delete_repository	Deletes a repository
delete_repository_permissions_policy	Deletes the resource policy that is set on a repository
describe_domain	Returns a DomainDescription object that contains information about the reque
describe_package	Returns a PackageDescription object that contains information about the reque
describe_package_group	Returns a PackageGroupDescription object that contains information about the
describe_package_version	Returns a PackageVersionDescription object that contains information about th
describe_repository	Returns a RepositoryDescription object that contains detailed information abou
disassociate_external_connection	Removes an existing external connection from a repository
dispose_package_versions	Deletes the assets in package versions and sets the package versions' status to
get_associated_package_group	Returns the most closely associated package group to the specified package

get_authorization_token	Generates a temporary authorization token for accessing repositories in the do
get_domain_permissions_policy	Returns the resource policy attached to the specified domain
get_package_version_asset	Returns an asset (or file) that is in a package
get_package_version_readme	Gets the readme file or descriptive text for a package version
get_repository_endpoint	Returns the endpoint of a repository for a specific package format
get_repository_permissions_policy	Returns the resource policy that is set on a repository
list_allowed_repositories_for_group	Lists the repositories in the added repositories list of the specified restriction ty
list_associated_packages	Returns a list of packages associated with the requested package group
list_domains	Returns a list of DomainSummary objects for all domains owned by the Amaz
list_package_groups	Returns a list of package groups in the requested domain
list_packages	Returns a list of PackageSummary objects for packages in a repository that ma
list_package_version_assets	Returns a list of AssetSummary objects for assets in a package version
list_package_version_dependencies	Returns the direct dependencies for a package version
list_package_versions	Returns a list of PackageVersionSummary objects for package versions in a re
list_repositories	Returns a list of RepositorySummary objects
list_repositories_in_domain	Returns a list of RepositorySummary objects
list_sub_package_groups	Returns a list of direct children of the specified package group
list_tags_for_resource	Gets information about Amazon Web Services tags for a specified Amazon Re
publish_package_version	Creates a new package version containing one or more assets (or files)
put_domain_permissions_policy	Sets a resource policy on a domain that specifies permissions to access it
put_package_origin_configuration	Sets the package origin configuration for a package
put_repository_permissions_policy	Sets the resource policy on a repository that specifies permissions to access it
tag_resource	Adds or updates tags for a resource in CodeArtifact
untag_resource	Removes tags from a resource in CodeArtifact
update_package_group	Updates a package group
update_package_group_origin_configuration	Updates the package origin configuration for a package group
update_package_versions_status	Updates the status of one or more versions of a package
update_repository	Update the properties of a repository

Examples

```
## Not run:
svc <- codeartifact()
svc$associate_external_connection(
  Foo = 123
)

## End(Not run)
```

---

codebuild *AWS CodeBuild*

Description

CodeBuild

CodeBuild is a fully managed build service in the cloud. CodeBuild compiles your source code, runs unit tests, and produces artifacts that are ready to deploy. CodeBuild eliminates the need to provision, manage, and scale your own build servers. It provides prepackaged build environments for the most popular programming languages and build tools, such as Apache Maven, Gradle, and more. You can also fully customize build environments in CodeBuild to use your own build tools. CodeBuild scales automatically to meet peak build requests. You pay only for the build time you consume. For more information about CodeBuild, see the *CodeBuild User Guide*.

Usage

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

Operations

batch_delete_builds	Deletes one or more builds
batch_get_build_batches	Retrieves information about one or more batch builds
batch_get_builds	Gets information about one or more builds
batch_get_fleets	Gets information about one or more compute fleets
batch_get_projects	Gets information about one or more build projects
batch_get_report_groups	Returns an array of report groups
batch_get_reports	Returns an array of reports
create_fleet	Creates a compute fleet
create_project	Creates a build project
create_report_group	Creates a report group
create_webhook	For an existing CodeBuild build project that has its source code stored in a GitHub or Bitl
delete_build_batch	Deletes a batch build
delete_fleet	Deletes a compute fleet
delete_project	Deletes a build project
delete_report	Deletes a report
delete_report_group	Deletes a report group
delete_resource_policy	Deletes a resource policy that is identified by its resource ARN
delete_source_credentials	Deletes a set of GitHub, GitHub Enterprise, or Bitbucket source credentials
delete_webhook	For an existing CodeBuild build project that has its source code stored in a GitHub or Bitl
describe_code_coverages	Retrieves one or more code coverage reports
describe_test_cases	Returns a list of details about test cases for a report
get_report_group_trend	Analyzes and accumulates test report values for the specified test reports
get_resource_policy	Gets a resource policy that is identified by its resource ARN
import_source_credentials	Imports the source repository credentials for an CodeBuild project that has its source code
invalidate_project_cache	Resets the cache for a project
list_build_batches	Retrieves the identifiers of your build batches in the current region
list_build_batches_for_project	Retrieves the identifiers of the build batches for a specific project
list_builds	Gets a list of build IDs, with each build ID representing a single build
list_builds_for_project	Gets a list of build identifiers for the specified build project, with each build identifier repr
list_curated_environment_images	Gets information about Docker images that are managed by CodeBuild
list_fleets	Gets a list of compute fleet names with each compute fleet name representing a single con
list_projects	Gets a list of build project names, with each build project name representing a single build
list_report_groups	Gets a list ARNs for the report groups in the current Amazon Web Services account
list_reports	Returns a list of ARNs for the reports in the current Amazon Web Services account
list_reports_for_report_group	Returns a list of ARNs for the reports that belong to a ReportGroup
list_shared_projects	Gets a list of projects that are shared with other Amazon Web Services accounts or users
list_shared_report_groups	Gets a list of report groups that are shared with other Amazon Web Services accounts or u
list_source_credentials	Returns a list of SourceCredentialsInfo objects
put_resource_policy	Stores a resource policy for the ARN of a Project or ReportGroup object
retry_build	Restarts a build
retry_build_batch	Restarts a failed batch build
start_build	Starts running a build with the settings defined in the project
start_build_batch	Starts a batch build for a project
stop_build	Attempts to stop running a build
stop_build_batch	Stops a running batch build
update_fleet	Updates a compute fleet
update_project	Changes the settings of a build project
update_project_visibility	Changes the public visibility for a project

update_report_group	Updates a report group
update_webhook	Updates the webhook associated with an CodeBuild build project

Examples

```
## Not run:
svc <- codebuild()
# The following example gets information about builds with the specified
# build IDs.
svc$batch_get_builds(
  ids = list(
    "codebuild-demo-project:9b0ac37f-d19e-4254-9079-f47e9a389eEX",
    "codebuild-demo-project:b79a46f7-1473-4636-a23f-da9c45c208EX"
  )
)

## End(Not run)
```

---

codecatalyst *Amazon CodeCatalyst*

Description

Welcome to the Amazon CodeCatalyst API reference. This reference provides descriptions of operations and data types for Amazon CodeCatalyst. You can use the Amazon CodeCatalyst API to work with the following objects.

Spaces, by calling the following:

- delete_space, which deletes a space.
- get_space, which returns information about a space.
- get_subscription, which returns information about the Amazon Web Services account used for billing purposes and the billing plan for the space.
- list_spaces, which retrieves a list of spaces.
- update_space, which changes one or more values for a space.

Projects, by calling the following:

- create_project which creates a project in a specified space.
- get_project, which returns information about a project.
- list_projects, which retrieves a list of projects in a space.

Users, by calling the following:

- get_user_details, which returns information about a user in Amazon CodeCatalyst.

Source repositories, by calling the following:

- create_source_repository, which creates an empty Git-based source repository in a specified project.
- create_source_repository_branch, which creates a branch in a specified repository where you can work on code.
- delete_source_repository, which deletes a source repository.
- get_source_repository, which returns information about a source repository.
- get_source_repository_clone_urls, which returns information about the URLs that can be used with a Git client to clone a source repository.
- list_source_repositories, which retrieves a list of source repositories in a project.
- list_source_repository_branches, which retrieves a list of branches in a source repository.

Dev Environments and the Amazon Web Services Toolkits, by calling the following:

- create_dev_environment, which creates a Dev Environment, where you can quickly work on the code stored in the source repositories of your project.
- delete_dev_environment, which deletes a Dev Environment.
- get_dev_environment, which returns information about a Dev Environment.
- list_dev_environments, which retrieves a list of Dev Environments in a project.
- list_dev_environment_sessions, which retrieves a list of active Dev Environment sessions in a project.
- start_dev_environment, which starts a specified Dev Environment and puts it into an active state.
- start_dev_environment_session, which starts a session to a specified Dev Environment.
- stop_dev_environment, which stops a specified Dev Environment and puts it into an stopped state.
- stop_dev_environment_session, which stops a session for a specified Dev Environment.
- update_dev_environment, which changes one or more values for a Dev Environment.

Workflows, by calling the following:

- get_workflow, which returns information about a workflow.
- get_workflow_run, which returns information about a specified run of a workflow.
- list_workflow_runs, which retrieves a list of runs of a specified workflow.
- list_workflows, which retrieves a list of workflows in a specified project.
- start_workflow_run, which starts a run of a specified workflow.

Security, activity, and resource management in Amazon CodeCatalyst, by calling the following:

- create_access_token, which creates a personal access token (PAT) for the current user.
- delete_access_token, which deletes a specified personal access token (PAT).
- list_access_tokens, which lists all personal access tokens (PATs) associated with a user.

- list_event_logs, which retrieves a list of events that occurred during a specified time period in a space.
- verify_session, which verifies whether the calling user has a valid Amazon CodeCatalyst login and session.

If you are using the Amazon CodeCatalyst APIs with an SDK or the CLI, you must configure your computer to work with Amazon CodeCatalyst and single sign-on (SSO). For more information, see Setting up to use the CLI with Amazon CodeCatalyst and the SSO documentation for your SDK.

Usage

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

Operations

create_access_token	Creates a personal access token (PAT) for the current user
create_dev_environment	Creates a Dev Environment in Amazon CodeCatalyst, a cloud-based development enviror
create_project	Creates a project in a specified space

create_source_repository	Creates an empty Git-based source repository in a specified project
create_source_repository_branch	Creates a branch in a specified source repository in Amazon CodeCatalyst
delete_access_token	Deletes a specified personal access token (PAT)
delete_dev_environment	Deletes a Dev Environment
delete_project	Deletes a project in a space
delete_source_repository	Deletes a source repository in Amazon CodeCatalyst
delete_space	Deletes a space
get_dev_environment	Returns information about a Dev Environment for a source repository in a project
get_project	Returns information about a project
get_source_repository	Returns information about a source repository
get_source_repository_clone_urls	Returns information about the URLs that can be used with a Git client to clone a source r
get_space	Returns information about an space
get_subscription	Returns information about the Amazon Web Services account used for billing purposes a
get_user_details	Returns information about a user
get_workflow	Returns information about a workflow
get_workflow_run	Returns information about a specified run of a workflow
list_access_tokens	Lists all personal access tokens (PATs) associated with the user who calls the API
list_dev_environments	Retrieves a list of Dev Environments in a project
list_dev_environment_sessions	Retrieves a list of active sessions for a Dev Environment in a project
list_event_logs	Retrieves a list of events that occurred during a specific time in a space
list_projects	Retrieves a list of projects
list_source_repositories	Retrieves a list of source repositories in a project
list_source_repository_branches	Retrieves a list of branches in a specified source repository
list_spaces	Retrieves a list of spaces
list_workflow_runs	Retrieves a list of workflow runs of a specified workflow
list_workflows	Retrieves a list of workflows in a specified project
start_dev_environment	Starts a specified Dev Environment and puts it into an active state
start_dev_environment_session	Starts a session for a specified Dev Environment
start_workflow_run	Begins a run of a specified workflow
stop_dev_environment	Pauses a specified Dev Environment and places it in a non-running state
stop_dev_environment_session	Stops a session for a specified Dev Environment
update_dev_environment	Changes one or more values for a Dev Environment
update_project	Changes one or more values for a project
update_space	Changes one or more values for a space
verify_session	Verifies whether the calling user has a valid Amazon CodeCatalyst login and session

Examples

```
## Not run:
svc <- codecatalyst()
svc$create_access_token(
  Foo = 123
)

## End(Not run)
```

codecommit *AWS CodeCommit*

Description

CodeCommit

This is the *CodeCommit API Reference*. This reference provides descriptions of the operations and data types for CodeCommit API along with usage examples.

You can use the CodeCommit API to work with the following objects:

Repositories, by calling the following:

- batch_get_repositories, which returns information about one or more repositories associated with your Amazon Web Services account.
- create_repository, which creates an CodeCommit repository.
- delete_repository, which deletes an CodeCommit repository.
- get_repository, which returns information about a specified repository.
- list_repositories, which lists all CodeCommit repositories associated with your Amazon Web Services account.
- update_repository_description, which sets or updates the description of the repository.
- update_repository_encryption_key, which updates the Key Management Service encryption key used to encrypt and decrypt a repository.
- update_repository_name, which changes the name of the repository. If you change the name of a repository, no other users of that repository can access it until you send them the new HTTPS or SSH URL to use.

Branches, by calling the following:

- create_branch, which creates a branch in a specified repository.
- delete_branch, which deletes the specified branch in a repository unless it is the default branch.
- get_branch, which returns information about a specified branch.
- list_branches, which lists all branches for a specified repository.
- update_default_branch, which changes the default branch for a repository.

Files, by calling the following:

- delete_file, which deletes the content of a specified file from a specified branch.
- get_blob, which returns the base-64 encoded content of an individual Git blob object in a repository.
- get_file, which returns the base-64 encoded content of a specified file.
- get_folder, which returns the contents of a specified folder or directory.
- list_file_commit_history, which retrieves a list of commits and changes to a specified file.

- put_file, which adds or modifies a single file in a specified repository and branch.

Commits, by calling the following:

- batch_get_commits, which returns information about one or more commits in a repository.
- create_commit, which creates a commit for changes to a repository.
- get_commit, which returns information about a commit, including commit messages and author and committer information.
- get_differences, which returns information about the differences in a valid commit specifier (such as a branch, tag, HEAD, commit ID, or other fully qualified reference).

Merges, by calling the following:

- batch_describe_merge_conflicts, which returns information about conflicts in a merge between commits in a repository.
- create_unreferenced_merge_commit, which creates an unreferenced commit between two branches or commits for the purpose of comparing them and identifying any potential conflicts.
- describe_merge_conflicts, which returns information about merge conflicts between the base, source, and destination versions of a file in a potential merge.
- get_merge_commit, which returns information about the merge between a source and destination commit.
- get_merge_conflicts, which returns information about merge conflicts between the source and destination branch in a pull request.
- get_merge_options, which returns information about the available merge options between two branches or commit specifiers.
- merge_branches_by_fast_forward, which merges two branches using the fast-forward merge option.
- merge_branches_by_squash, which merges two branches using the squash merge option.
- merge_branches_by_three_way, which merges two branches using the three-way merge option.

Pull requests, by calling the following:

- create_pull_request, which creates a pull request in a specified repository.
- create_pull_request_approval_rule, which creates an approval rule for a specified pull request.
- delete_pull_request_approval_rule, which deletes an approval rule for a specified pull request.
- describe_pull_request_events, which returns information about one or more pull request events.
- evaluate_pull_request_approval_rules, which evaluates whether a pull request has met all the conditions specified in its associated approval rules.
- get_comments_for_pull_request, which returns information about comments on a specified pull request.

- get_pull_request, which returns information about a specified pull request.
- get_pull_request_approval_states, which returns information about the approval states for a specified pull request.
- get_pull_request_override_state, which returns information about whether approval rules have been set aside (overriden) for a pull request, and if so, the Amazon Resource Name (ARN) of the user or identity that overrode the rules and their requirements for the pull request.
- list_pull_requests, which lists all pull requests for a repository.
- merge_pull_request_by_fast_forward, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the fast-forward merge option.
- merge_pull_request_by_squash, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the squash merge option.
- merge_pull_request_by_three_way, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the three-way merge option.
- override_pull_request_approval_rules, which sets aside all approval rule requirements for a pull request.
- post_comment_for_pull_request, which posts a comment to a pull request at the specified line, file, or request.
- update_pull_request_approval_rule_content, which updates the structure of an approval rule for a pull request.
- update_pull_request_approval_state, which updates the state of an approval on a pull request.
- update_pull_request_description, which updates the description of a pull request.
- update_pull_request_status, which updates the status of a pull request.
- update_pull_request_title, which updates the title of a pull request.

Approval rule templates, by calling the following:

- associate_approval_rule_template_with_repository, which associates a template with a specified repository. After the template is associated with a repository, CodeCommit creates approval rules that match the template conditions on every pull request created in the specified repository.
- batch_associate_approval_rule_template_with_repositories, which associates a template with one or more specified repositories. After the template is associated with a repository, CodeCommit creates approval rules that match the template conditions on every pull request created in the specified repositories.
- batch_disassociate_approval_rule_template_from_repositories, which removes the association between a template and specified repositories so that approval rules based on the template are not automatically created when pull requests are created in those repositories.
- create_approval_rule_template, which creates a template for approval rules that can then be associated with one or more repositories in your Amazon Web Services account.

- delete_approval_rule_template, which deletes the specified template. It does not remove approval rules on pull requests already created with the template.
- disassociate_approval_rule_template_from_repository, which removes the association between a template and a repository so that approval rules based on the template are not automatically created when pull requests are created in the specified repository.
- get_approval_rule_template, which returns information about an approval rule template.
- list_approval_rule_templates, which lists all approval rule templates in the Amazon Web Services Region in your Amazon Web Services account.
- list_associated_approval_rule_templates_for_repository, which lists all approval rule templates that are associated with a specified repository.
- list_repositories_for_approval_rule_template, which lists all repositories associated with the specified approval rule template.
- update_approval_rule_template_description, which updates the description of an approval rule template.
- update_approval_rule_template_name, which updates the name of an approval rule template.
- update_approval_rule_template_content, which updates the content of an approval rule template.

Comments in a repository, by calling the following:

- delete_comment_content, which deletes the content of a comment on a commit in a repository.
- get_comment, which returns information about a comment on a commit.
- get_comment_reactions, which returns information about emoji reactions to comments.
- get_comments_for_compared_commit, which returns information about comments on the comparison between two commit specifiers in a repository.
- post_comment_for_compared_commit, which creates a comment on the comparison between two commit specifiers in a repository.
- post_comment_reply, which creates a reply to a comment.
- put_comment_reaction, which creates or updates an emoji reaction to a comment.
- update_comment, which updates the content of a comment on a commit in a repository.

Tags used to tag resources in CodeCommit (not Git tags), by calling the following:

- list_tags_for_resource, which gets information about Amazon Web Servicestags for a specified Amazon Resource Name (ARN) in CodeCommit.
- tag_resource, which adds or updates tags for a resource in CodeCommit.
- untag_resource, which removes tags for a resource in CodeCommit.

Triggers, by calling the following:

- get_repository_triggers, which returns information about triggers configured for a repository.

- put_repository_triggers, which replaces all triggers for a repository and can be used to create or delete triggers.
- test_repository_triggers, which tests the functionality of a repository trigger by sending data to the trigger target.

For information about how to use CodeCommit, see the CodeCommit User Guide.

Usage

```
codecommit(
  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 <- codecommit(
  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_approval_rule_template_with_repository	Creates an association between an approval rule template and
batch_associate_approval_rule_template_with_repositories	Creates an association between an approval rule template and
batch_describe_merge_conflicts	Returns information about one or more merge conflicts in the
batch_disassociate_approval_rule_template_from_repositories	Removes the association between an approval rule template a
batch_get_commits	Returns information about the contents of one or more comm
batch_get_repositories	Returns information about one or more repositories
create_approval_rule_template	Creates a template for approval rules that can then be associa
create_branch	Creates a branch in a repository and points the branch to a co

create_commit	Creates a commit for a repository on the tip of a specified bra
create_pull_request	Creates a pull request in the specified repository
create_pull_request_approval_rule	Creates an approval rule for a pull request
create_repository	Creates a new, empty repository
create_unreferenced_merge_commit	Creates an unreferenced commit that represents the result of
delete_approval_rule_template	Deletes a specified approval rule template
delete_branch	Deletes a branch from a repository, unless that branch is the
delete_comment_content	Deletes the content of a comment made on a change, file, or
delete_file	Deletes a specified file from a specified branch
delete_pull_request_approval_rule	Deletes an approval rule from a specified pull request
delete_repository	Deletes a repository
describe_merge_conflicts	Returns information about one or more merge conflicts in the
describe_pull_request_events	Returns information about one or more pull request events
disassociate_approval_rule_template_from_repository	Removes the association between a template and a repository
evaluate_pull_request_approval_rules	Evaluates whether a pull request has met all the conditions sp
get_approval_rule_template	Returns information about a specified approval rule template
get_blob	Returns the base-64 encoded content of an individual blob in
get_branch	Returns information about a repository branch, including its
get_comment	Returns the content of a comment made on a change, file, or
get_comment_reactions	Returns information about reactions to a specified comment
get_comments_for_compared_commit	Returns information about comments made on the comparisc
get_comments_for_pull_request	Returns comments made on a pull request
get_commit	Returns information about a commit, including commit mess
get_differences	Returns information about the differences in a valid commit
get_file	Returns the base-64 encoded contents of a specified file and i
get_folder	Returns the contents of a specified folder in a repository
get_merge_commit	Returns information about a specified merge commit
get_merge_conflicts	Returns information about merge conflicts between the befor
get_merge_options	Returns information about the merge options available for m
get_pull_request	Gets information about a pull request in a specified repositor
get_pull_request_approval_states	Gets information about the approval states for a specified pul
get_pull_request_override_state	Returns information about whether approval rules have been
get_repository	Returns information about a repository
get_repository_triggers	Gets information about triggers configured for a repository
list_approval_rule_templates	Lists all approval rule templates in the specified Amazon We
list_associated_approval_rule_templates_for_repository	Lists all approval rule templates that are associated with a sp
list_branches	Gets information about one or more branches in a repository
list_file_commit_history	Retrieves a list of commits and changes to a specified file
list_pull_requests	Returns a list of pull requests for a specified repository
list_repositories	Gets information about one or more repositories
list_repositories_for_approval_rule_template	Lists all repositories associated with the specified approval ru
list_tags_for_resource	Gets information about Amazon Web Servicestags for a spec
merge_branches_by_fast_forward	Merges two branches using the fast-forward merge strategy
merge_branches_by_squash	Merges two branches using the squash merge strategy
merge_branches_by_three_way	Merges two specified branches using the three-way merge str
merge_pull_request_by_fast_forward	Attempts to merge the source commit of a pull request into th
merge_pull_request_by_squash	Attempts to merge the source commit of a pull request into th
merge_pull_request_by_three_way	Attempts to merge the source commit of a pull request into th

override_pull_request_approval_rules — Sets aside (overrides) all approval rule requirements for a spe
post_comment_for_compared_commit — Posts a comment on the comparison between two commits
post_comment_for_pull_request — Posts a comment on a pull request
post_comment_reply — Posts a comment in reply to an existing comment on a compa
put_comment_reaction — Adds or updates a reaction to a specified comment for the us
put_file — Adds or updates a file in a branch in an CodeCommit reposit
put_repository_triggers — Replaces all triggers for a repository
tag_resource — Adds or updates tags for a resource in CodeCommit
test_repository_triggers — Tests the functionality of repository triggers by sending infor
untag_resource — Removes tags for a resource in CodeCommit
update_approval_rule_template_content — Updates the content of an approval rule template
update_approval_rule_template_description — Updates the description for a specified approval rule templat
update_approval_rule_template_name — Updates the name of a specified approval rule template
update_comment — Replaces the contents of a comment
update_default_branch — Sets or changes the default branch name for the specified rep
update_pull_request_approval_rule_content — Updates the structure of an approval rule created specifically
update_pull_request_approval_state — Updates the state of a user's approval on a pull request
update_pull_request_description — Replaces the contents of the description of a pull request
update_pull_request_status — Updates the status of a pull request
update_pull_request_title — Replaces the title of a pull request
update_repository_description — Sets or changes the comment or description for a repository
update_repository_encryption_key — Updates the Key Management Service encryption key used t
update_repository_name — Renames a repository

Examples

```
## Not run:
svc <- codecommit()
svc$associate_approval_rule_template_with_repository(
  Foo = 123
)

## End(Not run)
```

`codeconnections` *AWS CodeConnections*

Description

This Amazon Web Services CodeConnections API Reference provides descriptions and usage examples of the operations and data types for the Amazon Web Services CodeConnections API. You can use the connections API to work with connections and installations.

Connections are configurations that you use to connect Amazon Web Services resources to external code repositories. Each connection is a resource that can be given to services such as CodePipeline

to connect to a third-party repository such as Bitbucket. For example, you can add the connection in CodePipeline so that it triggers your pipeline when a code change is made to your third-party code repository. Each connection is named and associated with a unique ARN that is used to reference the connection.

When you create a connection, the console initiates a third-party connection handshake. *Installations* are the apps that are used to conduct this handshake. For example, the installation for the Bitbucket provider type is the Bitbucket app. When you create a connection, you can choose an existing installation or create one.

When you want to create a connection to an installed provider type such as GitHub Enterprise Server, you create a *host* for your connections.

You can work with connections by calling:

- `create_connection`, which creates a uniquely named connection that can be referenced by services such as CodePipeline.
- `delete_connection`, which deletes the specified connection.
- `get_connection`, which returns information about the connection, including the connection status.
- `list_connections`, which lists the connections associated with your account.

You can work with hosts by calling:

- `create_host`, which creates a host that represents the infrastructure where your provider is installed.
- `delete_host`, which deletes the specified host.
- `get_host`, which returns information about the host, including the setup status.
- `list_hosts`, which lists the hosts associated with your account.

You can work with tags in Amazon Web Services CodeConnections by calling the following:

- `list_tags_for_resource`, which gets information about Amazon Web Services tags for a specified Amazon Resource Name (ARN) in Amazon Web Services CodeConnections.
- `tag_resource`, which adds or updates tags for a resource in Amazon Web Services CodeConnections.
- `untag_resource`, which removes tags for a resource in Amazon Web Services CodeConnections.

For information about how to use Amazon Web Services CodeConnections, see the Developer Tools User Guide.

Usage

```
codeconnections(
  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 <- codeconnections(
  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_connection	Creates a connection that can then be given to other Amazon Web Services services like Co
create_host	Creates a resource that represents the infrastructure where a third-party provider is installed
create_repository_link	Creates a link to a specified external Git repository
create_sync_configuration	Creates a sync configuration which allows Amazon Web Services to sync content from a Gi
delete_connection	The connection to be deleted
delete_host	The host to be deleted
delete_repository_link	Deletes the association between your connection and a specified external Git repository
delete_sync_configuration	Deletes the sync configuration for a specified repository and connection
get_connection	Returns the connection ARN and details such as status, owner, and provider type
get_host	Returns the host ARN and details such as status, provider type, endpoint, and, if applicable
get_repository_link	Returns details about a repository link
get_repository_sync_status	Returns details about the sync status for a repository
get_resource_sync_status	Returns the status of the sync with the Git repository for a specific Amazon Web Services r
get_sync_blocker_summary	Returns a list of the most recent sync blockers
get_sync_configuration	Returns details about a sync configuration, including the sync type and resource name
list_connections	Lists the connections associated with your account
list_hosts	Lists the hosts associated with your account
list_repository_links	Lists the repository links created for connections in your account
list_repository_sync_definitions	Lists the repository sync definitions for repository links in your account
list_sync_configurations	Returns a list of sync configurations for a specified repository

list_tags_for_resource	Gets the set of key-value pairs (metadata) that are used to manage the resource
tag_resource	Adds to or modifies the tags of the given resource
untag_resource	Removes tags from an Amazon Web Services resource
update_host	Updates a specified host with the provided configurations
update_repository_link	Updates the association between your connection and a specified external Git repository
update_sync_blocker	Allows you to update the status of a sync blocker, resolving the blocker and allowing synci
update_sync_configuration	Updates the sync configuration for your connection and a specified external Git repository

Examples

```
## Not run:
svc <- codeconnections()
svc$create_connection(
  Foo = 123
)

## End(Not run)
```

`codedeploy` *AWS CodeDeploy*

Description

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances running in your own facility, serverless Lambda functions, or applications in an Amazon ECS service.

You can deploy a nearly unlimited variety of application content, such as an updated Lambda function, updated applications in an Amazon ECS service, code, web and configuration files, executables, packages, scripts, multimedia files, and so on. CodeDeploy can deploy application content stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. You do not need to make changes to your existing code before you can use CodeDeploy.

CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

CodeDeploy Components

Use the information in this guide to help you work with the following CodeDeploy components:

- **Application**: A name that uniquely identifies the application you want to deploy. CodeDeploy uses this name, which functions as a container, to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.

- **Deployment group**: A set of individual instances, CodeDeploy Lambda deployment configuration settings, or an Amazon ECS service and network details. A Lambda deployment group specifies how to route traffic to a new version of a Lambda function. An Amazon ECS deployment group specifies the service created in Amazon ECS to deploy, a load balancer, and a listener to reroute production traffic to an updated containerized application. An Amazon EC2/On-premises deployment group contains individually tagged instances, Amazon EC2 instances in Amazon EC2 Auto Scaling groups, or both. All deployment groups can specify optional trigger, alarm, and rollback settings.
- **Deployment configuration**: A set of deployment rules and deployment success and failure conditions used by CodeDeploy during a deployment.
- **Deployment**: The process and the components used when updating a Lambda function, a containerized application in an Amazon ECS service, or of installing content on one or more instances.
- **Application revisions**: For an Lambda deployment, this is an AppSpec file that specifies the Lambda function to be updated and one or more functions to validate deployment lifecycle events. For an Amazon ECS deployment, this is an AppSpec file that specifies the Amazon ECS task definition, container, and port where production traffic is rerouted. For an EC2/On-premises deployment, this is an archive file that contains source content—source code, webpages, executable files, and deployment scripts—along with an AppSpec file. Revisions are stored in Amazon S3 buckets or GitHub repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.

This guide also contains information to help you get details about the instances in your deployments, to make on-premises instances available for CodeDeploy deployments, to get details about a Lambda function deployment, and to get details about Amazon ECS service deployments.

CodeDeploy Information Resources

- CodeDeploy User Guide
- CodeDeploy API Reference Guide
- CLI Reference for CodeDeploy
- CodeDeploy Developer Forum

Usage

```
codedeploy(
  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 <- codedeploy(
  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_on_premises_instances	Adds tags to on-premises instances
batch_get_application_revisions	Gets information about one or more application revisions
batch_get_applications	Gets information about one or more applications
batch_get_deployment_groups	Gets information about one or more deployment groups
batch_get_deployment_instances	This method works, but is deprecated
batch_get_deployments	Gets information about one or more deployments
batch_get_deployment_targets	Returns an array of one or more targets associated with a deployment
batch_get_on_premises_instances	Gets information about one or more on-premises instances
continue_deployment	For a blue/green deployment, starts the process of rerouting traffic from instanc
create_application	Creates an application
create_deployment	Deploys an application revision through the specified deployment group
create_deployment_config	Creates a deployment configuration
create_deployment_group	Creates a deployment group to which application revisions are deployed
delete_application	Deletes an application
delete_deployment_config	Deletes a deployment configuration
delete_deployment_group	Deletes a deployment group
delete_git_hub_account_token	Deletes a GitHub account connection
delete_resources_by_external_id	Deletes resources linked to an external ID
deregister_on_premises_instance	Deregisters an on-premises instance
get_application	Gets information about an application
get_application_revision	Gets information about an application revision
get_deployment	Gets information about a deployment
get_deployment_config	Gets information about a deployment configuration
get_deployment_group	Gets information about a deployment group
get_deployment_instance	Gets information about an instance as part of a deployment

get_deployment_target	Returns information about a deployment target
get_on_premises_instance	Gets information about an on-premises instance
list_application_revisions	Lists information about revisions for an application
list_applications	Lists the applications registered with the user or Amazon Web Services account
list_deployment_configs	Lists the deployment configurations with the user or Amazon Web Services acc
list_deployment_groups	Lists the deployment groups for an application registered with the Amazon Web
list_deployment_instances	The newer BatchGetDeploymentTargets should be used instead because it work
list_deployments	Lists the deployments in a deployment group for an application registered with
list_deployment_targets	Returns an array of target IDs that are associated a deployment
list_git_hub_account_token_names	Lists the names of stored connections to GitHub accounts
list_on_premises_instances	Gets a list of names for one or more on-premises instances
list_tags_for_resource	Returns a list of tags for the resource identified by a specified Amazon Resource
put_lifecycle_event_hook_execution_status	Sets the result of a Lambda validation function
register_application_revision	Registers with CodeDeploy a revision for the specified application
register_on_premises_instance	Registers an on-premises instance
remove_tags_from_on_premises_instances	Removes one or more tags from one or more on-premises instances
skip_wait_time_for_instance_termination	In a blue/green deployment, overrides any specified wait time and starts termina
stop_deployment	Attempts to stop an ongoing deployment
tag_resource	Associates the list of tags in the input Tags parameter with the resource identifie
untag_resource	Disassociates a resource from a list of tags
update_application	Changes the name of an application
update_deployment_group	Changes information about a deployment group

Examples

```
## Not run:
svc <- codedeploy()
svc$add_tags_to_on_premises_instances(
  Foo = 123
)

## End(Not run)
```

`codeguruprofiler` *Amazon CodeGuru Profiler*

Description

This section provides documentation for the Amazon CodeGuru Profiler API operations.

Amazon CodeGuru Profiler collects runtime performance data from your live applications, and provides recommendations that can help you fine-tune your application performance. Using machine learning algorithms, CodeGuru Profiler can help you find your most expensive lines of code and suggest ways you can improve efficiency and remove CPU bottlenecks.

Amazon CodeGuru Profiler provides different visualizations of profiling data to help you identify what code is running on the CPU, see how much time is consumed, and suggest ways to reduce CPU utilization.

Amazon CodeGuru Profiler currently supports applications written in all Java virtual machine (JVM) languages and Python. While CodeGuru Profiler supports both visualizations and recommendations for applications written in Java, it can also generate visualizations and a subset of recommendations for applications written in other JVM languages and Python.

For more information, see What is Amazon CodeGuru Profiler in the *Amazon CodeGuru Profiler User Guide*.

Usage

```
codeguruprofiler(
  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 <- codeguruprofiler(
  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_notification_channels	Add up to 2 anomaly notifications channels for a profiling group
batch_get_frame_metric_data	Returns the time series of values for a requested list of frame metrics from a time pe
configure_agent	Used by profiler agents to report their current state and to receive remote configurati
create_profiling_group	Creates a profiling group
delete_profiling_group	Deletes a profiling group
describe_profiling_group	Returns a ProfilingGroupDescription object that contains information about the requ
get_findings_report_account_summary	Returns a list of FindingsReportSummary objects that contain analysis results for all
get_notification_configuration	Get the current configuration for anomaly notifications for a profiling group
get_policy	Returns the JSON-formatted resource-based policy on a profiling group
get_profile	Gets the aggregated profile of a profiling group for a specified time range
get_recommendations	Returns a list of Recommendation objects that contain recommendations for a profil
list_findings_reports	List the available reports for a given profiling group and time range
list_profile_times	Lists the start times of the available aggregated profiles of a profiling group for an ag
list_profiling_groups	Returns a list of profiling groups
list_tags_for_resource	Returns a list of the tags that are assigned to a specified resource
post_agent_profile	Submits profiling data to an aggregated profile of a profiling group
put_permission	Adds permissions to a profiling group's resource-based policy that are provided usin
remove_notification_channel	Remove one anomaly notifications channel for a profiling group
remove_permission	Removes permissions from a profiling group's resource-based policy that are provid
submit_feedback	Sends feedback to CodeGuru Profiler about whether the anomaly detected by the ana
tag_resource	Use to assign one or more tags to a resource
untag_resource	Use to remove one or more tags from a resource
update_profiling_group	Updates a profiling group

Examples

```
## Not run:
svc <- codeguruprofiler()
svc$add_notification_channels(
  Foo = 123
)

## End(Not run)
```

`codegurureviewer` *Amazon CodeGuru Reviewer*

Description

This section provides documentation for the Amazon CodeGuru Reviewer API operations. CodeGuru Reviewer is a service that uses program analysis and machine learning to detect potential defects that are difficult for developers to find and recommends fixes in your Java and Python code.

By proactively detecting and providing recommendations for addressing code defects and implementing best practices, CodeGuru Reviewer improves the overall quality and maintainability of

your code base during the code review stage. For more information about CodeGuru Reviewer, see the *Amazon CodeGuru Reviewer User Guide*.

To improve the security of your CodeGuru Reviewer API calls, you can establish a private connection between your VPC and CodeGuru Reviewer by creating an *interface VPC endpoint*. For more information, see CodeGuru Reviewer and interface VPC endpoints (Amazon Web Services PrivateLink) in the *Amazon CodeGuru Reviewer User Guide*.

Usage

```
codegurureviewer(
  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 <- codegurureviewer(
  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_repository	Use to associate an Amazon Web Services CodeCommit repository or a repository mar
create_code_review	Use to create a code review with a CodeReviewType of RepositoryAnalysis
describe_code_review	Returns the metadata associated with the code review along with its status
describe_recommendation_feedback	Describes the customer feedback for a CodeGuru Reviewer recommendation

describe_repository_association	Returns a RepositoryAssociation object that contains information about the requested r
disassociate_repository	Removes the association between Amazon CodeGuru Reviewer and a repository
list_code_reviews	Lists all the code reviews that the customer has created in the past 90 days
list_recommendation_feedback	Returns a list of RecommendationFeedbackSummary objects that contain customer rec
list_recommendations	Returns the list of all recommendations for a completed code review
list_repository_associations	Returns a list of RepositoryAssociationSummary objects that contain summary informa
list_tags_for_resource	Returns the list of tags associated with an associated repository resource
put_recommendation_feedback	Stores customer feedback for a CodeGuru Reviewer recommendation
tag_resource	Adds one or more tags to an associated repository
untag_resource	Removes a tag from an associated repository

Examples

```
## Not run:
svc <- codegurureviewer()
svc$associate_repository(
  Foo = 123
)

## End(Not run)
```

`codegurusecurity` *Amazon CodeGuru Security*

Description

Amazon CodeGuru Security is in preview release and is subject to change.

This section provides documentation for the Amazon CodeGuru Security API operations. CodeGuru Security is a service that uses program analysis and machine learning to detect security policy violations and vulnerabilities, and recommends ways to address these security risks.

By proactively detecting and providing recommendations for addressing security risks, CodeGuru Security improves the overall security of your application code. For more information about CodeGuru Security, see the Amazon CodeGuru Security User Guide.

Usage

```
codegurusecurity(
  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 <- codegurusecurity(
  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_findings	Returns a list of requested findings from standard scans
create_scan	Use to create a scan using code uploaded to an Amazon S3 bucket
create_upload_url	Generates a pre-signed URL, request headers used to upload a code resource, and code artifa
get_account_configuration	Use to get the encryption configuration for an account
get_findings	Returns a list of all findings generated by a particular scan
get_metrics_summary	Returns a summary of metrics for an account from a specified date, including number of oper
get_scan	Returns details about a scan, including whether or not a scan has completed
list_findings_metrics	Returns metrics about all findings in an account within a specified time range
list_scans	Returns a list of all scans in an account
list_tags_for_resource	Returns a list of all tags associated with a scan
tag_resource	Use to add one or more tags to an existing scan
untag_resource	Use to remove one or more tags from an existing scan
update_account_configuration	Use to update the encryption configuration for an account

Examples

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

```
svc$batch_get_findings(
  Foo = 123
)

## End(Not run)
```

`codepipeline` *AWS CodePipeline*

Description

CodePipeline

Overview

This is the CodePipeline API Reference. This guide provides descriptions of the actions and data types for CodePipeline. Some functionality for your pipeline can only be configured through the API. For more information, see the CodePipeline User Guide.

You can use the CodePipeline API to work with pipelines, stages, actions, and transitions.

Pipelines are models of automated release processes. Each pipeline is uniquely named, and consists of stages, actions, and transitions.

You can work with pipelines by calling:

- `create_pipeline`, which creates a uniquely named pipeline.
- `delete_pipeline`, which deletes the specified pipeline.
- `get_pipeline`, which returns information about the pipeline structure and pipeline metadata, including the pipeline Amazon Resource Name (ARN).
- `get_pipeline_execution`, which returns information about a specific execution of a pipeline.
- `get_pipeline_state`, which returns information about the current state of the stages and actions of a pipeline.
- `list_action_executions`, which returns action-level details for past executions. The details include full stage and action-level details, including individual action duration, status, any errors that occurred during the execution, and input and output artifact location details.
- `list_pipelines`, which gets a summary of all of the pipelines associated with your account.
- `list_pipeline_executions`, which gets a summary of the most recent executions for a pipeline.
- `start_pipeline_execution`, which runs the most recent revision of an artifact through the pipeline.
- `stop_pipeline_execution`, which stops the specified pipeline execution from continuing through the pipeline.
- `update_pipeline`, which updates a pipeline with edits or changes to the structure of the pipeline.

Pipelines include *stages*. Each stage contains one or more actions that must complete before the next stage begins. A stage results in success or failure. If a stage fails, the pipeline stops at that stage and remains stopped until either a new version of an artifact appears in the source location, or a user takes action to rerun the most recent artifact through the pipeline. You can call `get_pipeline_state`, which displays the status of a pipeline, including the status of stages in the pipeline, or `get_pipeline`, which returns the entire structure of the pipeline, including the stages of that pipeline. For more information about the structure of stages and actions, see CodePipeline Pipeline Structure Reference.

Pipeline stages include *actions* that are categorized into categories such as source or build actions performed in a stage of a pipeline. For example, you can use a source action to import artifacts into a pipeline from a source such as Amazon S3. Like stages, you do not work with actions directly in most cases, but you do define and interact with actions when working with pipeline operations such as `create_pipeline` and `get_pipeline_state`. Valid action categories are:

- Source
- Build
- Test
- Deploy
- Approval
- Invoke
- Compute

Pipelines also include *transitions*, which allow the transition of artifacts from one stage to the next in a pipeline after the actions in one stage complete.

You can work with transitions by calling:

- `disable_stage_transition`, which prevents artifacts from transitioning to the next stage in a pipeline.
- `enable_stage_transition`, which enables transition of artifacts between stages in a pipeline.

Using the API to integrate with CodePipeline

For third-party integrators or developers who want to create their own integrations with CodePipeline, the expected sequence varies from the standard API user. To integrate with CodePipeline, developers need to work with the following items:

Jobs, which are instances of an action. For example, a job for a source action might import a revision of an artifact from a source.

You can work with jobs by calling:

- `acknowledge_job`, which confirms whether a job worker has received the specified job.
- `get_job_details`, which returns the details of a job.
- `poll_for_jobs`, which determines whether there are any jobs to act on.
- `put_job_failure_result`, which provides details of a job failure.
- `put_job_success_result`, which provides details of a job success.

Third party jobs, which are instances of an action created by a partner action and integrated into CodePipeline. Partner actions are created by members of the Amazon Web Services Partner Network.

You can work with third party jobs by calling:

- acknowledge_third_party_job, which confirms whether a job worker has received the specified job.
- get_third_party_job_details, which requests the details of a job for a partner action.
- poll_for_third_party_jobs, which determines whether there are any jobs to act on.
- put_third_party_job_failure_result, which provides details of a job failure.
- put_third_party_job_success_result, which provides details of a job success.

Usage

```
codepipeline(
  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 <- codepipeline(
  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

acknowledge_job	Returns information about a specified job and whether that job has been received by t
acknowledge_third_party_job	Confirms a job worker has received the specified job
create_custom_action_type	Creates a new custom action that can be used in all pipelines associated with the Ama
create_pipeline	Creates a pipeline
delete_custom_action_type	Marks a custom action as deleted
delete_pipeline	Deletes the specified pipeline
delete_webhook	Deletes a previously created webhook by name
deregister_webhook_with_third_party	Removes the connection between the webhook that was created by CodePipeline and
disable_stage_transition	Prevents artifacts in a pipeline from transitioning to the next stage in the pipeline
enable_stage_transition	Enables artifacts in a pipeline to transition to a stage in a pipeline
get_action_type	Returns information about an action type created for an external provider, where the a
get_job_details	Returns information about a job
get_pipeline	Returns the metadata, structure, stages, and actions of a pipeline
get_pipeline_execution	Returns information about an execution of a pipeline, including details about artifacts
get_pipeline_state	Returns information about the state of a pipeline, including the stages and actions
get_third_party_job_details	Requests the details of a job for a third party action
list_action_executions	Lists the action executions that have occurred in a pipeline
list_action_types	Gets a summary of all CodePipeline action types associated with your account
list_pipeline_executions	Gets a summary of the most recent executions for a pipeline
list_pipelines	Gets a summary of all of the pipelines associated with your account
list_rule_executions	Lists the rule executions that have occurred in a pipeline configured for conditions wi
list_rule_types	Lists the rules for the condition
list_tags_for_resource	Gets the set of key-value pairs (metadata) that are used to manage the resource
list_webhooks	Gets a listing of all the webhooks in this Amazon Web Services Region for this accou
override_stage_condition	Used to override a stage condition
poll_for_jobs	Returns information about any jobs for CodePipeline to act on
poll_for_third_party_jobs	Determines whether there are any third party jobs for a job worker to act on
put_action_revision	Provides information to CodePipeline about new revisions to a source
put_approval_result	Provides the response to a manual approval request to CodePipeline
put_job_failure_result	Represents the failure of a job as returned to the pipeline by a job worker
put_job_success_result	Represents the success of a job as returned to the pipeline by a job worker
put_third_party_job_failure_result	Represents the failure of a third party job as returned to the pipeline by a job worker
put_third_party_job_success_result	Represents the success of a third party job as returned to the pipeline by a job worker
put_webhook	Defines a webhook and returns a unique webhook URL generated by CodePipeline
register_webhook_with_third_party	Configures a connection between the webhook that was created and the external tool
retry_stage_execution	You can retry a stage that has failed without having to run a pipeline again from the b
rollback_stage	Rolls back a stage execution
start_pipeline_execution	Starts the specified pipeline
stop_pipeline_execution	Stops the specified pipeline execution
tag_resource	Adds to or modifies the tags of the given resource
untag_resource	Removes tags from an Amazon Web Services resource
update_action_type	Updates an action type that was created with any supported integration model, where
update_pipeline	Updates a specified pipeline with edits or changes to its structure

Examples

```
## Not run:
svc <- codepipeline()
svc$acknowledge_job(
  Foo = 123
)

## End(Not run)
```

`codestarconnections` *AWS CodeStar connections*

Description

AWS CodeStar Connections

This Amazon Web Services CodeStar Connections API Reference provides descriptions and usage examples of the operations and data types for the Amazon Web Services CodeStar Connections API. You can use the connections API to work with connections and installations.

Connections are configurations that you use to connect Amazon Web Services resources to external code repositories. Each connection is a resource that can be given to services such as CodePipeline to connect to a third-party repository such as Bitbucket. For example, you can add the connection in CodePipeline so that it triggers your pipeline when a code change is made to your third-party code repository. Each connection is named and associated with a unique ARN that is used to reference the connection.

When you create a connection, the console initiates a third-party connection handshake. *Installations* are the apps that are used to conduct this handshake. For example, the installation for the Bitbucket provider type is the Bitbucket app. When you create a connection, you can choose an existing installation or create one.

When you want to create a connection to an installed provider type such as GitHub Enterprise Server, you create a *host* for your connections.

You can work with connections by calling:

- `create_connection`, which creates a uniquely named connection that can be referenced by services such as CodePipeline.
- `delete_connection`, which deletes the specified connection.
- `get_connection`, which returns information about the connection, including the connection status.
- `list_connections`, which lists the connections associated with your account.

You can work with hosts by calling:

- `create_host`, which creates a host that represents the infrastructure where your provider is installed.
- `delete_host`, which deletes the specified host.

- get_host, which returns information about the host, including the setup status.
- list_hosts, which lists the hosts associated with your account.

You can work with tags in Amazon Web Services CodeStar Connections by calling the following:

- list_tags_for_resource, which gets information about Amazon Web Services tags for a specified Amazon Resource Name (ARN) in Amazon Web Services CodeStar Connections.
- tag_resource, which adds or updates tags for a resource in Amazon Web Services CodeStar Connections.
- untag_resource, which removes tags for a resource in Amazon Web Services CodeStar Connections.

For information about how to use Amazon Web Services CodeStar Connections, see the Developer Tools User Guide.

Usage

```
codestarconnections(
  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 <- codestarconnections(
  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_connection	Creates a connection that can then be given to other Amazon Web Services services like Co
create_host	Creates a resource that represents the infrastructure where a third-party provider is installed
create_repository_link	Creates a link to a specified external Git repository
create_sync_configuration	Creates a sync configuration which allows Amazon Web Services to sync content from a Gi
delete_connection	The connection to be deleted
delete_host	The host to be deleted
delete_repository_link	Deletes the association between your connection and a specified external Git repository
delete_sync_configuration	Deletes the sync configuration for a specified repository and connection
get_connection	Returns the connection ARN and details such as status, owner, and provider type
get_host	Returns the host ARN and details such as status, provider type, endpoint, and, if applicable,
get_repository_link	Returns details about a repository link
get_repository_sync_status	Returns details about the sync status for a repository
get_resource_sync_status	Returns the status of the sync with the Git repository for a specific Amazon Web Services r
get_sync_blocker_summary	Returns a list of the most recent sync blockers
get_sync_configuration	Returns details about a sync configuration, including the sync type and resource name
list_connections	Lists the connections associated with your account
list_hosts	Lists the hosts associated with your account
list_repository_links	Lists the repository links created for connections in your account
list_repository_sync_definitions	Lists the repository sync definitions for repository links in your account
list_sync_configurations	Returns a list of sync configurations for a specified repository
list_tags_for_resource	Gets the set of key-value pairs (metadata) that are used to manage the resource
tag_resource	Adds to or modifies the tags of the given resource
untag_resource	Removes tags from an Amazon Web Services resource
update_host	Updates a specified host with the provided configurations
update_repository_link	Updates the association between your connection and a specified external Git repository
update_sync_blocker	Allows you to update the status of a sync blocker, resolving the blocker and allowing synci
update_sync_configuration	Updates the sync configuration for your connection and a specified external Git repository

Examples

```
## Not run:
svc <- codestarconnections()
svc$create_connection(
  Foo = 123
)

## End(Not run)
```

`codestarnotifications` *AWS CodeStar Notifications*

Description

This AWS CodeStar Notifications API Reference provides descriptions and usage examples of the operations and data types for the AWS CodeStar Notifications API. You can use the AWS CodeStar Notifications API to work with the following objects:

Notification rules, by calling the following:

- create_notification_rule, which creates a notification rule for a resource in your account.
- delete_notification_rule, which deletes a notification rule.
- describe_notification_rule, which provides information about a notification rule.
- list_notification_rules, which lists the notification rules associated with your account.
- update_notification_rule, which changes the name, events, or targets associated with a notification rule.
- subscribe, which subscribes a target to a notification rule.
- unsubscribe, which removes a target from a notification rule.

Targets, by calling the following:

- delete_target, which removes a notification rule target from a notification rule.
- list_targets, which lists the targets associated with a notification rule.

Events, by calling the following:

- list_event_types, which lists the event types you can include in a notification rule.

Tags, by calling the following:

- list_tags_for_resource, which lists the tags already associated with a notification rule in your account.
- tag_resource, which associates a tag you provide with a notification rule in your account.
- untag_resource, which removes a tag from a notification rule in your account.

For information about how to use AWS CodeStar Notifications, see the Amazon Web Services Developer Tools Console User Guide.

Usage

```
codestarnotifications(
  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 <- codestarnotifications(
  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_notification_rule	Creates a notification rule for a resource
delete_notification_rule	Deletes a notification rule for a resource
delete_target	Deletes a specified target for notifications
describe_notification_rule	Returns information about a specified notification rule
list_event_types	Returns information about the event types available for configuring notifications
list_notification_rules	Returns a list of the notification rules for an Amazon Web Services account
list_tags_for_resource	Returns a list of the tags associated with a notification rule
list_targets	Returns a list of the notification rule targets for an Amazon Web Services account
subscribe	Creates an association between a notification rule and an Chatbot topic or Chatbot client so that tl
tag_resource	Associates a set of provided tags with a notification rule
unsubscribe	Removes an association between a notification rule and an Chatbot topic so that subscribers to tha
untag_resource	Removes the association between one or more provided tags and a notification rule
update_notification_rule	Updates a notification rule for a resource

Examples

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

```
svc$create_notification_rule(
  Foo = 123
)

## End(Not run)
```

---

devopsguru *Amazon DevOps Guru*

Description

Amazon DevOps Guru is a fully managed service that helps you identify anomalous behavior in business critical operational applications. You specify the Amazon Web Services resources that you want DevOps Guru to cover, then the Amazon CloudWatch metrics and Amazon Web Services CloudTrail events related to those resources are analyzed. When anomalous behavior is detected, DevOps Guru creates an *insight* that includes recommendations, related events, and related metrics that can help you improve your operational applications. For more information, see What is Amazon DevOps Guru.

You can specify 1 or 2 Amazon Simple Notification Service topics so you are notified every time a new insight is created. You can also enable DevOps Guru to generate an OpsItem in Amazon Web Services Systems Manager for each insight to help you manage and track your work addressing insights.

To learn about the DevOps Guru workflow, see How DevOps Guru works. To learn about DevOps Guru concepts, see Concepts in DevOps Guru.

Usage

```
devopsguru(
  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 <- devopsguru(
  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_notification_channel	Adds a notification channel to DevOps Guru
delete_insight	Deletes the insight along with the associated anomalies, events and recon
describe_account_health	Returns the number of open reactive insights, the number of open proacti
describe_account_overview	For the time range passed in, returns the number of open reactive insight
describe_anomaly	Returns details about an anomaly that you specify using its ID
describe_event_sources_config	Returns the integration status of services that are integrated with DevOps
describe_feedback	Returns the most recent feedback submitted in the current Amazon Web S
describe_insight	Returns details about an insight that you specify using its ID
describe_organization_health	Returns active insights, predictive insights, and resource hours analyzed i
describe_organization_overview	Returns an overview of your organization's history based on the specified
describe_organization_resource_collection_health	Provides an overview of your system's health
describe_resource_collection_health	Returns the number of open proactive insights, open reactive insights, and
describe_service_integration	Returns the integration status of services that are integrated with DevOps
get_cost_estimation	Returns an estimate of the monthly cost for DevOps Guru to analyze you
get_resource_collection	Returns lists Amazon Web Services resources that are of the specified res
list_anomalies_for_insight	Returns a list of the anomalies that belong to an insight that you specify u
list_anomalous_log_groups	Returns the list of log groups that contain log anomalies
list_events	Returns a list of the events emitted by the resources that are evaluated by
list_insights	Returns a list of insights in your Amazon Web Services account
list_monitored_resources	Returns the list of all log groups that are being monitored and tagged by l
list_notification_channels	Returns a list of notification channels configured for DevOps Guru
list_organization_insights	Returns a list of insights associated with the account or OU Id
list_recommendations	Returns a list of a specified insight's recommendations
put_feedback	Collects customer feedback about the specified insight
remove_notification_channel	Removes a notification channel from DevOps Guru
search_insights	Returns a list of insights in your Amazon Web Services account
search_organization_insights	Returns a list of insights in your organization
start_cost_estimation	Starts the creation of an estimate of the monthly cost to analyze your Am
update_event_sources_config	Enables or disables integration with a service that can be integrated with
update_resource_collection	Updates the collection of resources that DevOps Guru analyzes
update_service_integration	Enables or disables integration with a service that can be integrated with

Examples

```
## Not run:
svc <- devopsguru()
svc$add_notification_channel(
  Foo = 123
)

## End(Not run)
```

`drs` *Elastic Disaster Recovery Service*

Description

AWS Elastic Disaster Recovery Service.

Usage

```
drs(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 <- drs(
  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_source_network_stack	Associate a Source Network to an existing CloudFormation Stack and modify
create_extended_source_server	Create an extended source server in the target Account based on the source se
create_launch_configuration_template	Creates a new Launch Configuration Template
create_replication_configuration_template	Creates a new ReplicationConfigurationTemplate
create_source_network	Create a new Source Network resource for a provided VPC ID
delete_job	Deletes a single Job by ID
delete_launch_action	Deletes a resource launch action
delete_launch_configuration_template	Deletes a single Launch Configuration Template by ID
delete_recovery_instance	Deletes a single Recovery Instance by ID
delete_replication_configuration_template	Deletes a single Replication Configuration Template by ID
delete_source_network	Delete Source Network resource
delete_source_server	Deletes a single Source Server by ID
describe_job_log_items	Retrieves a detailed Job log with pagination
describe_jobs	Returns a list of Jobs
describe_launch_configuration_templates	Lists all Launch Configuration Templates, filtered by Launch Configuration T
describe_recovery_instances	Lists all Recovery Instances or multiple Recovery Instances by ID
describe_recovery_snapshots	Lists all Recovery Snapshots for a single Source Server
describe_replication_configuration_templates	Lists all ReplicationConfigurationTemplates, filtered by Source Server IDs
describe_source_networks	Lists all Source Networks or multiple Source Networks filtered by ID
describe_source_servers	Lists all Source Servers or multiple Source Servers filtered by ID
disconnect_recovery_instance	Disconnect a Recovery Instance from Elastic Disaster Recovery
disconnect_source_server	Disconnects a specific Source Server from Elastic Disaster Recovery
export_source_network_cfn_template	Export the Source Network CloudFormation template to an S3 bucket
get_failback_replication_configuration	Lists all Failback ReplicationConfigurations, filtered by Recovery Instance ID
get_launch_configuration	Gets a LaunchConfiguration, filtered by Source Server IDs
get_replication_configuration	Gets a ReplicationConfiguration, filtered by Source Server ID
initialize_service	Initialize Elastic Disaster Recovery
list_extensible_source_servers	Returns a list of source servers on a staging account that are extensible, which
list_launch_actions	Lists resource launch actions
list_staging_accounts	Returns an array of staging accounts for existing extended source servers
list_tags_for_resource	List all tags for your Elastic Disaster Recovery resources
put_launch_action	Puts a resource launch action
retry_data_replication	WARNING: RetryDataReplication is deprecated
reverse_replication	Start replication to origin / target region - applies only to protected instances t
start_failback_launch	Initiates a Job for launching the machine that is being failed back to from the
start_recovery	Launches Recovery Instances for the specified Source Servers
start_replication	Starts replication for a stopped Source Server
start_source_network_recovery	Deploy VPC for the specified Source Network and modify launch templates t
start_source_network_replication	Starts replication for a Source Network
stop_failback	Stops the failback process for a specified Recovery Instance
stop_replication	Stops replication for a Source Server
stop_source_network_replication	Stops replication for a Source Network
tag_resource	Adds or overwrites only the specified tags for the specified Elastic Disaster Re
terminate_recovery_instances	Initiates a Job for terminating the EC2 resources associated with the specified
untag_resource	Deletes the specified set of tags from the specified set of Elastic Disaster Reco
update_failback_replication_configuration	Allows you to update the failback replication configuration of a Recovery Inst

update_launch_configuration	Updates a LaunchConfiguration by Source Server ID
update_launch_configuration_template	Updates an existing Launch Configuration Template by ID
update_replication_configuration	Allows you to update a ReplicationConfiguration by Source Server ID
update_replication_configuration_template	Updates a ReplicationConfigurationTemplate by ID

Examples

```
## Not run:
svc <- drs()
svc$associate_source_network_stack(
  Foo = 123
)

## End(Not run)
```

`fis` *AWS Fault Injection Simulator*

Description

Amazon Web Services Fault Injection Service is a managed service that enables you to perform fault injection experiments on your Amazon Web Services workloads. For more information, see the Fault Injection Service User Guide.

Usage

```
fis(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 <- fis(
  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_experiment_template	Creates an experiment template
create_target_account_configuration	Creates a target account configuration for the experiment template
delete_experiment_template	Deletes the specified experiment template
delete_target_account_configuration	Deletes the specified target account configuration of the experiment template
get_action	Gets information about the specified FIS action
get_experiment	Gets information about the specified experiment
get_experiment_target_account_configuration	Gets information about the specified target account configuration of the expe
get_experiment_template	Gets information about the specified experiment template
get_safety_lever	Gets information about the specified safety lever
get_target_account_configuration	Gets information about the specified target account configuration of the expe
get_target_resource_type	Gets information about the specified resource type
list_actions	Lists the available FIS actions
list_experiment_resolved_targets	Lists the resolved targets information of the specified experiment
list_experiments	Lists your experiments
list_experiment_target_account_configurations	Lists the target account configurations of the specified experiment
list_experiment_templates	Lists your experiment templates
list_tags_for_resource	Lists the tags for the specified resource
list_target_account_configurations	Lists the target account configurations of the specified experiment template
list_target_resource_types	Lists the target resource types
start_experiment	Starts running an experiment from the specified experiment template
stop_experiment	Stops the specified experiment
tag_resource	Applies the specified tags to the specified resource
untag_resource	Removes the specified tags from the specified resource
update_experiment_template	Updates the specified experiment template
update_safety_lever_state	Updates the specified safety lever state
update_target_account_configuration	Updates the target account configuration for the specified experiment templa

Examples

```
## Not run:
svc <- fis()
svc$create_experiment_template(
  Foo = 123
)
```

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

`wellarchitected` *AWS Well-Architected Tool*

Description

Well-Architected Tool

This is the *Well-Architected Tool API Reference*. The WA Tool API provides programmatic access to the Well-Architected Tool in the Amazon Web Services Management Console. For information about the Well-Architected Tool, see the Well-Architected Tool User Guide.

Usage

```
wellarchitected(
  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 <- wellarchitected(
  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_lenses	Associate a lens to a workload
associate_profiles	Associate a profile with a workload
create_lens_share	Create a lens share
create_lens_version	Create a new lens version
create_milestone	Create a milestone for an existing workload
create_profile	Create a profile
create_profile_share	Create a profile share
create_review_template	Create a review template
create_template_share	Create a review template share
create_workload	Create a new workload
create_workload_share	Create a workload share
delete_lens	Delete an existing lens
delete_lens_share	Delete a lens share
delete_profile	Delete a profile
delete_profile_share	Delete a profile share
delete_review_template	Delete a review template
delete_template_share	Delete a review template share
delete_workload	Delete an existing workload
delete_workload_share	Delete a workload share
disassociate_lenses	Disassociate a lens from a workload
disassociate_profiles	Disassociate a profile from a workload
export_lens	Export an existing lens
get_answer	Get the answer to a specific question in a workload review
get_consolidated_report	Get a consolidated report of your workloads
get_global_settings	Global settings for all workloads
get_lens	Get an existing lens
get_lens_review	Get lens review
get_lens_review_report	Get lens review report
get_lens_version_difference	Get lens version differences
get_milestone	Get a milestone for an existing workload
get_profile	Get profile information
get_profile_template	Get profile template
get_review_template	Get review template
get_review_template_answer	Get review template answer
get_review_template_lens_review	Get a lens review associated with a review template
get_workload	Get an existing workload
import_lens	Import a new custom lens or update an existing custom lens
list_answers	List of answers for a particular workload and lens
list_check_details	List of Trusted Advisor check details by account related to the workload
list_check_summaries	List of Trusted Advisor checks summarized for all accounts related to the workload
list_lenses	List the available lenses
list_lens_review_improvements	List the improvements of a particular lens review
list_lens_reviews	List lens reviews for a particular workload
list_lens_shares	List the lens shares associated with the lens
list_milestones	List all milestones for an existing workload
list_notifications	List lens notifications

list_profile_notifications	List profile notifications
list_profiles	List profiles
list_profile_shares	List profile shares
list_review_template_answers	List the answers of a review template
list_review_templates	List review templates
list_share_invitations	List the share invitations
list_tags_for_resource	List the tags for a resource
list_template_shares	List review template shares
list_workloads	Paginated list of workloads
list_workload_shares	List the workload shares associated with the workload
tag_resource	Adds one or more tags to the specified resource
untag_resource	Deletes specified tags from a resource
update_answer	Update the answer to a specific question in a workload review
update_global_settings	Update whether the Amazon Web Services account is opted into organization sharin
update_integration	Update integration features
update_lens_review	Update lens review for a particular workload
update_profile	Update a profile
update_review_template	Update a review template
update_review_template_answer	Update a review template answer
update_review_template_lens_review	Update a lens review associated with a review template
update_share_invitation	Update a workload or custom lens share invitation
update_workload	Update an existing workload
update_workload_share	Update a workload share
upgrade_lens_review	Upgrade lens review for a particular workload
upgrade_profile_version	Upgrade a profile
upgrade_review_template_lens_review	Upgrade the lens review of a review template

Examples

```
## Not run:
svc <- wellarchitected()
svc$associate_lenses(
  Foo = 123
)

## End(Not run)
```

`xray` *AWS X-Ray*

Description

Amazon Web Services X-Ray provides APIs for managing debug traces and retrieving service maps and other data created by processing those traces.

Usage

```
xray(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 <- xray(
  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_traces	You cannot find traces through this API if Transaction Search is enabled since trace is no
cancel_trace_retrieval	Cancels an ongoing trace retrieval job initiated by StartTraceRetrieval using the provided
create_group	Creates a group resource with a name and a filter expression
create_sampling_rule	Creates a rule to control sampling behavior for instrumented applications
delete_group	Deletes a group resource
delete_resource_policy	Deletes a resource policy from the target Amazon Web Services account
delete_sampling_rule	Deletes a sampling rule
get_encryption_config	Retrieves the current encryption configuration for X-Ray data
get_group	Retrieves group resource details
get_groups	Retrieves all active group details
get_indexing_rules	Retrieves all indexing rules
get_insight	Retrieves the summary information of an insight
get_insight_events	X-Ray reevaluates insights periodically until they're resolved, and records each intermed

get_insight_impact_graph	Retrieves a service graph structure filtered by the specified insight
get_insight_summaries	Retrieves the summaries of all insights in the specified group matching the provided filte
get_retrieved_traces_graph	Retrieves a service graph for traces based on the specified RetrievalToken from the Cloud
get_sampling_rules	Retrieves all sampling rules
get_sampling_statistic_summaries	Retrieves information about recent sampling results for all sampling rules
get_sampling_targets	Requests a sampling quota for rules that the service is using to sample requests
get_service_graph	Retrieves a document that describes services that process incoming requests, and downst
get_time_series_service_statistics	Get an aggregation of service statistics defined by a specific time range
get_trace_graph	Retrieves a service graph for one or more specific trace IDs
get_trace_segment_destination	Retrieves the current destination of data sent to PutTraceSegments and OpenTelemetry A
get_trace_summaries	Retrieves IDs and annotations for traces available for a specified time frame using an opt
list_resource_policies	Returns the list of resource policies in the target Amazon Web Services account
list_retrieved_traces	Retrieves a list of traces for a given RetrievalToken from the CloudWatch log group gene
list_tags_for_resource	Returns a list of tags that are applied to the specified Amazon Web Services X-Ray group
put_encryption_config	Updates the encryption configuration for X-Ray data
put_resource_policy	Sets the resource policy to grant one or more Amazon Web Services services and accoun
put_telemetry_records	Used by the Amazon Web Services X-Ray daemon to upload telemetry
put_trace_segments	Uploads segment documents to Amazon Web Services X-Ray
start_trace_retrieval	Initiates a trace retrieval process using the specified time range and for the give trace IDs
tag_resource	Applies tags to an existing Amazon Web Services X-Ray group or sampling rule
untag_resource	Removes tags from an Amazon Web Services X-Ray group or sampling rule
update_group	Updates a group resource
update_indexing_rule	Modifies an indexing rule's configuration
update_sampling_rule	Modifies a sampling rule's configuration
update_trace_segment_destination	Modifies the destination of data sent to PutTraceSegments

Examples

```
## Not run:
svc <- xray()
svc$batch_get_traces(
  Foo = 123
)

## End(Not run)
```

Index