

Package ‘paws.networking’

May 10, 2024

Title 'Amazon Web Services' Networking & Content Delivery Services

Version 0.6.0

Description Interface to 'Amazon Web Services' networking and content delivery services, including 'Route 53' Domain Name System service, 'CloudFront' content delivery, load balancing, and more
<<https://aws.amazon.com/>>.

License Apache License (>= 2.0)

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

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

Imports paws.common (>= 0.6.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.1

Collate 'apigateway_service.R' 'apigateway_interfaces.R'
'apigateway_operations.R' 'apigatewaymanagementapi_service.R'
'apigatewaymanagementapi_interfaces.R'
'apigatewaymanagementapi_operations.R' 'apigatewayv2_service.R'
'apigatewayv2_interfaces.R' 'apigatewayv2_operations.R'
'appfabric_service.R' 'appfabric_interfaces.R'
'appfabric_operations.R' 'appmesh_service.R'
'appmesh_interfaces.R' 'appmesh_operations.R'
'arczonalshift_service.R' 'arczonalshift_interfaces.R'
'arczonalshift_operations.R' 'backupgateway_service.R'
'backupgateway_interfaces.R' 'backupgateway_operations.R'
'cloudfront_service.R' 'cloudfront_interfaces.R'
'cloudfront_operations.R' 'directconnect_service.R'
'directconnect_interfaces.R' 'directconnect_operations.R'
'elb_service.R' 'elb_interfaces.R' 'elb_operations.R'
'elbv2_service.R' 'elbv2_interfaces.R' 'elbv2_operations.R'
'globalaccelerator_service.R' 'globalaccelerator_interfaces.R'
'globalaccelerator_operations.R' 'networkfirewall_service.R'
'networkfirewall_interfaces.R' 'networkfirewall_operations.R'

'networkmanager_service.R' 'networkmanager_interfaces.R'
 'networkmanager_operations.R' 'reexports_paws.common.R'
 'route53_service.R' 'route53_interfaces.R'
 'route53_operations.R' 'route53domains_service.R'
 'route53domains_interfaces.R' 'route53domains_operations.R'
 'route53recoverycluster_service.R'
 'route53recoverycluster_interfaces.R'
 'route53recoverycluster_operations.R'
 'route53recoverycontrolconfig_service.R'
 'route53recoverycontrolconfig_interfaces.R'
 'route53recoverycontrolconfig_operations.R'
 'route53recoveryreadiness_service.R'
 'route53recoveryreadiness_interfaces.R'
 'route53recoveryreadiness_operations.R'
 'route53resolver_service.R' 'route53resolver_interfaces.R'
 'route53resolver_operations.R' 'servicediscovery_service.R'
 'servicediscovery_interfaces.R' 'servicediscovery_operations.R'
 'telconetworkbuilder_service.R'
 'telconetworkbuilder_interfaces.R'
 'telconetworkbuilder_operations.R' 'vpclattice_service.R'
 'vpclattice_interfaces.R' 'vpclattice_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 2024-05-10 16:50:02 UTC

R topics documented:

apigateway	3
apigatewaymanagementapi	8
apigatewayv2	11
appfabric	15
appmesh	18
arczonalshift	21
backupgateway	24
cloudfront	27
directconnect	31
elb	35
elbv2	38
globalaccelerator	42
networkfirewall	47
networkmanager	50

route53	54
route53domains	58
route53recoverycluster	61
route53recoverycontrolconfig	64
route53recoveryreadiness	67
route53resolver	70
servicediscovery	74
telconetworkbuilder	77
vpclattice	80

Index	84
--------------	-----------

apigateway	<i>Amazon API Gateway</i>
------------	---------------------------

Description

Amazon API Gateway helps developers deliver robust, secure, and scalable mobile and web application back ends. API Gateway allows developers to securely connect mobile and web applications to APIs that run on Lambda, Amazon EC2, or other publicly addressable web services that are hosted outside of AWS.

Usage

```
apigateway(
  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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- apigateway(
  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_api_key	Create an ApiKey resource
create_authorizer	Adds a new Authorizer resource to an existing RestApi resource
create_base_path_mapping	Creates a new BasePathMapping resource
create_deployment	Creates a Deployment resource, which makes a specified RestApi callable over the internet
create_documentation_part	Creates a documentation part
create_documentation_version	Creates a documentation version
create_domain_name	Creates a new domain name
create_model	Adds a new Model resource to an existing RestApi resource
create_request_validator	Creates a RequestValidator of a given RestApi
create_resource	Creates a Resource resource
create_rest_api	Creates a new RestApi resource
create_stage	Creates a new Stage resource that references a pre-existing Deployment for the API
create_usage_plan	Creates a usage plan with the throttle and quota limits, as well as the associated API stages,
create_usage_plan_key	Creates a usage plan key for adding an existing API key to a usage plan
create_vpc_link	Creates a VPC link, under the caller's account in a selected region, in an asynchronous oper
delete_api_key	Deletes the ApiKey resource
delete_authorizer	Deletes an existing Authorizer resource
delete_base_path_mapping	Deletes the BasePathMapping resource
delete_client_certificate	Deletes the ClientCertificate resource
delete_deployment	Deletes a Deployment resource
delete_documentation_part	Deletes a documentation part
delete_documentation_version	Deletes a documentation version
delete_domain_name	Deletes the DomainName resource
delete_gateway_response	Clears any customization of a GatewayResponse of a specified response type on the given R
delete_integration	Represents a delete integration
delete_integration_response	Represents a delete integration response
delete_method	Deletes an existing Method resource
delete_method_response	Deletes an existing MethodResponse resource
delete_model	Deletes a model
delete_request_validator	Deletes a RequestValidator of a given RestApi
delete_resource	Deletes a Resource resource
delete_rest_api	Deletes the specified API
delete_stage	Deletes a Stage resource
delete_usage_plan	Deletes a usage plan of a given plan Id
delete_usage_plan_key	Deletes a usage plan key and remove the underlying API key from the associated usage plan
delete_vpc_link	Deletes an existing VpcLink of a specified identifier

flush_stage_authorizers_cache	Flushes all authorizer cache entries on a stage
flush_stage_cache	Flushes a stage's cache
generate_client_certificate	Generates a ClientCertificate resource
get_account	Gets information about the current Account resource
get_api_key	Gets information about the current ApiKey resource
get_api_keys	Gets information about the current ApiKeys resource
get_authorizer	Describe an existing Authorizer resource
get_authorizers	Describe an existing Authorizers resource
get_base_path_mapping	Describe a BasePathMapping resource
get_base_path_mappings	Represents a collection of BasePathMapping resources
get_client_certificate	Gets information about the current ClientCertificate resource
get_client_certificates	Gets a collection of ClientCertificate resources
get_deployment	Gets information about a Deployment resource
get_deployments	Gets information about a Deployments collection
get_documentation_part	Gets a documentation part
get_documentation_parts	Gets documentation parts
get_documentation_version	Gets a documentation version
get_documentation_versions	Gets documentation versions
get_domain_name	Represents a domain name that is contained in a simpler, more intuitive URL that can be called
get_domain_names	Represents a collection of DomainName resources
get_export	Exports a deployed version of a RestApi in a specified format
get_gateway_response	Gets a GatewayResponse of a specified response type on the given RestApi
get_gateway_responses	Gets the GatewayResponses collection on the given RestApi
get_integration	Get the integration settings
get_integration_response	Represents a get integration response
get_method	Describe an existing Method resource
get_method_response	Describes a MethodResponse resource
get_model	Describes an existing model defined for a RestApi resource
get_models	Describes existing Models defined for a RestApi resource
get_model_template	Generates a sample mapping template that can be used to transform a payload into the structure
get_request_validator	Gets a RequestValidator of a given RestApi
get_request_validators	Gets the RequestValidators collection of a given RestApi
get_resource	Lists information about a resource
get_resources	Lists information about a collection of Resource resources
get_rest_api	Lists the RestApi resource in the collection
get_rest_apis	Lists the RestApis resources for your collection
get_sdk	Generates a client SDK for a RestApi and Stage
get_sdk_type	Gets an SDK type
get_sdk_types	Gets SDK types
get_stage	Gets information about a Stage resource
get_stages	Gets information about one or more Stage resources
get_tags	Gets the Tags collection for a given resource
get_usage	Gets the usage data of a usage plan in a specified time interval
get_usage_plan	Gets a usage plan of a given plan identifier
get_usage_plan_key	Gets a usage plan key of a given key identifier
get_usage_plan_keys	Gets all the usage plan keys representing the API keys added to a specified usage plan
get_usage_plans	Gets all the usage plans of the caller's account
get_vpc_link	Gets a specified VPC link under the caller's account in a region

get_vpc_links	Gets the VpcLinks collection under the caller's account in a selected region
import_api_keys	Import API keys from an external source, such as a CSV-formatted file
import_documentation_parts	Imports documentation parts
import_rest_api	A feature of the API Gateway control service for creating a new API from an external API d
put_gateway_response	Creates a customization of a GatewayResponse of a specified response type and status code
put_integration	Sets up a method's integration
put_integration_response	Represents a put integration
put_method	Add a method to an existing Resource resource
put_method_response	Adds a MethodResponse to an existing Method resource
put_rest_api	A feature of the API Gateway control service for updating an existing API with an input of c
tag_resource	Adds or updates a tag on a given resource
test_invoke_authorizer	Simulate the execution of an Authorizer in your RestApi with headers, parameters, and an in
test_invoke_method	Simulate the invocation of a Method in your RestApi with headers, parameters, and an incor
untag_resource	Removes a tag from a given resource
update_account	Changes information about the current Account resource
update_api_key	Changes information about an ApiKey resource
update_authorizer	Updates an existing Authorizer resource
update_base_path_mapping	Changes information about the BasePathMapping resource
update_client_certificate	Changes information about an ClientCertificate resource
update_deployment	Changes information about a Deployment resource
update_documentation_part	Updates a documentation part
update_documentation_version	Updates a documentation version
update_domain_name	Changes information about the DomainName resource
update_gateway_response	Updates a GatewayResponse of a specified response type on the given RestApi
update_integration	Represents an update integration
update_integration_response	Represents an update integration response
update_method	Updates an existing Method resource
update_method_response	Updates an existing MethodResponse resource
update_model	Changes information about a model
update_request_validator	Updates a RequestValidator of a given RestApi
update_resource	Changes information about a Resource resource
update_rest_api	Changes information about the specified API
update_stage	Changes information about a Stage resource
update_usage	Grants a temporary extension to the remaining quota of a usage plan associated with a speci
update_usage_plan	Updates a usage plan of a given plan Id
update_vpc_link	Updates an existing VpcLink of a specified identifier

Examples

```
## Not run:
svc <- apigateway()
svc$create_api_key(
  Foo = 123
)

## End(Not run)
```

 apigatewaymanagementapi

AmazonApiGatewayManagementApi

Description

The Amazon API Gateway Management API allows you to directly manage runtime aspects of your deployed APIs. To use it, you must explicitly set the SDK's endpoint to point to the endpoint of your deployed API. The endpoint will be of the form `https://{api-id}.execute-api.{region}.amazonaws.com/{stage}`, or will be the endpoint corresponding to your API's custom domain and base path, if applicable.

Usage

```
apigatewaymanagementapi(
  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-endpoint.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 <- apigatewaymanagementapi(
  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

<code>delete_connection</code>	Delete the connection with the provided id
<code>get_connection</code>	Get information about the connection with the provided id
<code>post_to_connection</code>	Sends the provided data to the specified connection

Examples

```
## Not run:
svc <- apigatewaymanagementapi()
svc$delete_connection(
  Foo = 123
)

## End(Not run)
```

apigatewayv2

AmazonApiGatewayV2

Description

Amazon API Gateway V2

Usage

```
apigatewayv2(
  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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- apigatewayv2(
  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_api	Creates an Api resource
create_api_mapping	Creates an API mapping
create_authorizer	Creates an Authorizer for an API
create_deployment	Creates a Deployment for an API
create_domain_name	Creates a domain name
create_integration	Creates an Integration
create_integration_response	Creates an IntegrationResponses
create_model	Creates a Model for an API
create_route	Creates a Route for an API
create_route_response	Creates a RouteResponse for a Route
create_stage	Creates a Stage for an API
create_vpc_link	Creates a VPC link
delete_access_log_settings	Deletes the AccessLogSettings for a Stage
delete_api	Deletes an Api resource
delete_api_mapping	Deletes an API mapping
delete_authorizer	Deletes an Authorizer
delete_cors_configuration	Deletes a CORS configuration
delete_deployment	Deletes a Deployment
delete_domain_name	Deletes a domain name
delete_integration	Deletes an Integration
delete_integration_response	Deletes an IntegrationResponses
delete_model	Deletes a Model
delete_route	Deletes a Route
delete_route_request_parameter	Deletes a route request parameter
delete_route_response	Deletes a RouteResponse
delete_route_settings	Deletes the RouteSettings for a stage
delete_stage	Deletes a Stage
delete_vpc_link	Deletes a VPC link
export_api	Export api
get_api	Gets an Api resource
get_api_mapping	Gets an API mapping
get_api_mappings	Gets API mappings
get_apis	Gets a collection of Api resources
get_authorizer	Gets an Authorizer

<code>get_authorizers</code>	Gets the Authorizers for an API
<code>get_deployment</code>	Gets a Deployment
<code>get_deployments</code>	Gets the Deployments for an API
<code>get_domain_name</code>	Gets a domain name
<code>get_domain_names</code>	Gets the domain names for an AWS account
<code>get_integration</code>	Gets an Integration
<code>get_integration_response</code>	Gets an IntegrationResponses
<code>get_integration_responses</code>	Gets the IntegrationResponses for an Integration
<code>get_integrations</code>	Gets the Integrations for an API
<code>get_model</code>	Gets a Model
<code>get_models</code>	Gets the Models for an API
<code>get_model_template</code>	Gets a model template
<code>get_route</code>	Gets a Route
<code>get_route_response</code>	Gets a RouteResponse
<code>get_route_responses</code>	Gets the RouteResponses for a Route
<code>get_routes</code>	Gets the Routes for an API
<code>get_stage</code>	Gets a Stage
<code>get_stages</code>	Gets the Stages for an API
<code>get_tags</code>	Gets a collection of Tag resources
<code>get_vpc_link</code>	Gets a VPC link
<code>get_vpc_links</code>	Gets a collection of VPC links
<code>import_api</code>	Imports an API
<code>reimport_api</code>	Puts an Api resource
<code>reset_authorizers_cache</code>	Resets all authorizer cache entries on a stage
<code>tag_resource</code>	Creates a new Tag resource to represent a tag
<code>untag_resource</code>	Deletes a Tag
<code>update_api</code>	Updates an Api resource
<code>update_api_mapping</code>	The API mapping
<code>update_authorizer</code>	Updates an Authorizer
<code>update_deployment</code>	Updates a Deployment
<code>update_domain_name</code>	Updates a domain name
<code>update_integration</code>	Updates an Integration
<code>update_integration_response</code>	Updates an IntegrationResponses
<code>update_model</code>	Updates a Model
<code>update_route</code>	Updates a Route
<code>update_route_response</code>	Updates a RouteResponse
<code>update_stage</code>	Updates a Stage
<code>update_vpc_link</code>	Updates a VPC link

Examples

```
## Not run:
svc <- apigatewayv2()
svc$create_api(
  Foo = 123
)
```

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

```
appfabric
```

```
AppFabric
```

Description

Amazon Web Services AppFabric quickly connects software as a service (SaaS) applications across your organization. This allows IT and security teams to easily manage and secure applications using a standard schema, and employees can complete everyday tasks faster using generative artificial intelligence (AI). You can use these APIs to complete AppFabric tasks, such as setting up audit log ingestions or viewing user access. For more information about AppFabric, including the required permissions to use the service, see the [Amazon Web Services AppFabric Administration Guide](#). For more information about using the Command Line Interface (CLI) to manage your AppFabric resources, see the [AppFabric section of the CLI Reference](#).

Usage

```
appfabric(
  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`.

	<ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- appfabric(
  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_user_access_tasks	Gets user access details in a batch request
connect_app_authorization	Establishes a connection between Amazon Web Services AppFabric and an application, which
create_app_authorization	Creates an app authorization within an app bundle, which allows AppFabric to connect to an a
create_app_bundle	Creates an app bundle to collect data from an application using AppFabric
create_ingestion	Creates a data ingestion for an application
create_ingestion_destination	Creates an ingestion destination, which specifies how an application's ingested data is process
delete_app_authorization	Deletes an app authorization
delete_app_bundle	Deletes an app bundle
delete_ingestion	Deletes an ingestion
delete_ingestion_destination	Deletes an ingestion destination
get_app_authorization	Returns information about an app authorization
get_app_bundle	Returns information about an app bundle
get_ingestion	Returns information about an ingestion
get_ingestion_destination	Returns information about an ingestion destination
list_app_authorizations	Returns a list of all app authorizations configured for an app bundle
list_app_bundles	Returns a list of app bundles
list_ingestion_destinations	Returns a list of all ingestion destinations configured for an ingestion
list_ingestions	Returns a list of all ingestions configured for an app bundle
list_tags_for_resource	Returns a list of tags for a resource
start_ingestion	Starts (enables) an ingestion, which collects data from an application
start_user_access_tasks	Starts the tasks to search user access status for a specific email address
stop_ingestion	Stops (disables) an ingestion
tag_resource	Assigns one or more tags (key-value pairs) to the specified resource
untag_resource	Removes a tag or tags from a resource
update_app_authorization	Updates an app authorization within an app bundle, which allows AppFabric to connect to an a
update_ingestion_destination	Updates an ingestion destination, which specifies how an application's ingested data is process

Examples

```

## Not run:
svc <- appfabric()
svc$batch_get_user_access_tasks(
  Foo = 123
)

## End(Not run)

```

appmesh

AWS App Mesh

Description

App Mesh is a service mesh based on the Envoy proxy that makes it easy to monitor and control microservices. App Mesh standardizes how your microservices communicate, giving you end-to-end visibility and helping to ensure high availability for your applications.

App Mesh gives you consistent visibility and network traffic controls for every microservice in an application. You can use App Mesh with Amazon Web Services Fargate, Amazon ECS, Amazon EKS, Kubernetes on Amazon Web Services, and Amazon EC2.

App Mesh supports microservice applications that use service discovery naming for their components. For more information about service discovery on Amazon ECS, see [Service Discovery](#) in the *Amazon Elastic Container Service Developer Guide*. Kubernetes kube-dns and coredns are supported. For more information, see [DNS for Services and Pods](#) in the Kubernetes documentation.

Usage

```
appmesh(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • 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 <- appmesh(
  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

<code>create_gateway_route</code>	Creates a gateway route
<code>create_mesh</code>	Creates a service mesh
<code>create_route</code>	Creates a route that is associated with a virtual router
<code>create_virtual_gateway</code>	Creates a virtual gateway
<code>create_virtual_node</code>	Creates a virtual node within a service mesh
<code>create_virtual_router</code>	Creates a virtual router within a service mesh
<code>create_virtual_service</code>	Creates a virtual service within a service mesh
<code>delete_gateway_route</code>	Deletes an existing gateway route
<code>delete_mesh</code>	Deletes an existing service mesh
<code>delete_route</code>	Deletes an existing route
<code>delete_virtual_gateway</code>	Deletes an existing virtual gateway
<code>delete_virtual_node</code>	Deletes an existing virtual node
<code>delete_virtual_router</code>	Deletes an existing virtual router
<code>delete_virtual_service</code>	Deletes an existing virtual service
<code>describe_gateway_route</code>	Describes an existing gateway route
<code>describe_mesh</code>	Describes an existing service mesh
<code>describe_route</code>	Describes an existing route
<code>describe_virtual_gateway</code>	Describes an existing virtual gateway
<code>describe_virtual_node</code>	Describes an existing virtual node
<code>describe_virtual_router</code>	Describes an existing virtual router
<code>describe_virtual_service</code>	Describes an existing virtual service
<code>list_gateway_routes</code>	Returns a list of existing gateway routes that are associated to a virtual gateway
<code>list_meshes</code>	Returns a list of existing service meshes
<code>list_routes</code>	Returns a list of existing routes in a service mesh
<code>list_tags_for_resource</code>	List the tags for an App Mesh resource
<code>list_virtual_gateways</code>	Returns a list of existing virtual gateways in a service mesh
<code>list_virtual_nodes</code>	Returns a list of existing virtual nodes
<code>list_virtual_routers</code>	Returns a list of existing virtual routers in a service mesh
<code>list_virtual_services</code>	Returns a list of existing virtual services in a service mesh
<code>tag_resource</code>	Associates the specified tags to a resource with the specified resourceArn
<code>untag_resource</code>	Deletes specified tags from a resource
<code>update_gateway_route</code>	Updates an existing gateway route that is associated to a specified virtual gateway in a service mesh
<code>update_mesh</code>	Updates an existing service mesh
<code>update_route</code>	Updates an existing route for a specified service mesh and virtual router
<code>update_virtual_gateway</code>	Updates an existing virtual gateway in a specified service mesh
<code>update_virtual_node</code>	Updates an existing virtual node in a specified service mesh
<code>update_virtual_router</code>	Updates an existing virtual router in a specified service mesh
<code>update_virtual_service</code>	Updates an existing virtual service in a specified service mesh

Examples

```
## Not run:
svc <- appmesh()
svc$create_gateway_route(
  Foo = 123
```

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

arczonalshift

AWS ARC - Zonal Shift

Description

Welcome to the Zonal Shift API Reference Guide for Amazon Route 53 Application Recovery Controller (Route 53 ARC).

You can start a zonal shift to move traffic for a load balancer resource away from an Availability Zone to help your application recover quickly from an impairment in an Availability Zone. For example, you can recover your application from a developer's bad code deployment or from an Amazon Web Services infrastructure failure in a single Availability Zone.

You can also configure zonal autoshift for a load balancer resource. Zonal autoshift is a capability in Route 53 ARC where Amazon Web Services shifts away application resource traffic from an Availability Zone, on your behalf, to help reduce your time to recovery during events. Amazon Web Services shifts away traffic for resources that are enabled for zonal autoshift whenever Amazon Web Services determines that there's an issue in the Availability Zone that could potentially affect customers.

To ensure that zonal autoshift is safe for your application, you must also configure practice runs when you enable zonal autoshift for a resource. Practice runs start weekly zonal shifts for a resource, to shift traffic for the resource out of an Availability Zone. Practice runs make sure, on a regular basis, that you have enough capacity in all the Availability Zones in an Amazon Web Services Region for your application to continue to operate normally when traffic for a resource is shifted away from one Availability Zone.

You must prescale resource capacity in all Availability Zones in the Region where your application is deployed, before you configure practice runs or enable zonal autoshift for a resource. You should not rely on scaling on demand when an autoshift or practice run starts.

For more information about using zonal shift and zonal autoshift, see the [Amazon Route 53 Application Recovery Controller Developer Guide](#).

Usage

```
arczonalshift(  
  config = list(),  
  credentials = list(),  
  endpoint = NULL,  
  region = NULL  
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- arczonalshift(
  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_zonal_shift	Cancel a zonal shift in Amazon Route 53 Application Recovery Controller
create_practice_run_configuration	A practice run configuration for zonal autoshift is required when you enable zonal auto
delete_practice_run_configuration	Deletes the practice run configuration for a resource
get_managed_resource	Get information about a resource that's been registered for zonal shifts with Amazon
list_autoshifts	Returns the active autoshifts for a specified resource
list_managed_resources	Lists all the resources in your Amazon Web Services account in this Amazon Web Se
list_zonal_shifts	Lists all active and completed zonal shifts in Amazon Route 53 Application Recovery
start_zonal_shift	You start a zonal shift to temporarily move load balancer traffic away from an Availab
update_practice_run_configuration	Update a practice run configuration to change one or more of the following: add, char
update_zonal_autoshift_configuration	You can update the zonal autoshift status for a resource, to enable or disable zonal aut
update_zonal_shift	Update an active zonal shift in Amazon Route 53 Application Recovery Controller in

Examples

```

## Not run:
svc <- arczonalshift()
svc$cancel_zonal_shift(
  Foo = 123
)

```

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

backupgateway	<i>AWS Backup Gateway</i>
---------------	---------------------------

Description

Backup gateway

Backup gateway connects Backup to your hypervisor, so you can create, store, and restore backups of your virtual machines (VMs) anywhere, whether on-premises or in the VMware Cloud (VMC) on Amazon Web Services.

Add on-premises resources by connecting to a hypervisor through a gateway. Backup will automatically discover the resources in your hypervisor.

Use Backup to assign virtual or on-premises resources to a backup plan, or run on-demand backups. Once you have backed up your resources, you can view them and restore them like any resource supported by Backup.

To download the Amazon Web Services software to get started, navigate to the Backup console, choose **Gateways**, then choose **Create gateway**.

Usage

```
backupgateway(
  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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- backupgateway(
  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_gateway_to_server	Associates a backup gateway with your server
create_gateway	Creates a backup gateway
delete_gateway	Deletes a backup gateway
delete_hypervisor	Deletes a hypervisor
disassociate_gateway_from_server	Disassociates a backup gateway from the specified server
get_bandwidth_rate_limit_schedule	Retrieves the bandwidth rate limit schedule for a specified gateway
get_gateway	By providing the ARN (Amazon Resource Name), this API returns the gateway
get_hypervisor	This action requests information about the specified hypervisor to which the gateway
get_hypervisor_property_mappings	This action retrieves the property mappings for the specified hypervisor
get_virtual_machine	By providing the ARN (Amazon Resource Name), this API returns the virtual machine
import_hypervisor_configuration	Connect to a hypervisor by importing its configuration
list_gateways	Lists backup gateways owned by an Amazon Web Services account in an Amazon Web Services
list_hypervisors	Lists your hypervisors
list_tags_for_resource	Lists the tags applied to the resource identified by its Amazon Resource Name (ARN)
list_virtual_machines	Lists your virtual machines
put_bandwidth_rate_limit_schedule	This action sets the bandwidth rate limit schedule for a specified gateway
put_hypervisor_property_mappings	This action sets the property mappings for the specified hypervisor
put_maintenance_start_time	Set the maintenance start time for a gateway
start_virtual_machines_metadata_sync	This action sends a request to sync metadata across the specified virtual machines
tag_resource	Tag the resource
test_hypervisor_configuration	Tests your hypervisor configuration to validate that backup gateway can connect with
untag_resource	Removes tags from the resource
update_gateway_information	Updates a gateway's name
update_gateway_software_now	Updates the gateway virtual machine (VM) software
update_hypervisor	Updates a hypervisor metadata, including its host, username, and password

Examples

```

## Not run:
svc <- backupgateway()
svc$associate_gateway_to_server(
  Foo = 123
)

```

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

cloudfront

Amazon CloudFront

Description

This is the *Amazon CloudFront API Reference*. This guide is for developers who need detailed information about CloudFront API actions, data types, and errors. For detailed information about CloudFront features, see the [Amazon CloudFront Developer Guide](#).

Usage

```
cloudfront(
  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-endpoint.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 <- cloudfront(
  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_alias	Associates an alias (also known as a CNAME or an alternate domain name) to a CloudFront distribution
copy_distribution	Creates a staging distribution using the configuration of the provided primary distribution
create_cache_policy	Creates a cache policy
create_cloud_front_origin_access_identity	Creates a new origin access identity
create_continuous_deployment_policy	Creates a continuous deployment policy that distributes traffic for a custom domain
create_distribution	Creates a CloudFront distribution
create_distribution_with_tags	Create a new distribution with tags
create_field_level_encryption_config	Create a new field-level encryption configuration
create_field_level_encryption_profile	Create a field-level encryption profile
create_function	Creates a CloudFront function
create_invalidation	Create a new invalidation
create_key_group	Creates a key group that you can use with CloudFront signed URLs and signed cookies
create_key_value_store	Specifies the key value store resource to add to your account
create_monitoring_subscription	Enables additional CloudWatch metrics for the specified CloudFront distribution
create_origin_access_control	Creates a new origin access control in CloudFront
create_origin_request_policy	Creates an origin request policy
create_public_key	Uploads a public key to CloudFront that you can use with signed URLs and signed cookies
create_realtime_log_config	Creates a real-time log configuration
create_response_headers_policy	Creates a response headers policy
create_streaming_distribution	This API is deprecated
create_streaming_distribution_with_tags	This API is deprecated
delete_cache_policy	Deletes a cache policy
delete_cloud_front_origin_access_identity	Delete an origin access identity
delete_continuous_deployment_policy	Deletes a continuous deployment policy
delete_distribution	Delete a distribution
delete_field_level_encryption_config	Remove a field-level encryption configuration
delete_field_level_encryption_profile	Remove a field-level encryption profile
delete_function	Deletes a CloudFront function
delete_key_group	Deletes a key group
delete_key_value_store	Specifies the key value store to delete
delete_monitoring_subscription	Disables additional CloudWatch metrics for the specified CloudFront distribution
delete_origin_access_control	Deletes a CloudFront origin access control
delete_origin_request_policy	Deletes an origin request policy
delete_public_key	Remove a public key you previously added to CloudFront
delete_realtime_log_config	Deletes a real-time log configuration
delete_response_headers_policy	Deletes a response headers policy
delete_streaming_distribution	Delete a streaming distribution
describe_function	Gets configuration information and metadata about a CloudFront function
describe_key_value_store	Specifies the key value store and its configuration
get_cache_policy	Gets a cache policy, including the following metadata:
get_cache_policy_config	Gets a cache policy configuration
get_cloud_front_origin_access_identity	Get the information about an origin access identity
get_cloud_front_origin_access_identity_config	Get the configuration information about an origin access identity
get_continuous_deployment_policy	Gets a continuous deployment policy, including metadata (the policy's id)
get_continuous_deployment_policy_config	Gets configuration information about a continuous deployment policy
get_distribution	Get the information about a distribution

<code>get_distribution_config</code>	Get the configuration information about a distribution
<code>get_field_level_encryption</code>	Get the field-level encryption configuration information
<code>get_field_level_encryption_config</code>	Get the field-level encryption configuration information
<code>get_field_level_encryption_profile</code>	Get the field-level encryption profile information
<code>get_field_level_encryption_profile_config</code>	Get the field-level encryption profile configuration information
<code>get_function</code>	Gets the code of a CloudFront function
<code>get_invalidation</code>	Get the information about an invalidation
<code>get_key_group</code>	Gets a key group, including the date and time when the key group was last used
<code>get_key_group_config</code>	Gets a key group configuration
<code>get_monitoring_subscription</code>	Gets information about whether additional CloudWatch metrics are enabled for a distribution
<code>get_origin_access_control</code>	Gets a CloudFront origin access control, including its unique identifier
<code>get_origin_access_control_config</code>	Gets a CloudFront origin access control configuration
<code>get_origin_request_policy</code>	Gets an origin request policy, including the following metadata:
<code>get_origin_request_policy_config</code>	Gets an origin request policy configuration
<code>get_public_key</code>	Gets a public key
<code>get_public_key_config</code>	Gets a public key configuration
<code>get_realtime_log_config</code>	Gets a real-time log configuration
<code>get_response_headers_policy</code>	Gets a response headers policy, including metadata (the policy's identifier)
<code>get_response_headers_policy_config</code>	Gets a response headers policy configuration
<code>get_streaming_distribution</code>	Gets information about a specified RTMP distribution, including the distribution ID
<code>get_streaming_distribution_config</code>	Get the configuration information about a streaming distribution
<code>list_cache_policies</code>	Gets a list of cache policies
<code>list_cloud_front_origin_access_identities</code>	Lists origin access identities
<code>list_conflicting_aliases</code>	Gets a list of aliases (also called CNAMEs or alternate domain names) that conflict with a distribution's domain name
<code>list_continuous_deployment_policies</code>	Gets a list of the continuous deployment policies in your Amazon Web Services account
<code>list_distributions</code>	List CloudFront distributions
<code>list_distributions_by_cache_policy_id</code>	Gets a list of distribution IDs for distributions that have a cache behavior with the specified cache policy ID
<code>list_distributions_by_key_group</code>	Gets a list of distribution IDs for distributions that have a cache behavior with the specified key group
<code>list_distributions_by_origin_request_policy_id</code>	Gets a list of distribution IDs for distributions that have a cache behavior with the specified origin request policy ID
<code>list_distributions_by_realtime_log_config</code>	Gets a list of distributions that have a cache behavior that's associated with the specified real-time log configuration
<code>list_distributions_by_response_headers_policy_id</code>	Gets a list of distribution IDs for distributions that have a cache behavior with the specified response headers policy ID
<code>list_distributions_by_web_acl_id</code>	List the distributions that are associated with a specified WAF web ACL
<code>list_field_level_encryption_configs</code>	List all field-level encryption configurations that have been created in CloudFront
<code>list_field_level_encryption_profiles</code>	Request a list of field-level encryption profiles that have been created in CloudFront
<code>list_functions</code>	Gets a list of all CloudFront functions in your Amazon Web Services account
<code>list_invalidations</code>	Lists invalidation batches
<code>list_key_groups</code>	Gets a list of key groups
<code>list_key_value_stores</code>	Specifies the key value stores to list
<code>list_origin_access_controls</code>	Gets the list of CloudFront origin access controls in this Amazon Web Services account
<code>list_origin_request_policies</code>	Gets a list of origin request policies
<code>list_public_keys</code>	List all public keys that have been added to CloudFront for this account
<code>list_realtime_log_configs</code>	Gets a list of real-time log configurations
<code>list_response_headers_policies</code>	Gets a list of response headers policies
<code>list_streaming_distributions</code>	List streaming distributions
<code>list_tags_for_resource</code>	List tags for a CloudFront resource
<code>publish_function</code>	Publishes a CloudFront function by copying the function code from the DDB resource
<code>tag_resource</code>	Add tags to a CloudFront resource
<code>test_function</code>	Tests a CloudFront function

<code>untag_resource</code>	Remove tags from a CloudFront resource
<code>update_cache_policy</code>	Updates a cache policy configuration
<code>update_cloud_front_origin_access_identity</code>	Update an origin access identity
<code>update_continuous_deployment_policy</code>	Updates a continuous deployment policy
<code>update_distribution</code>	Updates the configuration for a CloudFront distribution
<code>update_distribution_with_staging_config</code>	Copies the staging distribution's configuration to its corresponding primary distribution
<code>update_field_level_encryption_config</code>	Update a field-level encryption configuration
<code>update_field_level_encryption_profile</code>	Update a field-level encryption profile
<code>update_function</code>	Updates a CloudFront function
<code>update_key_group</code>	Updates a key group
<code>update_key_value_store</code>	Specifies the key value store to update
<code>update_origin_access_control</code>	Updates a CloudFront origin access control
<code>update_origin_request_policy</code>	Updates an origin request policy configuration
<code>update_public_key</code>	Update public key information
<code>update_realtime_log_config</code>	Updates a real-time log configuration
<code>update_response_headers_policy</code>	Updates a response headers policy
<code>update_streaming_distribution</code>	Update a streaming distribution

Examples

```
## Not run:
svc <- cloudfront()
# Use the following command to create a function.
svc$create_function(
  FunctionCode = "function-code.js",
  FunctionConfig = list(
    Comment = "my-function-comment",
    KeyValueStoreAssociations = list(
      Items = list(
        list(
          KeyValueStoreARN = "arn:aws:cloudfront::123456789012:key-value-st..."
        )
      ),
      Quantity = 1L
    ),
    Runtime = "cloudfront-js-2.0"
  ),
  Name = "my-function-name"
)

## End(Not run)
```

Description

Direct Connect links your internal network to an Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an Direct Connect router. With this connection in place, you can create virtual interfaces directly to the Amazon Web Services Cloud (for example, to Amazon EC2 and Amazon S3) and to Amazon VPC, bypassing Internet service providers in your network path. A connection provides access to all Amazon Web Services Regions except the China (Beijing) and (China) Ningxia Regions. Amazon Web Services resources in the China Regions can only be accessed through locations associated with those Regions.

Usage

```
directconnect(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- | | |
|-------------|---|
| config | <p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html |
| credentials | <p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- directconnect(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

[accept_direct_connect_gateway_association_proposal](#)

Accepts a proposal request to attach a virtual private gateway or tr

allocate_connection_on_interconnect	Deprecated
allocate_hosted_connection	Creates a hosted connection on the specified interconnect or a link aggregation group (LAG).
allocate_private_virtual_interface	Provisions a private virtual interface to be owned by the specified Amazon Web Services account.
allocate_public_virtual_interface	Provisions a public virtual interface to be owned by the specified Amazon Web Services account.
allocate_transit_virtual_interface	Provisions a transit virtual interface to be owned by the specified Amazon Web Services account.
associate_connection_with_lag	Associates an existing connection with a link aggregation group (LAG).
associate_hosted_connection	Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG).
associate_mac_sec_key	Associates a MAC Security (MACsec) Connection Key Name (CKN) with a connection.
associate_virtual_interface	Associates a virtual interface with a specified link aggregation group (LAG).
confirm_connection	Confirms the creation of the specified hosted connection on an interconnect.
confirm_customer_agreement	The confirmation of the terms of agreement when creating the connection.
confirm_private_virtual_interface	Accepts ownership of a private virtual interface created by another Amazon Web Services account.
confirm_public_virtual_interface	Accepts ownership of a public virtual interface created by another Amazon Web Services account.
confirm_transit_virtual_interface	Accepts ownership of a transit virtual interface created by another Amazon Web Services account.
create_bgp_peer	Creates a BGP peer on the specified virtual interface.
create_connection	Creates a connection between a customer network and a specific Direct Connect gateway.
create_direct_connect_gateway	Creates a Direct Connect gateway, which is an intermediate object between a customer network and a virtual private gateway.
create_direct_connect_gateway_association	Creates an association between a Direct Connect gateway and a virtual private gateway.
create_direct_connect_gateway_association_proposal	Creates a proposal to associate the specified virtual private gateway with the specified Direct Connect gateway.
create_interconnect	Creates an interconnect between an Amazon Web Services account and a Direct Connect Partner's network.
create_lag	Creates a link aggregation group (LAG) with the specified number of virtual interfaces.
create_private_virtual_interface	Creates a private virtual interface.
create_public_virtual_interface	Creates a public virtual interface.
create_transit_virtual_interface	Creates a transit virtual interface.
delete_bgp_peer	Deletes the specified BGP peer on the specified virtual interface.
delete_connection	Deletes the specified connection.
delete_direct_connect_gateway	Deletes the specified Direct Connect gateway.
delete_direct_connect_gateway_association	Deletes the association between the specified Direct Connect gateway and virtual private gateway.
delete_direct_connect_gateway_association_proposal	Deletes the association proposal request between the specified Direct Connect gateway and virtual private gateway.
delete_interconnect	Deletes the specified interconnect.
delete_lag	Deletes the specified link aggregation group (LAG).
delete_virtual_interface	Deletes a virtual interface.
describe_connection_loa	Deprecated
describe_connections	Displays the specified connection or all connections in this Region.
describe_connections_on_interconnect	Deprecated
describe_customer_metadata	Get and view a list of customer agreements, along with their signed versions.
describe_direct_connect_gateway_association_proposals	Describes one or more association proposals for connection between a Direct Connect gateway and a virtual private gateway.
describe_direct_connect_gateway_associations	Lists the associations between your Direct Connect gateways and virtual private gateways.
describe_direct_connect_gateway_attachments	Lists the attachments between your Direct Connect gateways and virtual private gateways.
describe_direct_connect_gateways	Lists all your Direct Connect gateways or only the specified Direct Connect gateway.
describe_hosted_connections	Lists the hosted connections that have been provisioned on the specified interconnect.
describe_interconnect_loa	Deprecated
describe_interconnects	Lists the interconnects owned by the Amazon Web Services account.
describe_lags	Describes all your link aggregation groups (LAG) or the specified LAG.
describe_loa	Gets the LOA-CFA for a connection, interconnect, or link aggregation group.
describe_locations	Lists the Direct Connect locations in the current Amazon Web Services Region.
describe_router_configuration	Details about the router.
describe_tags	Describes the tags associated with the specified Direct Connect resource.

describe_virtual_gateways	Lists the virtual private gateways owned by the Amazon Web Serv
describe_virtual_interfaces	Displays all virtual interfaces for an Amazon Web Services account
disassociate_connection_from_lag	Disassociates a connection from a link aggregation group (LAG)
disassociate_mac_sec_key	Removes the association between a MAC Security (MACsec) security
list_virtual_interface_test_history	Lists the virtual interface failover test history
start_bgp_failover_test	Starts the virtual interface failover test that verifies your configurat
stop_bgp_failover_test	Stops the virtual interface failover test
tag_resource	Adds the specified tags to the specified Direct Connect resource
untag_resource	Removes one or more tags from the specified Direct Connect resource
update_connection	Updates the Direct Connect dedicated connection configuration
update_direct_connect_gateway	Updates the name of a current Direct Connect gateway
update_direct_connect_gateway_association	Updates the specified attributes of the Direct Connect gateway associ
update_lag	Updates the attributes of the specified link aggregation group (LAG)
update_virtual_interface_attributes	Updates the specified attributes of the specified virtual private inter

Examples

```
## Not run:
svc <- directconnect()
svc$accept_direct_connect_gateway_association_proposal(
  Foo = 123
)

## End(Not run)
```

 elb

Elastic Load Balancing

Description

A load balancer can distribute incoming traffic across your EC2 instances. This enables you to increase the availability of your application. The load balancer also monitors the health of its registered instances and ensures that it routes traffic only to healthy instances. You configure your load balancer to accept incoming traffic by specifying one or more listeners, which are configured with a protocol and port number for connections from clients to the load balancer and a protocol and port number for connections from the load balancer to the instances.

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs. For more information, see the [Elastic Load Balancing User Guide](#).

This reference covers the 2012-06-01 API, which supports Classic Load Balancers. The 2015-12-01 API supports Application Load Balancers and Network Load Balancers.

To get started, create a load balancer with one or more listeners using [create_load_balancer](#). Register your instances with the load balancer using [register_instances_with_load_balancer](#).

All Elastic Load Balancing operations are *idempotent*, which means that they complete at most one time. If you repeat an operation, it succeeds with a 200 OK response code.

Usage

```
elb(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- elb(
  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](#)

Adds the specified tags to the specified load balancer

[apply_security_groups_to_load_balancer](#)

Associates one or more security groups with your load balancer in a virtual

[attach_load_balancer_to_subnets](#)

Adds one or more subnets to the set of configured subnets for the specified

[configure_health_check](#)

Specifies the health check settings to use when evaluating the health state of

[create_app_cookie_stickiness_policy](#)

Generates a stickiness policy with sticky session lifetimes that follow that of

[create_lb_cookie_stickiness_policy](#)

Generates a stickiness policy with sticky session lifetimes controlled by the

[create_load_balancer](#)

Creates a Classic Load Balancer

[create_load_balancer_listeners](#)

Creates one or more listeners for the specified load balancer

[create_load_balancer_policy](#)

Creates a policy with the specified attributes for the specified load balancer

[delete_load_balancer](#)

Deletes the specified load balancer

[delete_load_balancer_listeners](#)

Deletes the specified listeners from the specified load balancer

[delete_load_balancer_policy](#)

Deletes the specified policy from the specified load balancer

[deregister_instances_from_load_balancer](#)

Deregisters the specified instances from the specified load balancer

[describe_account_limits](#)

Describes the current Elastic Load Balancing resource limits for your AWS

describe_instance_health	Describes the state of the specified instances with respect to the specified load balancer
describe_load_balancer_attributes	Describes the attributes for the specified load balancer
describe_load_balancer_policies	Describes the specified policies
describe_load_balancer_policy_types	Describes the specified load balancer policy types or all load balancer policy types
describe_load_balancers	Describes the specified load balancers
describe_tags	Describes the tags associated with the specified load balancers
detach_load_balancer_from_subnets	Removes the specified subnets from the set of configured subnets for the specified load balancer
disable_availability_zones_for_load_balancer	Removes the specified Availability Zones from the set of Availability Zones for the specified load balancer
enable_availability_zones_for_load_balancer	Adds the specified Availability Zones to the set of Availability Zones for the specified load balancer
modify_load_balancer_attributes	Modifies the attributes of the specified load balancer
register_instances_with_load_balancer	Adds the specified instances to the specified load balancer
remove_tags	Removes one or more tags from the specified load balancer
set_load_balancer_listener_ssl_certificate	Sets the certificate that terminates the specified listener's SSL connections
set_load_balancer_policies_for_backend_server	Replaces the set of policies associated with the specified port on which the specified load balancer listens
set_load_balancer_policies_of_listener	Replaces the current set of policies for the specified load balancer port with the specified policies

Examples

```
## Not run:
svc <- elb()
# This example adds two tags to the specified load balancer.
svc$add_tags(
  LoadBalancerNames = list(
    "my-load-balancer"
  ),
  Tags = list(
    list(
      Key = "project",
      Value = "lima"
    ),
    list(
      Key = "department",
      Value = "digital-media"
    )
  )
)
## End(Not run)
```

Description

A load balancer distributes incoming traffic across targets, such as your EC2 instances. This enables you to increase the availability of your application. The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets. You configure your load balancer to accept incoming traffic by specifying one or more listeners, which are configured with a protocol and port number for connections from clients to the load balancer. You configure a target group with a protocol and port number for connections from the load balancer to the targets, and with health check settings to be used when checking the health status of the targets.

Elastic Load Balancing supports the following types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. This reference covers the following load balancer types:

- Application Load Balancer - Operates at the application layer (layer 7) and supports HTTP and HTTPS.
- Network Load Balancer - Operates at the transport layer (layer 4) and supports TCP, TLS, and UDP.
- Gateway Load Balancer - Operates at the network layer (layer 3).

For more information, see the [Elastic Load Balancing User Guide](#).

All Elastic Load Balancing operations are idempotent, which means that they complete at most one time. If you repeat an operation, it succeeds.

Usage

```
elbv2(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`.

	<ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- elbv2(
  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_listener_certificates	Adds the specified SSL server certificate to the certificate list for the specified HTTP
add_tags	Adds the specified tags to the specified Elastic Load Balancing resource
add_trust_store_revocations	Adds the specified revocation file to the specified trust store
create_listener	Creates a listener for the specified Application Load Balancer, Network Load Balancer,
create_load_balancer	Creates an Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
create_rule	Creates a rule for the specified listener
create_target_group	Creates a target group
create_trust_store	Creates a trust store
delete_listener	Deletes the specified listener
delete_load_balancer	Deletes the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
delete_rule	Deletes the specified rule
delete_target_group	Deletes the specified target group
delete_trust_store	Deletes a trust store
deregister_targets	Deregisters the specified targets from the specified target group
describe_account_limits	Describes the current Elastic Load Balancing resource limits for your Amazon Web Services
describe_listener_certificates	Describes the default certificate and the certificate list for the specified HTTPS or TLS
describe_listeners	Describes the specified listeners or the listeners for the specified Application Load Balancer,
describe_load_balancer_attributes	Describes the attributes for the specified Application Load Balancer, Network Load Balancer,
describe_load_balancers	Describes the specified load balancers or all of your load balancers
describe_rules	Describes the specified rules or the rules for the specified listener
describe_ssl_policies	Describes the specified policies or all policies used for SSL negotiation
describe_tags	Describes the tags for the specified Elastic Load Balancing resources
describe_target_group_attributes	Describes the attributes for the specified target group
describe_target_groups	Describes the specified target groups or all of your target groups
describe_target_health	Describes the health of the specified targets or all of your targets
describe_trust_store_associations	Describes all resources associated with the specified trust store
describe_trust_store_revocations	Describes the revocation files in use by the specified trust store arn, or revocation ID
describe_trust_stores	Describes all trust stores for a given account by trust store arn's or name
get_trust_store_ca_certificates_bundle	Retrieves the ca certificate bundle
get_trust_store_revocation_content	Retrieves the specified revocation file
modify_listener	Replaces the specified properties of the specified listener
modify_load_balancer_attributes	Modifies the specified attributes of the specified Application Load Balancer, Network Load Balancer,
modify_rule	Replaces the specified properties of the specified rule
modify_target_group	Modifies the health checks used when evaluating the health state of the targets in the
modify_target_group_attributes	Modifies the specified attributes of the specified target group
modify_trust_store	Update the ca certificate bundle for a given trust store
register_targets	Registers the specified targets with the specified target group
remove_listener_certificates	Removes the specified certificate from the certificate list for the specified HTTPS or TLS
remove_tags	Removes the specified tags from the specified Elastic Load Balancing resources
remove_trust_store_revocations	Removes the specified revocation file from the specified trust store

set_ip_address_type	Sets the type of IP addresses used by the subnets of the specified load balancer
set_rule_priorities	Sets the priorities of the specified rules
set_security_groups	Associates the specified security groups with the specified Application Load Balancer
set_subnets	Enables the Availability Zones for the specified public subnets for the specified Appl

Examples

```
## Not run:
svc <- elbv2()
# This example adds the specified tags to the specified load balancer.
svc$add_tags(
  ResourceArns = list(
    "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/m..."
  ),
  Tags = list(
    list(
      Key = "project",
      Value = "lima"
    ),
    list(
      Key = "department",
      Value = "digital-media"
    )
  )
)
## End(Not run)
```

globalaccelerator *AWS Global Accelerator*

Description

Global Accelerator

This is the *Global Accelerator API Reference*. This guide is for developers who need detailed information about Global Accelerator API actions, data types, and errors. For more information about Global Accelerator features, see the [Global Accelerator Developer Guide](#).

Global Accelerator is a service in which you create *accelerators* to improve the performance of your applications for local and global users. Depending on the type of accelerator you choose, you can gain additional benefits.

- By using a standard accelerator, you can improve availability of your internet applications that are used by a global audience. With a standard accelerator, Global Accelerator directs traffic to optimal endpoints over the Amazon Web Services global network.

- For other scenarios, you might choose a custom routing accelerator. With a custom routing accelerator, you can use application logic to directly map one or more users to a specific endpoint among many endpoints.

Global Accelerator is a global service that supports endpoints in multiple Amazon Web Services Regions but you must specify the US West (Oregon) Region to create, update, or otherwise work with accelerators. That is, for example, specify `--region us-west-2` on Amazon Web Services CLI commands.

By default, Global Accelerator provides you with static IP addresses that you associate with your accelerator. The static IP addresses are anycast from the Amazon Web Services edge network. For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. With a standard accelerator for IPv4, instead of using the addresses that Global Accelerator provides, you can configure these entry points to be IPv4 addresses from your own IP address ranges that you bring to Global Accelerator (BYOIP).

For a standard accelerator, they distribute incoming application traffic across multiple endpoint resources in multiple Amazon Web Services Regions, which increases the availability of your applications. Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that are located in one Amazon Web Services Region or multiple Amazon Web Services Regions. For custom routing accelerators, you map traffic that arrives to the static IP addresses to specific Amazon EC2 servers in endpoints that are virtual private cloud (VPC) subnets.

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies like tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see [Tag-based policies](#).

For standard accelerators, Global Accelerator uses the Amazon Web Services global network to route traffic to the optimal regional endpoint based on health, client location, and policies that you configure. The service reacts instantly to changes in health or configuration to ensure that internet traffic from clients is always directed to healthy endpoints.

For more information about understanding and using Global Accelerator, see the [Global Accelerator Developer Guide](#).

Usage

```
globalaccelerator(  
    config = list(),  
    credentials = list(),  
    endpoint = NULL,  
    region = NULL  
)
```

Arguments

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

	<ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- globalaccelerator(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
    ),
  ),
```

```

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

```

Operations

add_custom_routing_endpoints	Associate a virtual private cloud (VPC) subnet endpoint with your custom routing accelerator
add_endpoints	Add endpoints to an endpoint group
advertise_byoip_cidr	Advertises an IPv4 address range that is provisioned for use with your custom routing accelerator
allow_custom_routing_traffic	Specify the Amazon EC2 instance (destination) IP addresses and ports to allow traffic to a custom routing accelerator
create_accelerator	Create an accelerator
create_cross_account_attachment	Create a cross-account attachment in Global Accelerator
create_custom_routing_accelerator	Create a custom routing accelerator
create_custom_routing_endpoint_group	Create an endpoint group for the specified listener for a custom routing accelerator
create_custom_routing_listener	Create a listener to process inbound connections from clients to a custom routing accelerator
create_endpoint_group	Create an endpoint group for the specified listener
create_listener	Create a listener to process inbound connections from clients to an accelerator
delete_accelerator	Delete an accelerator
delete_cross_account_attachment	Delete a cross-account attachment
delete_custom_routing_accelerator	Delete a custom routing accelerator
delete_custom_routing_endpoint_group	Delete an endpoint group from a listener for a custom routing accelerator
delete_custom_routing_listener	Delete a listener for a custom routing accelerator
delete_endpoint_group	Delete an endpoint group from a listener
delete_listener	Delete a listener from an accelerator
deny_custom_routing_traffic	Specify the Amazon EC2 instance (destination) IP addresses and ports to deny traffic to a custom routing accelerator
deprovision_byoip_cidr	Releases the specified address range that you provisioned to use with your custom routing accelerator
describe_accelerator	Describe an accelerator
describe_accelerator_attributes	Describe the attributes of an accelerator
describe_cross_account_attachment	Gets configuration information about a cross-account attachment

<code>describe_custom_routing_accelerator</code>	Describe a custom routing accelerator
<code>describe_custom_routing_accelerator_attributes</code>	Describe the attributes of a custom routing accelerator
<code>describe_custom_routing_endpoint_group</code>	Describe an endpoint group for a custom routing accelerator
<code>describe_custom_routing_listener</code>	The description of a listener for a custom routing accelerator
<code>describe_endpoint_group</code>	Describe an endpoint group
<code>describe_listener</code>	Describe a listener
<code>list_accelerators</code>	List the accelerators for an Amazon Web Services account
<code>list_byoip_cidrs</code>	Lists the IP address ranges that were specified in calls to ProvisionByoip
<code>list_cross_account_attachments</code>	List the cross-account attachments that have been created in Global Acc
<code>list_cross_account_resource_accounts</code>	List the accounts that have cross-account resources
<code>list_cross_account_resources</code>	List the cross-account resources available to work with
<code>list_custom_routing_accelerators</code>	List the custom routing accelerators for an Amazon Web Services acco
<code>list_custom_routing_endpoint_groups</code>	List the endpoint groups that are associated with a listener for a custom
<code>list_custom_routing_listeners</code>	List the listeners for a custom routing accelerator
<code>list_custom_routing_port_mappings</code>	Provides a complete mapping from the public accelerator IP address an
<code>list_custom_routing_port_mappings_by_destination</code>	List the port mappings for a specific EC2 instance (destination) in a VP
<code>list_endpoint_groups</code>	List the endpoint groups that are associated with a listener
<code>list_listeners</code>	List the listeners for an accelerator
<code>list_tags_for_resource</code>	List all tags for an accelerator
<code>provision_byoip_cidr</code>	Provisions an IP address range to use with your Amazon Web Services
<code>remove_custom_routing_endpoints</code>	Remove endpoints from a custom routing accelerator
<code>remove_endpoints</code>	Remove endpoints from an endpoint group
<code>tag_resource</code>	Add tags to an accelerator resource
<code>untag_resource</code>	Remove tags from a Global Accelerator resource
<code>update_accelerator</code>	Update an accelerator to make changes, such as the following:
<code>update_accelerator_attributes</code>	Update the attributes for an accelerator
<code>update_cross_account_attachment</code>	Update a cross-account attachment to add or remove principals or resou
<code>update_custom_routing_accelerator</code>	Update a custom routing accelerator
<code>update_custom_routing_accelerator_attributes</code>	Update the attributes for a custom routing accelerator
<code>update_custom_routing_listener</code>	Update a listener for a custom routing accelerator
<code>update_endpoint_group</code>	Update an endpoint group
<code>update_listener</code>	Update a listener
<code>withdraw_byoip_cidr</code>	Stops advertising an address range that is provisioned as an address po

Examples

```
## Not run:
svc <- globalaccelerator()
svc$add_custom_routing_endpoints(
  Foo = 123
)

## End(Not run)
```

Description

This is the API Reference for Network Firewall. This guide is for developers who need detailed information about the Network Firewall API actions, data types, and errors.

- The REST API requires you to handle connection details, such as calculating signatures, handling request retries, and error handling. For general information about using the Amazon Web Services REST APIs, see [Amazon Web Services APIs](#).

To access Network Firewall using the REST API endpoint: `https://network-firewall.<region>.amazonaws.com`

- Alternatively, you can use one of the Amazon Web Services SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see [Amazon Web Services SDKs](#).
- For descriptions of Network Firewall features, including and step-by-step instructions on how to use them through the Network Firewall console, see the [Network Firewall Developer Guide](#).

Network Firewall is a stateful, managed, network firewall and intrusion detection and prevention service for Amazon Virtual Private Cloud (Amazon VPC). With Network Firewall, you can filter traffic at the perimeter of your VPC. This includes filtering traffic going to and coming from an internet gateway, NAT gateway, or over VPN or Direct Connect. Network Firewall uses rules that are compatible with Suricata, a free, open source network analysis and threat detection engine. Network Firewall supports Suricata version 6.0.9. For information about Suricata, see the [Suricata website](#).

You can use Network Firewall to monitor and protect your VPC traffic in a number of ways. The following are just a few examples:

- Allow domains or IP addresses for known Amazon Web Services service endpoints, such as Amazon S3, and block all other forms of traffic.
- Use custom lists of known bad domains to limit the types of domain names that your applications can access.
- Perform deep packet inspection on traffic entering or leaving your VPC.
- Use stateful protocol detection to filter protocols like HTTPS, regardless of the port used.

To enable Network Firewall for your VPCs, you perform steps in both Amazon VPC and in Network Firewall. For information about using Amazon VPC, see [Amazon VPC User Guide](#).

To start using Network Firewall, do the following:

1. (Optional) If you don't already have a VPC that you want to protect, create it in Amazon VPC.
2. In Amazon VPC, in each Availability Zone where you want to have a firewall endpoint, create a subnet for the sole use of Network Firewall.
3. In Network Firewall, create stateless and stateful rule groups, to define the components of the network traffic filtering behavior that you want your firewall to have.
4. In Network Firewall, create a firewall policy that uses your rule groups and specifies additional default traffic filtering behavior.

5. In Network Firewall, create a firewall and specify your new firewall policy and VPC subnets. Network Firewall creates a firewall endpoint in each subnet that you specify, with the behavior that's defined in the firewall policy.
6. In Amazon VPC, use ingress routing enhancements to route traffic through the new firewall endpoints.

Usage

```
networkfirewall(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- networkfirewall(
  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_firewall_policy](#)

[associate_subnets](#)

[create_firewall](#)

[create_firewall_policy](#)

[create_rule_group](#)

[create_tls_inspection_configuration](#)

[delete_firewall](#)

[delete_firewall_policy](#)

Associates a FirewallPolicy to a Firewall

Associates the specified subnets in the Amazon VPC to the firewall

Creates an Network Firewall Firewall and accompanying FirewallStatus for a VPC

Creates the firewall policy for the firewall according to the specifications

Creates the specified stateless or stateful rule group, which includes the rules for

Creates an Network Firewall TLS inspection configuration

Deletes the specified Firewall and its FirewallStatus

Deletes the specified FirewallPolicy

<code>delete_resource_policy</code>	Deletes a resource policy that you created in a PutResourcePolicy request
<code>delete_rule_group</code>	Deletes the specified RuleGroup
<code>delete_tls_inspection_configuration</code>	Deletes the specified TLSInspectionConfiguration
<code>describe_firewall</code>	Returns the data objects for the specified firewall
<code>describe_firewall_policy</code>	Returns the data objects for the specified firewall policy
<code>describe_logging_configuration</code>	Returns the logging configuration for the specified firewall
<code>describe_resource_policy</code>	Retrieves a resource policy that you created in a PutResourcePolicy request
<code>describe_rule_group</code>	Returns the data objects for the specified rule group
<code>describe_rule_group_metadata</code>	High-level information about a rule group, returned by operations like create and
<code>describe_tls_inspection_configuration</code>	Returns the data objects for the specified TLS inspection configuration
<code>disassociate_subnets</code>	Removes the specified subnet associations from the firewall
<code>list_firewall_policies</code>	Retrieves the metadata for the firewall policies that you have defined
<code>list_firewalls</code>	Retrieves the metadata for the firewalls that you have defined
<code>list_rule_groups</code>	Retrieves the metadata for the rule groups that you have defined
<code>list_tags_for_resource</code>	Retrieves the tags associated with the specified resource
<code>list_tls_inspection_configurations</code>	Retrieves the metadata for the TLS inspection configurations that you have defin
<code>put_resource_policy</code>	Creates or updates an IAM policy for your rule group or firewall policy
<code>tag_resource</code>	Adds the specified tags to the specified resource
<code>untag_resource</code>	Removes the tags with the specified keys from the specified resource
<code>update_firewall_delete_protection</code>	Modifies the flag, DeleteProtection, which indicates whether it is possible to dele
<code>update_firewall_description</code>	Modifies the description for the specified firewall
<code>update_firewall_encryption_configuration</code>	A complex type that contains settings for encryption of your firewall resources
<code>update_firewall_policy</code>	Updates the properties of the specified firewall policy
<code>update_firewall_policy_change_protection</code>	Modifies the flag, ChangeProtection, which indicates whether it is possible to cha
<code>update_logging_configuration</code>	Sets the logging configuration for the specified firewall
<code>update_rule_group</code>	Updates the rule settings for the specified rule group
<code>update_subnet_change_protection</code>	Update subnet change protection
<code>update_tls_inspection_configuration</code>	Updates the TLS inspection configuration settings for the specified TLS inspecti

Examples

```
## Not run:
svc <- networkfirewall()
svc$associate_firewall_policy(
  Foo = 123
)

## End(Not run)
```

Description

Amazon Web Services enables you to centrally manage your Amazon Web Services Cloud WAN core network and your Transit Gateway network across Amazon Web Services accounts, Regions, and on-premises locations.

Usage

```
networkmanager(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- networkmanager(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

[accept_attachment](#)

[associate_connect_peer](#)

[associate_customer_gateway](#)

[associate_link](#)

[associate_transit_gateway_connect_peer](#)

[create_connect_attachment](#)

[create_connection](#)

[create_connect_peer](#)

Accepts a core network attachment request

Associates a core network Connect peer with a device and optionally, with a

Associates a customer gateway with a device and optionally, with a link

Associates a link to a device

Associates a transit gateway Connect peer with a device, and optionally, with

Creates a core network Connect attachment from a specified core network a

Creates a connection between two devices

Creates a core network Connect peer for a specified core network connect a

create_core_network	Creates a core network as part of your global network, and optionally, with
create_device	Creates a new device in a global network
create_global_network	Creates a new, empty global network
create_link	Creates a new link for a specified site
create_site	Creates a new site in a global network
create_site_to_site_vpn_attachment	Creates an Amazon Web Services site-to-site VPN attachment on an edge l
create_transit_gateway_peering	Creates a transit gateway peering connection
create_transit_gateway_route_table_attachment	Creates a transit gateway route table attachment
create_vpc_attachment	Creates a VPC attachment on an edge location of a core network
delete_attachment	Deletes an attachment
delete_connection	Deletes the specified connection in your global network
delete_connect_peer	Deletes a Connect peer
delete_core_network	Deletes a core network along with all core network policies
delete_core_network_policy_version	Deletes a policy version from a core network
delete_device	Deletes an existing device
delete_global_network	Deletes an existing global network
delete_link	Deletes an existing link
delete_peering	Deletes an existing peering connection
delete_resource_policy	Deletes a resource policy for the specified resource
delete_site	Deletes an existing site
deregister_transit_gateway	Deregisters a transit gateway from your global network
describe_global_networks	Describes one or more global networks
disassociate_connect_peer	Disassociates a core network Connect peer from a device and a link
disassociate_customer_gateway	Disassociates a customer gateway from a device and a link
disassociate_link	Disassociates an existing device from a link
disassociate_transit_gateway_connect_peer	Disassociates a transit gateway Connect peer from a device and link
execute_core_network_change_set	Executes a change set on your core network
get_connect_attachment	Returns information about a core network Connect attachment
get_connections	Gets information about one or more of your connections in a global network
get_connect_peer	Returns information about a core network Connect peer
get_connect_peer_associations	Returns information about a core network Connect peer associations
get_core_network	Returns information about the LIVE policy for a core network
get_core_network_change_events	Returns information about a core network change event
get_core_network_change_set	Returns a change set between the LIVE core network policy and a submitte
get_core_network_policy	Returns details about a core network policy
get_customer_gateway_associations	Gets the association information for customer gateways that are associated
get_devices	Gets information about one or more of your devices in a global network
get_link_associations	Gets the link associations for a device or a link
get_links	Gets information about one or more links in a specified global network
get_network_resource_counts	Gets the count of network resources, by resource type, for the specified glo
get_network_resource_relationships	Gets the network resource relationships for the specified global network
get_network_resources	Describes the network resources for the specified global network
get_network_routes	Gets the network routes of the specified global network
get_network_telemetry	Gets the network telemetry of the specified global network
get_resource_policy	Returns information about a resource policy
get_route_analysis	Gets information about the specified route analysis
get_sites	Gets information about one or more of your sites in a global network
get_site_to_site_vpn_attachment	Returns information about a site-to-site VPN attachment

<code>get_transit_gateway_connect_peer_associations</code>	Gets information about one or more of your transit gateway Connect peer a
<code>get_transit_gateway_peering</code>	Returns information about a transit gateway peer
<code>get_transit_gateway_registrations</code>	Gets information about the transit gateway registrations in a specified globa
<code>get_transit_gateway_route_table_attachment</code>	Returns information about a transit gateway route table attachment
<code>get_vpc_attachment</code>	Returns information about a VPC attachment
<code>list_attachments</code>	Returns a list of core network attachments
<code>list_connect_peers</code>	Returns a list of core network Connect peers
<code>list_core_network_policy_versions</code>	Returns a list of core network policy versions
<code>list_core_networks</code>	Returns a list of owned and shared core networks
<code>list_organization_service_access_status</code>	Gets the status of the Service Linked Role (SLR) deployment for the accoun
<code>list_peerings</code>	Lists the peerings for a core network
<code>list_tags_for_resource</code>	Lists the tags for a specified resource
<code>put_core_network_policy</code>	Creates a new, immutable version of a core network policy
<code>put_resource_policy</code>	Creates or updates a resource policy
<code>register_transit_gateway</code>	Registers a transit gateway in your global network
<code>reject_attachment</code>	Rejects a core network attachment request
<code>restore_core_network_policy_version</code>	Restores a previous policy version as a new, immutable version of a core ne
<code>start_organization_service_access_update</code>	Enables the Network Manager service for an Amazon Web Services Organi
<code>start_route_analysis</code>	Starts analyzing the routing path between the specified source and destinati
<code>tag_resource</code>	Tags a specified resource
<code>untag_resource</code>	Removes tags from a specified resource
<code>update_connection</code>	Updates the information for an existing connection
<code>update_core_network</code>	Updates the description of a core network
<code>update_device</code>	Updates the details for an existing device
<code>update_global_network</code>	Updates an existing global network
<code>update_link</code>	Updates the details for an existing link
<code>update_network_resource_metadata</code>	Updates the resource metadata for the specified global network
<code>update_site</code>	Updates the information for an existing site
<code>update_vpc_attachment</code>	Updates a VPC attachment

Examples

```
## Not run:
svc <- networkmanager()
svc$accept_attachment(
  Foo = 123
)

## End(Not run)
```

Description

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service.

You can use Route 53 to:

- Register domain names.
For more information, see [How domain registration works](#).
- Route internet traffic to the resources for your domain
For more information, see [How internet traffic is routed to your website or web application](#).
- Check the health of your resources.
For more information, see [How Route 53 checks the health of your resources](#).

Usage

```
route53(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

[activate_key_signing_key](#)
[associate_vpc_with_hosted_zone](#)
[change_cidr_collection](#)
[change_resource_record_sets](#)
[change_tags_for_resource](#)
[create_cidr_collection](#)
[create_health_check](#)
[create_hosted_zone](#)

Activates a key-signing key (KSK) so that it can be used for signing by DNSSEC
 Associates an Amazon VPC with a private hosted zone
 Creates, changes, or deletes CIDR blocks within a collection
 Creates, changes, or deletes a resource record set, which contains authoritative DNS records
 Adds, edits, or deletes tags for a health check or a hosted zone
 Creates a CIDR collection in the current Amazon Web Services account
 Creates a new health check
 Creates a new public or private hosted zone

create_key_signing_key	Creates a new key-signing key (KSK) associated with a hosted zone
create_query_logging_config	Creates a configuration for DNS query logging
create_reusable_delegation_set	Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones
create_traffic_policy	Creates a traffic policy, which you use to create multiple DNS resource record sets
create_traffic_policy_instance	Creates resource record sets in a specified hosted zone based on the settings in a traffic policy
create_traffic_policy_version	Creates a new version of an existing traffic policy
create_vpc_association_authorization	Authorizes the Amazon Web Services account that created a specified VPC to submit an AssociateVPCWithHostedZone request to Route 53
deactivate_key_signing_key	Deactivates a key-signing key (KSK) so that it will not be used for signing by Route 53
delete_cidr_collection	Deletes a CIDR collection in the current Amazon Web Services account
delete_health_check	Deletes a health check
delete_hosted_zone	Deletes a hosted zone
delete_key_signing_key	Deletes a key-signing key (KSK)
delete_query_logging_config	Deletes a configuration for DNS query logging
delete_reusable_delegation_set	Deletes a reusable delegation set
delete_traffic_policy	Deletes a traffic policy
delete_traffic_policy_instance	Deletes a traffic policy instance and all of the resource record sets that Amazon Route 53 creates for the instance
delete_vpc_association_authorization	Removes authorization to submit an AssociateVPCWithHostedZone request to Route 53
disable_hosted_zone_dnssec	Disables DNSSEC signing in a specific hosted zone
disassociate_vpc_from_hosted_zone	Disassociates an Amazon Virtual Private Cloud (Amazon VPC) from an Amazon Route 53 hosted zone
enable_hosted_zone_dnssec	Enables DNSSEC signing in a specific hosted zone
get_account_limit	Gets the specified limit for the current account, for example, the maximum number of hosted zones that you can associate with the specified account
get_change	Returns the current status of a change batch request
get_checker_ip_ranges	Route 53 does not perform authorization for this API because it retrieves information from Amazon Route 53
get_dnssec	Returns information about DNSSEC for a specific hosted zone, including the status of DNSSEC signing and the status of DNSSEC validation
get_geo_location	Gets information about whether a specified geographic location is supported for a hosted zone
get_health_check	Gets information about a specified health check
get_health_check_count	Retrieves the number of health checks that are associated with the current Amazon Web Services account
get_health_check_last_failure_reason	Gets the reason that a specified health check failed most recently
get_health_check_status	Gets status of a specified health check
get_hosted_zone	Gets information about a specified hosted zone including the four name servers that are associated with the zone
get_hosted_zone_count	Retrieves the number of hosted zones that are associated with the current Amazon Web Services account
get_hosted_zone_limit	Gets the specified limit for a specified hosted zone, for example, the maximum number of hosted zones that you can associate with the specified account
get_query_logging_config	Gets information about a specified configuration for DNS query logging
get_reusable_delegation_set	Retrieves information about a specified reusable delegation set, including the four name servers that are associated with the set
get_reusable_delegation_set_limit	Gets the maximum number of hosted zones that you can associate with the specified reusable delegation set
get_traffic_policy	Gets information about a specific traffic policy version
get_traffic_policy_instance	Gets information about a specified traffic policy instance
get_traffic_policy_instance_count	Gets the number of traffic policy instances that are associated with the current Amazon Web Services account
list_cidr_blocks	Returns a paginated list of location objects and their CIDR blocks
list_cidr_collections	Returns a paginated list of CIDR collections in the Amazon Web Services account
list_cidr_locations	Returns a paginated list of CIDR locations for the given collection (metadata about the location)
list_geo_locations	Retrieves a list of supported geographic locations
list_health_checks	Retrieve a list of the health checks that are associated with the current Amazon Web Services account
list_hosted_zones	Retrieves a list of the public and private hosted zones that are associated with the current Amazon Web Services account
list_hosted_zones_by_name	Retrieves a list of your hosted zones in lexicographic order
list_hosted_zones_by_vpc	Lists all the private hosted zones that a specified VPC is associated with, regardless of whether the zones are authorized to be associated with the VPC
list_query_logging_configs	Lists the configurations for DNS query logging that are associated with the current Amazon Web Services account
list_resource_record_sets	Lists the resource record sets in a specified hosted zone

list_reusable_delegation_sets	Retrieves a list of the reusable delegation sets that are associated with the current account
list_tags_for_resource	Lists tags for one health check or hosted zone
list_tags_for_resources	Lists tags for up to 10 health checks or hosted zones
list_traffic_policies	Gets information about the latest version for every traffic policy that is associated with the current account
list_traffic_policy_instances	Gets information about the traffic policy instances that you created by using the current account
list_traffic_policy_instances_by_hosted_zone	Gets information about the traffic policy instances that you created in a specified hosted zone
list_traffic_policy_instances_by_policy	Gets information about the traffic policy instances that you created by using a specified traffic policy
list_traffic_policy_versions	Gets information about all of the versions for a specified traffic policy
list_vpc_association_authorizations	Gets a list of the VPCs that were created by other accounts and that can be associated with the current account
test_dns_answer	Gets the value that Amazon Route 53 returns in response to a DNS request for a specified hosted zone
update_health_check	Updates an existing health check
update_hosted_zone_comment	Updates the comment for a specified hosted zone
update_traffic_policy_comment	Updates the comment for a specified traffic policy version
update_traffic_policy_instance	After you submit a UpdateTrafficPolicyInstance request, there's a brief delay before the new traffic policy is active

Examples

```
## Not run:
svc <- route53()
# The following example associates the VPC with ID vpc-1a2b3c4d with the
# hosted zone with ID Z3M3LMPEXAMPLE.
svc$associate_vpc_with_hosted_zone(
  Comment = "",
  HostedZoneId = "Z3M3LMPEXAMPLE",
  VPC = list(
    VPCId = "vpc-1a2b3c4d",
    VPCRegion = "us-east-2"
  )
)

## End(Not run)
```

route53domains

Amazon Route 53 Domains

Description

Amazon Route 53 API actions let you register domain names and perform related operations.

Usage

```
route53domains(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53domains(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```

```

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

```

Operations

[accept_domain_transfer_from_another_aws_account](#)
[associate_delegation_signer_to_domain](#)
[cancel_domain_transfer_to_another_aws_account](#)
[check_domain_availability](#)
[check_domain_transferability](#)
[delete_domain](#)
[delete_tags_for_domain](#)
[disable_domain_auto_renew](#)
[disable_domain_transfer_lock](#)
[disassociate_delegation_signer_from_domain](#)
[enable_domain_auto_renew](#)
[enable_domain_transfer_lock](#)
[get_contact_reachability_status](#)
[get_domain_detail](#)
[get_domain_suggestions](#)
[get_operation_detail](#)
[list_domains](#)
[list_operations](#)
[list_prices](#)
[list_tags_for_domain](#)

Accepts the transfer of a domain from another Amazon Web Services account.
 Creates a delegation signer (DS) record in the registry zone for this domain.
 Cancels the transfer of a domain from the current Amazon Web Services account.
 This operation checks the availability of one domain name.
 Checks whether a domain name can be transferred to Amazon Route 53.
 This operation deletes the specified domain.
 This operation deletes the specified tags for a domain.
 This operation disables automatic renewal of domain registration for this domain.
 This operation removes the transfer lock on the domain (specifically the client delete action).
 Deletes a delegation signer (DS) record in the registry zone for this domain.
 This operation configures Amazon Route 53 to automatically renew the domain.
 This operation sets the transfer lock on the domain (specifically the client delete action).
 For operations that require confirmation that the email address for the registrant is correct.
 This operation returns detailed information about a specified domain.
 The GetDomainSuggestions operation returns a list of suggested domain names.
 This operation returns the current status of an operation that is not complete.
 This operation returns all the domain names registered with Amazon Route 53.
 Returns information about all of the operations that return an operation ID.
 Lists the following prices for either all the TLDs supported by Route 53 or a specific TLD.
 This operation returns all of the tags that are associated with the specified domain.

push_domain	Moves a domain from Amazon Web Services to another registrar
register_domain	This operation registers a domain
reject_domain_transfer_from_another_aws_account	Rejects the transfer of a domain from another Amazon Web Services account
renew_domain	This operation renews a domain for the specified number of years
resend_contact_reachability_email	For operations that require confirmation that the email address for the registrar is correct, resend the form of authorization email for this operation
resend_operation_authorization	Resend the form of authorization email for this operation
retrieve_domain_auth_code	This operation returns the authorization code for the domain
transfer_domain	Transfers a domain from another registrar to Amazon Route 53
transfer_domain_to_another_aws_account	Transfers a domain from the current Amazon Web Services account to another Amazon Web Services account
update_domain_contact	This operation updates the contact information for a particular domain
update_domain_contact_privacy	This operation updates the specified domain contact's privacy setting
update_domain_nameservers	This operation replaces the current set of name servers for the domain
update_tags_for_domain	This operation adds or updates tags for a specified domain
view_billing	Returns all the domain-related billing records for the current Amazon Web Services account

Examples

```
## Not run:
svc <- route53domains()
svc$accept_domain_transfer_from_another_aws_account(
  Foo = 123
)

## End(Not run)
```

route53recoverycluster

Route53 Recovery Cluster

Description

Welcome to the Routing Control (Recovery Cluster) API Reference Guide for Amazon Route 53 Application Recovery Controller.

With Route 53 ARC, you can use routing control with extreme reliability to recover applications by rerouting traffic across Availability Zones or Amazon Web Services Regions. Routing controls are simple on/off switches hosted on a highly available cluster in Route 53 ARC. A cluster provides a set of five redundant Regional endpoints against which you can run API calls to get or update the state of routing controls. To implement failover, you set one routing control to ON and another one to OFF, to reroute traffic from one Availability Zone or Amazon Web Services Region to another.

Be aware that you must specify a Regional endpoint for a cluster when you work with API cluster operations to get or update routing control states in Route 53 ARC. In addition, you must specify the US West (Oregon) Region for Route 53 ARC API calls. For example, use the parameter `--region us-west-2` with AWS CLI commands. For more information, see [Get and update routing](#)

control states using the API in the Amazon Route 53 Application Recovery Controller Developer Guide.

This API guide includes information about the API operations for how to get and update routing control states in Route 53 ARC. To work with routing control in Route 53 ARC, you must first create the required components (clusters, control panels, and routing controls) using the recovery cluster configuration API.

For more information about working with routing control in Route 53 ARC, see the following:

- Create clusters, control panels, and routing controls by using API operations. For more information, see the [Recovery Control Configuration API Reference Guide for Amazon Route 53 Application Recovery Controller](#).
- Learn about the components in recovery control, including clusters, routing controls, and control panels, and how to work with Route 53 ARC in the Amazon Web Services console. For more information, see [Recovery control components](#) in the Amazon Route 53 Application Recovery Controller Developer Guide.
- Route 53 ARC also provides readiness checks that continually audit resources to help make sure that your applications are scaled and ready to handle failover traffic. For more information about the related API operations, see the [Recovery Readiness API Reference Guide for Amazon Route 53 Application Recovery Controller](#).
- For more information about creating resilient applications and preparing for recovery readiness with Route 53 ARC, see the [Amazon Route 53 Application Recovery Controller Developer Guide](#).

Usage

```
route53recoverycluster(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- | | |
|--------|---|
| config | Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. |
|--------|---|

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53recoverycluster(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
```

```

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

```

Operations

get_routing_control_state	Get the state for a routing control
list_routing_controls	List routing control names and Amazon Resource Names (ARNs), as well as the routing control state
update_routing_control_state	Set the state of the routing control to reroute traffic
update_routing_control_states	Set multiple routing control states

Examples

```

## Not run:
svc <- route53recoverycluster()
svc$get_routing_control_state(
  Foo = 123
)

## End(Not run)

```

route53recoverycontrolconfig
AWS Route53 Recovery Control Config

Description

Recovery Control Configuration API Reference for Amazon Route 53 Application Recovery Controller

Usage

```

route53recoverycontrolconfig(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)

```


Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53recoverycontrolconfig(
  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_cluster	Create a new cluster
create_control_panel	Creates a new control panel
create_routing_control	Creates a new routing control
create_safety_rule	Creates a safety rule in a control panel
delete_cluster	Delete a cluster
delete_control_panel	Deletes a control panel
delete_routing_control	Deletes a routing control
delete_safety_rule	Deletes a safety rule
describe_cluster	Display the details about a cluster
describe_control_panel	Displays details about a control panel
describe_routing_control	Displays details about a routing control
describe_safety_rule	Returns information about a safety rule
get_resource_policy	Get information about the resource policy for a cluster
list_associated_route_53_health_checks	Returns an array of all Amazon Route 53 health checks associated with a specific resource
list_clusters	Returns an array of all the clusters in an account
list_control_panels	Returns an array of control panels in an account or in a cluster
list_routing_controls	Returns an array of routing controls for a control panel
list_safety_rules	List the safety rules (the assertion rules and gating rules) that you've defined for the control panel
list_tags_for_resource	Lists the tags for a resource
tag_resource	Adds a tag to a resource

untag_resource	Removes a tag from a resource
update_control_panel	Updates a control panel
update_routing_control	Updates a routing control
update_safety_rule	Update a safety rule (an assertion rule or gating rule)

Examples

```
## Not run:
svc <- route53recoverycontrolconfig()
svc$create_cluster(
  Foo = 123
)

## End(Not run)
```

route53recoveryreadiness

AWS Route53 Recovery Readiness

Description

Recovery readiness

Usage

```
route53recoveryreadiness(
  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.

	<ul style="list-style-type: none"> • 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53recoveryreadiness(
  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_cell	Creates a cell in an account
create_cross_account_authorization	Creates a cross-account readiness authorization
create_readiness_check	Creates a readiness check in an account
create_recovery_group	Creates a recovery group in an account
create_resource_set	Creates a resource set
delete_cell	Delete a cell
delete_cross_account_authorization	Deletes cross account readiness authorization
delete_readiness_check	Deletes a readiness check
delete_recovery_group	Deletes a recovery group
delete_resource_set	Deletes a resource set
get_architecture_recommendations	Gets recommendations about architecture designs for improving resiliency for an account
get_cell	Gets information about a cell including cell name, cell Amazon Resource Name (ARN), and cell type
get_cell_readiness_summary	Gets readiness for a cell
get_readiness_check	Gets details about a readiness check
get_readiness_check_resource_status	Gets individual readiness status for a readiness check
get_readiness_check_status	Gets the readiness status for an individual readiness check
get_recovery_group	Gets details about a recovery group, including a list of the cells that are included in the group
get_recovery_group_readiness_summary	Displays a summary of information about a recovery group's readiness status
get_resource_set	Displays the details about a resource set, including a list of the resources in the set
list_cells	Lists the cells for an account
list_cross_account_authorizations	Lists the cross-account readiness authorizations that are in place for an account
list_readiness_checks	Lists the readiness checks for an account
list_recovery_groups	Lists the recovery groups in an account
list_resource_sets	Lists the resource sets in an account
list_rules	Lists all readiness rules, or lists the readiness rules for a specific resource type
list_tags_for_resources	Lists the tags for a resource
tag_resource	Adds a tag to a resource
untag_resource	Removes a tag from a resource
update_cell	Updates a cell to replace the list of nested cells with a new list of nested cells
update_readiness_check	Updates a readiness check
update_recovery_group	Updates a recovery group
update_resource_set	Updates a resource set

Examples

```
## Not run:
svc <- route53recoveryreadiness()
svc$create_cell(
  Foo = 123
)

## End(Not run)
```

route53resolver

Amazon Route 53 Resolver

Description

When you create a VPC using Amazon VPC, you automatically get DNS resolution within the VPC from Route 53 Resolver. By default, Resolver answers DNS queries for VPC domain names such as domain names for EC2 instances or Elastic Load Balancing load balancers. Resolver performs recursive lookups against public name servers for all other domain names.

You can also configure DNS resolution between your VPC and your network over a Direct Connect or VPN connection:

Forward DNS queries from resolvers on your network to Route 53 Resolver

DNS resolvers on your network can forward DNS queries to Resolver in a specified VPC. This allows your DNS resolvers to easily resolve domain names for Amazon Web Services resources such as EC2 instances or records in a Route 53 private hosted zone. For more information, see [How DNS Resolvers on Your Network Forward DNS Queries to Route 53 Resolver](#) in the *Amazon Route 53 Developer Guide*.

Conditionally forward queries from a VPC to resolvers on your network

You can configure Resolver to forward queries that it receives from EC2 instances in your VPCs to DNS resolvers on your network. To forward selected queries, you create Resolver rules that specify the domain names for the DNS queries that you want to forward (such as example.com), and the IP addresses of the DNS resolvers on your network that you want to forward the queries to. If a query matches multiple rules (example.com, acme.example.com), Resolver chooses the rule with the most specific match (acme.example.com) and forwards the query to the IP addresses that you specified in that rule. For more information, see [How Route 53 Resolver Forwards DNS Queries from Your VPCs to Your Network](#) in the *Amazon Route 53 Developer Guide*.

Like Amazon VPC, Resolver is Regional. In each Region where you have VPCs, you can choose whether to forward queries from your VPCs to your network (outbound queries), from your network to your VPCs (inbound queries), or both.

Usage

```
route53resolver(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- route53resolver(
  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_firewall_rule_group	Associates a FirewallRuleGroup with a VPC, to provide DNS filtering for the VPC
associate_resolver_endpoint_ip_address	Adds IP addresses to an inbound or an outbound Resolver endpoint
associate_resolver_query_log_config	Associates an Amazon VPC with a specified query logging configuration
associate_resolver_rule	Associates a Resolver rule with a VPC
create_firewall_domain_list	Creates an empty firewall domain list for use in DNS Firewall rules
create_firewall_rule	Creates a single DNS Firewall rule in the specified rule group, using the specified domain list
create_firewall_rule_group	Creates an empty DNS Firewall rule group for filtering DNS network traffic in a VPC
create_outpost_resolver	Creates a Route 53 Resolver on an Outpost
create_resolver_endpoint	Creates a Resolver endpoint
create_resolver_query_log_config	Creates a Resolver query logging configuration, which defines where you want to log DNS queries
create_resolver_rule	For DNS queries that originate in your VPCs, specifies which Resolver endpoint to use
delete_firewall_domain_list	Deletes the specified domain list
delete_firewall_rule	Deletes the specified firewall rule
delete_firewall_rule_group	Deletes the specified firewall rule group

<code>delete_outpost_resolver</code>	Deletes a Resolver on the Outpost
<code>delete_resolver_endpoint</code>	Deletes a Resolver endpoint
<code>delete_resolver_query_log_config</code>	Deletes a query logging configuration
<code>delete_resolver_rule</code>	Deletes a Resolver rule
<code>disassociate_firewall_rule_group</code>	Disassociates a FirewallRuleGroup from a VPC, to remove DNS filtering from the VPC
<code>disassociate_resolver_endpoint_ip_address</code>	Removes IP addresses from an inbound or an outbound Resolver endpoint
<code>disassociate_resolver_query_log_config</code>	Disassociates a VPC from a query logging configuration
<code>disassociate_resolver_rule</code>	Removes the association between a specified Resolver rule and a specified VPC
<code>get_firewall_config</code>	Retrieves the configuration of the firewall behavior provided by DNS Firewall
<code>get_firewall_domain_list</code>	Retrieves the specified firewall domain list
<code>get_firewall_rule_group</code>	Retrieves the specified firewall rule group
<code>get_firewall_rule_group_association</code>	Retrieves a firewall rule group association, which enables DNS filtering for a VPC
<code>get_firewall_rule_group_policy</code>	Returns the Identity and Access Management (Amazon Web Services IAM) policy for a firewall rule group
<code>get_outpost_resolver</code>	Gets information about a specified Resolver on the Outpost, such as its instance ID
<code>get_resolver_config</code>	Retrieves the behavior configuration of Route 53 Resolver behavior for a single resource
<code>get_resolver_dnssec_config</code>	Gets DNSSEC validation information for a specified resource
<code>get_resolver_endpoint</code>	Gets information about a specified Resolver endpoint, such as whether it's an inbound or outbound endpoint
<code>get_resolver_query_log_config</code>	Gets information about a specified Resolver query logging configuration, such as the logging policy
<code>get_resolver_query_log_config_association</code>	Gets information about a specified association between a Resolver query logging configuration and a VPC
<code>get_resolver_query_log_config_policy</code>	Gets information about a query logging policy
<code>get_resolver_rule</code>	Gets information about a specified Resolver rule, such as the domain name that the rule applies to
<code>get_resolver_rule_association</code>	Gets information about an association between a specified Resolver rule and a VPC
<code>get_resolver_rule_policy</code>	Gets information about the Resolver rule policy for a specified rule
<code>import_firewall_domains</code>	Imports domain names from a file into a domain list, for use in a DNS firewall
<code>list_firewall_configs</code>	Retrieves the firewall configurations that you have defined
<code>list_firewall_domain_lists</code>	Retrieves the firewall domain lists that you have defined
<code>list_firewall_domains</code>	Retrieves the domains that you have defined for the specified firewall domain list
<code>list_firewall_rule_group_associations</code>	Retrieves the firewall rule group associations that you have defined
<code>list_firewall_rule_groups</code>	Retrieves the minimal high-level information for the rule groups that you have defined
<code>list_firewall_rules</code>	Retrieves the firewall rules that you have defined for the specified firewall rule group
<code>list_outpost_resolvers</code>	Lists all the Resolvers on Outposts that were created using the current Amazon Web Services account
<code>list_resolver_configs</code>	Retrieves the Resolver configurations that you have defined
<code>list_resolver_dnssec_configs</code>	Lists the configurations for DNSSEC validation that are associated with the current Amazon Web Services account
<code>list_resolver_endpoint_ip_addresses</code>	Gets the IP addresses for a specified Resolver endpoint
<code>list_resolver_endpoints</code>	Lists all the Resolver endpoints that were created using the current Amazon Web Services account
<code>list_resolver_query_log_config_associations</code>	Lists information about associations between Amazon VPCs and query logging configurations
<code>list_resolver_query_log_configs</code>	Lists information about the specified query logging configurations
<code>list_resolver_rule_associations</code>	Lists the associations that were created between Resolver rules and VPCs using the current Amazon Web Services account
<code>list_resolver_rules</code>	Lists the Resolver rules that were created using the current Amazon Web Services account
<code>list_tags_for_resource</code>	Lists the tags that you associated with the specified resource
<code>put_firewall_rule_group_policy</code>	Attaches an Identity and Access Management (Amazon Web Services IAM) policy to a firewall rule group
<code>put_resolver_query_log_config_policy</code>	Specifies an Amazon Web Services account that you want to share a query logging configuration with
<code>put_resolver_rule_policy</code>	Specifies an Amazon Web Services rule that you want to share with another Amazon Web Services account
<code>tag_resource</code>	Adds one or more tags to a specified resource
<code>untag_resource</code>	Removes one or more tags from a specified resource
<code>update_firewall_config</code>	Updates the configuration of the firewall behavior provided by DNS Firewall
<code>update_firewall_domains</code>	Updates the firewall domain list from an array of domain specifications
<code>update_firewall_rule</code>	Updates the specified firewall rule

update_firewall_rule_group_association	Changes the association of a FirewallRuleGroup with a VPC
update_outpost_resolver	You can use UpdateOutpostResolver to update the instance count, type, or name
update_resolver_config	Updates the behavior configuration of Route 53 Resolver behavior for a single
update_resolver_dnssec_config	Updates an existing DNSSEC validation configuration
update_resolver_endpoint	Updates the name, or endpoint type for an inbound or an outbound Resolver en
update_resolver_rule	Updates settings for a specified Resolver rule

Examples

```
## Not run:
svc <- route53resolver()
svc$associate_firewall_rule_group(
  Foo = 123
)

## End(Not run)
```

servicediscovery

AWS Cloud Map

Description

Cloud Map

With Cloud Map, you can configure public DNS, private DNS, or HTTP namespaces that your microservice applications run in. When an instance becomes available, you can call the Cloud Map API to register the instance with Cloud Map. For public or private DNS namespaces, Cloud Map automatically creates DNS records and an optional health check. Clients that submit public or private DNS queries, or HTTP requests, for the service receive an answer that contains up to eight healthy records.

Usage

```
servicediscovery(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials:**
 - **creds:**

	<ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- servicediscovery(
  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_http_namespace	Creates an HTTP namespace
create_private_dns_namespace	Creates a private namespace based on DNS, which is visible only inside a specified Amazon VPC
create_public_dns_namespace	Creates a public namespace based on DNS, which is visible on the internet
create_service	Creates a service
delete_namespace	Deletes a namespace from the current account
delete_service	Deletes a specified service
deregister_instance	Deletes the Amazon Route 53 DNS records and health check, if any, that Cloud Map registered for the instance
discover_instances	Discovers registered instances for a specified namespace and service
discover_instances_revision	Discovers the increasing revision associated with an instance
get_instance	Gets information about a specified instance
get_instances_health_status	Gets the current health status (Healthy, Unhealthy, or Unknown) of one or more instances
get_namespace	Gets information about a namespace
get_operation	Gets information about any operation that returns an operation ID in the response, such as create_namespace
get_service	Gets the settings for a specified service
list_instances	Lists summary information about the instances that you registered by using a specified namespace and service
list_namespaces	Lists summary information about the namespaces that were created by the current Amazon account
list_operations	Lists operations that match the criteria that you specify
list_services	Lists summary information for all the services that are associated with one or more namespaces
list_tags_for_resource	Lists tags for the specified resource
register_instance	Creates or updates one or more records and, optionally, creates a health check based on the specified settings
tag_resource	Adds one or more tags to the specified resource
untag_resource	Removes one or more tags from the specified resource
update_http_namespace	Updates an HTTP namespace
update_instance_custom_health_status	Submits a request to change the health status of a custom health check to healthy or unhealthy
update_private_dns_namespace	Updates a private DNS namespace

update_public_dns_namespace
update_service

Updates a public DNS namespace
Submits a request to perform the following operations:

Examples

```
## Not run:
svc <- servicediscovery()
# This example creates an HTTP namespace.
svc$create_http_namespace(
  CreatorRequestId = "example-creator-request-id-0001",
  Description = "Example.com AWS Cloud Map HTTP Namespace",
  Name = "example-http.com"
)

## End(Not run)
```

telconetworkbuilder *AWS Telco Network Builder*

Description

Amazon Web Services Telco Network Builder (TNB) is a network automation service that helps you deploy and manage telecom networks. AWS TNB helps you with the lifecycle management of your telecommunication network functions throughout planning, deployment, and post-deployment activities.

Usage

```
telconetworkbuilder(
  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

	<ul style="list-style-type: none"> – 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- telconetworkbuilder(
  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_sol_network_operation	Cancels a network operation
create_sol_function_package	Creates a function package
create_sol_network_instance	Creates a network instance
create_sol_network_package	Creates a network package
delete_sol_function_package	Deletes a function package
delete_sol_network_instance	Deletes a network instance
delete_sol_network_package	Deletes network package
get_sol_function_instance	Gets the details of a network function instance, including the instantiation state and
get_sol_function_package	Gets the details of an individual function package, such as the operational state and
get_sol_function_package_content	Gets the contents of a function package
get_sol_function_package_descriptor	Gets a function package descriptor in a function package
get_sol_network_instance	Gets the details of the network instance
get_sol_network_operation	Gets the details of a network operation, including the tasks involved in the network
get_sol_network_package	Gets the details of a network package
get_sol_network_package_content	Gets the contents of a network package
get_sol_network_package_descriptor	Gets the content of the network service descriptor
instantiate_sol_network_instance	Instantiates a network instance
list_sol_function_instances	Lists network function instances
list_sol_function_packages	Lists information about function packages
list_sol_network_instances	Lists your network instances
list_sol_network_operations	Lists details for a network operation, including when the operation started and the s
list_sol_network_packages	Lists network packages
list_tags_for_resource	Lists tags for AWS TNB resources
put_sol_function_package_content	Uploads the contents of a function package
put_sol_network_package_content	Uploads the contents of a network package
tag_resource	Tags an AWS TNB resource
terminate_sol_network_instance	Terminates a network instance
untag_resource	Untags an AWS TNB resource
update_sol_function_package	Updates the operational state of function package

<code>update_sol_network_instance</code>	Update a network instance
<code>update_sol_network_package</code>	Updates the operational state of a network package
<code>validate_sol_function_package_content</code>	Validates function package content
<code>validate_sol_network_package_content</code>	Validates network package content

Examples

```
## Not run:
svc <- telconetworkbuilder()
svc$cancel_sol_network_operation(
  Foo = 123
)

## End(Not run)
```

vpclattice

Amazon VPC Lattice

Description

Amazon VPC Lattice is a fully managed application networking service that you use to connect, secure, and monitor all of your services across multiple accounts and virtual private clouds (VPCs). Amazon VPC Lattice interconnects your microservices and legacy services within a logical boundary, so that you can discover and manage them more efficiently. For more information, see the [Amazon VPC Lattice User Guide](#)

Usage

```
vpclattice(
  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

	<ul style="list-style-type: none"> – 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- vpclattice(
  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_update_rule	Updates the listener rules in a batch
create_access_log_subscription	Enables access logs to be sent to Amazon CloudWatch, Amazon S3, and Amazon
create_listener	Creates a listener for a service
create_rule	Creates a listener rule
create_service	Creates a service
create_service_network	Creates a service network
create_service_network_service_association	Associates a service with a service network
create_service_network_vpc_association	Associates a VPC with a service network
create_target_group	Creates a target group
delete_access_log_subscription	Deletes the specified access log subscription
delete_auth_policy	Deletes the specified auth policy
delete_listener	Deletes the specified listener
delete_resource_policy	Deletes the specified resource policy
delete_rule	Deletes a listener rule
delete_service	Deletes a service
delete_service_network	Deletes a service network
delete_service_network_service_association	Deletes the association between a specified service and the specific service network
delete_service_network_vpc_association	Disassociates the VPC from the service network
delete_target_group	Deletes a target group
deregister_targets	Deregisters the specified targets from the specified target group
get_access_log_subscription	Retrieves information about the specified access log subscription
get_auth_policy	Retrieves information about the auth policy for the specified service or service
get_listener	Retrieves information about the specified listener for the specified service
get_resource_policy	Retrieves information about the resource policy
get_rule	Retrieves information about listener rules
get_service	Retrieves information about the specified service
get_service_network	Retrieves information about the specified service network
get_service_network_service_association	Retrieves information about the specified association between a service network
get_service_network_vpc_association	Retrieves information about the association between a service network and a V

<code>get_target_group</code>	Retrieves information about the specified target group
<code>list_access_log_subscriptions</code>	Lists all access log subscriptions for the specified service network or service
<code>list_listeners</code>	Lists the listeners for the specified service
<code>list_rules</code>	Lists the rules for the listener
<code>list_service_networks</code>	Lists the service networks owned by the caller account or shared with the caller
<code>list_service_network_service_associations</code>	Lists the associations between the service network and the service
<code>list_service_network_vpc_associations</code>	Lists the service network and VPC associations
<code>list_services</code>	Lists the services owned by the caller account or shared with the caller account
<code>list_tags_for_resource</code>	Lists the tags for the specified resource
<code>list_target_groups</code>	Lists your target groups
<code>list_targets</code>	Lists the targets for the target group
<code>put_auth_policy</code>	Creates or updates the auth policy
<code>put_resource_policy</code>	Attaches a resource-based permission policy to a service or service network
<code>register_targets</code>	Registers the targets with the target group
<code>tag_resource</code>	Adds the specified tags to the specified resource
<code>untag_resource</code>	Removes the specified tags from the specified resource
<code>update_access_log_subscription</code>	Updates the specified access log subscription
<code>update_listener</code>	Updates the specified listener for the specified service
<code>update_rule</code>	Updates a rule for the listener
<code>update_service</code>	Updates the specified service
<code>update_service_network</code>	Updates the specified service network
<code>update_service_network_vpc_association</code>	Updates the service network and VPC association
<code>update_target_group</code>	Updates the specified target group

Examples

```
## Not run:
svc <- vpclattice()
svc$batch_update_rule(
  Foo = 123
)

## End(Not run)
```

Index

accept_attachment, [52](#)
accept_direct_connect_gateway_association_proposal, [72](#)
[33](#)
accept_domain_transfer_from_another_aws_account, [72](#)
[60](#)
activate_key_signing_key, [56](#)
add_custom_routing_endpoints, [45](#)
add_endpoints, [45](#)
add_listener_certificates, [41](#)
add_tags, [37](#), [41](#)
add_trust_store_revocations, [41](#)
advertise_byoip_cidr, [45](#)
allocate_connection_on_interconnect, [34](#)
allocate_hosted_connection, [34](#)
allocate_private_virtual_interface, [34](#)
allocate_public_virtual_interface, [34](#)
allocate_transit_virtual_interface, [34](#)
allow_custom_routing_traffic, [45](#)
apigateway, [3](#)
apigatewaymanagementapi, [8](#)
apigatewayv2, [11](#)
appfabric, [15](#)
apply_security_groups_to_load_balancer, [37](#)
appmesh, [18](#)
arczonalshift, [21](#)
associate_alias, [29](#)
associate_connect_peer, [52](#)
associate_connection_with_lag, [34](#)
associate_customer_gateway, [52](#)
associate_delegation_signer_to_domain, [60](#)
associate_firewall_policy, [49](#)
associate_firewall_rule_group, [72](#)
associate_gateway_to_server, [26](#)
associate_hosted_connection, [34](#)
associate_link, [52](#)
associate_mac_sec_key, [34](#)
associate_resolver_endpoint_ip_address,
associate_resolver_query_log_config,
associate_resolver_rule, [72](#)
associate_subnets, [49](#)
associate_transit_gateway_connect_peer,
[52](#)
associate_virtual_interface, [34](#)
associate_vpc_with_hosted_zone, [56](#)
attach_load_balancer_to_subnets, [37](#)
backupgateway, [24](#)
batch_get_user_access_tasks, [17](#)
batch_update_rule, [82](#)
cancel_domain_transfer_to_another_aws_account,
[60](#)
cancel_sol_network_operation, [79](#)
cancel_zonal_shift, [23](#)
change_cidr_collection, [56](#)
change_resource_record_sets, [56](#)
change_tags_for_resource, [56](#)
check_domain_availability, [60](#)
check_domain_transferability, [60](#)
cloudfront, [27](#)
configure_health_check, [37](#)
confirm_connection, [34](#)
confirm_customer_agreement, [34](#)
confirm_private_virtual_interface, [34](#)
confirm_public_virtual_interface, [34](#)
confirm_transit_virtual_interface, [34](#)
connect_app_authorization, [17](#)
copy_distribution, [29](#)
create_accelerator, [45](#)
create_access_log_subscription, [82](#)
create_api, [13](#)
create_api_key, [5](#)
create_api_mapping, [13](#)
create_app_authorization, [17](#)

- create_app_bundle, 17
- create_app_cookie_stickiness_policy, 37
- create_authorizer, 5, 13
- create_base_path_mapping, 5
- create_bgp_peer, 34
- create_cache_policy, 29
- create_cell, 69
- create_cidr_collection, 56
- create_cloud_front_origin_access_identity, 29
- create_cluster, 66
- create_connect_attachment, 52
- create_connect_peer, 52
- create_connection, 34, 52
- create_continuous_deployment_policy, 29
- create_control_panel, 66
- create_core_network, 53
- create_cross_account_attachment, 45
- create_cross_account_authorization, 69
- create_custom_routing_accelerator, 45
- create_custom_routing_endpoint_group, 45
- create_custom_routing_listener, 45
- create_deployment, 5, 13
- create_device, 53
- create_direct_connect_gateway, 34
- create_direct_connect_gateway_association, 34
- create_direct_connect_gateway_association_proposal, 34
- create_distribution, 29
- create_distribution_with_tags, 29
- create_documentation_part, 5
- create_documentation_version, 5
- create_domain_name, 5, 13
- create_endpoint_group, 45
- create_field_level_encryption_config, 29
- create_field_level_encryption_profile, 29
- create_firewall, 49
- create_firewall_domain_list, 72
- create_firewall_policy, 49
- create_firewall_rule, 72
- create_firewall_rule_group, 72
- create_function, 29
- create_gateway, 26
- create_gateway_route, 20
- create_global_network, 53
- create_health_check, 56
- create_hosted_zone, 56
- create_http_namespace, 76
- create_ingestion, 17
- create_ingestion_destination, 17
- create_integration, 13
- create_integration_response, 13
- create_interconnect, 34
- create_invalidations, 29
- create_key_group, 29
- create_key_signing_key, 57
- create_key_value_store, 29
- create_lag, 34
- create_lb_cookie_stickiness_policy, 37
- create_link, 53
- create_listener, 41, 45, 82
- create_load_balancer, 35, 37, 41
- create_load_balancer_listeners, 37
- create_load_balancer_policy, 37
- create_mesh, 20
- create_model, 5, 13
- create_monitoring_subscription, 29
- create_origin_access_control, 29
- create_origin_request_policy, 29
- create_outpost_resolver, 72
- create_practice_run_configuration, 23
- create_private_dns_namespace, 76
- create_private_virtual_interface, 34
- create_public_dns_namespace, 76
- create_public_key, 29
- create_public_virtual_interface, 34
- create_query_logging_config, 57
- create_readiness_check, 69
- create_realtime_log_config, 29
- create_recovery_group, 69
- create_request_validator, 5
- create_resolver_endpoint, 72
- create_resolver_query_log_config, 72
- create_resolver_rule, 72
- create_resource, 5
- create_resource_set, 69
- create_response_headers_policy, 29
- create_rest_api, 5
- create_reusable_delegation_set, 57
- create_route, 13, 20

- create_route_response, [13](#)
- create_routing_control, [66](#)
- create_rule, [41](#), [82](#)
- create_rule_group, [49](#)
- create_safety_rule, [66](#)
- create_service, [76](#), [82](#)
- create_service_network, [82](#)
- create_service_network_service_association, [82](#)
- create_service_network_vpc_association, [82](#)
- create_site, [53](#)
- create_site_to_site_vpn_attachment, [53](#)
- create_sol_function_package, [79](#)
- create_sol_network_instance, [79](#)
- create_sol_network_package, [79](#)
- create_stage, [5](#), [13](#)
- create_streaming_distribution, [29](#)
- create_streaming_distribution_with_tags, [29](#)
- create_target_group, [41](#), [82](#)
- create_tls_inspection_configuration, [49](#)
- create_traffic_policy, [57](#)
- create_traffic_policy_instance, [57](#)
- create_traffic_policy_version, [57](#)
- create_transit_gateway_peering, [53](#)
- create_transit_gateway_route_table_attachment, [53](#)
- create_transit_virtual_interface, [34](#)
- create_trust_store, [41](#)
- create_usage_plan, [5](#)
- create_usage_plan_key, [5](#)
- create_virtual_gateway, [20](#)
- create_virtual_node, [20](#)
- create_virtual_router, [20](#)
- create_virtual_service, [20](#)
- create_vpc_association_authorization, [57](#)
- create_vpc_attachment, [53](#)
- create_vpc_link, [5](#), [13](#)

- deactivate_key_signing_key, [57](#)
- delete_accelerator, [45](#)
- delete_access_log_settings, [13](#)
- delete_access_log_subscription, [82](#)
- delete_api, [13](#)
- delete_api_key, [5](#)
- delete_api_mapping, [13](#)

- delete_app_authorization, [17](#)
- delete_app_bundle, [17](#)
- delete_attachment, [53](#)
- delete_auth_policy, [82](#)
- delete_authorizer, [5](#), [13](#)
- delete_base_path_mapping, [5](#)
- delete_bgp_peer, [34](#)
- delete_cache_policy, [29](#)
- delete_cell, [69](#)
- delete_cidr_collection, [57](#)
- delete_client_certificate, [5](#)
- delete_cloud_front_origin_access_identity, [29](#)
- delete_cluster, [66](#)
- delete_connect_peer, [53](#)
- delete_connection, [11](#), [34](#), [53](#)
- delete_continuous_deployment_policy, [29](#)
- delete_control_panel, [66](#)
- delete_core_network, [53](#)
- delete_core_network_policy_version, [53](#)
- delete_cors_configuration, [13](#)
- delete_cross_account_attachment, [45](#)
- delete_cross_account_authorization, [69](#)
- delete_custom_routing_accelerator, [45](#)
- delete_custom_routing_endpoint_group, [45](#)
- delete_custom_routing_listener, [45](#)
- delete_deployment, [5](#), [13](#)
- delete_device, [53](#)
- delete_direct_connect_gateway, [34](#)
- delete_direct_connect_gateway_association, [34](#)
- delete_direct_connect_gateway_association_proposal, [34](#)
- delete_distribution, [29](#)
- delete_documentation_part, [5](#)
- delete_documentation_version, [5](#)
- delete_domain, [60](#)
- delete_domain_name, [5](#), [13](#)
- delete_endpoint_group, [45](#)
- delete_field_level_encryption_config, [29](#)
- delete_field_level_encryption_profile, [29](#)
- delete_firewall, [49](#)
- delete_firewall_domain_list, [72](#)
- delete_firewall_policy, [49](#)

- delete_firewall_rule, 72
- delete_firewall_rule_group, 72
- delete_function, 29
- delete_gateway, 26
- delete_gateway_response, 5
- delete_gateway_route, 20
- delete_global_network, 53
- delete_health_check, 57
- delete_hosted_zone, 57
- delete_hypervisor, 26
- delete_ingestion, 17
- delete_ingestion_destination, 17
- delete_integration, 5, 13
- delete_integration_response, 5, 13
- delete_interconnect, 34
- delete_key_group, 29
- delete_key_signing_key, 57
- delete_key_value_store, 29
- delete_lag, 34
- delete_link, 53
- delete_listener, 41, 45, 82
- delete_load_balancer, 37, 41
- delete_load_balancer_listeners, 37
- delete_load_balancer_policy, 37
- delete_mesh, 20
- delete_method, 5
- delete_method_response, 5
- delete_model, 5, 13
- delete_monitoring_subscription, 29
- delete_namespace, 76
- delete_origin_access_control, 29
- delete_origin_request_policy, 29
- delete_outpost_resolver, 73
- delete_peering, 53
- delete_practice_run_configuration, 23
- delete_public_key, 29
- delete_query_logging_config, 57
- delete_readiness_check, 69
- delete_realtime_log_config, 29
- delete_recovery_group, 69
- delete_request_validator, 5
- delete_resolver_endpoint, 73
- delete_resolver_query_log_config, 73
- delete_resolver_rule, 73
- delete_resource, 5
- delete_resource_policy, 50, 53, 82
- delete_resource_set, 69
- delete_response_headers_policy, 29
- delete_rest_api, 5
- delete_reusable_delegation_set, 57
- delete_route, 13, 20
- delete_route_request_parameter, 13
- delete_route_response, 13
- delete_route_settings, 13
- delete_routing_control, 66
- delete_rule, 41, 82
- delete_rule_group, 50
- delete_safety_rule, 66
- delete_service, 76, 82
- delete_service_network, 82
- delete_service_network_service_association, 82
- delete_service_network_vpc_association, 82
- delete_site, 53
- delete_sol_function_package, 79
- delete_sol_network_instance, 79
- delete_sol_network_package, 79
- delete_stage, 5, 13
- delete_streaming_distribution, 29
- delete_tags_for_domain, 60
- delete_target_group, 41, 82
- delete_tls_inspection_configuration, 50
- delete_traffic_policy, 57
- delete_traffic_policy_instance, 57
- delete_trust_store, 41
- delete_usage_plan, 5
- delete_usage_plan_key, 5
- delete_virtual_gateway, 20
- delete_virtual_interface, 34
- delete_virtual_node, 20
- delete_virtual_router, 20
- delete_virtual_service, 20
- delete_vpc_association_authorization, 57
- delete_vpc_link, 5, 13
- deny_custom_routing_traffic, 45
- deprovision_byoip_cidr, 45
- deregister_instance, 76
- deregister_instances_from_load_balancer, 37
- deregister_targets, 41, 82
- deregister_transit_gateway, 53
- describe_accelerator, 45
- describe_accelerator_attributes, 45

- describe_account_limits, [37, 41](#)
- describe_cluster, [66](#)
- describe_connection_loa, [34](#)
- describe_connections, [34](#)
- describe_connections_on_interconnect, [34](#)
- describe_control_panel, [66](#)
- describe_cross_account_attachment, [45](#)
- describe_custom_routing_accelerator, [46](#)
- describe_custom_routing_accelerator_attributes, [46](#)
- describe_custom_routing_endpoint_group, [46](#)
- describe_custom_routing_listener, [46](#)
- describe_customer_metadata, [34](#)
- describe_direct_connect_gateway_association_proposals, [34](#)
- describe_direct_connect_gateway_associations, [34](#)
- describe_direct_connect_gateway_attachments, [34](#)
- describe_direct_connect_gateways, [34](#)
- describe_endpoint_group, [46](#)
- describe_firewall, [50](#)
- describe_firewall_policy, [50](#)
- describe_function, [29](#)
- describe_gateway_route, [20](#)
- describe_global_networks, [53](#)
- describe_hosted_connections, [34](#)
- describe_instance_health, [38](#)
- describe_interconnect_loa, [34](#)
- describe_interconnects, [34](#)
- describe_key_value_store, [29](#)
- describe_lags, [34](#)
- describe_listener, [46](#)
- describe_listener_certificates, [41](#)
- describe_listeners, [41](#)
- describe_loa, [34](#)
- describe_load_balancer_attributes, [38, 41](#)
- describe_load_balancer_policies, [38](#)
- describe_load_balancer_policy_types, [38](#)
- describe_load_balancers, [38, 41](#)
- describe_locations, [34](#)
- describe_logging_configuration, [50](#)
- describe_mesh, [20](#)
- describe_resource_policy, [50](#)
- describe_route, [20](#)
- describe_router_configuration, [34](#)
- describe_routing_control, [66](#)
- describe_rule_group, [50](#)
- describe_rule_group_metadata, [50](#)
- describe_rules, [41](#)
- describe_safety_rule, [66](#)
- describe_ssl_policies, [41](#)
- describe_tags, [34, 38, 41](#)
- describe_target_group_attributes, [41](#)
- describe_target_groups, [41](#)
- describe_target_health, [41](#)
- describe_tls_inspection_configuration, [50](#)
- describe_trust_store_associations, [41](#)
- describe_trust_store_revocations, [41](#)
- describe_trust_stores, [41](#)
- describe_virtual_gateway, [20](#)
- describe_virtual_gateways, [35](#)
- describe_virtual_interfaces, [35](#)
- describe_virtual_node, [20](#)
- describe_virtual_router, [20](#)
- describe_virtual_service, [20](#)
- detach_load_balancer_from_subnets, [38](#)
- directconnect, [31](#)
- disable_availability_zones_for_load_balancer, [38](#)
- disable_domain_auto_renew, [60](#)
- disable_domain_transfer_lock, [60](#)
- disable_hosted_zone_dnssec, [57](#)
- disassociate_connect_peer, [53](#)
- disassociate_connection_from_lag, [35](#)
- disassociate_customer_gateway, [53](#)
- disassociate_delegation_signer_from_domain, [60](#)
- disassociate_firewall_rule_group, [73](#)
- disassociate_gateway_from_server, [26](#)
- disassociate_link, [53](#)
- disassociate_mac_sec_key, [35](#)
- disassociate_resolver_endpoint_ip_address, [73](#)
- disassociate_resolver_query_log_config, [73](#)
- disassociate_resolver_rule, [73](#)
- disassociate_subnets, [50](#)
- disassociate_transit_gateway_connect_peer, [53](#)

- disassociate_vpc_from_hosted_zone, [57](#)
- discover_instances, [76](#)
- discover_instances_revision, [76](#)
- elb, [35](#)
- elbv2, [38](#)
- enable_availability_zones_for_load_balancer, [38](#)
- enable_domain_auto_renew, [60](#)
- enable_domain_transfer_lock, [60](#)
- enable_hosted_zone_dnssec, [57](#)
- execute_core_network_change_set, [53](#)
- export_api, [13](#)
- flush_stage_authorizers_cache, [6](#)
- flush_stage_cache, [6](#)
- generate_client_certificate, [6](#)
- get_access_log_subscription, [82](#)
- get_account, [6](#)
- get_account_limit, [57](#)
- get_api, [13](#)
- get_api_key, [6](#)
- get_api_keys, [6](#)
- get_api_mapping, [13](#)
- get_api_mappings, [13](#)
- get_apis, [13](#)
- get_app_authorization, [17](#)
- get_app_bundle, [17](#)
- get_architecture_recommendations, [69](#)
- get_auth_policy, [82](#)
- get_authorizer, [6, 13](#)
- get_authorizers, [6, 14](#)
- get_bandwidth_rate_limit_schedule, [26](#)
- get_base_path_mapping, [6](#)
- get_base_path_mappings, [6](#)
- get_cache_policy, [29](#)
- get_cache_policy_config, [29](#)
- get_cell, [69](#)
- get_cell_readiness_summary, [69](#)
- get_change, [57](#)
- get_checker_ip_ranges, [57](#)
- get_client_certificate, [6](#)
- get_client_certificates, [6](#)
- get_cloud_front_origin_access_identity, [29](#)
- get_cloud_front_origin_access_identity_config, [29](#)
- get_connect_attachment, [53](#)
- get_connect_peer, [53](#)
- get_connect_peer_associations, [53](#)
- get_connection, [11](#)
- get_connections, [53](#)
- get_contact_reachability_status, [60](#)
- get_continuous_deployment_policy, [29](#)
- get_continuous_deployment_policy_config, [29](#)
- get_core_network, [53](#)
- get_core_network_change_events, [53](#)
- get_core_network_change_set, [53](#)
- get_core_network_policy, [53](#)
- get_customer_gateway_associations, [53](#)
- get_deployment, [6, 14](#)
- get_deployments, [6, 14](#)
- get_devices, [53](#)
- get_distribution, [29](#)
- get_distribution_config, [30](#)
- get_dnssec, [57](#)
- get_documentation_part, [6](#)
- get_documentation_parts, [6](#)
- get_documentation_version, [6](#)
- get_documentation_versions, [6](#)
- get_domain_detail, [60](#)
- get_domain_name, [6, 14](#)
- get_domain_names, [6, 14](#)
- get_domain_suggestions, [60](#)
- get_export, [6](#)
- get_field_level_encryption, [30](#)
- get_field_level_encryption_config, [30](#)
- get_field_level_encryption_profile, [30](#)
- get_field_level_encryption_profile_config, [30](#)
- get_firewall_config, [73](#)
- get_firewall_domain_list, [73](#)
- get_firewall_rule_group, [73](#)
- get_firewall_rule_group_association, [73](#)
- get_firewall_rule_group_policy, [73](#)
- get_function, [30](#)
- get_gateway, [26](#)
- get_gateway_response, [6](#)
- get_gateway_responses, [6](#)
- get_geo_location, [57](#)
- get_health_check, [57](#)
- get_health_check_count, [57](#)
- get_health_check_last_failure_reason, [57](#)

get_health_check_status, 57
get_hosted_zone, 57
get_hosted_zone_count, 57
get_hosted_zone_limit, 57
get_hypervisor, 26
get_hypervisor_property_mappings, 26
get_ingestion, 17
get_ingestion_destination, 17
get_instance, 76
get_instances_health_status, 76
get_integration, 6, 14
get_integration_response, 6, 14
get_integration_responses, 14
get_integrations, 14
get_invalidation, 30
get_key_group, 30
get_key_group_config, 30
get_link_associations, 53
get_links, 53
get_listener, 82
get_managed_resource, 23
get_method, 6
get_method_response, 6
get_model, 6, 14
get_model_template, 6, 14
get_models, 6, 14
get_monitoring_subscription, 30
get_namespace, 76
get_network_resource_counts, 53
get_network_resource_relationships, 53
get_network_resources, 53
get_network_routes, 53
get_network_telemetry, 53
get_operation, 76
get_operation_detail, 60
get_origin_access_control, 30
get_origin_access_control_config, 30
get_origin_request_policy, 30
get_origin_request_policy_config, 30
get_outpost_resolver, 73
get_public_key, 30
get_public_key_config, 30
get_query_logging_config, 57
get_readiness_check, 69
get_readiness_check_resource_status, 69
get_readiness_check_status, 69
get_realtime_log_config, 30
get_recovery_group, 69
get_recovery_group_readiness_summary, 69
get_request_validator, 6
get_request_validators, 6
get_resolver_config, 73
get_resolver_dnssec_config, 73
get_resolver_endpoint, 73
get_resolver_query_log_config, 73
get_resolver_query_log_config_association, 73
get_resolver_query_log_config_policy, 73
get_resolver_rule, 73
get_resolver_rule_association, 73
get_resolver_rule_policy, 73
get_resource, 6
get_resource_policy, 53, 66, 82
get_resource_set, 69
get_resources, 6
get_response_headers_policy, 30
get_response_headers_policy_config, 30
get_rest_api, 6
get_rest_apis, 6
get_reusable_delegation_set, 57
get_reusable_delegation_set_limit, 57
get_route, 14
get_route_analysis, 53
get_route_response, 14
get_route_responses, 14
get_routes, 14
get_routing_control_state, 64
get_rule, 82
get_sdk, 6
get_sdk_type, 6
get_sdk_types, 6
get_service, 76, 82
get_service_network, 82
get_service_network_service_association, 82
get_service_network_vpc_association, 82
get_site_to_site_vpn_attachment, 53
get_sites, 53
get_sol_function_instance, 79
get_sol_function_package, 79
get_sol_function_package_content, 79
get_sol_function_package_descriptor,

- 79
- get_sol_network_instance, 79
- get_sol_network_operation, 79
- get_sol_network_package, 79
- get_sol_network_package_content, 79
- get_sol_network_package_descriptor, 79
- get_stage, 6, 14
- get_stages, 6, 14
- get_streaming_distribution, 30
- get_streaming_distribution_config, 30
- get_tags, 6, 14
- get_target_group, 83
- get_traffic_policy, 57
- get_traffic_policy_instance, 57
- get_traffic_policy_instance_count, 57
- get_transit_gateway_connect_peer_associations, 54
- get_transit_gateway_peering, 54
- get_transit_gateway_registrations, 54
- get_transit_gateway_route_table_attachment, 54
- get_trust_store_ca_certificates_bundle, 41
- get_trust_store_revocation_content, 41
- get_usage, 6
- get_usage_plan, 6
- get_usage_plan_key, 6
- get_usage_plan_keys, 6
- get_usage_plans, 6
- get_virtual_machine, 26
- get_vpc_attachment, 54
- get_vpc_link, 6, 14
- get_vpc_links, 7, 14
- globalaccelerator, 42
- import_api, 14
- import_api_keys, 7
- import_documentation_parts, 7
- import_firewall_domains, 73
- import_hypervisor_configuration, 26
- import_rest_api, 7
- instantiate_sol_network_instance, 79
- list_accelerators, 46
- list_access_log_subscriptions, 83
- list_app_authorizations, 17
- list_app_bundles, 17
- list_associated_route_53_health_checks, 66
- list_attachments, 54
- list_autoshifts, 23
- list_byoip_cidrs, 46
- list_cache_policies, 30
- list_cells, 69
- list_cidr_blocks, 57
- list_cidr_collections, 57
- list_cidr_locations, 57
- list_cloud_front_origin_access_identities, 30
- list_clusters, 66
- list_conflicting_aliases, 30
- list_connect_peers, 54
- list_continuous_deployment_policies, 30
- list_control_panels, 66
- list_core_network_policy_versions, 54
- list_core_networks, 54
- list_cross_account_attachments, 46
- list_cross_account_authorizations, 69
- list_cross_account_resource_accounts, 46
- list_cross_account_resources, 46
- list_custom_routing_accelerators, 46
- list_custom_routing_endpoint_groups, 46
- list_custom_routing_listeners, 46
- list_custom_routing_port_mappings, 46
- list_custom_routing_port_mappings_by_destination, 46
- list_distributions, 30
- list_distributions_by_cache_policy_id, 30
- list_distributions_by_key_group, 30
- list_distributions_by_origin_request_policy_id, 30
- list_distributions_by_realtime_log_config, 30
- list_distributions_by_response_headers_policy_id, 30
- list_distributions_by_web_acl_id, 30
- list_domains, 60
- list_endpoint_groups, 46
- list_field_level_encryption_configs, 30
- list_field_level_encryption_profiles, 30
- list_firewall_configs, 73

- list_firewall_domain_lists, [73](#)
- list_firewall_domains, [73](#)
- list_firewall_policies, [50](#)
- list_firewall_rule_group_associations, [73](#)
- list_firewall_rule_groups, [73](#)
- list_firewall_rules, [73](#)
- list_firewalls, [50](#)
- list_functions, [30](#)
- list_gateway_routes, [20](#)
- list_gateways, [26](#)
- list_geo_locations, [57](#)
- list_health_checks, [57](#)
- list_hosted_zones, [57](#)
- list_hosted_zones_by_name, [57](#)
- list_hosted_zones_by_vpc, [57](#)
- list_hypervisors, [26](#)
- list_ingestion_destinations, [17](#)
- list_ingestions, [17](#)
- list_instances, [76](#)
- list_invalidations, [30](#)
- list_key_groups, [30](#)
- list_key_value_stores, [30](#)
- list_listeners, [46, 83](#)
- list_managed_resources, [23](#)
- list_meshes, [20](#)
- list_namespaces, [76](#)
- list_operations, [60, 76](#)
- list_organization_service_access_status, [54](#)
- list_origin_access_controls, [30](#)
- list_origin_request_policies, [30](#)
- list_outpost_resolvers, [73](#)
- list_peerings, [54](#)
- list_prices, [60](#)
- list_public_keys, [30](#)
- list_query_logging_configs, [57](#)
- list_readiness_checks, [69](#)
- list_realtime_log_configs, [30](#)
- list_recovery_groups, [69](#)
- list_resolver_configs, [73](#)
- list_resolver_dnssec_configs, [73](#)
- list_resolver_endpoint_ip_addresses, [73](#)
- list_resolver_endpoints, [73](#)
- list_resolver_query_log_config_associations, [73](#)
- list_resolver_query_log_configs, [73](#)
- list_resolver_rule_associations, [73](#)
- list_resolver_rules, [73](#)
- list_resource_record_sets, [57](#)
- list_resource_sets, [69](#)
- list_response_headers_policies, [30](#)
- list_reusable_delegation_sets, [58](#)
- list_routes, [20](#)
- list_routing_controls, [64, 66](#)
- list_rule_groups, [50](#)
- list_rules, [69, 83](#)
- list_safety_rules, [66](#)
- list_service_network_service_associations, [83](#)
- list_service_network_vpc_associations, [83](#)
- list_service_networks, [83](#)
- list_services, [76, 83](#)
- list_sol_function_instances, [79](#)
- list_sol_function_packages, [79](#)
- list_sol_network_instances, [79](#)
- list_sol_network_operations, [79](#)
- list_sol_network_packages, [79](#)
- list_streaming_distributions, [30](#)
- list_tags_for_domain, [60](#)
- list_tags_for_resource, [17, 20, 26, 30, 46, 50, 54, 58, 66, 73, 76, 79, 83](#)
- list_tags_for_resources, [58, 69](#)
- list_target_groups, [83](#)
- list_targets, [83](#)
- list_tls_inspection_configurations, [50](#)
- list_traffic_policies, [58](#)
- list_traffic_policy_instances, [58](#)
- list_traffic_policy_instances_by_hosted_zone, [58](#)
- list_traffic_policy_instances_by_policy, [58](#)
- list_traffic_policy_versions, [58](#)
- list_virtual_gateways, [20](#)
- list_virtual_interface_test_history, [35](#)
- list_virtual_machines, [26](#)
- list_virtual_nodes, [20](#)
- list_virtual_routers, [20](#)
- list_virtual_services, [20](#)
- list_vpc_association_authorizations, [58](#)
- list_zonal_shifts, [23](#)
- modify_listener, [41](#)

- modify_load_balancer_attributes, [38, 41](#)
- modify_rule, [41](#)
- modify_target_group, [41](#)
- modify_target_group_attributes, [41](#)
- modify_trust_store, [41](#)
- networkfirewall, [47](#)
- networkmanager, [50](#)
- post_to_connection, [11](#)
- provision_byoip_cidr, [46](#)
- publish_function, [30](#)
- push_domain, [61](#)
- put_auth_policy, [83](#)
- put_bandwidth_rate_limit_schedule, [26](#)
- put_core_network_policy, [54](#)
- put_firewall_rule_group_policy, [73](#)
- put_gateway_response, [7](#)
- put_hypervisor_property_mappings, [26](#)
- put_integration, [7](#)
- put_integration_response, [7](#)
- put_maintenance_start_time, [26](#)
- put_method, [7](#)
- put_method_response, [7](#)
- put_resolver_query_log_config_policy, [73](#)
- put_resolver_rule_policy, [73](#)
- put_resource_policy, [50, 54, 83](#)
- put_rest_api, [7](#)
- put_sol_function_package_content, [79](#)
- put_sol_network_package_content, [79](#)
- register_domain, [61](#)
- register_instance, [76](#)
- register_instances_with_load_balancer, [35, 38](#)
- register_targets, [41, 83](#)
- register_transit_gateway, [54](#)
- reimport_api, [14](#)
- reject_attachment, [54](#)
- reject_domain_transfer_from_another_aws_account, [61](#)
- remove_custom_routing_endpoints, [46](#)
- remove_endpoints, [46](#)
- remove_listener_certificates, [41](#)
- remove_tags, [38, 41](#)
- remove_trust_store_revocations, [41](#)
- renew_domain, [61](#)
- resend_contact_reachability_email, [61](#)
- resend_operation_authorization, [61](#)
- reset_authorizers_cache, [14](#)
- restore_core_network_policy_version, [54](#)
- retrieve_domain_auth_code, [61](#)
- route53, [54](#)
- route53domains, [58](#)
- route53recoverycluster, [61](#)
- route53recoverycontrolconfig, [64](#)
- route53recoveryreadiness, [67](#)
- route53resolver, [70](#)
- servicediscovery, [74](#)
- set_ip_address_type, [42](#)
- set_load_balancer_listener_ssl_certificate, [38](#)
- set_load_balancer_policies_for_backend_server, [38](#)
- set_load_balancer_policies_of_listener, [38](#)
- set_rule_priorities, [42](#)
- set_security_groups, [42](#)
- set_subnets, [42](#)
- start_bgp_failover_test, [35](#)
- start_ingestion, [17](#)
- start_organization_service_access_update, [54](#)
- start_route_analysis, [54](#)
- start_user_access_tasks, [17](#)
- start_virtual_machines_metadata_sync, [26](#)
- start_zonal_shift, [23](#)
- stop_bgp_failover_test, [35](#)
- stop_ingestion, [17](#)
- tag_resource, [7, 14, 17, 20, 26, 30, 35, 46, 50, 54, 66, 69, 73, 76, 79, 83](#)
- telconetworkbuilder, [77](#)
- terminate_sol_network_instance, [79](#)
- test_dns_answer, [58](#)
- test_function, [30](#)
- test_hypervisor_configuration, [26](#)
- test_invoke_authorizer, [7](#)
- test_invoke_method, [7](#)
- transfer_domain, [61](#)
- transfer_domain_to_another_aws_account, [61](#)
- untag_resource, [7, 14, 17, 20, 26, 31, 35, 46, 50, 54, 67, 69, 73, 76, 79, 83](#)

- update_accelerator, [46](#)
- update_accelerator_attributes, [46](#)
- update_access_log_subscription, [83](#)
- update_account, [7](#)
- update_api, [14](#)
- update_api_key, [7](#)
- update_api_mapping, [14](#)
- update_app_authorization, [17](#)
- update_authorizer, [7](#), [14](#)
- update_base_path_mapping, [7](#)
- update_cache_policy, [31](#)
- update_cell, [69](#)
- update_client_certificate, [7](#)
- update_cloud_front_origin_access_identity, [31](#)
- update_connection, [35](#), [54](#)
- update_continuous_deployment_policy, [31](#)
- update_control_panel, [67](#)
- update_core_network, [54](#)
- update_cross_account_attachment, [46](#)
- update_custom_routing_accelerator, [46](#)
- update_custom_routing_accelerator_attributes, [46](#)
- update_custom_routing_listener, [46](#)
- update_deployment, [7](#), [14](#)
- update_device, [54](#)
- update_direct_connect_gateway, [35](#)
- update_direct_connect_gateway_association, [35](#)
- update_distribution, [31](#)
- update_distribution_with_staging_config, [31](#)
- update_documentation_part, [7](#)
- update_documentation_version, [7](#)
- update_domain_contact, [61](#)
- update_domain_contact_privacy, [61](#)
- update_domain_name, [7](#), [14](#)
- update_domain_nameservers, [61](#)
- update_endpoint_group, [46](#)
- update_field_level_encryption_config, [31](#)
- update_field_level_encryption_profile, [31](#)
- update_firewall_config, [73](#)
- update_firewall_delete_protection, [50](#)
- update_firewall_description, [50](#)
- update_firewall_domains, [73](#)
- update_firewall_encryption_configuration, [50](#)
- update_firewall_policy, [50](#)
- update_firewall_policy_change_protection, [50](#)
- update_firewall_rule, [73](#)
- update_firewall_rule_group_association, [74](#)
- update_function, [31](#)
- update_gateway_information, [26](#)
- update_gateway_response, [7](#)
- update_gateway_route, [20](#)
- update_gateway_software_now, [26](#)
- update_global_network, [54](#)
- update_health_check, [58](#)
- update_hosted_zone_comment, [58](#)
- update_http_namespace, [76](#)
- update_hypervisor, [26](#)
- update_ingestion_destination, [17](#)
- update_instance_custom_health_status, [76](#)
- update_integration, [7](#), [14](#)
- update_integration_response, [7](#), [14](#)
- update_key_group, [31](#)
- update_key_value_store, [31](#)
- update_lag, [35](#)
- update_link, [54](#)
- update_listener, [46](#), [83](#)
- update_logging_configuration, [50](#)
- update_mesh, [20](#)
- update_method, [7](#)
- update_method_response, [7](#)
- update_model, [7](#), [14](#)
- update_network_resource_metadata, [54](#)
- update_origin_access_control, [31](#)
- update_origin_request_policy, [31](#)
- update_outpost_resolver, [74](#)
- update_practice_run_configuration, [23](#)
- update_private_dns_namespace, [76](#)
- update_public_dns_namespace, [77](#)
- update_public_key, [31](#)
- update_readiness_check, [69](#)
- update_realtime_log_config, [31](#)
- update_recovery_group, [69](#)
- update_request_validator, [7](#)
- update_resolver_config, [74](#)
- update_resolver_dnssec_config, [74](#)
- update_resolver_endpoint, [74](#)

update_resolver_rule, [74](#)
update_resource, [7](#)
update_resource_set, [69](#)
update_response_headers_policy, [31](#)
update_rest_api, [7](#)
update_route, [14](#), [20](#)
update_route_response, [14](#)
update_routing_control, [67](#)
update_routing_control_state, [64](#)
update_routing_control_states, [64](#)
update_rule, [83](#)
update_rule_group, [50](#)
update_safety_rule, [67](#)
update_service, [77](#), [83](#)
update_service_network, [83](#)
update_service_network_vpc_association,
[83](#)
update_site, [54](#)
update_sol_function_package, [79](#)
update_sol_network_instance, [80](#)
update_sol_network_package, [80](#)
update_stage, [7](#), [14](#)
update_streaming_distribution, [31](#)
update_subnet_change_protection, [50](#)
update_tags_for_domain, [61](#)
update_target_group, [83](#)
update_tls_inspection_configuration,
[50](#)
update_traffic_policy_comment, [58](#)
update_traffic_policy_instance, [58](#)
update_usage, [7](#)
update_usage_plan, [7](#)
update_virtual_gateway, [20](#)
update_virtual_interface_attributes,
[35](#)
update_virtual_node, [20](#)
update_virtual_router, [20](#)
update_virtual_service, [20](#)
update_vpc_attachment, [54](#)
update_vpc_link, [7](#), [14](#)
update_zonal_autoshift_configuration,
[23](#)
update_zonal_shift, [23](#)

validate_sol_function_package_content,
[80](#)
validate_sol_network_package_content,
[80](#)
view_billing, [61](#)

vpclattice, [80](#)
withdraw_byoip_cidr, [46](#)