

Package ‘paws.networking’

October 7, 2026

Title 'Amazon Web Services' Networking & Content Delivery Services

Version 0.11.0

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>,
<https://paws-r.r-universe.dev/paws.networking>,
<https://www.paws-r-sdk.com>

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

Depends R (>= 3.5.0)

Imports paws.common (>= 0.9.0)

Suggests testthat

Encoding UTF-8

Config/roxygen2/version 8.1.0

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' 'cloudfrontkeyvaluestore_service.R'
'cloudfrontkeyvaluestore_interfaces.R'
'cloudfrontkeyvaluestore_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'
'route53profiles_service.R' 'route53profiles_interfaces.R'
'route53profiles_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 2026-10-07 19:50:02 UTC

Contents

apigateway	3
apigatewaymanagementapi	8
apigatewayv2	11
appfabric	15
appmesh	18

arczonalshift	21
backupgateway	24
cloudfront	27
cloudfrontkeyvaluestore	33
directconnect	36
elb	40
elbv2	43
globalaccelerator	47
networkfirewall	51
networkmanager	57
route53	61
route53domains	65
route53profiles	68
route53recoverycluster	71
route53recoverycontrolconfig	74
route53recoveryreadiness	77
route53resolver	80
servicediscovery	85
telconetworkbuilder	88
vpclattice	91

Index	96
--------------	-----------

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. • 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
use_dual_stack = "logical"
),
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
create_documentation_part	Creates a documentation part
create_documentation_version	Creates a documentation version
create_domain_name	Creates a new domain name
create_domain_name_access_association	Creates a domain name access association resource between an access association
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 A
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 asynchro
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_domain_name_access_association	Deletes the DomainNameAccessAssociation resource
delete_gateway_response	Clears any customization of a GatewayResponse of a specified response type on the given RestApi
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 than the RestApi
get_domain_name_access_associations	Represents a collection on DomainNameAccessAssociations resources
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
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 p
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 exte
put_gateway_response	Creates a customization of a GatewayResponse of a specified response type and st
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 ar
reject_domain_name_access_association	Rejects a domain name access association with a private custom domain name
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
test_invoke_method	Simulate the invocation of a Method in your RestApi with headers, parameters, an
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

<code>update_model</code>	Changes information about a model
<code>update_request_validator</code>	Updates a RequestValidator of a given RestApi
<code>update_resource</code>	Changes information about a Resource resource
<code>update_rest_api</code>	Changes information about the specified API
<code>update_stage</code>	Changes information about a Stage resource
<code>update_usage</code>	Grants a temporary extension to the remaining quota of a usage plan associated w
<code>update_usage_plan</code>	Updates a usage plan of a given plan Id
<code>update_vpc_link</code>	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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
use_dual_stack = "logical"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

```

Operations

delete_connection	Delete the connection with the provided id
get_connection	Get information about the connection with the provided id
post_to_connection	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.

• **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>

• **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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 <- 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",
    use_dual_stack = "logical"
  ),
  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_portal	Creates a portal
create_portal_product	Creates a new portal product
create_product_page	Creates a new product page for a portal product
create_product_rest_endpoint_page	Creates a product REST endpoint page for a portal product
create_route	Creates a Route for an API
create_route_response	Creates a RouteResponse for a Route
create_routing_rule	Creates a RoutingRule
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_portal	Deletes a portal
delete_portal_product	Deletes a portal product
delete_portal_product_sharing_policy	Deletes the sharing policy for a portal product
delete_product_page	Deletes a product page of a portal product
delete_product_rest_endpoint_page	Deletes a product REST endpoint page
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_routing_rule	Deletes a routing rule
delete_stage	Deletes a Stage
delete_vpc_link	Deletes a VPC link
disable_portal	Deletes the publication of a portal portal
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_portal</code>	Gets a portal
<code>get_portal_product</code>	Gets a portal product
<code>get_portal_product_sharing_policy</code>	Gets the sharing policy for a portal product
<code>get_product_page</code>	Gets a product page of a portal product
<code>get_product_rest_endpoint_page</code>	Gets a product REST endpoint page
<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_routing_rule</code>	Gets a routing rule
<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>list_portal_products</code>	Lists portal products
<code>list_portals</code>	Lists portals
<code>list_product_pages</code>	Lists the product pages for a portal product
<code>list_product_rest_endpoint_pages</code>	Lists the product REST endpoint pages of a portal product
<code>list_routing_rules</code>	Lists routing rules
<code>preview_portal</code>	Creates a portal preview
<code>publish_portal</code>	Publishes a portal
<code>put_portal_product_sharing_policy</code>	Updates the sharing policy for a portal product
<code>put_routing_rule</code>	Updates a routing rule
<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

update_model	Updates a Model
update_portal	Updates a portal
update_portal_product	Updates the portal product
update_product_page	Updates a product page of a portal product
update_product_rest_endpoint_page	Updates a product REST endpoint page
update_route	Updates a Route
update_route_response	Updates a RouteResponse
update_stage	Updates a Stage
update_vpc_link	Updates a VPC link

Examples

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

## End(Not run)
```

appfabric	<i>AppFabric</i>
-----------	------------------

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. <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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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

<code>get_ingestion</code>	Returns information about an ingestion
<code>get_ingestion_destination</code>	Returns information about an ingestion destination
<code>list_app_authorizations</code>	Returns a list of all app authorizations configured for an app bundle
<code>list_app_bundles</code>	Returns a list of app bundles
<code>list_ingestion_destinations</code>	Returns a list of all ingestion destinations configured for an ingestion
<code>list_ingestions</code>	Returns a list of all ingestions configured for an app bundle
<code>list_tags_for_resource</code>	Returns a list of tags for a resource
<code>start_ingestion</code>	Starts (enables) an ingestion, which collects data from an application
<code>start_user_access_tasks</code>	Starts the tasks to search user access status for a specific email address
<code>stop_ingestion</code>	Stops (disables) an ingestion
<code>tag_resource</code>	Assigns one or more tags (key-value pairs) to the specified resource
<code>untag_resource</code>	Removes a tag or tags from a resource
<code>update_app_authorization</code>	Updates an app authorization within an app bundle, which allows AppFabric to connect to an application
<code>update_ingestion_destination</code>	Updates an ingestion destination, which specifies how an application's ingested data is processed

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

```
apppmesh(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. <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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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

delete_virtual_router	Deletes an existing virtual router
delete_virtual_service	Deletes an existing virtual service
describe_gateway_route	Describes an existing gateway route
describe_mesh	Describes an existing service mesh
describe_route	Describes an existing route
describe_virtual_gateway	Describes an existing virtual gateway
describe_virtual_node	Describes an existing virtual node
describe_virtual_router	Describes an existing virtual router
describe_virtual_service	Describes an existing virtual service
list_gateway_routes	Returns a list of existing gateway routes that are associated to a virtual gateway
list_meshes	Returns a list of existing service meshes
list_routes	Returns a list of existing routes in a service mesh
list_tags_for_resource	List the tags for an App Mesh resource
list_virtual_gateways	Returns a list of existing virtual gateways in a service mesh
list_virtual_nodes	Returns a list of existing virtual nodes
list_virtual_routers	Returns a list of existing virtual routers in a service mesh
list_virtual_services	Returns a list of existing virtual services in a service mesh
tag_resource	Associates the specified tags to a resource with the specified resourceArn
untag_resource	Deletes specified tags from a resource
update_gateway_route	Updates an existing gateway route that is associated to a specified virtual gateway in a service mesh
update_mesh	Updates an existing service mesh
update_route	Updates an existing route for a specified service mesh and virtual router
update_virtual_gateway	Updates an existing virtual gateway in a specified service mesh
update_virtual_node	Updates an existing virtual node in a specified service mesh
update_virtual_router	Updates an existing virtual router in a specified service mesh
update_virtual_service	Updates an existing virtual service in a specified service mesh

Examples

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

## End(Not run)
```

Description

Welcome to the API Reference Guide for zonal shift and zonal autoshift in Amazon Application Recovery Controller (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 supported load balancer resources. Zonal autoshift is a capability in ARC where you authorize Amazon Web Services to shift away application resource traffic from an Availability Zone during events, on your behalf, to help reduce your time to recovery. Amazon Web Services starts an autoshift when internal telemetry indicates that there is an Availability Zone impairment that could potentially impact customers.

For more information about using zonal shift and zonal autoshift, see the [Amazon 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.

- **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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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 <- 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",
    use_dual_stack = "logical"
  ),
  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_practice_run
cancel_zonal_shift
create_practice_run_configuration
delete_practice_run_configuration
get_autoshift_observer_notification_status
get_managed_resource
list_autoshifts
list_managed_resources
list_zonal_shifts
start_practice_run
start_zonal_shift
update_autoshift_observer_notification_status
update_practice_run_configuration
update_zonal_autoshift_configuration
update_zonal_shift

```

Cancel an in-progress practice run zonal shift in Amazon Application Recovery Controller

Cancel a zonal shift in Amazon Application Recovery Controller

A practice run configuration for zonal autoshift is required when you enable zonal autoshift

Deletes the practice run configuration for a resource

Returns the status of the autoshift observer notification

Get information about a resource that's been registered for zonal shifts with Amazon Application Recovery Controller

Returns the autoshifts for an Amazon Web Services Region

Lists all the resources in your Amazon Web Services account in this Amazon Application Recovery Controller

Lists all active and completed zonal shifts in Amazon Application Recovery Controller

Start an on-demand practice run zonal shift in Amazon Application Recovery Controller

You start a zonal shift to temporarily move load balancer traffic away from an active zonal shift

Update the status of autoshift observer notification

Update a practice run configuration to change one or more of the following: a practice run configuration

The zonal autoshift configuration for a resource includes the practice run configuration

Update an active zonal shift in Amazon Application Recovery Controller in your Amazon Application Recovery Controller

Examples

```

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

## End(Not run)

```

backupgateway

AWS Backup Gateway

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.
- **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>
- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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 <- 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",
    use_dual_stack = "logical"
  ),
  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 W
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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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](#)

[associate_distribution_tenant_web_acl](#)

[associate_distribution_web_acl](#)

[copy_distribution](#)

[create_anycast_ip_list](#)

[create_cache_policy](#)

[create_cloud_front_origin_access_identity](#)

The AssociateAlias API operation only supports standard distributions

Associates the WAF web ACL with a distribution tenant

Associates the WAF web ACL with a distribution

Creates a staging distribution using the configuration of the provided principal

Creates an Anycast static IP list

Creates a cache policy

Creates a new origin access identity

<code>create_connection_function</code>	Creates a connection function
<code>create_connection_group</code>	Creates a connection group
<code>create_continuous_deployment_policy</code>	Creates a continuous deployment policy that distributes traffic for a custom
<code>create_distribution</code>	Creates a CloudFront distribution
<code>create_distribution_tenant</code>	Creates a distribution tenant
<code>create_distribution_with_tags</code>	Create a new distribution with tags
<code>create_field_level_encryption_config</code>	Create a new field-level encryption configuration
<code>create_field_level_encryption_profile</code>	Create a field-level encryption profile
<code>create_function</code>	Creates a CloudFront function
<code>create_invalidation</code>	Create a new invalidation
<code>create_invalidation_for_distribution_tenant</code>	Creates an invalidation for a distribution tenant
<code>create_key_group</code>	Creates a key group that you can use with CloudFront signed URLs and s
<code>create_key_value_store</code>	Specifies the key value store resource to add to your account
<code>create_monitoring_subscription</code>	Enables or disables additional Amazon CloudWatch metrics for the speci
<code>create_origin_access_control</code>	Creates a new origin access control in CloudFront
<code>create_origin_request_policy</code>	Creates an origin request policy
<code>create_public_key</code>	Uploads a public key to CloudFront that you can use with signed URLs a
<code>create_realtime_log_config</code>	Creates a real-time log configuration
<code>create_response_headers_policy</code>	Creates a response headers policy
<code>create_streaming_distribution</code>	This API is deprecated
<code>create_streaming_distribution_with_tags</code>	This API is deprecated
<code>create_trust_store</code>	Creates a trust store
<code>create_vpc_origin</code>	Create an Amazon CloudFront VPC origin
<code>delete_anycast_ip_list</code>	Deletes an Anycast static IP list
<code>delete_cache_policy</code>	Deletes a cache policy
<code>delete_cloud_front_origin_access_identity</code>	Delete an origin access identity
<code>delete_connection_function</code>	Deletes a connection function
<code>delete_connection_group</code>	Deletes a connection group
<code>delete_continuous_deployment_policy</code>	Deletes a continuous deployment policy
<code>delete_distribution</code>	Delete a distribution
<code>delete_distribution_tenant</code>	Deletes a distribution tenant
<code>delete_field_level_encryption_config</code>	Remove a field-level encryption configuration
<code>delete_field_level_encryption_profile</code>	Remove a field-level encryption profile
<code>delete_function</code>	Deletes a CloudFront function
<code>delete_key_group</code>	Deletes a key group
<code>delete_key_value_store</code>	Specifies the key value store to delete
<code>delete_monitoring_subscription</code>	Disables additional CloudWatch metrics for the specified CloudFront dist
<code>delete_origin_access_control</code>	Deletes a CloudFront origin access control
<code>delete_origin_request_policy</code>	Deletes an origin request policy
<code>delete_public_key</code>	Remove a public key you previously added to CloudFront
<code>delete_realtime_log_config</code>	Deletes a real-time log configuration
<code>delete_resource_policy</code>	Deletes the resource policy attached to the CloudFront resource
<code>delete_response_headers_policy</code>	Deletes a response headers policy
<code>delete_streaming_distribution</code>	Delete a streaming distribution
<code>delete_trust_store</code>	Deletes a trust store
<code>delete_vpc_origin</code>	Delete an Amazon CloudFront VPC origin
<code>describe_connection_function</code>	Describes a connection function
<code>describe_function</code>	Gets configuration information and metadata about a CloudFront function

<code>describe_key_value_store</code>	Specifies the key value store and its configuration
<code>disassociate_distribution_tenant_web_acl</code>	Disassociates a distribution tenant from the WAF web ACL
<code>disassociate_distribution_web_acl</code>	Disassociates a distribution from the WAF web ACL
<code>get_anycast_ip_list</code>	Gets an Anycast static IP list
<code>get_cache_policy</code>	Gets a cache policy, including the following metadata:
<code>get_cache_policy_config</code>	Gets a cache policy configuration
<code>get_cloud_front_origin_access_identity</code>	Get the information about an origin access identity
<code>get_cloud_front_origin_access_identity_config</code>	Get the configuration information about an origin access identity
<code>get_connection_function</code>	Gets a connection function
<code>get_connection_group</code>	Gets information about a connection group
<code>get_connection_group_by_routing_endpoint</code>	Gets information about a connection group by using the endpoint that you specify
<code>get_continuous_deployment_policy</code>	Gets a continuous deployment policy, including metadata (the policy's identifier)
<code>get_continuous_deployment_policy_config</code>	Gets configuration information about a continuous deployment policy
<code>get_distribution</code>	Get the information about a distribution
<code>get_distribution_config</code>	Get the configuration information about a distribution
<code>get_distribution_tenant</code>	Gets information about a distribution tenant
<code>get_distribution_tenant_by_domain</code>	Gets information about a distribution tenant by the associated domain
<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_invalidation_for_distribution_tenant</code>	Gets information about a specific invalidation for a distribution tenant
<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_managed_certificate_details</code>	Gets details about the CloudFront managed ACM certificate
<code>get_monitoring_subscription</code>	Gets information about whether additional CloudWatch metrics are enabled
<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_resource_policy</code>	Retrieves the resource policy for the specified CloudFront resource that you specify
<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's identifier
<code>get_streaming_distribution_config</code>	Get the configuration information about a streaming distribution
<code>get_trust_store</code>	Gets a trust store
<code>get_vpc_origin</code>	Get the details of an Amazon CloudFront VPC origin
<code>list_anycast_ip_lists</code>	Lists your Anycast static IP lists
<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>	The ListConflictingAliases API operation only supports standard distributions
<code>list_connection_functions</code>	Lists connection functions
<code>list_connection_groups</code>	Lists the connection groups in your Amazon Web Services account

list_continuous_deployment_policies	Gets a list of the continuous deployment policies in your Amazon Web Services account
list_distributions	List CloudFront distributions
list_distributions_by_anycast_ip_list_id	Lists the distributions in your account that are associated with the specified IP list
list_distributions_by_cache_policy_id	Gets a list of distribution IDs for distributions that have a cache behavior with the specified cache policy
list_distributions_by_connection_function	Lists distributions by connection function
list_distributions_by_connection_mode	Lists the distributions by the connection mode that you specify
list_distributions_by_key_group	Gets a list of distribution IDs for distributions that have a cache behavior with the specified key group
list_distributions_by_origin_request_policy_id	Gets a list of distribution IDs for distributions that have a cache behavior with the specified origin request policy
list_distributions_by_owned_resource	Lists the CloudFront distributions that are associated with the specified resource
list_distributions_by_realtime_log_config	Gets a list of distributions that have a cache behavior that's associated with the specified log configuration
list_distributions_by_response_headers_policy_id	Gets a list of distribution IDs for distributions that have a cache behavior with the specified response headers policy
list_distributions_by_trust_store	Lists distributions by trust store
list_distributions_by_vpc_origin_id	List CloudFront distributions by their VPC origin ID
list_distributions_by_web_acl_id	List the distributions that are associated with a specified WAF web ACL
list_distribution_tenants	Lists the distribution tenants in your Amazon Web Services account
list_distribution_tenants_by_customization	Lists distribution tenants by the customization that you specify
list_domain_conflicts	We recommend that you use the ListDomainConflicts API operation to check for domain conflicts
list_field_level_encryption_configs	List all field-level encryption configurations that have been created in CloudFront
list_field_level_encryption_profiles	Request a list of field-level encryption profiles that have been created in CloudFront
list_functions	Gets a list of all CloudFront functions in your Amazon Web Services account
list_invalidations	Lists invalidation batches
list_invalidations_for_distribution_tenant	Lists the invalidations for a distribution tenant
list_key_groups	Gets a list of key groups
list_key_value_stores	Specifies the key value stores to list
list_origin_access_controls	Gets the list of CloudFront origin access controls (OACs) in this Amazon Web Services account
list_origin_request_policies	Gets a list of origin request policies
list_public_keys	List all public keys that have been added to CloudFront for this account
list_realtime_log_configs	Gets a list of real-time log configurations
list_response_headers_policies	Gets a list of response headers policies
list_streaming_distributions	List streaming distributions
list_tags_for_resource	List tags for a CloudFront resource
list_trust_stores	Lists trust stores
list_vpc_origins	List the CloudFront VPC origins in your account
publish_connection_function	Publishes a connection function
publish_function	Publishes a CloudFront function by copying the function code from the local file system
put_resource_policy	Creates a resource control policy for a given CloudFront resource
tag_resource	Add tags to a CloudFront resource
test_connection_function	Tests a connection function
test_function	Tests a CloudFront function
untag_resource	Remove tags from a CloudFront resource
update_anycast_ip_list	Updates an Anycast static IP list
update_cache_policy	Updates a cache policy configuration
update_cloud_front_origin_access_identity	Update an origin access identity
update_connection_function	Updates a connection function
update_connection_group	Updates a connection group
update_continuous_deployment_policy	Updates a continuous deployment policy
update_distribution	Updates the configuration for a CloudFront distribution
update_distribution_tenant	Updates a distribution tenant

update_distribution_with_staging_config	Copies the staging distribution's configuration to its corresponding primary distribution
update_domain_association	We recommend that you use the UpdateDomainAssociation API operation
update_field_level_encryption_config	Update a field-level encryption configuration
update_field_level_encryption_profile	Update a field-level encryption profile
update_function	Updates a CloudFront function
update_key_group	Updates a key group
update_key_value_store	Specifies the key value store to update
update_origin_access_control	Updates a CloudFront origin access control
update_origin_request_policy	Updates an origin request policy configuration
update_public_key	Update public key information
update_realtime_log_config	Updates a real-time log configuration
update_response_headers_policy	Updates a response headers policy
update_streaming_distribution	Update a streaming distribution
update_trust_store	Updates a trust store
update_vpc_origin	Update an Amazon CloudFront VPC origin in your account
verify_dns_configuration	Verify the DNS configuration for your domain names

Examples

```
## Not run:
svc <- cloudfront()
svc$associate_alias(
  Foo = 123
)

## End(Not run)
```

cloudfrontkeyvaluestore

Amazon CloudFront KeyValueStore

Description

Amazon CloudFront KeyValueStore Service to View and Update Data in a KVS Resource

Usage

```
cloudfrontkeyvaluestore(
  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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- cloudfrontkeyvaluestore(
  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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

delete_key	Deletes the key value pair specified by the key
describe_key_value_store	Returns metadata information about Key Value Store
get_key	Returns a key value pair
list_keys	Returns a list of key value pairs
put_key	Creates a new key value pair or replaces the value of an existing key
update_keys	Puts or Deletes multiple key value pairs in a single, all-or-nothing operation

Examples

```

## Not run:
svc <- cloudfrontkeyvaluestore()

```

```

svc$delete_key(
  Foo = 123
)

## End(Not run)

```

directconnect

AWS Direct Connect

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 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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
```

```

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](#)
[allocate_connection_on_interconnect](#)
[allocate_hosted_connection](#)
[allocate_private_virtual_interface](#)
[allocate_public_virtual_interface](#)
[allocate_transit_virtual_interface](#)
[associate_connections_to_resiliency_group](#)
[associate_connection_with_lag](#)
[associate_hosted_connection](#)
[associate_mac_sec_key](#)
[associate_virtual_interface](#)
[confirm_connection](#)
[confirm_customer_agreement](#)
[confirm_private_virtual_interface](#)
[confirm_public_virtual_interface](#)
[confirm_transit_virtual_interface](#)
[create_bgp_peer](#)
[create_connection](#)
[create_direct_connect_gateway](#)
[create_direct_connect_gateway_association](#)
[create_direct_connect_gateway_association_proposal](#)
[create_interconnect](#)
[create_lag](#)
[create_private_virtual_interface](#)
[create_public_virtual_interface](#)
[create_resiliency_group](#)
[create_transit_virtual_interface](#)
[delete_bgp_peer](#)
[delete_connection](#)
[delete_direct_connect_gateway](#)
[delete_direct_connect_gateway_association](#)
[delete_direct_connect_gateway_association_proposal](#)
[delete_interconnect](#)

Accepts a proposal request to attach a virtual private gateway or transit virtual gateway to a Direct Connect gateway.
 Deprecated
 Creates a hosted connection on the specified interconnect or a link aggregation group (LAG).
 Provisions a private virtual interface to be owned by the specified Amazon Web Services account.
 Provisions a public virtual interface to be owned by the specified Amazon Web Services account.
 Provisions a transit virtual interface to be owned by the specified Amazon Web Services account.
 Associates one or more connections with the specified resiliency group.
 Associates an existing connection with a link aggregation group (LAG).
 Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG).
 Associates a MAC Security (MACsec) Connection Key Name (CKN) with a virtual interface.
 Associates a virtual interface with a specified link aggregation group (LAG).
 Confirms the creation of the specified hosted connection on an interconnect.
 The confirmation of the terms of agreement when creating the connection.
 Accepts ownership of a private virtual interface created by another Amazon Web Services account.
 Accepts ownership of a public virtual interface created by another Amazon Web Services account.
 Accepts ownership of a transit virtual interface created by another Amazon Web Services account.
 Creates a BGP peer on the specified virtual interface.
 Creates a connection between a customer network and a specific Direct Connect endpoint.
 Creates a Direct Connect gateway, which is an intermediate object for creating Direct Connect connections.
 Creates an association between a Direct Connect gateway and a virtual private gateway or transit virtual gateway.
 Creates a proposal to associate the specified virtual private gateway or transit virtual gateway with a Direct Connect gateway.
 Creates an interconnect between an Direct Connect Partner's network and an Amazon Web Services account.
 Creates a link aggregation group (LAG) with the specified number of members.
 Creates a private virtual interface.
 Creates a public virtual interface.
 Creates a resiliency group.
 Creates a transit virtual interface.
 Deletes the specified BGP peer on the specified virtual interface with the specified BGP peer ID.
 Deletes the specified connection.
 Deletes the specified Direct Connect gateway.
 Deletes the association between the specified Direct Connect gateway and a virtual private gateway or transit virtual gateway.
 Deletes the association proposal request between the specified Direct Connect gateway and a virtual private gateway or transit virtual gateway.
 Deletes the specified interconnect.

<code>delete_lag</code>	Deletes the specified link aggregation group (LAG)
<code>delete_resiliency_group</code>	Deletes the specified resiliency group
<code>delete_virtual_interface</code>	Deletes a virtual interface
<code>describe_connection_loa</code>	Deprecated
<code>describe_connections</code>	Displays the specified connection or all connections in this Region
<code>describe_connections_on_interconnect</code>	Deprecated
<code>describe_customer_metadata</code>	Get and view a list of customer agreements, along with their signed versions
<code>describe_direct_connect_gateway_association_proposals</code>	Describes one or more association proposals for connection between your Direct Connect gateway and a virtual interface
<code>describe_direct_connect_gateway_associations</code>	Lists the associations between your Direct Connect gateways and virtual interfaces
<code>describe_direct_connect_gateway_attachments</code>	Lists the attachments between your Direct Connect gateways and virtual interfaces
<code>describe_direct_connect_gateways</code>	Lists all your Direct Connect gateways or only the specified Direct Connect gateway
<code>describe_hosted_connections</code>	Lists the hosted connections that have been provisioned on the specified Direct Connect gateway
<code>describe_interconnect_loa</code>	Deprecated
<code>describe_interconnects</code>	Lists the interconnects owned by the Amazon Web Services account
<code>describe_lags</code>	Describes all your link aggregation groups (LAG) or the specified LAG
<code>describe_loa</code>	Gets the LOA-CFA for a connection, interconnect, or link aggregation group
<code>describe_locations</code>	Lists the Direct Connect locations in the current Amazon Web Services Region
<code>describe_router_configuration</code>	Details about the router
<code>describe_tags</code>	Describes the tags associated with the specified Direct Connect resource
<code>describe_virtual_gateways</code>	Deprecated
<code>describe_virtual_interfaces</code>	Displays all virtual interfaces for an Amazon Web Services account
<code>disassociate_connection_from_lag</code>	Disassociates a connection from a link aggregation group (LAG)
<code>disassociate_connections_from_resiliency_group</code>	Disassociates one or more connections from the specified resiliency group
<code>disassociate_mac_sec_key</code>	Removes the association between a MAC Security (MACsec) security profile and a connection
<code>get_resiliency_group</code>	Gets information about the specified resiliency group
<code>list_resiliency_group_associations</code>	Lists the connection associations for the specified resiliency group
<code>list_resiliency_groups</code>	Lists the resiliency groups owned by your Amazon Web Services account
<code>list_virtual_interface_routes</code>	Lists the routes for the specified virtual interface
<code>list_virtual_interface_test_history</code>	Lists the virtual interface failover test history
<code>start_bgp_failover_test</code>	Starts the virtual interface failover test that verifies your configuration
<code>stop_bgp_failover_test</code>	Stops the virtual interface failover test
<code>tag_resource</code>	Adds the specified tags to the specified Direct Connect resource
<code>untag_resource</code>	Removes one or more tags from the specified Direct Connect resource
<code>update_connection</code>	Updates the Direct Connect connection configuration
<code>update_connections_billing_mode</code>	Updates the billing mode for the specified Direct Connect connection
<code>update_direct_connect_gateway</code>	Updates the name of a current Direct Connect gateway
<code>update_direct_connect_gateway_association</code>	Updates the specified attributes of the Direct Connect gateway association
<code>update_lag</code>	Updates the attributes of the specified link aggregation group (LAG)
<code>update_resiliency_group</code>	Updates the name of the specified resiliency group
<code>update_virtual_interface_attributes</code>	Updates the specified attributes of the specified virtual private interface

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 | 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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
        use_dual_stack = "logical"
    ),
    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 private cloud
attach_load_balancer_to_subnets	Adds one or more subnets to the set of configured subnets for the specified load balancer
configure_health_check	Specifies the health check settings to use when evaluating the health state of the specified instances
create_app_cookie_stickiness_policy	Generates a stickiness policy with sticky session lifetimes that follow that of the application
create_lb_cookie_stickiness_policy	Generates a stickiness policy with sticky session lifetimes controlled by the load balancer
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 account
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 the 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 load balancer
disable_availability_zones_for_load_balancer	Removes the specified Availability Zones from the set of Availability Zones for the load balancer
enable_availability_zones_for_load_balancer	Adds the specified Availability Zones to the set of Availability Zones for the 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 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, UDP, and QUIC.
- 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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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](#)
[add_tags](#)
[add_trust_store_revocations](#)
[create_listener](#)
[create_load_balancer](#)
[create_rule](#)
[create_target_group](#)

Adds the specified SSL server certificate to the certificate list for the specified HTTP
 Adds the specified tags to the specified Elastic Load Balancing resource
 Adds the specified revocation file to the specified trust store
 Creates a listener for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
 Creates an Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
 Creates a rule for the specified listener
 Creates a target group

<code>create_trust_store</code>	Creates a trust store
<code>delete_listener</code>	Deletes the specified listener
<code>delete_load_balancer</code>	Deletes the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
<code>delete_rule</code>	Deletes the specified rule
<code>delete_shared_trust_store_association</code>	Deletes a shared trust store association
<code>delete_target_group</code>	Deletes the specified target group
<code>delete_trust_store</code>	Deletes a trust store
<code>deregister_targets</code>	Deregisters the specified targets from the specified target group
<code>describe_account_limits</code>	Describes the current Elastic Load Balancing resource limits for your Amazon Web Services account
<code>describe_capacity_reservation</code>	Describes the capacity reservation status for the specified load balancer
<code>describe_listener_attributes</code>	Describes the attributes for the specified listener
<code>describe_listener_certificates</code>	Describes the default certificate and the certificate list for the specified HTTPS or TLS listener
<code>describe_listeners</code>	Describes the specified listeners or the listeners for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
<code>describe_load_balancer_attributes</code>	Describes the attributes for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
<code>describe_load_balancers</code>	Describes the specified load balancers or all of your load balancers
<code>describe_rules</code>	Describes the specified rules or the rules for the specified listener
<code>describe_ssl_policies</code>	Describes the specified policies or all policies used for SSL negotiation
<code>describe_tags</code>	Describes the tags for the specified Elastic Load Balancing resources
<code>describe_target_group_attributes</code>	Describes the attributes for the specified target group
<code>describe_target_groups</code>	Describes the specified target groups or all of your target groups
<code>describe_target_health</code>	Describes the health of the specified targets or all of your targets
<code>describe_trust_store_associations</code>	Describes all resources associated with the specified trust store
<code>describe_trust_store_revocations</code>	Describes the revocation files in use by the specified trust store or revocation files
<code>describe_trust_stores</code>	Describes all trust stores for the specified account
<code>get_resource_policy</code>	Retrieves the resource policy for a specified resource
<code>get_trust_store_ca_certificates_bundle</code>	Retrieves the ca certificate bundle
<code>get_trust_store_revocation_content</code>	Retrieves the specified revocation file
<code>modify_capacity_reservation</code>	Modifies the capacity reservation of the specified load balancer
<code>modify_ip_pools</code>	[Application Load Balancers] Modify the IP pool associated to a load balancer
<code>modify_listener</code>	Replaces the specified properties of the specified listener
<code>modify_listener_attributes</code>	Modifies the specified attributes of the specified listener
<code>modify_load_balancer_attributes</code>	Modifies the specified attributes of the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
<code>modify_rule</code>	Replaces the specified properties of the specified rule
<code>modify_target_group</code>	Modifies the health checks used when evaluating the health state of the targets in the specified target group
<code>modify_target_group_attributes</code>	Modifies the specified attributes of the specified target group
<code>modify_trust_store</code>	Update the ca certificate bundle for the specified trust store
<code>register_targets</code>	Registers the specified targets with the specified target group
<code>remove_listener_certificates</code>	Removes the specified certificate from the certificate list for the specified HTTPS or TLS listener
<code>remove_tags</code>	Removes the specified tags from the specified Elastic Load Balancing resources
<code>remove_trust_store_revocations</code>	Removes the specified revocation file from the specified trust store
<code>set_ip_address_type</code>	Sets the type of IP addresses used by the subnets of the specified load balancer
<code>set_rule_priorities</code>	Sets the priorities of the specified rules
<code>set_security_groups</code>	Associates the specified security groups with the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
<code>set_subnets</code>	Enables the Availability Zones for the specified public subnets for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer

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.
--------	---

- **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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
        use_dual_stack = "logical"
    ),
    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](#)
[add_endpoints](#)
[advertise_byoip_cidr](#)
[allow_custom_routing_traffic](#)
[create_accelerator](#)
[create_cross_account_attachment](#)
[create_custom_routing_accelerator](#)
[create_custom_routing_endpoint_group](#)
[create_custom_routing_listener](#)
[create_endpoint_group](#)
[create_listener](#)
[delete_accelerator](#)
[delete_cross_account_attachment](#)
[delete_custom_routing_accelerator](#)
[delete_custom_routing_endpoint_group](#)
[delete_custom_routing_listener](#)
[delete_endpoint_group](#)
[delete_listener](#)
[deny_custom_routing_traffic](#)
[deprovision_byoip_cidr](#)
[describe_accelerator](#)
[describe_accelerator_attributes](#)
[describe_cross_account_attachment](#)
[describe_custom_routing_accelerator](#)
[describe_custom_routing_accelerator_attributes](#)
[describe_custom_routing_endpoint_group](#)

Associate a virtual private cloud (VPC) subnet endpoint with your custom routing accelerator
 Add endpoints to an endpoint group
 Advertises an IPv4 address range that is provisioned for use with your custom routing accelerator
 Specify the Amazon EC2 instance (destination) IP addresses and ports for a custom routing accelerator
 Create an accelerator
 Create a cross-account attachment in Global Accelerator
 Create a custom routing accelerator
 Create an endpoint group for the specified listener for a custom routing accelerator
 Create a listener to process inbound connections from clients to a custom routing accelerator
 Create an endpoint group for the specified listener
 Create a listener to process inbound connections from clients to an accelerator
 Delete an accelerator
 Delete a cross-account attachment
 Delete a custom routing accelerator
 Delete an endpoint group from a listener for a custom routing accelerator
 Delete a listener for a custom routing accelerator
 Delete an endpoint group from a listener
 Delete a listener from an accelerator
 Specify the Amazon EC2 instance (destination) IP addresses and ports for a custom routing accelerator
 Releases the specified address range that you provisioned to use with your custom routing accelerator
 Describe an accelerator
 Describe the attributes of an accelerator
 Gets configuration information about a cross-account attachment
 Describe a custom routing accelerator
 Describe the attributes of a custom routing accelerator
 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 ProvisionByoipCidr
<code>list_cross_account_attachments</code>	List the cross-account attachments that have been created in Global Accelerator
<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 account
<code>list_custom_routing_endpoint_groups</code>	List the endpoint groups that are associated with a listener for a custom routing accelerator
<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 and port to the private IP address and port
<code>list_custom_routing_port_mappings_by_destination</code>	List the port mappings for a specific EC2 instance (destination) in a VPC
<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 account
<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 resources
<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 pool

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 view the complete list of Amazon Web Services Regions where Network Firewall is available, see [Service endpoints and quotas](#) in the *Amazon Web Services General Reference*.

To access Network Firewall using the IPv4 REST API endpoint: `https://network-firewall.<region>.amazonaws.com`

To access Network Firewall using the Dualstack (IPv4 and IPv6) 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 7.0.8. For information about Suricata, see the [Suricata website](#) and the [Suricata User Guide](#).

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, define the firewall behavior as follows:
 - (a) Create stateless and stateful rule groups, to define the components of the network traffic filtering behavior that you want your firewall to have.
 - (b) Create a firewall policy that uses your rule groups and specifies additional default traffic filtering behavior.

4. 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.
5. In Amazon VPC, use ingress routing enhancements to route traffic through the new firewall endpoints.

After your firewall is established, you can add firewall endpoints for new Availability Zones by following the prior steps for the Amazon VPC setup and firewall subnet definitions. You can also add endpoints to Availability Zones that you're using in the firewall, either for the same VPC or for another VPC, by following the prior steps for the Amazon VPC setup, and defining the new VPC subnets as VPC endpoint associations.

Usage

```
networkfirewall(
  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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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 <- 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",
    use_dual_stack = "logical"
  ),
  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>accept_network_firewall_transit_gateway_attachment</code>	Accepts a transit gateway attachment request for Network Firewall
<code>associate_availability_zones</code>	Associates the specified Availability Zones with a transit gateway-atta
<code>associate_firewall_policy</code>	Associates a FirewallPolicy to a Firewall
<code>associate_subnets</code>	Associates the specified subnets in the Amazon VPC to the firewall
<code>attach_rule_groups_to_proxy_configuration</code>	Attaches ProxyRuleGroup resources to a ProxyConfiguration
<code>create_container_association</code>	Creates a Network Firewall container association
<code>create_firewall</code>	Creates an Network Firewall Firewall and accompanying FirewallStat
<code>create_firewall_policy</code>	Creates the firewall policy for the firewall according to the specificatio
<code>create_proxy</code>	Creates an Network Firewall Proxy
<code>create_proxy_configuration</code>	Creates an Network Firewall ProxyConfiguration
<code>create_proxy_rule_group</code>	Creates an Network Firewall ProxyRuleGroup
<code>create_proxy_rules</code>	Creates Network Firewall ProxyRule resources
<code>create_rule_group</code>	Creates the specified stateless or stateful rule group, which includes th
<code>create_tls_inspection_configuration</code>	Creates an Network Firewall TLS inspection configuration
<code>create_vpc_endpoint_association</code>	Creates a firewall endpoint for an Network Firewall firewall
<code>delete_container_association</code>	Deletes a container association
<code>delete_firewall</code>	Deletes the specified Firewall and its FirewallStatus
<code>delete_firewall_policy</code>	Deletes the specified FirewallPolicy
<code>delete_network_firewall_transit_gateway_attachment</code>	Deletes a transit gateway attachment from a Network Firewall
<code>delete_proxy</code>	Deletes the specified Proxy
<code>delete_proxy_configuration</code>	Deletes the specified ProxyConfiguration
<code>delete_proxy_rule_group</code>	Deletes the specified ProxyRuleGroup
<code>delete_proxy_rules</code>	Deletes the specified ProxyRule(s)
<code>delete_resource_policy</code>	Deletes a resource policy that you created in a PutResourcePolicy requ
<code>delete_rule_group</code>	Deletes the specified RuleGroup
<code>delete_tls_inspection_configuration</code>	Deletes the specified TLSInspectionConfiguration
<code>delete_vpc_endpoint_association</code>	Deletes the specified VpcEndpointAssociation
<code>describe_container_association</code>	Retrieves the configuration and status of a container association
<code>describe_firewall</code>	Returns the data objects for the specified firewall
<code>describe_firewall_metadata</code>	Returns the high-level information about a firewall, including the Avail
<code>describe_firewall_policy</code>	Returns the data objects for the specified firewall policy
<code>describe_flow_operation</code>	Returns key information about a specific flow operation
<code>describe_logging_configuration</code>	Returns the logging configuration for the specified firewall
<code>describe_proxy</code>	Returns the data objects for the specified proxy
<code>describe_proxy_configuration</code>	Returns the data objects for the specified proxy configuration
<code>describe_proxy_rule</code>	Returns the data objects for the specified proxy configuration for the s
<code>describe_proxy_rule_group</code>	Returns the data objects for the specified proxy rule group
<code>describe_resource_policy</code>	Retrieves a resource policy that you created in a PutResourcePolicy re
<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
<code>describe_rule_group_summary</code>	Returns detailed information for a stateful rule group
<code>describe_tls_inspection_configuration</code>	Returns the data objects for the specified TLS inspection configuration
<code>describe_vpc_endpoint_association</code>	Returns the data object for the specified VPC endpoint association
<code>detach_rule_groups_from_proxy_configuration</code>	Detaches ProxyRuleGroup resources from a ProxyConfiguration

<code>disassociate_availability_zones</code>	Removes the specified Availability Zone associations from a transit gateway
<code>disassociate_subnets</code>	Removes the specified subnet associations from the firewall
<code>get_analysis_report_results</code>	The results of a COMPLETED analysis report generated with StartAnalysis
<code>list_analysis_reports</code>	Returns a list of all traffic analysis reports generated within the last 30 days
<code>list_container_associations</code>	Lists the container associations in your account and Region
<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_flow_operation_results</code>	Returns the results of a specific flow operation
<code>list_flow_operations</code>	Returns a list of all flow operations ran in a specific firewall
<code>list_proxies</code>	Retrieves the metadata for the proxies that you have defined
<code>list_proxy_configurations</code>	Retrieves the metadata for the proxy configuration that you have defined
<code>list_proxy_rule_groups</code>	Retrieves the metadata for the proxy rule groups 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 defined
<code>list_vpc_endpoint_associations</code>	Retrieves the metadata for the VPC endpoint associations that you have defined
<code>put_resource_policy</code>	Creates or updates an IAM policy for your rule group, firewall policy, or transit gateway
<code>reject_network_firewall_transit_gateway_attachment</code>	Rejects a transit gateway attachment request for Network Firewall
<code>start_analysis_report</code>	Generates a traffic analysis report for the timeframe and traffic type you specify
<code>start_flow_capture</code>	Begins capturing the flows in a firewall, according to the filters you define
<code>start_flow_flush</code>	Begins the flushing of traffic from the firewall, according to the filters you define
<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_availability_zone_change_protection</code>	Modifies the AvailabilityZoneChangeProtection setting for a transit gateway
<code>update_container_association</code>	Updates the monitoring configurations and description of a container association
<code>update_firewall_analysis_settings</code>	Enables specific types of firewall analysis on a specific firewall you define
<code>update_firewall_delete_protection</code>	Modifies the flag, DeleteProtection, which indicates whether it is possible to delete the firewall
<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 rule groups
<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 change the firewall policy
<code>update_logging_configuration</code>	Sets the logging configuration for the specified firewall
<code>update_proxy</code>	Updates the properties of the specified proxy
<code>update_proxy_configuration</code>	Updates the properties of the specified proxy configuration
<code>update_proxy_rule</code>	Updates the properties of the specified proxy rule
<code>update_proxy_rule_group_priorities</code>	Updates proxy rule group priorities within a proxy configuration
<code>update_proxy_rule_priorities</code>	Updates proxy rule priorities within a proxy rule group
<code>update_proxy_settings</code>	Modifies the proxy listener configuration of a proxy mode 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 inspection configuration

Examples

```
## Not run:
svc <- networkfirewall()
svc$accept_network_firewall_transit_gateway_attachment(
```

```

    Foo = 123
)

## End(Not run)

```

networkmanager

AWS Network Manager

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.

- **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>

	• use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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>accept_attachment</code>	Accepts a core network attachment request
<code>associate_connect_peer</code>	Associates a core network Connect peer with a device and optionally, with a link
<code>associate_customer_gateway</code>	Associates a customer gateway with a device and optionally, with a link
<code>associate_link</code>	Associates a link to a device
<code>associate_transit_gateway_connect_peer</code>	Associates a transit gateway Connect peer with a device, and optionally, with a link
<code>create_connect_attachment</code>	Creates a core network Connect attachment from a specified core network connect peer
<code>create_connection</code>	Creates a connection between two devices
<code>create_connect_peer</code>	Creates a core network Connect peer for a specified core network connect peer
<code>create_core_network</code>	Creates a core network as part of your global network, and optionally, with a link
<code>create_core_network_prefix_list_association</code>	Creates an association between a core network and a prefix list for routing
<code>create_device</code>	Creates a new device in a global network
<code>create_direct_connect_gateway_attachment</code>	Creates an Amazon Web Services Direct Connect gateway attachment
<code>create_global_network</code>	Creates a new, empty global network
<code>create_link</code>	Creates a new link for a specified site
<code>create_site</code>	Creates a new site in a global network
<code>create_site_to_site_vpn_attachment</code>	Creates an Amazon Web Services site-to-site VPN attachment on an edge location
<code>create_transit_gateway_peering</code>	Creates a transit gateway peering connection
<code>create_transit_gateway_route_table_attachment</code>	Creates a transit gateway route table attachment
<code>create_vpc_attachment</code>	Creates a VPC attachment on an edge location of a core network
<code>delete_attachment</code>	Deletes an attachment
<code>delete_connection</code>	Deletes the specified connection in your global network
<code>delete_connect_peer</code>	Deletes a Connect peer
<code>delete_core_network</code>	Deletes a core network along with all core network policies
<code>delete_core_network_policy_version</code>	Deletes a policy version from a core network
<code>delete_core_network_prefix_list_association</code>	Deletes an association between a core network and a prefix list
<code>delete_device</code>	Deletes an existing device
<code>delete_global_network</code>	Deletes an existing global network
<code>delete_link</code>	Deletes an existing link
<code>delete_peering</code>	Deletes an existing peering connection
<code>delete_resource_policy</code>	Deletes a resource policy for the specified resource
<code>delete_site</code>	Deletes an existing site
<code>deregister_transit_gateway</code>	Deregisters a transit gateway from your global network
<code>describe_global_networks</code>	Describes one or more global networks
<code>disassociate_connect_peer</code>	Disassociates a core network Connect peer from a device and a link
<code>disassociate_customer_gateway</code>	Disassociates a customer gateway from a device and a link
<code>disassociate_link</code>	Disassociates an existing device from a link
<code>disassociate_transit_gateway_connect_peer</code>	Disassociates a transit gateway Connect peer from a device and link
<code>execute_core_network_change_set</code>	Executes a change set on your core network

<code>get_connect_attachment</code>	Returns information about a core network Connect attachment
<code>get_connections</code>	Gets information about one or more of your connections in a global network
<code>get_connect_peer</code>	Returns information about a core network Connect peer
<code>get_connect_peer_associations</code>	Returns information about a core network Connect peer associations
<code>get_core_network</code>	Returns information about the LIVE policy for a core network
<code>get_core_network_change_events</code>	Returns information about a core network change event
<code>get_core_network_change_set</code>	Returns a change set between the LIVE core network policy and a submitted change set
<code>get_core_network_policy</code>	Returns details about a core network policy
<code>get_customer_gateway_associations</code>	Gets the association information for customer gateways that are associated with a core network
<code>get_devices</code>	Gets information about one or more of your devices in a global network
<code>get_direct_connect_gateway_attachment</code>	Returns information about a specific Amazon Web Services Direct Connect gateway attachment
<code>get_link_associations</code>	Gets the link associations for a device or a link
<code>get_links</code>	Gets information about one or more links in a specified global network
<code>get_network_resource_counts</code>	Gets the count of network resources, by resource type, for the specified global network
<code>get_network_resource_relationships</code>	Gets the network resource relationships for the specified global network
<code>get_network_resources</code>	Describes the network resources for the specified global network
<code>get_network_routes</code>	Gets the network routes of the specified global network
<code>get_network_telemetry</code>	Gets the network telemetry of the specified global network
<code>get_resource_policy</code>	Returns information about a resource policy
<code>get_route_analysis</code>	Gets information about the specified route analysis
<code>get_sites</code>	Gets information about one or more of your sites in a global network
<code>get_site_to_site_vpn_attachment</code>	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 associations
<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 global network
<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_attachment_routing_policy_associations</code>	Lists the routing policy associations for attachments in a core network
<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_network_prefix_list_associations</code>	Lists the prefix list associations for a core network
<code>list_core_network_routing_information</code>	Lists routing information for a core network, including routes and their attributes
<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 account
<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_attachment_routing_policy_label</code>	Applies a routing policy label to an attachment for traffic routing decisions
<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>remove_attachment_routing_policy_label</code>	Removes a routing policy label from an attachment
<code>restore_core_network_policy_version</code>	Restores a previous policy version as a new, immutable version of a core network policy
<code>start_organization_service_access_update</code>	Enables the Network Manager service for an Amazon Web Services Organization
<code>start_route_analysis</code>	Starts analyzing the routing path between the specified source and destination
<code>tag_resource</code>	Tags a specified resource
<code>untag_resource</code>	Removes tags from a specified resource

update_connection	Updates the information for an existing connection
update_core_network	Updates the description of a core network
update_device	Updates the details for an existing device
update_direct_connect_gateway_attachment	Updates the edge locations associated with an Amazon Web Services Direct Connect gateway
update_global_network	Updates an existing global network
update_link	Updates the details for an existing link
update_network_resource_metadata	Updates the resource metadata for the specified global network
update_site	Updates the information for an existing site
update_vpc_attachment	Updates a VPC attachment

Examples

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

## End(Not run)
```

route53	Amazon Route 53
---------	-----------------

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	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. <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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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](#)
[create_key_signing_key](#)
[create_query_logging_config](#)
[create_reusable_delegation_set](#)
[create_traffic_policy](#)

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
 Creates a new key-signing key (KSK) associated with a hosted zone
 Creates a configuration for DNS query logging
 Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones
 Creates a traffic policy, which you use to create multiple DNS resource record sets

<code>create_traffic_policy_instance</code>	Creates resource record sets in a specified hosted zone based on the settings in the specified traffic policy
<code>create_traffic_policy_version</code>	Creates a new version of an existing traffic policy
<code>create_vpc_association_authorization</code>	Authorizes the Amazon Web Services account that created a specified VPC to associate the VPC with a hosted zone
<code>deactivate_key_signing_key</code>	Deactivates a key-signing key (KSK) so that it will not be used for signing by Route 53
<code>delete_cidr_collection</code>	Deletes a CIDR collection in the current Amazon Web Services account
<code>delete_health_check</code>	Deletes a health check
<code>delete_hosted_zone</code>	Deletes a hosted zone
<code>delete_key_signing_key</code>	Deletes a key-signing key (KSK)
<code>delete_query_logging_config</code>	Deletes a configuration for DNS query logging
<code>delete_reusable_delegation_set</code>	Deletes a reusable delegation set
<code>delete_traffic_policy</code>	Deletes a traffic policy
<code>delete_traffic_policy_instance</code>	Deletes a traffic policy instance and all of the resource record sets that Amazon Route 53 has created for the instance
<code>delete_vpc_association_authorization</code>	Removes authorization to submit an AssociateVPCWithHostedZone request to the current Amazon Web Services account
<code>disable_hosted_zone_dnssec</code>	Disables DNSSEC signing in a specific hosted zone
<code>disassociate_vpc_from_hosted_zone</code>	Disassociates an Amazon Virtual Private Cloud (Amazon VPC) from an Amazon Route 53 hosted zone
<code>enable_hosted_zone_dnssec</code>	Enables DNSSEC signing in a specific hosted zone
<code>get_account_limit</code>	Gets the specified limit for the current account, for example, the maximum number of hosted zones that you can associate with the specified account
<code>get_change</code>	Returns the current status of a change batch request
<code>get_checker_ip_ranges</code>	Route 53 does not perform authorization for this API because it retrieves information from Amazon Route 53
<code>get_dnssec</code>	Returns information about DNSSEC for a specific hosted zone, including the status of the zone's DNSSEC signing
<code>get_geo_location</code>	Gets information about whether a specified geographic location is supported for a specific hosted zone
<code>get_health_check</code>	Gets information about a specified health check
<code>get_health_check_count</code>	Retrieves the number of health checks that are associated with the current Amazon Web Services account
<code>get_health_check_last_failure_reason</code>	Gets the reason that a specified health check failed most recently
<code>get_health_check_status</code>	Gets status of a specified health check
<code>get_hosted_zone</code>	Gets information about a specified hosted zone including the four name servers that are associated with the zone
<code>get_hosted_zone_count</code>	Retrieves the number of hosted zones that are associated with the current Amazon Web Services account
<code>get_hosted_zone_limit</code>	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
<code>get_query_logging_config</code>	Gets information about a specified configuration for DNS query logging
<code>get_reusable_delegation_set</code>	Retrieves information about a specified reusable delegation set, including the status of the set
<code>get_reusable_delegation_set_limit</code>	Gets the maximum number of hosted zones that you can associate with the specified account
<code>get_traffic_policy</code>	Gets information about a specific traffic policy version
<code>get_traffic_policy_instance</code>	Gets information about a specified traffic policy instance
<code>get_traffic_policy_instance_count</code>	Gets the number of traffic policy instances that are associated with the current Amazon Web Services account
<code>list_cidr_blocks</code>	Returns a paginated list of location objects and their CIDR blocks
<code>list_cidr_collections</code>	Returns a paginated list of CIDR collections in the Amazon Web Services account
<code>list_cidr_locations</code>	Returns a paginated list of CIDR locations for the given collection (metadata only)
<code>list_geo_locations</code>	Retrieves a list of supported geographic locations
<code>list_health_checks</code>	Retrieve a list of the health checks that are associated with the current Amazon Web Services account
<code>list_hosted_zones</code>	Retrieves a list of the public and private hosted zones that are associated with the current Amazon Web Services account
<code>list_hosted_zones_by_name</code>	Retrieves a list of your hosted zones in lexicographic order
<code>list_hosted_zones_by_vpc</code>	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
<code>list_query_logging_configs</code>	Lists the configurations for DNS query logging that are associated with the current Amazon Web Services account
<code>list_resource_record_sets</code>	Lists the resource record sets in a specified hosted zone
<code>list_reusable_delegation_sets</code>	Retrieves a list of the reusable delegation sets that are associated with the current Amazon Web Services account
<code>list_tags_for_resource</code>	Lists tags for one health check or hosted zone
<code>list_tags_for_resources</code>	Lists tags for up to 10 health checks or hosted zones
<code>list_traffic_policies</code>	Gets information about the latest version for every traffic policy that is associated with the current Amazon Web Services account

list_traffic_policy_instances	Gets information about the traffic policy instances that you created by using the
list_traffic_policy_instances_by_hosted_zone	Gets information about the traffic policy instances that you created in a specified
list_traffic_policy_instances_by_policy	Gets information about the traffic policy instances that you created by using a
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
test_dns_answer	Gets the value that Amazon Route 53 returns in response to a DNS request for a
update_health_check	Updates an existing health check
update_hosted_zone_comment	Updates the comment for a specified hosted zone
update_hosted_zone_features	Updates the features configuration for a 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

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	<i>Amazon Route 53 Domains</i>
----------------	--------------------------------

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. • 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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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](#)

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 transfer lock)
 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 transfer lock)

get_contact_reachability_status	For operations that require confirmation that the email address for the r
get_domain_detail	This operation returns detailed information about a specified domain th
get_domain_suggestions	The GetDomainSuggestions operation returns a list of suggested domain
get_operation_detail	This operation returns the current status of an operation that is not com
list_domains	This operation returns all the domain names registered with Amazon R
list_operations	Returns information about all of the operations that return an operation
list_prices	Lists the following prices for either all the TLDs supported by Route 53
list_tags_for_domain	This operation returns all of the tags that are associated with the specifi
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 a
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 r
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
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 v
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 V

Examples

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

## End(Not run)
```

route53profiles	Route 53 Profiles
-----------------	-------------------

Description

With Amazon Route 53 Profiles you can share Route 53 configurations with VPCs and AWS ac-
counts

Usage

```
route53profiles(
  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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- route53profiles(
  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",
    use_dual_stack = "logical"
  ),
  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_profile	Associates a Route 53 Profiles profile with a VPC
associate_resource_to_profile	Associates a DNS resource configuration to a Route 53 Profile
create_profile	Creates an empty Route 53 Profile
delete_profile	Deletes the specified Route 53 Profile
disassociate_profile	Dissociates a specified Route 53 Profile from the specified VPC
disassociate_resource_from_profile	Dissoociated a specified resource, from the Route 53 Profile
get_profile	Returns information about a specified Route 53 Profile, such as whether whether the Pr

get_profile_association	Retrieves a Route 53 Profile association for a VPC
get_profile_resource_association	Returns information about a specified Route 53 Profile resource association
list_profile_associations	Lists all the VPCs that the specified Route 53 Profile is associated with
list_profile_resource_associations	Lists all the resource associations for the specified Route 53 Profile
list_profiles	Lists all the Route 53 Profiles associated with your Amazon Web Services account
list_tags_for_resource	Lists the tags that you associated with the specified resource
tag_resource	Adds one or more tags to a specified resource
untag_resource	Removes one or more tags from a specified resource
update_profile_resource_association	Updates the specified Route 53 Profile resource association

Examples

```
## Not run:
svc <- route53profiles()
svc$associate_profile(
  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.

- **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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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. • 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. <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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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 resource
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_cluster	Updates an existing cluster
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. • 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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 status

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. |
|--------|---|
- **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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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](#)
[associate_resolver_endpoint_ip_address](#)
[associate_resolver_query_log_config](#)
[associate_resolver_rule](#)
[batch_create_firewall_rule](#)
[batch_delete_firewall_rule](#)
[batch_update_firewall_rule](#)
[create_firewall_domain_list](#)
[create_firewall_rule](#)
[create_firewall_rule_group](#)
[create_outpost_resolver](#)
[create_resolver_endpoint](#)
[create_resolver_query_log_config](#)
[create_resolver_rule](#)
[delete_firewall_domain_list](#)
[delete_firewall_rule](#)
[delete_firewall_rule_group](#)
[delete_outpost_resolver](#)
[delete_resolver_endpoint](#)
[delete_resolver_query_log_config](#)
[delete_resolver_rule](#)
[disassociate_firewall_rule_group](#)
[disassociate_resolver_endpoint_ip_address](#)
[disassociate_resolver_query_log_config](#)
[disassociate_resolver_rule](#)
[get_firewall_config](#)
[get_firewall_domain_list](#)
[get_firewall_rule_group](#)
[get_firewall_rule_group_association](#)
[get_firewall_rule_group_policy](#)
[get_outpost_resolver](#)
[get_resolver_config](#)
[get_resolver_dnssec_config](#)
[get_resolver_endpoint](#)
[get_resolver_query_log_config](#)

Associates a FirewallRuleGroup with a VPC, to provide DNS filtering for the VPC
 Adds IP addresses to an inbound or an outbound Resolver endpoint
 Associates an Amazon VPC with a specified query logging configuration
 Associates a Resolver rule with a VPC
 Creates multiple DNS Firewall rules in the specified rule group
 Deletes multiple DNS Firewall rules from the specified rule group
 Updates multiple DNS Firewall rules in the specified rule group
 Creates an empty firewall domain list for use in DNS Firewall rules
 Creates a single DNS Firewall rule in the specified rule group
 Creates an empty DNS Firewall rule group for filtering DNS network traffic in a VPC
 Creates a Route 53 Resolver on an Outpost
 Creates a Resolver endpoint
 Creates a Resolver query logging configuration, which defines where you want to log DNS queries
 For DNS queries that originate in your VPCs, specifies which Resolver endpoint to use
 Deletes the specified domain list
 Deletes the specified firewall rule
 Deletes the specified firewall rule group
 Deletes a Resolver on the Outpost
 Deletes a Resolver endpoint
 Deletes a query logging configuration
 Deletes a Resolver rule
 Disassociates a FirewallRuleGroup from a VPC, to remove DNS filtering from the VPC
 Removes IP addresses from an inbound or an outbound Resolver endpoint
 Disassociates a VPC from a query logging configuration
 Removes the association between a specified Resolver rule and a specified VPC
 Retrieves the configuration of the firewall behavior provided by DNS Firewall
 Retrieves the specified firewall domain list
 Retrieves the specified firewall rule group
 Retrieves a firewall rule group association, which enables DNS filtering for a VPC
 Returns the Identity and Access Management (Amazon Web Services IAM) policy for a firewall rule group
 Gets information about a specified Resolver on the Outpost, such as its instance ID
 Retrieves the behavior configuration of Route 53 Resolver behavior for a single VPC
 Gets DNSSEC validation information for a specified resource
 Gets information about a specified Resolver endpoint, such as whether it's an inbound or outbound endpoint
 Gets information about a specified Resolver query logging configuration, such as the log bucket

<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_firewall_rule_types</code>	Retrieves the rule-type variants that can be used in the FirewallRuleType field of a FirewallRuleGroup
<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 FirewallRuleGroup
<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 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 for the specified VPC
<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
<code>update_firewall_rule_group_association</code>	Changes the association of a FirewallRuleGroup with a VPC
<code>update_outpost_resolver</code>	You can use UpdateOutpostResolver to update the instance count, type, or name of an Outpost Resolver
<code>update_resolver_config</code>	Updates the behavior configuration of Route 53 Resolver behavior for a single Amazon Web Services account
<code>update_resolver_dnssec_config</code>	Updates an existing DNSSEC validation configuration
<code>update_resolver_endpoint</code>	Updates the name, or endpoint type for an inbound or an outbound Resolver endpoint
<code>update_resolver_rule</code>	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:**

- * **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>

	• use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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_http_namespace</code>	Creates an HTTP namespace
<code>create_private_dns_namespace</code>	Creates a private namespace based on DNS, which is visible only inside a specified Amazon VPC
<code>create_public_dns_namespace</code>	Creates a public namespace based on DNS, which is visible on the internet
<code>create_service</code>	Creates a service
<code>delete_namespace</code>	Deletes a namespace from the current account
<code>delete_service</code>	Deletes a specified service and all associated service attributes
<code>delete_service_attributes</code>	Deletes specific attributes associated with a service
<code>deregister_instance</code>	Deletes the Amazon Route 53 DNS records and health check, if any, that Cloud Map created for the specified instance
<code>discover_instances</code>	Discovers registered instances for a specified namespace and service
<code>discover_instances_revision</code>	Discovers the increasing revision associated with an instance
<code>get_instance</code>	Gets information about a specified instance
<code>get_instances_health_status</code>	Gets the current health status (Healthy, Unhealthy, or Unknown) of one or more instances
<code>get_namespace</code>	Gets information about a namespace
<code>get_operation</code>	Gets information about any operation that returns an operation ID in the response, such as <code>create_namespace</code>
<code>get_service</code>	Gets the settings for a specified service
<code>get_service_attributes</code>	Returns the attributes associated with a specified service
<code>list_instances</code>	Lists summary information about the instances that you registered by using a specified namespace and service
<code>list_namespaces</code>	Lists summary information about the namespaces that were created by the current Amazon account
<code>list_operations</code>	Lists operations that match the criteria that you specify
<code>list_services</code>	Lists summary information for all the services that are associated with one or more namespaces
<code>list_tags_for_resource</code>	Lists tags for the specified resource
<code>register_instance</code>	Creates or updates one or more records and, optionally, creates a health check based on the specified settings
<code>tag_resource</code>	Adds one or more tags to the specified resource
<code>untag_resource</code>	Removes one or more tags from the specified resource
<code>update_http_namespace</code>	Updates an HTTP namespace
<code>update_instance_custom_health_status</code>	Submits a request to change the health status of a custom health check to healthy or unhealthy
<code>update_private_dns_namespace</code>	Updates a private DNS namespace
<code>update_public_dns_namespace</code>	Updates a public DNS namespace
<code>update_service</code>	Submits a request to perform the following operations:
<code>update_service_attributes</code>	Submits a request to update a specified service to add service-level attributes

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
 - **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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
  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>cancel_sol_network_operation</code>	Cancels a network operation
<code>create_sol_function_package</code>	Creates a function package
<code>create_sol_network_instance</code>	Creates a network instance
<code>create_sol_network_package</code>	Creates a network package
<code>delete_sol_function_package</code>	Deletes a function package
<code>delete_sol_network_instance</code>	Deletes a network instance
<code>delete_sol_network_package</code>	Deletes network package
<code>get_sol_function_instance</code>	Gets the details of a network function instance, including the instantiation state and
<code>get_sol_function_package</code>	Gets the details of an individual function package, such as the operational state and
<code>get_sol_function_package_content</code>	Gets the contents of a function package
<code>get_sol_function_package_descriptor</code>	Gets a function package descriptor in a function package
<code>get_sol_network_instance</code>	Gets the details of the network instance
<code>get_sol_network_operation</code>	Gets the details of a network operation, including the tasks involved in the network
<code>get_sol_network_package</code>	Gets the details of a network package
<code>get_sol_network_package_content</code>	Gets the contents of a network package
<code>get_sol_network_package_descriptor</code>	Gets the content of the network service descriptor
<code>instantiate_sol_network_instance</code>	Instantiates a network instance
<code>list_sol_function_instances</code>	Lists network function instances
<code>list_sol_function_packages</code>	Lists information about function packages
<code>list_sol_network_instances</code>	Lists your network instances
<code>list_sol_network_operations</code>	Lists details for a network operation, including when the operation started and the s
<code>list_sol_network_packages</code>	Lists network packages
<code>list_tags_for_resource</code>	Lists tags for AWS TNB resources
<code>put_sol_function_package_content</code>	Uploads the contents of a function package
<code>put_sol_network_package_content</code>	Uploads the contents of a network package
<code>tag_resource</code>	Tags an AWS TNB resource
<code>terminate_sol_network_instance</code>	Terminates a network instance
<code>untag_resource</code>	Untags an AWS TNB resource
<code>update_sol_function_package</code>	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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
    use_dual_stack = "logical"
  ),
```

```

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 Athena
create_listener	Creates a listener for a service
create_resource_configuration	Creates a resource configuration
create_resource_gateway	A resource gateway is a point of ingress into the VPC where a resource resides
create_rule	Creates a listener rule
create_service	Creates a service
create_service_network	Creates a service network
create_service_network_resource_association	Associates the specified service network with the specified resource configuration
create_service_network_service_association	Associates the specified service with the specified 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_domain_verification	Deletes the specified domain verification
delete_listener	Deletes the specified listener
delete_resource_configuration	Deletes the specified resource configuration
delete_resource_endpoint_association	Disassociates the resource configuration from the resource VPC endpoint
delete_resource_gateway	Deletes the specified resource gateway
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_resource_association	Deletes the association between a service network and a resource configuration
delete_service_network_service_association	Deletes the association between a service and a 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 network
get_domain_verification	Retrieves information about a domain verification
get_listener	Retrieves information about the specified listener for the specified service
get_resource_configuration	Retrieves information about the specified resource configuration

<code>get_resource_gateway</code>	Retrieves information about the specified resource gateway
<code>get_resource_policy</code>	Retrieves information about the specified resource policy
<code>get_rule</code>	Retrieves information about the specified listener rules
<code>get_service</code>	Retrieves information about the specified service
<code>get_service_network</code>	Retrieves information about the specified service network
<code>get_service_network_resource_association</code>	Retrieves information about the specified association between a service ne
<code>get_service_network_service_association</code>	Retrieves information about the specified association between a service ne
<code>get_service_network_vpc_association</code>	Retrieves information about the specified association between a service ne
<code>get_target_group</code>	Retrieves information about the specified target group
<code>list_access_log_subscriptions</code>	Lists the access log subscriptions for the specified service network or servi
<code>list_domain_verifications</code>	Lists the domain verifications
<code>list_listeners</code>	Lists the listeners for the specified service
<code>list_resource_configurations</code>	Lists the resource configurations owned by or shared with this account
<code>list_resource_endpoint_associations</code>	Lists the associations for the specified VPC endpoint
<code>list_resource_gateways</code>	Lists the resource gateways that you own or that were shared with you
<code>list_rules</code>	Lists the rules for the specified listener
<code>list_service_network_resource_associations</code>	Lists the associations between a service network and a resource configurat
<code>list_service_networks</code>	Lists the service networks owned by or shared with this account
<code>list_service_network_service_associations</code>	Lists the associations between a service network and a service
<code>list_service_network_vpc_associations</code>	Lists the associations between a service network and a VPC
<code>list_service_network_vpc_endpoint_associations</code>	Lists the associations between a service network and a VPC endpoint
<code>list_services</code>	Lists the services owned by the caller account or shared with the caller acc
<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>start_domain_verification</code>	Starts the domain verification process for a custom domain name
<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_resource_configuration</code>	Updates the specified resource configuration
<code>update_resource_gateway</code>	Updates the specified resource gateway
<code>update_rule</code>	Updates a specified 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, [59](#)
accept_direct_connect_gateway_association_proposal, [38](#)
accept_domain_transfer_from_another_aws_account, [67](#)
accept_network_firewall_transit_gateway_attachment, [55](#)
activate_key_signing_key, [63](#)
add_custom_routing_endpoints, [50](#)
add_endpoints, [50](#)
add_listener_certificates, [45](#)
add_tags, [42](#), [45](#)
add_trust_store_revocations, [45](#)
advertise_byoip_cidr, [50](#)
allocate_connection_on_interconnect, [38](#)
allocate_hosted_connection, [38](#)
allocate_private_virtual_interface, [38](#)
allocate_public_virtual_interface, [38](#)
allocate_transit_virtual_interface, [38](#)
allow_custom_routing_traffic, [50](#)
apigateway, [3](#)
apigatewaymanagementapi, [8](#)
apigatewayv2, [11](#)
appfabric, [15](#)
apply_security_groups_to_load_balancer, [42](#)
appmesh, [18](#)
arczonalshift, [21](#)
associate_alias, [29](#)
associate_availability_zones, [55](#)
associate_connect_peer, [59](#)
associate_connection_with_lag, [38](#)
associate_connections_to_resiliency_group, [38](#)
associate_customer_gateway, [59](#)
associate_delegation_signer_to_domain, [67](#)
associate_distribution_tenant_web_acl, [29](#)
associate_distribution_web_acl, [29](#)
associate_firewall_policy, [55](#)
associate_firewall_rule_group, [83](#)
associate_gateway_to_server, [27](#)
associate_hosted_connection, [38](#)
associate_link, [59](#)
associate_mac_sec_key, [38](#)
associate_profile, [70](#)
associate_resolver_endpoint_ip_address, [83](#)
associate_resolver_query_log_config, [83](#)
associate_resolver_rule, [83](#)
associate_resource_to_profile, [70](#)
associate_subnets, [55](#)
associate_transit_gateway_connect_peer, [59](#)
associate_virtual_interface, [38](#)
associate_vpc_with_hosted_zone, [63](#)
attach_load_balancer_to_subnets, [42](#)
attach_rule_groups_to_proxy_configuration, [55](#)

backupgateway, [24](#)
batch_create_firewall_rule, [83](#)
batch_delete_firewall_rule, [83](#)
batch_get_user_access_tasks, [17](#)
batch_update_firewall_rule, [83](#)
batch_update_rule, [93](#)

cancel_domain_transfer_to_another_aws_account, [67](#)
cancel_practice_run, [24](#)
cancel_sol_network_operation, [90](#)
cancel_zonal_shift, [24](#)
change_cidr_collection, [63](#)
change_resource_record_sets, [63](#)
change_tags_for_resource, [63](#)
check_domain_availability, [67](#)

check_domain_transferability, 67
cloudfront, 27
cloudfrontkeyvaluestore, 33
configure_health_check, 42
confirm_connection, 38
confirm_customer_agreement, 38
confirm_private_virtual_interface, 38
confirm_public_virtual_interface, 38
confirm_transit_virtual_interface, 38
connect_app_authorization, 17
copy_distribution, 29
create_accelerator, 50
create_access_log_subscription, 93
create_anycast_ip_list, 29
create_api, 13
create_api_key, 5
create_api_mapping, 13
create_app_authorization, 17
create_app_bundle, 17
create_app_cookie_stickiness_policy, 42
create_authorizer, 5, 13
create_base_path_mapping, 5
create_bgp_peer, 38
create_cache_policy, 29
create_cell, 79
create_cidr_collection, 63
create_cloud_front_origin_access_identity, 29
create_cluster, 76
create_connect_attachment, 59
create_connect_peer, 59
create_connection, 38, 59
create_connection_function, 30
create_connection_group, 30
create_container_association, 55
create_continuous_deployment_policy, 30
create_control_panel, 76
create_core_network, 59
create_core_network_prefix_list_association, 59
create_cross_account_attachment, 50
create_cross_account_authorization, 79
create_custom_routing_accelerator, 50
create_custom_routing_endpoint_group, 50
create_custom_routing_listener, 50
create_deployment, 5, 13
create_device, 59
create_direct_connect_gateway, 38
create_direct_connect_gateway_association, 38
create_direct_connect_gateway_association_proposal, 38
create_direct_connect_gateway_attachment, 59
create_distribution, 30
create_distribution_tenant, 30
create_distribution_with_tags, 30
create_documentation_part, 5
create_documentation_version, 5
create_domain_name, 5, 13
create_domain_name_access_association, 5
create_endpoint_group, 50
create_field_level_encryption_config, 30
create_field_level_encryption_profile, 30
create_firewall, 55
create_firewall_domain_list, 83
create_firewall_policy, 55
create_firewall_rule, 83
create_firewall_rule_group, 83
create_function, 30
create_gateway, 27
create_gateway_route, 20
create_global_network, 59
create_health_check, 63
create_hosted_zone, 63
create_http_namespace, 87
create_ingestion, 17
create_ingestion_destination, 17
create_integration, 13
create_integration_response, 13
create_interconnect, 38
create_invalidation, 30
create_invalidation_for_distribution_tenant, 30
create_key_group, 30
create_key_signing_key, 63
create_key_value_store, 30
create_lag, 38
create_lb_cookie_stickiness_policy, 42
create_link, 59

- create_listener, [45](#), [50](#), [93](#)
- create_load_balancer, [40](#), [42](#), [45](#)
- create_load_balancer_listeners, [42](#)
- create_load_balancer_policy, [42](#)
- create_mesh, [20](#)
- create_model, [5](#), [13](#)
- create_monitoring_subscription, [30](#)
- create_origin_access_control, [30](#)
- create_origin_request_policy, [30](#)
- create_outpost_resolver, [83](#)
- create_portal, [13](#)
- create_portal_product, [13](#)
- create_practice_run_configuration, [24](#)
- create_private_dns_namespace, [87](#)
- create_private_virtual_interface, [38](#)
- create_product_page, [13](#)
- create_product_rest_endpoint_page, [13](#)
- create_profile, [70](#)
- create_proxy, [55](#)
- create_proxy_configuration, [55](#)
- create_proxy_rule_group, [55](#)
- create_proxy_rules, [55](#)
- create_public_dns_namespace, [87](#)
- create_public_key, [30](#)
- create_public_virtual_interface, [38](#)
- create_query_logging_config, [63](#)
- create_readiness_check, [79](#)
- create_realtime_log_config, [30](#)
- create_recovery_group, [79](#)
- create_request_validator, [5](#)
- create_resiliency_group, [38](#)
- create_resolver_endpoint, [83](#)
- create_resolver_query_log_config, [83](#)
- create_resolver_rule, [83](#)
- create_resource, [5](#)
- create_resource_configuration, [93](#)
- create_resource_gateway, [93](#)
- create_resource_set, [79](#)
- create_response_headers_policy, [30](#)
- create_rest_api, [5](#)
- create_reusable_delegation_set, [63](#)
- create_route, [13](#), [20](#)
- create_route_response, [13](#)
- create_routing_control, [76](#)
- create_routing_rule, [13](#)
- create_rule, [45](#), [93](#)
- create_rule_group, [55](#)
- create_safety_rule, [76](#)
- create_service, [87](#), [93](#)
- create_service_network, [93](#)
- create_service_network_resource_association, [93](#)
- create_service_network_service_association, [93](#)
- create_service_network_vpc_association, [93](#)
- create_site, [59](#)
- create_site_to_site_vpn_attachment, [59](#)
- create_sol_function_package, [90](#)
- create_sol_network_instance, [90](#)
- create_sol_network_package, [90](#)
- create_stage, [5](#), [13](#)
- create_streaming_distribution, [30](#)
- create_streaming_distribution_with_tags, [30](#)
- create_target_group, [45](#), [93](#)
- create_tls_inspection_configuration, [55](#)
- create_traffic_policy, [63](#)
- create_traffic_policy_instance, [64](#)
- create_traffic_policy_version, [64](#)
- create_transit_gateway_peering, [59](#)
- create_transit_gateway_route_table_attachment, [59](#)
- create_transit_virtual_interface, [38](#)
- create_trust_store, [30](#), [46](#)
- 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, [64](#)
- create_vpc_attachment, [59](#)
- create_vpc_endpoint_association, [55](#)
- create_vpc_link, [5](#), [13](#)
- create_vpc_origin, [30](#)
- deactivate_key_signing_key, [64](#)
- delete_accelerator, [50](#)
- delete_access_log_settings, [13](#)
- delete_access_log_subscription, [93](#)
- delete_anycast_ip_list, [30](#)
- delete_api, [13](#)
- delete_api_key, [5](#)
- delete_api_mapping, [13](#)

- delete_app_authorization, [17](#)
- delete_app_bundle, [17](#)
- delete_attachment, [59](#)
- delete_auth_policy, [93](#)
- delete_authorizer, [5](#), [13](#)
- delete_base_path_mapping, [5](#)
- delete_bgp_peer, [38](#)
- delete_cache_policy, [30](#)
- delete_cell, [79](#)
- delete_cidr_collection, [64](#)
- delete_client_certificate, [5](#)
- delete_cloud_front_origin_access_identity, [30](#)
- delete_cluster, [76](#)
- delete_connect_peer, [59](#)
- delete_connection, [10](#), [38](#), [59](#)
- delete_connection_function, [30](#)
- delete_connection_group, [30](#)
- delete_container_association, [55](#)
- delete_continuous_deployment_policy, [30](#)
- delete_control_panel, [76](#)
- delete_core_network, [59](#)
- delete_core_network_policy_version, [59](#)
- delete_core_network_prefix_list_association, [59](#)
- delete_cors_configuration, [13](#)
- delete_cross_account_attachment, [50](#)
- delete_cross_account_authorization, [79](#)
- delete_custom_routing_accelerator, [50](#)
- delete_custom_routing_endpoint_group, [50](#)
- delete_custom_routing_listener, [50](#)
- delete_deployment, [6](#), [13](#)
- delete_device, [59](#)
- delete_direct_connect_gateway, [38](#)
- delete_direct_connect_gateway_association, [38](#)
- delete_direct_connect_gateway_association_proposal, [38](#)
- delete_distribution, [30](#)
- delete_distribution_tenant, [30](#)
- delete_documentation_part, [6](#)
- delete_documentation_version, [6](#)
- delete_domain, [67](#)
- delete_domain_name, [6](#), [13](#)
- delete_domain_name_access_association, [6](#)
- delete_domain_verification, [93](#)
- delete_endpoint_group, [50](#)
- delete_field_level_encryption_config, [30](#)
- delete_field_level_encryption_profile, [30](#)
- delete_firewall, [55](#)
- delete_firewall_domain_list, [83](#)
- delete_firewall_policy, [55](#)
- delete_firewall_rule, [83](#)
- delete_firewall_rule_group, [83](#)
- delete_function, [30](#)
- delete_gateway, [27](#)
- delete_gateway_response, [6](#)
- delete_gateway_route, [20](#)
- delete_global_network, [59](#)
- delete_health_check, [64](#)
- delete_hosted_zone, [64](#)
- delete_hypervisor, [27](#)
- delete_ingestion, [17](#)
- delete_ingestion_destination, [17](#)
- delete_integration, [6](#), [13](#)
- delete_integration_response, [6](#), [13](#)
- delete_interconnect, [38](#)
- delete_key, [35](#)
- delete_key_group, [30](#)
- delete_key_signing_key, [64](#)
- delete_key_value_store, [30](#)
- delete_lag, [39](#)
- delete_link, [59](#)
- delete_listener, [46](#), [50](#), [93](#)
- delete_load_balancer, [42](#), [46](#)
- delete_load_balancer_listeners, [42](#)
- delete_load_balancer_policy, [42](#)
- delete_mesh, [20](#)
- delete_method, [6](#)
- delete_method_response, [6](#)
- delete_model, [6](#), [13](#)
- delete_monitoring_subscription, [30](#)
- delete_namespace, [87](#)
- delete_network_firewall_transit_gateway_attachment, [55](#)
- delete_origin_access_control, [30](#)
- delete_origin_request_policy, [30](#)
- delete_outpost_resolver, [83](#)
- delete_peering, [59](#)
- delete_portal, [13](#)
- delete_portal_product, [13](#)

- delete_portal_product_sharing_policy, [13](#)
- delete_practice_run_configuration, [24](#)
- delete_product_page, [13](#)
- delete_product_rest_endpoint_page, [13](#)
- delete_profile, [70](#)
- delete_proxy, [55](#)
- delete_proxy_configuration, [55](#)
- delete_proxy_rule_group, [55](#)
- delete_proxy_rules, [55](#)
- delete_public_key, [30](#)
- delete_query_logging_config, [64](#)
- delete_readiness_check, [79](#)
- delete_realtime_log_config, [30](#)
- delete_recovery_group, [79](#)
- delete_request_validator, [6](#)
- delete_resiliency_group, [39](#)
- delete_resolver_endpoint, [83](#)
- delete_resolver_query_log_config, [83](#)
- delete_resolver_rule, [83](#)
- delete_resource, [6](#)
- delete_resource_configuration, [93](#)
- delete_resource_endpoint_association, [93](#)
- delete_resource_gateway, [93](#)
- delete_resource_policy, [30](#), [55](#), [59](#), [93](#)
- delete_resource_set, [79](#)
- delete_response_headers_policy, [30](#)
- delete_rest_api, [6](#)
- delete_reusable_delegation_set, [64](#)
- delete_route, [13](#), [20](#)
- delete_route_request_parameter, [13](#)
- delete_route_response, [13](#)
- delete_route_settings, [13](#)
- delete_routing_control, [76](#)
- delete_routing_rule, [13](#)
- delete_rule, [46](#), [93](#)
- delete_rule_group, [55](#)
- delete_safety_rule, [76](#)
- delete_service, [87](#), [93](#)
- delete_service_attributes, [87](#)
- delete_service_network, [93](#)
- delete_service_network_resource_association, [93](#)
- delete_service_network_service_association, [93](#)
- delete_service_network_vpc_association, [93](#)
- delete_shared_trust_store_association, [46](#)
- delete_site, [59](#)
- delete_sol_function_package, [90](#)
- delete_sol_network_instance, [90](#)
- delete_sol_network_package, [90](#)
- delete_stage, [6](#), [13](#)
- delete_streaming_distribution, [30](#)
- delete_tags_for_domain, [67](#)
- delete_target_group, [46](#), [93](#)
- delete_tls_inspection_configuration, [55](#)
- delete_traffic_policy, [64](#)
- delete_traffic_policy_instance, [64](#)
- delete_trust_store, [30](#), [46](#)
- delete_usage_plan, [6](#)
- delete_usage_plan_key, [6](#)
- delete_virtual_gateway, [20](#)
- delete_virtual_interface, [39](#)
- delete_virtual_node, [20](#)
- delete_virtual_router, [21](#)
- delete_virtual_service, [21](#)
- delete_vpc_association_authorization, [64](#)
- delete_vpc_endpoint_association, [55](#)
- delete_vpc_link, [6](#), [13](#)
- delete_vpc_origin, [30](#)
- deny_custom_routing_traffic, [50](#)
- deprovision_byoip_cidr, [50](#)
- deregister_instance, [87](#)
- deregister_instances_from_load_balancer, [42](#)
- deregister_targets, [46](#), [93](#)
- deregister_transit_gateway, [59](#)
- describe_accelerator, [50](#)
- describe_accelerator_attributes, [50](#)
- describe_account_limits, [42](#), [46](#)
- describe_capacity_reservation, [46](#)
- describe_cluster, [76](#)
- describe_connection_function, [30](#)
- describe_connection_loa, [39](#)
- describe_connections, [39](#)
- describe_connections_on_interconnect, [39](#)
- describe_container_association, [55](#)
- describe_control_panel, [76](#)
- describe_cross_account_attachment, [50](#)
- describe_custom_routing_accelerator,

- [50](#)
- [describe_custom_routing_accelerator_attributes](#), [50](#)
- [describe_custom_routing_endpoint_group](#), [50](#)
- [describe_custom_routing_listener](#), [51](#)
- [describe_customer_metadata](#), [39](#)
- [describe_direct_connect_gateway_association_proposals](#), [39](#)
- [describe_direct_connect_gateway_associations](#), [39](#)
- [describe_direct_connect_gateway_attachments](#), [39](#)
- [describe_direct_connect_gateways](#), [39](#)
- [describe_endpoint_group](#), [51](#)
- [describe_firewall](#), [55](#)
- [describe_firewall_metadata](#), [55](#)
- [describe_firewall_policy](#), [55](#)
- [describe_flow_operation](#), [55](#)
- [describe_function](#), [30](#)
- [describe_gateway_route](#), [21](#)
- [describe_global_networks](#), [59](#)
- [describe_hosted_connections](#), [39](#)
- [describe_instance_health](#), [42](#)
- [describe_interconnect_loa](#), [39](#)
- [describe_interconnects](#), [39](#)
- [describe_key_value_store](#), [31, 35](#)
- [describe_lags](#), [39](#)
- [describe_listener](#), [51](#)
- [describe_listener_attributes](#), [46](#)
- [describe_listener_certificates](#), [46](#)
- [describe_listeners](#), [46](#)
- [describe_loa](#), [39](#)
- [describe_load_balancer_attributes](#), [42, 46](#)
- [describe_load_balancer_policies](#), [42](#)
- [describe_load_balancer_policy_types](#), [42](#)
- [describe_load_balancers](#), [42, 46](#)
- [describe_locations](#), [39](#)
- [describe_logging_configuration](#), [55](#)
- [describe_mesh](#), [21](#)
- [describe_proxy](#), [55](#)
- [describe_proxy_configuration](#), [55](#)
- [describe_proxy_rule](#), [55](#)
- [describe_proxy_rule_group](#), [55](#)
- [describe_resource_policy](#), [55](#)
- [describe_route](#), [21](#)
- [describe_router_configuration](#), [39](#)
- [describe_routing_control](#), [76](#)
- [describe_rule_group](#), [55](#)
- [describe_rule_group_metadata](#), [55](#)
- [describe_rule_group_summary](#), [55](#)
- [describe_rules](#), [46](#)
- [describe_safety_rule](#), [76](#)
- [describe_ssl_policies](#), [46](#)
- [describe_tags](#), [39, 42, 46](#)
- [describe_target_group_attributes](#), [46](#)
- [describe_target_groups](#), [46](#)
- [describe_target_health](#), [46](#)
- [describe_tls_inspection_configuration](#), [55](#)
- [describe_trust_store_associations](#), [46](#)
- [describe_trust_store_revocations](#), [46](#)
- [describe_trust_stores](#), [46](#)
- [describe_virtual_gateway](#), [21](#)
- [describe_virtual_gateways](#), [39](#)
- [describe_virtual_interfaces](#), [39](#)
- [describe_virtual_node](#), [21](#)
- [describe_virtual_router](#), [21](#)
- [describe_virtual_service](#), [21](#)
- [describe_vpc_endpoint_association](#), [55](#)
- [detach_load_balancer_from_subnets](#), [42](#)
- [detach_rule_groups_from_proxy_configuration](#), [55](#)
- [directconnect](#), [36](#)
- [disable_availability_zones_for_load_balancer](#), [42](#)
- [disable_domain_auto_renew](#), [67](#)
- [disable_domain_transfer_lock](#), [67](#)
- [disable_hosted_zone_dnssec](#), [64](#)
- [disable_portal](#), [13](#)
- [disassociate_availability_zones](#), [56](#)
- [disassociate_connect_peer](#), [59](#)
- [disassociate_connection_from_lag](#), [39](#)
- [disassociate_connections_from_resiliency_group](#), [39](#)
- [disassociate_customer_gateway](#), [59](#)
- [disassociate_delegation_signer_from_domain](#), [67](#)
- [disassociate_distribution_tenant_web_acl](#), [31](#)
- [disassociate_distribution_web_acl](#), [31](#)
- [disassociate_firewall_rule_group](#), [83](#)
- [disassociate_gateway_from_server](#), [27](#)
- [disassociate_link](#), [59](#)

- disassociate_mac_sec_key, [39](#)
- disassociate_profile, [70](#)
- disassociate_resolver_endpoint_ip_address, [83](#)
- disassociate_resolver_query_log_config, [83](#)
- disassociate_resolver_rule, [83](#)
- disassociate_resource_from_profile, [70](#)
- disassociate_subnets, [56](#)
- disassociate_transit_gateway_connect_peer, [59](#)
- disassociate_vpc_from_hosted_zone, [64](#)
- discover_instances, [87](#)
- discover_instances_revision, [87](#)
- elb, [40](#)
- elbv2, [43](#)
- enable_availability_zones_for_load_balancer, [42](#)
- enable_domain_auto_renew, [67](#)
- enable_domain_transfer_lock, [67](#)
- enable_hosted_zone_dnssec, [64](#)
- execute_core_network_change_set, [59](#)
- export_api, [13](#)
- flush_stage_authorizers_cache, [6](#)
- flush_stage_cache, [6](#)
- generate_client_certificate, [6](#)
- get_access_log_subscription, [93](#)
- get_account, [6](#)
- get_account_limit, [64](#)
- get_analysis_report_results, [56](#)
- get_anycast_ip_list, [31](#)
- 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, [79](#)
- get_auth_policy, [93](#)
- get_authorizer, [6, 13](#)
- get_authorizers, [6, 14](#)
- get_autoshift_observer_notification_status, [24](#)
- get_bandwidth_rate_limit_schedule, [27](#)
- get_base_path_mapping, [6](#)
- get_base_path_mappings, [6](#)
- get_cache_policy, [31](#)
- get_cache_policy_config, [31](#)
- get_cell, [79](#)
- get_cell_readiness_summary, [80](#)
- get_change, [64](#)
- get_checker_ip_ranges, [64](#)
- get_client_certificate, [6](#)
- get_client_certificates, [6](#)
- get_cloud_front_origin_access_identity, [31](#)
- get_cloud_front_origin_access_identity_config, [31](#)
- get_connect_attachment, [60](#)
- get_connect_peer, [60](#)
- get_connect_peer_associations, [60](#)
- get_connection, [10](#)
- get_connection_function, [31](#)
- get_connection_group, [31](#)
- get_connection_group_by_routing_endpoint, [31](#)
- get_connections, [60](#)
- get_contact_reachability_status, [68](#)
- get_continuous_deployment_policy, [31](#)
- get_continuous_deployment_policy_config, [31](#)
- get_core_network, [60](#)
- get_core_network_change_events, [60](#)
- get_core_network_change_set, [60](#)
- get_core_network_policy, [60](#)
- get_customer_gateway_associations, [60](#)
- get_deployment, [6, 14](#)
- get_deployments, [6, 14](#)
- get_devices, [60](#)
- get_direct_connect_gateway_attachment, [60](#)
- get_distribution, [31](#)
- get_distribution_config, [31](#)
- get_distribution_tenant, [31](#)
- get_distribution_tenant_by_domain, [31](#)
- get_dnssec, [64](#)
- get_documentation_part, [6](#)
- get_documentation_parts, [6](#)
- get_documentation_version, [6](#)
- get_documentation_versions, [6](#)
- get_domain_detail, [68](#)
- get_domain_name, [6, 14](#)

get_domain_name_access_associations, [6](#)
get_domain_names, [6](#), [14](#)
get_domain_suggestions, [68](#)
get_domain_verification, [93](#)
get_export, [6](#)
get_field_level_encryption, [31](#)
get_field_level_encryption_config, [31](#)
get_field_level_encryption_profile, [31](#)
get_field_level_encryption_profile_config, [31](#)
get_firewall_config, [83](#)
get_firewall_domain_list, [83](#)
get_firewall_rule_group, [83](#)
get_firewall_rule_group_association, [83](#)
get_firewall_rule_group_policy, [83](#)
get_function, [31](#)
get_gateway, [27](#)
get_gateway_response, [6](#)
get_gateway_responses, [6](#)
get_geo_location, [64](#)
get_health_check, [64](#)
get_health_check_count, [64](#)
get_health_check_last_failure_reason, [64](#)
get_health_check_status, [64](#)
get_hosted_zone, [64](#)
get_hosted_zone_count, [64](#)
get_hosted_zone_limit, [64](#)
get_hypervisor, [27](#)
get_hypervisor_property_mappings, [27](#)
get_ingestion, [18](#)
get_ingestion_destination, [18](#)
get_instance, [87](#)
get_instances_health_status, [87](#)
get_integration, [6](#), [14](#)
get_integration_response, [6](#), [14](#)
get_integration_responses, [14](#)
get_integrations, [14](#)
get_invalidation, [31](#)
get_invalidation_for_distribution_tenant, [31](#)
get_key, [35](#)
get_key_group, [31](#)
get_key_group_config, [31](#)
get_link_associations, [60](#)
get_links, [60](#)
get_listener, [93](#)
get_managed_certificate_details, [31](#)
get_managed_resource, [24](#)
get_method, [6](#)
get_method_response, [6](#)
get_model, [6](#), [14](#)
get_model_template, [7](#), [14](#)
get_models, [6](#), [14](#)
get_monitoring_subscription, [31](#)
get_namespace, [87](#)
get_network_resource_counts, [60](#)
get_network_resource_relationships, [60](#)
get_network_resources, [60](#)
get_network_routes, [60](#)
get_network_telemetry, [60](#)
get_operation, [87](#)
get_operation_detail, [68](#)
get_origin_access_control, [31](#)
get_origin_access_control_config, [31](#)
get_origin_request_policy, [31](#)
get_origin_request_policy_config, [31](#)
get_outpost_resolver, [83](#)
get_portal, [14](#)
get_portal_product, [14](#)
get_portal_product_sharing_policy, [14](#)
get_product_page, [14](#)
get_product_rest_endpoint_page, [14](#)
get_profile, [70](#)
get_profile_association, [71](#)
get_profile_resource_association, [71](#)
get_public_key, [31](#)
get_public_key_config, [31](#)
get_query_logging_config, [64](#)
get_readiness_check, [80](#)
get_readiness_check_resource_status, [80](#)
get_readiness_check_status, [80](#)
get_realtime_log_config, [31](#)
get_recovery_group, [80](#)
get_recovery_group_readiness_summary, [80](#)
get_request_validator, [7](#)
get_request_validators, [7](#)
get_resiliency_group, [39](#)
get_resolver_config, [83](#)
get_resolver_dnssec_config, [83](#)
get_resolver_endpoint, [83](#)
get_resolver_query_log_config, [83](#)
get_resolver_query_log_config_association,

- [84](#)
- [get_resolver_query_log_config_policy, 84](#)
- [get_resolver_rule, 84](#)
- [get_resolver_rule_association, 84](#)
- [get_resolver_rule_policy, 84](#)
- [get_resource, 7](#)
- [get_resource_configuration, 93](#)
- [get_resource_gateway, 94](#)
- [get_resource_policy, 31, 46, 60, 77, 94](#)
- [get_resource_set, 80](#)
- [get_resources, 7](#)
- [get_response_headers_policy, 31](#)
- [get_response_headers_policy_config, 31](#)
- [get_rest_api, 7](#)
- [get_rest_apis, 7](#)
- [get_reusable_delegation_set, 64](#)
- [get_reusable_delegation_set_limit, 64](#)
- [get_route, 14](#)
- [get_route_analysis, 60](#)
- [get_route_response, 14](#)
- [get_route_responses, 14](#)
- [get_routes, 14](#)
- [get_routing_control_state, 74](#)
- [get_routing_rule, 14](#)
- [get_rule, 94](#)
- [get_sdk, 7](#)
- [get_sdk_type, 7](#)
- [get_sdk_types, 7](#)
- [get_service, 87, 94](#)
- [get_service_attributes, 87](#)
- [get_service_network, 94](#)
- [get_service_network_resource_association, 94](#)
- [get_service_network_service_association, 94](#)
- [get_service_network_vpc_association, 94](#)
- [get_site_to_site_vpn_attachment, 60](#)
- [get_sites, 60](#)
- [get_sol_function_instance, 90](#)
- [get_sol_function_package, 90](#)
- [get_sol_function_package_content, 90](#)
- [get_sol_function_package_descriptor, 90](#)
- [get_sol_network_instance, 90](#)
- [get_sol_network_operation, 90](#)
- [get_sol_network_package, 90](#)
- [get_sol_network_package_content, 90](#)
- [get_sol_network_package_descriptor, 90](#)
- [get_stage, 7, 14](#)
- [get_stages, 7, 14](#)
- [get_streaming_distribution, 31](#)
- [get_streaming_distribution_config, 31](#)
- [get_tags, 7, 14](#)
- [get_target_group, 94](#)
- [get_traffic_policy, 64](#)
- [get_traffic_policy_instance, 64](#)
- [get_traffic_policy_instance_count, 64](#)
- [get_transit_gateway_connect_peer_associations, 60](#)
- [get_transit_gateway_peering, 60](#)
- [get_transit_gateway_registrations, 60](#)
- [get_transit_gateway_route_table_attachment, 60](#)
- [get_trust_store, 31](#)
- [get_trust_store_ca_certificates_bundle, 46](#)
- [get_trust_store_revocation_content, 46](#)
- [get_usage, 7](#)
- [get_usage_plan, 7](#)
- [get_usage_plan_key, 7](#)
- [get_usage_plan_keys, 7](#)
- [get_usage_plans, 7](#)
- [get_virtual_machine, 27](#)
- [get_vpc_attachment, 60](#)
- [get_vpc_link, 7, 14](#)
- [get_vpc_links, 7, 14](#)
- [get_vpc_origin, 31](#)
- [globalaccelerator, 47](#)
- [import_api, 14](#)
- [import_api_keys, 7](#)
- [import_documentation_parts, 7](#)
- [import_firewall_domains, 84](#)
- [import_hypervisor_configuration, 27](#)
- [import_rest_api, 7](#)
- [instantiate_sol_network_instance, 90](#)
- [list_accelerators, 51](#)
- [list_access_log_subscriptions, 94](#)
- [list_analysis_reports, 56](#)
- [list_anycast_ip_lists, 31](#)
- [list_app_authorizations, 18](#)
- [list_app_bundles, 18](#)
- [list_associated_route_53_health_checks, 77](#)

- [list_attachment_routing_policy_associations,](#) [32](#)
- [60](#)
- [list_attachments,](#) [60](#)
- [list_autoshifts,](#) [24](#)
- [list_byoip_cidrs,](#) [51](#)
- [list_cache_policies,](#) [31](#)
- [list_cells,](#) [80](#)
- [list_cidr_blocks,](#) [64](#)
- [list_cidr_collections,](#) [64](#)
- [list_cidr_locations,](#) [64](#)
- [list_cloud_front_origin_access_identities,](#) [31](#)
- [list_clusters,](#) [77](#)
- [list_conflicting_aliases,](#) [31](#)
- [list_connect_peers,](#) [60](#)
- [list_connection_functions,](#) [31](#)
- [list_connection_groups,](#) [31](#)
- [list_container_associations,](#) [56](#)
- [list_continuous_deployment_policies,](#) [32](#)
- [list_control_panels,](#) [77](#)
- [list_core_network_policy_versions,](#) [60](#)
- [list_core_network_prefix_list_associations,](#) [60](#)
- [list_core_network_routing_information,](#) [60](#)
- [list_core_networks,](#) [60](#)
- [list_cross_account_attachments,](#) [51](#)
- [list_cross_account_authorizations,](#) [80](#)
- [list_cross_account_resource_accounts,](#) [51](#)
- [list_cross_account_resources,](#) [51](#)
- [list_custom_routing_accelerators,](#) [51](#)
- [list_custom_routing_endpoint_groups,](#) [51](#)
- [list_custom_routing_listeners,](#) [51](#)
- [list_custom_routing_port_mappings,](#) [51](#)
- [list_custom_routing_port_mappings_by_destination,](#) [51](#)
- [list_distribution_tenants,](#) [32](#)
- [list_distribution_tenants_by_customization,](#) [32](#)
- [list_distributions,](#) [32](#)
- [list_distributions_by_anycast_ip_list_id,](#) [32](#)
- [list_distributions_by_cache_policy_id,](#) [32](#)
- [list_distributions_by_connection_function,](#) [32](#)
- [list_distributions_by_connection_mode,](#) [32](#)
- [list_distributions_by_key_group,](#) [32](#)
- [list_distributions_by_origin_request_policy_id,](#) [32](#)
- [list_distributions_by_owned_resource,](#) [32](#)
- [list_distributions_by_realtime_log_config,](#) [32](#)
- [list_distributions_by_response_headers_policy_id,](#) [32](#)
- [list_distributions_by_trust_store,](#) [32](#)
- [list_distributions_by_vpc_origin_id,](#) [32](#)
- [list_distributions_by_web_acl_id,](#) [32](#)
- [list_domain_conflicts,](#) [32](#)
- [list_domain_verifications,](#) [94](#)
- [list_domains,](#) [68](#)
- [list_endpoint_groups,](#) [51](#)
- [list_field_level_encryption_configs,](#) [32](#)
- [list_field_level_encryption_profiles,](#) [32](#)
- [list_firewall_configs,](#) [84](#)
- [list_firewall_domain_lists,](#) [84](#)
- [list_firewall_domains,](#) [84](#)
- [list_firewall_policies,](#) [56](#)
- [list_firewall_rule_group_associations,](#) [84](#)
- [list_firewall_rule_groups,](#) [84](#)
- [list_firewall_rule_types,](#) [84](#)
- [list_firewall_rules,](#) [84](#)
- [list_firewalls,](#) [56](#)
- [list_flow_operation_results,](#) [56](#)
- [list_flow_operations,](#) [56](#)
- [list_functions,](#) [32](#)
- [list_gateway_routes,](#) [21](#)
- [list_gateways,](#) [27](#)
- [list_geo_locations,](#) [64](#)
- [list_health_checks,](#) [64](#)
- [list_hosted_zones,](#) [64](#)
- [list_hosted_zones_by_name,](#) [64](#)
- [list_hosted_zones_by_vpc,](#) [64](#)
- [list_hypervisors,](#) [27](#)
- [list_ingestion_destinations,](#) [18](#)
- [list_ingestions,](#) [18](#)
- [list_instances,](#) [87](#)

- `list_invalidations`, 32
- `list_invalidations_for_distribution_tenant`, 32
- `list_key_groups`, 32
- `list_key_value_stores`, 32
- `list_keys`, 35
- `list_listeners`, 51, 94
- `list_managed_resources`, 24
- `list_meshes`, 21
- `list_namespaces`, 87
- `list_operations`, 68, 87
- `list_organization_service_access_status`, 60
- `list_origin_access_controls`, 32
- `list_origin_request_policies`, 32
- `list_outpost_resolvers`, 84
- `list_peerings`, 60
- `list_portal_products`, 14
- `list_portals`, 14
- `list_prices`, 68
- `list_product_pages`, 14
- `list_product_rest_endpoint_pages`, 14
- `list_profile_associations`, 71
- `list_profile_resource_associations`, 71
- `list_profiles`, 71
- `list_proxies`, 56
- `list_proxy_configurations`, 56
- `list_proxy_rule_groups`, 56
- `list_public_keys`, 32
- `list_query_logging_configs`, 64
- `list_readiness_checks`, 80
- `list_realtime_log_configs`, 32
- `list_recovery_groups`, 80
- `list_resiliency_group_associations`, 39
- `list_resiliency_groups`, 39
- `list_resolver_configs`, 84
- `list_resolver_dnssec_configs`, 84
- `list_resolver_endpoint_ip_addresses`, 84
- `list_resolver_endpoints`, 84
- `list_resolver_query_log_config_associations`, 84
- `list_resolver_query_log_configs`, 84
- `list_resolver_rule_associations`, 84
- `list_resolver_rules`, 84
- `list_resource_configurations`, 94
- `list_resource_endpoint_associations`, 94
- `list_resource_gateways`, 94
- `list_resource_record_sets`, 64
- `list_resource_sets`, 80
- `list_response_headers_policies`, 32
- `list_reusable_delegation_sets`, 64
- `list_routes`, 21
- `list_routing_controls`, 74, 77
- `list_routing_rules`, 14
- `list_rule_groups`, 56
- `list_rules`, 80, 94
- `list_safety_rules`, 77
- `list_service_network_resource_associations`, 94
- `list_service_network_service_associations`, 94
- `list_service_network_vpc_associations`, 94
- `list_service_network_vpc_endpoint_associations`, 94
- `list_service_networks`, 94
- `list_services`, 87, 94
- `list_sol_function_instances`, 90
- `list_sol_function_packages`, 90
- `list_sol_network_instances`, 90
- `list_sol_network_operations`, 90
- `list_sol_network_packages`, 90
- `list_streaming_distributions`, 32
- `list_tags_for_domain`, 68
- `list_tags_for_resource`, 18, 21, 27, 32, 51, 56, 60, 64, 71, 77, 84, 87, 90, 94
- `list_tags_for_resources`, 64, 80
- `list_target_groups`, 94
- `list_targets`, 94
- `list_tls_inspection_configurations`, 56
- `list_traffic_policies`, 64
- `list_traffic_policy_instances`, 65
- `list_traffic_policy_instances_by_hosted_zone`, 65
- `list_traffic_policy_instances_by_policy`, 65
- `list_traffic_policy_versions`, 65
- `list_trust_stores`, 32
- `list_virtual_gateways`, 21
- `list_virtual_interface_routes`, 39
- `list_virtual_interface_test_history`, 39
- `list_virtual_machines`, 27
- `list_virtual_nodes`, 21

- list_virtual_routers, [21](#)
- list_virtual_services, [21](#)
- list_vpc_association_authorizations, [65](#)
- list_vpc_endpoint_associations, [56](#)
- list_vpc_origins, [32](#)
- list_zonal_shifts, [24](#)

- modify_capacity_reservation, [46](#)
- modify_ip_pools, [46](#)
- modify_listener, [46](#)
- modify_listener_attributes, [46](#)
- modify_load_balancer_attributes, [42, 46](#)
- modify_rule, [46](#)
- modify_target_group, [46](#)
- modify_target_group_attributes, [46](#)
- modify_trust_store, [46](#)

- networkfirewall, [51](#)
- networkmanager, [57](#)

- post_to_connection, [10](#)
- preview_portal, [14](#)
- provision_byoip_cidr, [51](#)
- publish_connection_function, [32](#)
- publish_function, [32](#)
- publish_portal, [14](#)
- push_domain, [68](#)
- put_attachment_routing_policy_label, [60](#)
- put_auth_policy, [94](#)
- put_bandwidth_rate_limit_schedule, [27](#)
- put_core_network_policy, [60](#)
- put_firewall_rule_group_policy, [84](#)
- put_gateway_response, [7](#)
- put_hypervisor_property_mappings, [27](#)
- put_integration, [7](#)
- put_integration_response, [7](#)
- put_key, [35](#)
- put_maintenance_start_time, [27](#)
- put_method, [7](#)
- put_method_response, [7](#)
- put_portal_product_sharing_policy, [14](#)
- put_resolver_query_log_config_policy, [84](#)
- put_resolver_rule_policy, [84](#)
- put_resource_policy, [32, 56, 60, 94](#)
- put_rest_api, [7](#)
- put_routing_rule, [14](#)
- put_sol_function_package_content, [90](#)
- put_sol_network_package_content, [90](#)

- register_domain, [68](#)
- register_instance, [87](#)
- register_instances_with_load_balancer, [40, 42](#)
- register_targets, [46, 94](#)
- register_transit_gateway, [60](#)
- reimport_api, [14](#)
- reject_attachment, [60](#)
- reject_domain_name_access_association, [7](#)
- reject_domain_transfer_from_another_aws_account, [68](#)
- reject_network_firewall_transit_gateway_attachment, [56](#)
- remove_attachment_routing_policy_label, [60](#)
- remove_custom_routing_endpoints, [51](#)
- remove_endpoints, [51](#)
- remove_listener_certificates, [46](#)
- remove_tags, [42, 46](#)
- remove_trust_store_revocations, [46](#)
- renew_domain, [68](#)
- resend_contact_reachability_email, [68](#)
- resend_operation_authorization, [68](#)
- reset_authorizers_cache, [14](#)
- restore_core_network_policy_version, [60](#)
- retrieve_domain_auth_code, [68](#)
- route53, [61](#)
- route53domains, [65](#)
- route53profiles, [68](#)
- route53recoverycluster, [71](#)
- route53recoverycontrolconfig, [74](#)
- route53recoveryreadiness, [77](#)
- route53resolver, [80](#)

- servicediscovery, [85](#)
- set_ip_address_type, [46](#)
- set_load_balancer_listener_ssl_certificate, [42](#)
- set_load_balancer_policies_for_backend_server, [42](#)
- set_load_balancer_policies_of_listener, [42](#)
- set_rule_priorities, [46](#)
- set_security_groups, [46](#)

- set_subnets, [46](#)
- start_analysis_report, [56](#)
- start_bgp_failover_test, [39](#)
- start_domain_verification, [94](#)
- start_flow_capture, [56](#)
- start_flow_flush, [56](#)
- start_ingestion, [18](#)
- start_organization_service_access_update, [60](#)
- start_practice_run, [24](#)
- start_route_analysis, [60](#)
- start_user_access_tasks, [18](#)
- start_virtual_machines_metadata_sync, [27](#)
- start_zonal_shift, [24](#)
- stop_bgp_failover_test, [39](#)
- stop_ingestion, [18](#)
- tag_resource, [7](#), [14](#), [18](#), [21](#), [27](#), [32](#), [39](#), [51](#), [56](#), [60](#), [71](#), [77](#), [80](#), [84](#), [87](#), [90](#), [94](#)
- telconetworkbuilder, [88](#)
- terminate_sol_network_instance, [90](#)
- test_connection_function, [32](#)
- test_dns_answer, [65](#)
- test_function, [32](#)
- test_hypervisor_configuration, [27](#)
- test_invoke_authorizer, [7](#)
- test_invoke_method, [7](#)
- transfer_domain, [68](#)
- transfer_domain_to_another_aws_account, [68](#)
- untag_resource, [7](#), [14](#), [18](#), [21](#), [27](#), [32](#), [39](#), [51](#), [56](#), [60](#), [71](#), [77](#), [80](#), [84](#), [87](#), [90](#), [94](#)
- update_accelerator, [51](#)
- update_accelerator_attributes, [51](#)
- update_access_log_subscription, [94](#)
- update_account, [7](#)
- update_anycast_ip_list, [32](#)
- update_api, [14](#)
- update_api_key, [7](#)
- update_api_mapping, [14](#)
- update_app_authorization, [18](#)
- update_authorizer, [7](#), [14](#)
- update_autoshift_observer_notification_status, [24](#)
- update_availability_zone_change_protection, [56](#)
- update_base_path_mapping, [7](#)
- update_cache_policy, [32](#)
- update_cell, [80](#)
- update_client_certificate, [7](#)
- update_cloud_front_origin_access_identity, [32](#)
- update_cluster, [77](#)
- update_connection, [39](#), [61](#)
- update_connection_function, [32](#)
- update_connection_group, [32](#)
- update_connections_billing_mode, [39](#)
- update_container_association, [56](#)
- update_continuous_deployment_policy, [32](#)
- update_control_panel, [77](#)
- update_core_network, [61](#)
- update_cross_account_attachment, [51](#)
- update_custom_routing_accelerator, [51](#)
- update_custom_routing_accelerator_attributes, [51](#)
- update_custom_routing_listener, [51](#)
- update_deployment, [7](#), [14](#)
- update_device, [61](#)
- update_direct_connect_gateway, [39](#)
- update_direct_connect_gateway_association, [39](#)
- update_direct_connect_gateway_attachment, [61](#)
- update_distribution, [32](#)
- update_distribution_tenant, [32](#)
- update_distribution_with_staging_config, [33](#)
- update_documentation_part, [7](#)
- update_documentation_version, [7](#)
- update_domain_association, [33](#)
- update_domain_contact, [68](#)
- update_domain_contact_privacy, [68](#)
- update_domain_name, [7](#), [14](#)
- update_domain_nameservers, [68](#)
- update_endpoint_group, [51](#)
- update_field_level_encryption_config, [33](#)
- update_field_level_encryption_profile, [33](#)
- update_firewall_analysis_settings, [56](#)
- update_firewall_config, [84](#)
- update_firewall_delete_protection, [56](#)
- update_firewall_description, [56](#)
- update_firewall_domains, [84](#)

- update_firewall_encryption_configuration, 56
- update_firewall_policy, 56
- update_firewall_policy_change_protection, 56
- update_firewall_rule, 84
- update_firewall_rule_group_association, 84
- update_function, 33
- update_gateway_information, 27
- update_gateway_response, 7
- update_gateway_route, 21
- update_gateway_software_now, 27
- update_global_network, 61
- update_health_check, 65
- update_hosted_zone_comment, 65
- update_hosted_zone_features, 65
- update_http_namespace, 87
- update_hypervisor, 27
- update_ingestion_destination, 18
- update_instance_custom_health_status, 87
- update_integration, 7, 14
- update_integration_response, 7, 14
- update_key_group, 33
- update_key_value_store, 33
- update_keys, 35
- update_lag, 39
- update_link, 61
- update_listener, 51, 94
- update_logging_configuration, 56
- update_mesh, 21
- update_method, 7
- update_method_response, 7
- update_model, 8, 15
- update_network_resource_metadata, 61
- update_origin_access_control, 33
- update_origin_request_policy, 33
- update_outpost_resolver, 84
- update_portal, 15
- update_portal_product, 15
- update_practice_run_configuration, 24
- update_private_dns_namespace, 87
- update_product_page, 15
- update_product_rest_endpoint_page, 15
- update_profile_resource_association, 71
- update_proxy, 56
- update_proxy_configuration, 56
- update_proxy_rule, 56
- update_proxy_rule_group_priorities, 56
- update_proxy_rule_priorities, 56
- update_proxy_settings, 56
- update_public_dns_namespace, 87
- update_public_key, 33
- update_readiness_check, 80
- update_realtime_log_config, 33
- update_recovery_group, 80
- update_request_validator, 8
- update_resiliency_group, 39
- update_resolver_config, 84
- update_resolver_dnssec_config, 84
- update_resolver_endpoint, 84
- update_resolver_rule, 84
- update_resource, 8
- update_resource_configuration, 94
- update_resource_gateway, 94
- update_resource_set, 80
- update_response_headers_policy, 33
- update_rest_api, 8
- update_route, 15, 21
- update_route_response, 15
- update_routing_control, 77
- update_routing_control_state, 74
- update_routing_control_states, 74
- update_rule, 94
- update_rule_group, 56
- update_safety_rule, 77
- update_service, 87, 94
- update_service_attributes, 87
- update_service_network, 94
- update_service_network_vpc_association, 94
- update_site, 61
- update_sol_function_package, 90
- update_sol_network_instance, 90
- update_sol_network_package, 90
- update_stage, 8, 15
- update_streaming_distribution, 33
- update_subnet_change_protection, 56
- update_tags_for_domain, 68
- update_target_group, 94
- update_tls_inspection_configuration, 56
- update_traffic_policy_comment, 65
- update_traffic_policy_instance, 65

- update_trust_store, [33](#)
- update_usage, [8](#)
- update_usage_plan, [8](#)
- update_virtual_gateway, [21](#)
- update_virtual_interface_attributes,
[39](#)
- update_virtual_node, [21](#)
- update_virtual_router, [21](#)
- update_virtual_service, [21](#)
- update_vpc_attachment, [61](#)
- update_vpc_link, [8](#), [15](#)
- update_vpc_origin, [33](#)
- update_zonal_autoshift_configuration,
[24](#)
- update_zonal_shift, [24](#)

- validate_sol_function_package_content,
[90](#)
- validate_sol_network_package_content,
[90](#)

- verify_dns_configuration, [33](#)
- view_billing, [68](#)
- vpclattice, [91](#)

- withdraw_byoip_cidr, [51](#)