

Package ‘paws.networking’

August 21, 2019

Title Amazon Web Services Networking & Content Delivery APIs

Version 0.1.4

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

License Apache License (>= 2.0)

Imports paws.common (>= 0.2.0)

Suggests testthat

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Collate 'apigateway_service.R' 'apigateway_interfaces.R'
'apigateway_operations.R' 'apigatewaymanagementapi_service.R'
'apigatewaymanagementapi_interfaces.R'
'apigatewaymanagementapi_operations.R' 'apigatewayv2_service.R'
'apigatewayv2_interfaces.R' 'apigatewayv2_operations.R'
'appmesh_service.R' 'appmesh_interfaces.R'
'appmesh_operations.R' 'cloudfront_service.R'
'cloudfront_interfaces.R' 'cloudfront_operations.R'
'directconnect_service.R' 'directconnect_interfaces.R'
'directconnect_operations.R' 'elb_service.R' 'elb_interfaces.R'
'elb_operations.R' 'elbv2_service.R' 'elbv2_interfaces.R'
'elbv2_operations.R' 'globalaccelerator_service.R'
'globalaccelerator_interfaces.R'
'globalaccelerator_operations.R' 'route53_service.R'
'route53_interfaces.R' 'route53_operations.R'
'route53domains_service.R' 'route53domains_interfaces.R'
'route53domains_operations.R' 'route53resolver_service.R'
'route53resolver_interfaces.R' 'route53resolver_operations.R'
'servicediscovery_service.R' 'servicediscovery_interfaces.R'
'servicediscovery_operations.R'

NeedsCompilation no

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

Maintainer David Kretch <david.kretch@gmail.com>

Repository CRAN

Date/Publication 2019-08-21 10:50:23 UTC

R topics documented:

apigateway	2
apigatewaymanagementapi	5
apigatewayv2	6
appmesh	8
cloudfront	9
directconnect	10
elb	12
elbv2	14
globalaccelerator	16
route53	18
route53domains	20
route53resolver	21
servicediscovery	22
Index	24

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 AWS Lambda, Amazon EC2, or other publicly addressable web services that are hosted outside of AWS.

Usage

```
apigateway()
```

Operations

create_api_key	Create an ApiKey resource
create_authorizer	Adds a new Authorizer resource to an existing RestApi resource
create_base_path_mapping	Creates a new BasePathMapping resource
create_deployment	Creates a Deployment resource, which makes a specified RestApi callable over the internet
create_documentation_part	Create documentation part

create_documentation_version	Create documentation version
create_domain_name	Creates a new domain name
create_model	Adds a new Model resource to an existing RestApi resource
create_request_validator	Creates a RequestValidator of a given RestApi
create_resource	Creates a Resource resource
create_rest_api	Creates a new RestApi resource
create_stage	Creates a new Stage resource that references a pre-existing Deployment for the API
create_usage_plan	Creates a usage plan with the throttle and quota limits, as well as the associated API stages,
create_usage_plan_key	Creates a usage plan key for adding an existing API key to a usage plan
create_vpc_link	Creates a VPC link, under the caller's account in a selected region, in an asynchronous oper
delete_api_key	Deletes the ApiKey resource
delete_authorizer	Deletes an existing Authorizer resource
delete_base_path_mapping	Deletes the BasePathMapping resource
delete_client_certificate	Deletes the ClientCertificate resource
delete_deployment	Deletes a Deployment resource
delete_documentation_part	Delete documentation part
delete_documentation_version	Delete documentation version
delete_domain_name	Deletes the DomainName resource
delete_gateway_response	Clears any customization of a GatewayResponse of a specified response type on the given R
delete_integration	Represents a delete integration
delete_integration_response	Represents a delete integration response
delete_method	Deletes an existing Method resource
delete_method_response	Deletes an existing MethodResponse resource
delete_model	Deletes a model
delete_request_validator	Deletes a RequestValidator of a given RestApi
delete_resource	Deletes a Resource resource
delete_rest_api	Deletes the specified API
delete_stage	Deletes a Stage resource
delete_usage_plan	Deletes a usage plan of a given plan Id
delete_usage_plan_key	Deletes a usage plan key and remove the underlying API key from the associated usage plan
delete_vpc_link	Deletes an existing VpcLink of a specified identifier
flush_stage_authorizers_cache	Flushes all authorizer cache entries on a stage
flush_stage_cache	Flushes a stage's cache
generate_client_certificate	Generates a ClientCertificate resource
get_account	Gets information about the current Account resource
get_api_key	Gets information about the current ApiKey resource
get_api_keys	Gets information about the current ApiKeys resource
get_authorizer	Describe an existing Authorizer resource
get_authorizers	Describe an existing Authorizers resource
get_base_path_mapping	Describe a BasePathMapping resource
get_base_path_mappings	Represents a collection of BasePathMapping resources
get_client_certificate	Gets information about the current ClientCertificate resource
get_client_certificates	Gets a collection of ClientCertificate resources
get_deployment	Gets information about a Deployment resource
get_deployments	Gets information about a Deployments collection
get_documentation_part	Get documentation part
get_documentation_parts	Get documentation parts
get_documentation_version	Get documentation version

get_documentation_versions	Get documentation versions
get_domain_name	Represents a domain name that is contained in a simpler, more intuitive URL that can be called
get_domain_names	Represents a collection of DomainName resources
get_export	Exports a deployed version of a RestApi in a specified format
get_gateway_response	Gets a GatewayResponse of a specified response type on the given RestApi
get_gateway_responses	Gets the GatewayResponses collection on the given RestApi
get_integration	Get the integration settings
get_integration_response	Represents a get integration response
get_method	Describe an existing Method resource
get_method_response	Describes a MethodResponse resource
get_model	Describes an existing model defined for a RestApi resource
get_model_template	Generates a sample mapping template that can be used to transform a payload into the structure of a RestApi resource
get_models	Describes existing Models defined for a RestApi resource
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	Get sdk type
get_sdk_types	Get sdk types
get_stage	Gets information about a Stage resource
get_stages	Gets information about one or more Stage resources
get_tags	Gets the Tags collection for a given resource
get_usage	Gets the usage data of a usage plan in a specified time interval
get_usage_plan	Gets a usage plan of a given plan identifier
get_usage_plan_key	Gets a usage plan key of a given key identifier
get_usage_plan_keys	Gets all the usage plan keys representing the API keys added to a specified usage plan
get_usage_plans	Gets all the usage plans of the caller's account
get_vpc_link	Gets a specified VPC link under the caller's account in a region
get_vpc_links	Gets the VpcLinks collection under the caller's account in a selected region
import_api_keys	Import API keys from an external source, such as a CSV-formatted file
import_documentation_parts	Import documentation parts
import_rest_api	A feature of the API Gateway control service for creating a new API from an external API definition
put_gateway_response	Creates a customization of a GatewayResponse of a specified response type and status code
put_integration	Sets up a method's integration
put_integration_response	Represents a put integration
put_method	Add a method to an existing Resource resource
put_method_response	Adds a MethodResponse to an existing Method resource
put_rest_api	A feature of the API Gateway control service for updating an existing API with an input of definition
tag_resource	Adds or updates a tag on a given resource
test_invoke_authorizer	Simulate the execution of an Authorizer in your RestApi with headers, parameters, and an incoming request
test_invoke_method	Simulate the execution of a Method in your RestApi with headers, parameters, and an incoming request
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

<code>update_base_path_mapping</code>	Changes information about the BasePathMapping resource
<code>update_client_certificate</code>	Changes information about an ClientCertificate resource
<code>update_deployment</code>	Changes information about a Deployment resource
<code>update_documentation_part</code>	Update documentation part
<code>update_documentation_version</code>	Update documentation version
<code>update_domain_name</code>	Changes information about the DomainName resource
<code>update_gateway_response</code>	Updates a GatewayResponse of a specified response type on the given RestApi
<code>update_integration</code>	Represents an update integration
<code>update_integration_response</code>	Represents an update integration response
<code>update_method</code>	Updates an existing Method resource
<code>update_method_response</code>	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 with a speci
<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

```
svc <- apigateway()
svc$create_api_key(
  Foo = 123
)
```

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()
```

Operations

`post_to_connection` Sends the provided data to the specified connection

Examples

```
svc <- apigatewaymanagementapi()
svc$post_to_connection(
  Foo = 123
)
```

apigatewayv2

AmazonApiGatewayV2

Description

Amazon API Gateway V2

Usage

```
apigatewayv2()
```

Operations

<code>create_api</code>	Creates an Api resource
<code>create_api_mapping</code>	Creates an API mapping
<code>create_authorizer</code>	Creates an Authorizer for an API
<code>create_deployment</code>	Creates a Deployment for an API
<code>create_domain_name</code>	Creates a domain name
<code>create_integration</code>	Creates an Integration
<code>create_integration_response</code>	Creates an IntegrationResponses
<code>create_model</code>	Creates a Model for an API
<code>create_route</code>	Creates a Route for an API
<code>create_route_response</code>	Creates a RouteResponse for a Route
<code>create_stage</code>	Creates a Stage for an API
<code>delete_api</code>	Deletes an Api resource
<code>delete_api_mapping</code>	Deletes an API mapping
<code>delete_authorizer</code>	Deletes an Authorizer
<code>delete_deployment</code>	Deletes a Deployment
<code>delete_domain_name</code>	Deletes a domain name
<code>delete_integration</code>	Deletes an Integration
<code>delete_integration_response</code>	Deletes an IntegrationResponses
<code>delete_model</code>	Deletes a Model
<code>delete_route</code>	Deletes a Route
<code>delete_route_response</code>	Deletes a RouteResponse
<code>delete_stage</code>	Deletes a Stage

<code>get_api</code>	Gets an Api resource
<code>get_api_mapping</code>	The API mapping
<code>get_api_mappings</code>	The API mappings
<code>get_apis</code>	Gets a collection of Api resources
<code>get_authorizer</code>	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_model_template</code>	Gets a model template
<code>get_models</code>	Gets the Models for an API
<code>get_route</code>	Gets a Route
<code>get_route_response</code>	Gets a RouteResponse
<code>get_route_responses</code>	Gets the RouteResponses for a Route
<code>get_routes</code>	Gets the Routes for an API
<code>get_stage</code>	Gets a Stage
<code>get_stages</code>	Gets the Stages for an API
<code>get_tags</code>	Gets the Tags for an API
<code>tag_resource</code>	Tag an APIGW resource
<code>untag_resource</code>	Untag an APIGW resource
<code>update_api</code>	Updates an Api resource
<code>update_api_mapping</code>	The API mapping
<code>update_authorizer</code>	Updates an Authorizer
<code>update_deployment</code>	Updates a Deployment
<code>update_domain_name</code>	Updates a domain name
<code>update_integration</code>	Updates an Integration
<code>update_integration_response</code>	Updates an IntegrationResponses
<code>update_model</code>	Updates a Model
<code>update_route</code>	Updates a Route
<code>update_route_response</code>	Updates a RouteResponse
<code>update_stage</code>	Updates a Stage

Examples

```
svc <- apigatewayv2()  
svc$create_api(  
  Foo = 123  
)
```

appmesh

*AWS App Mesh***Description**

AWS 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 AWS Fargate, Amazon ECS, Amazon EKS, Kubernetes on AWS, and Amazon EC2.

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

Usage

appmesh()

Operations

<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_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_mesh</code>	Deletes an existing service mesh
<code>delete_route</code>	Deletes an existing route
<code>delete_virtual_node</code>	Deletes an existing virtual node
<code>delete_virtual_router</code>	Deletes an existing virtual router
<code>delete_virtual_service</code>	Deletes an existing virtual service
<code>describe_mesh</code>	Describes an existing service mesh
<code>describe_route</code>	Describes an existing route
<code>describe_virtual_node</code>	Describes an existing virtual node
<code>describe_virtual_router</code>	Describes an existing virtual router
<code>describe_virtual_service</code>	Describes an existing virtual service
<code>list_meshes</code>	Returns a list of existing service meshes
<code>list_routes</code>	Returns a list of existing routes in a service mesh
<code>list_tags_for_resource</code>	List the tags for an App Mesh resource
<code>list_virtual_nodes</code>	Returns a list of existing virtual nodes
<code>list_virtual_routers</code>	Returns a list of existing virtual routers in a service mesh
<code>list_virtual_services</code>	Returns a list of existing virtual services in a service mesh
<code>tag_resource</code>	Associates the specified tags to a resource with the specified resourceArn
<code>untag_resource</code>	Deletes specified tags from a resource
<code>update_mesh</code>	Updates an existing service mesh

<code>update_route</code>	Updates an existing route for a specified service mesh and virtual router
<code>update_virtual_node</code>	Updates an existing virtual node in a specified service mesh
<code>update_virtual_router</code>	Updates an existing virtual router in a specified service mesh
<code>update_virtual_service</code>	Updates an existing virtual service in a specified service mesh

Examples

```
svc <- appmesh()
svc$create_mesh(
  Foo = 123
)
```

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()
```

Operations

<code>create_cloud_front_origin_access_identity</code>	Creates a new origin access identity
<code>create_distribution</code>	Creates a new web distribution
<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_invalidation</code>	Create a new invalidation
<code>create_public_key</code>	Add a new public key to CloudFront to use, for example, for field-level encryption
<code>create_streaming_distribution</code>	Creates a new RTMP distribution
<code>create_streaming_distribution_with_tags</code>	Create a new streaming distribution with tags
<code>delete_cloud_front_origin_access_identity</code>	Delete an origin access identity
<code>delete_distribution</code>	Delete a distribution
<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_public_key</code>	Remove a public key you previously added to CloudFront
<code>delete_streaming_distribution</code>	Delete a streaming distribution
<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

get_distribution	Get the information about a distribution
get_distribution_config	Get the configuration information about a distribution
get_field_level_encryption	Get the field-level encryption configuration information
get_field_level_encryption_config	Get the field-level encryption configuration information
get_field_level_encryption_profile	Get the field-level encryption profile information
get_field_level_encryption_profile_config	Get the field-level encryption profile configuration information
get_invalidation	Get the information about an invalidation
get_public_key	Get the public key information
get_public_key_config	Return public key configuration information
get_streaming_distribution	Gets information about a specified RTMP distribution, including the distribution ID
get_streaming_distribution_config	Get the configuration information about a streaming distribution
list_cloud_front_origin_access_identities	Lists origin access identities
list_distributions	List CloudFront distributions
list_distributions_by_web_acl_id	List the distributions that are associated with a specified AWS WAF web ACL
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_invalidations	Lists invalidation batches
list_public_keys	List all public keys that have been added to CloudFront for this account
list_streaming_distributions	List streaming distributions
list_tags_for_resource	List tags for a CloudFront resource
tag_resource	Add tags to a CloudFront resource
untag_resource	Remove tags from a CloudFront resource
update_cloud_front_origin_access_identity	Update an origin access identity
update_distribution	Updates the configuration for a web distribution
update_field_level_encryption_config	Update a field-level encryption configuration
update_field_level_encryption_profile	Update a field-level encryption profile
update_public_key	Update public key information
update_streaming_distribution	Update a streaming distribution

Examples

```

svc <- cloudfront()
svc$create_cloud_front_origin_access_identity(
  Foo = 123
)

```

directconnect

AWS Direct Connect

Description

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection in place, you can create virtual interfaces directly to

the AWS 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 AWS Regions except the China (Beijing) and (China) Ningxia Regions. AWS resources in the China Regions can only be accessed through locations associated with those Regions.

Usage

```
directconnect()
```

Operations

accept_direct_connect_gateway_association_proposal	Accepts a proposal request to attach a virtual private gateway or transit virtual gateway to a Direct Connect gateway.
allocate_connection_on_interconnect	Deprecated
allocate_hosted_connection	Creates a hosted connection on the specified interconnect or a link aggregation group (LAG).
allocate_private_virtual_interface	Provisions a private virtual interface to be owned by the specified customer.
allocate_public_virtual_interface	Provisions a public virtual interface to be owned by the specified customer.
allocate_transit_virtual_interface	Provisions a transit virtual interface to be owned by the specified customer.
associate_connection_with_lag	Associates an existing connection with a link aggregation group (LAG).
associate_hosted_connection	Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG).
associate_virtual_interface	Associates a virtual interface with a specified link aggregation group (LAG).
confirm_connection	Confirms the creation of the specified hosted connection on an interconnect.
confirm_private_virtual_interface	Accepts ownership of a private virtual interface created by another customer.
confirm_public_virtual_interface	Accepts ownership of a public virtual interface created by another customer.
confirm_transit_virtual_interface	Accepts ownership of a transit virtual interface created by another customer.
create_bgp_peer	Creates a BGP peer on the specified virtual interface.
create_connection	Creates a connection between a customer network and a specific AWS Region.
create_direct_connect_gateway	Creates a Direct Connect gateway, which is an intermediate object between a customer network and an AWS Region.
create_direct_connect_gateway_association	Creates an association between a Direct Connect gateway and a virtual private gateway or transit virtual gateway.
create_direct_connect_gateway_association_proposal	Creates a proposal to associate the specified virtual private gateway or transit virtual gateway with a Direct Connect gateway.
create_interconnect	Creates an interconnect between an AWS Direct Connect Partner's network and an AWS Region.
create_lag	Creates a link aggregation group (LAG) with the specified number of virtual interfaces.
create_private_virtual_interface	Creates a private virtual interface.
create_public_virtual_interface	Creates a public virtual interface.
create_transit_virtual_interface	Creates a transit virtual interface.
delete_bgp_peer	Deletes the specified BGP peer on the specified virtual interface.
delete_connection	Deletes the specified connection.
delete_direct_connect_gateway	Deletes the specified Direct Connect gateway.
delete_direct_connect_gateway_association	Deletes the association between the specified Direct Connect gateway and virtual private gateway or transit virtual gateway.
delete_direct_connect_gateway_association_proposal	Deletes the association proposal request between the specified Direct Connect gateway and virtual private gateway or transit virtual gateway.
delete_interconnect	Deletes the specified interconnect.
delete_lag	Deletes the specified link aggregation group (LAG).
delete_virtual_interface	Deletes a virtual interface.
describe_connection_loa	Deprecated
describe_connections	Displays the specified connection or all connections in this Region.
describe_connections_on_interconnect	Deprecated
describe_direct_connect_gateway_association_proposals	Describes one or more association proposals for connection between a Direct Connect gateway and a virtual private gateway or transit virtual gateway.
describe_direct_connect_gateway_associations	Lists the associations between your Direct Connect gateways and virtual private gateways or transit virtual gateways.
describe_direct_connect_gateway_attachments	Lists the attachments between your Direct Connect gateways and virtual private gateways or transit virtual gateways.

describe_direct_connect_gateways	Lists all your Direct Connect gateways or only the specified Direct Connect gateways
describe_hosted_connections	Lists the hosted connections that have been provisioned on the specified Direct Connect connection
describe_interconnect_loa	Deprecated
describe_interconnects	Lists the interconnects owned by the AWS account or only the specified interconnects
describe_lags	Describes all your link aggregation groups (LAG) or the specified link aggregation group (LAG)
describe_loa	Gets the LOA-CFA for a connection, interconnect, or link aggregation group (LAG)
describe_locations	Lists the AWS Direct Connect locations in the current AWS Region
describe_tags	Describes the tags associated with the specified AWS Direct Connect resource
describe_virtual_gateways	Lists the virtual private gateways owned by the AWS account
describe_virtual_interfaces	Displays all virtual interfaces for an AWS account
disassociate_connection_from_lag	Disassociates a connection from a link aggregation group (LAG)
tag_resource	Adds the specified tags to the specified AWS Direct Connect resource
untag_resource	Removes one or more tags from the specified AWS Direct Connect resource
update_direct_connect_gateway_association	Updates the specified attributes of the Direct Connect gateway association
update_lag	Updates the attributes of the specified link aggregation group (LAG)
update_virtual_interface_attributes	Updates the specified attributes of the specified virtual private interface

Examples

```
svc <- directconnect()
svc$accept_direct_connect_gateway_association_proposal(
  Foo = 123
)
```

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 `CreateLoadBalancer`. Register your instances with the load balancer using `RegisterInstancesWithLoadBalancer`.

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

```
e lb()
```

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 specified load balancer
disable_availability_zones_for_load_balancer	Removes the specified Availability Zones from the set of Availability Zones for the specified load balancer
enable_availability_zones_for_load_balancer	Adds the specified Availability Zones to the set of Availability Zones for the specified load balancer
modify_load_balancer_attributes	Modifies the attributes of the specified load balancer
register_instances_with_load_balancer	Adds the specified instances to the specified load balancer
remove_tags	Removes one or more tags from the specified load balancer
set_load_balancer_listener_ssl_certificate	Sets the certificate that terminates the specified listener's SSL connections
set_load_balancer_policies_for_backend_server	Replaces the set of policies associated with the specified port on which the specified load balancer listens
set_load_balancer_policies_of_listener	Replaces the current set of policies for the specified load balancer port with the specified policies

Examples

```
# This example adds two tags to the specified load balancer.
svc <- e lb()
svc$add_tags(
  LoadBalancerNames = list(
    "my-load-balancer"
  ),
  Tags = list(
    list(
      Key = "project",
```

```
        Value = "lima"
    ),
    list(
        Key = "department",
        Value = "digital-media"
    )
)
)
```

elbv2

Elastic Load Balancing

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, and Classic Load Balancers.

An Application Load Balancer makes routing and load balancing decisions at the application layer (HTTP/HTTPS). A Network Load Balancer makes routing and load balancing decisions at the transport layer (TCP/TLS). Both Application Load Balancers and Network Load Balancers can route requests to one or more ports on each EC2 instance or container instance in your virtual private cloud (VPC).

A Classic Load Balancer makes routing and load balancing decisions either at the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS), and supports either EC2-Classic or a VPC. For more information, see the [Elastic Load Balancing User Guide](#).

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

To get started, complete the following tasks:

1. Create a load balancer using `CreateLoadBalancer`.
2. Create a target group using `CreateTargetGroup`.
3. Register targets for the target group using `RegisterTargets`.
4. Create one or more listeners for your load balancer using `CreateListener`.

To delete a load balancer and its related resources, complete the following tasks:

1. Delete the load balancer using `DeleteLoadBalancer`.
2. Delete the target group using `DeleteTargetGroup`.

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()
```

Operations

add_listener_certificates	Adds the specified SSL server certificate to the certificate list for the specified HTTPS listener
add_tags	Adds the specified tags to the specified Elastic Load Balancing resource
create_listener	Creates a listener for the specified Application Load Balancer or Network Load Balancer
create_load_balancer	Creates an Application Load Balancer or a Network Load Balancer
create_rule	Creates a rule for the specified listener
create_target_group	Creates a target group
delete_listener	Deletes the specified listener
delete_load_balancer	Deletes the specified Application Load Balancer or Network Load Balancer and its attached listeners
delete_rule	Deletes the specified rule
delete_target_group	Deletes the specified target group
deregister_targets	Deregisters the specified targets from the specified target group
describe_account_limits	Describes the current Elastic Load Balancing resource limits for your AWS account
describe_listener_certificates	Describes the default certificate and the certificate list for the specified HTTPS listener
describe_listeners	Describes the specified listeners or the listeners for the specified Application Load Balancer or Network Load Balancer
describe_load_balancer_attributes	Describes the attributes for the specified Application Load Balancer or Network Load Balancer
describe_load_balancers	Describes the specified load balancers or all of your load balancers
describe_rules	Describes the specified rules or the rules for the specified listener
describe_ssl_policies	Describes the specified policies or all policies used for SSL negotiation
describe_tags	Describes the tags for the specified resources
describe_target_group_attributes	Describes the attributes for the specified target group
describe_target_groups	Describes the specified target groups or all of your target groups
describe_target_health	Describes the health of the specified targets or all of your targets
modify_listener	Modifies the specified properties of the specified listener
modify_load_balancer_attributes	Modifies the specified attributes of the specified Application Load Balancer or Network Load Balancer
modify_rule	Modifies the specified rule
modify_target_group	Modifies the health checks used when evaluating the health state of the targets in the specified target group
modify_target_group_attributes	Modifies the specified attributes of the specified target group
register_targets	Registers the specified targets with the specified target group
remove_listener_certificates	Removes the specified certificate from the certificate list for the specified HTTPS listener
remove_tags	Removes the specified tags from the specified Elastic Load Balancing resource
set_ip_address_type	Sets the type of IP addresses used by the subnets of the specified Application Load Balancer or Network Load Balancer
set_rule_priorities	Sets the priorities of the specified rules
set_security_groups	Associates the specified security groups with the specified Application Load Balancer or Network Load Balancer
set_subnets	Enables the Availability Zone for the specified public subnets for the specified Application Load Balancer or Network Load Balancer

Examples

```
# This example adds the specified tags to the specified load balancer.
svc <- elbv2()
svc$add_tags(
  ResourceArns = list(
```

```

    "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/5..."
  ),
  Tags = list(
    list(
      Key = "project",
      Value = "lima"
    ),
    list(
      Key = "department",
      Value = "digital-media"
    )
  )
)
)
)

```

globalaccelerator *AWS Global Accelerator*

Description

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

AWS Global Accelerator is a network layer service in which you create accelerators to improve availability and performance for internet applications used by a global audience.

Global Accelerator provides you with static IP addresses that you associate with your accelerator. These IP addresses are anycast from the AWS edge network and distribute incoming application traffic across multiple endpoint resources in multiple AWS Regions, which increases the availability of your applications. Endpoints can be Elastic IP addresses, Network Load Balancers, and Application Load Balancers that are located in one AWS Region or multiple Regions.

Global Accelerator uses the AWS 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 directed to only healthy endpoints.

Global Accelerator includes components that work together to help you improve performance and availability for your applications:

Static IP address

: AWS Global Accelerator provides you with a set of static IP addresses which are anycast from the AWS edge network and serve as the single fixed entry points for your clients. If you already have Elastic Load Balancing or Elastic IP address resources set up for your applications, you can easily add those to Global Accelerator to allow the resources to be accessed by a Global Accelerator static IP address.

Accelerator

: An accelerator directs traffic to optimal endpoints over the AWS global network to improve availability and performance for your internet applications that have a global audience. Each accelerator includes one or more listeners.

Network zone

: A network zone services the static IP addresses for your accelerator from a unique IP subnet. Similar to an AWS Availability Zone, a network zone is an isolated unit with its own set of physical infrastructure. When you configure an accelerator, Global Accelerator allocates two IPv4 addresses for it. If one IP address from a network zone becomes unavailable due to IP address blocking by certain client networks, or network disruptions, then client applications can retry on the healthy static IP address from the other isolated network zone.

Listener

: A listener processes inbound connections from clients to Global Accelerator, based on the protocol and port that you configure. Each listener has one or more endpoint groups associated with it, and traffic is forwarded to endpoints in one of the groups. You associate endpoint groups with listeners by specifying the Regions that you want to distribute traffic to. Traffic is distributed to optimal endpoints within the endpoint groups associated with a listener.

Endpoint group

: Each endpoint group is associated with a specific AWS Region. Endpoint groups include one or more endpoints in the Region. You can increase or reduce the percentage of traffic that would be otherwise directed to an endpoint group by adjusting a setting called a *traffic dial*. The traffic dial lets you easily do performance testing or blue/green deployment testing for new releases across different AWS Regions, for example.

Endpoint

: An endpoint is an Elastic IP address, Network Load Balancer, or Application Load Balancer. Traffic is routed to endpoints based on several factors, including the geo-proximity to the user, the health of the endpoint, and the configuration options that you choose, such as endpoint weights. For each endpoint, you can configure weights, which are numbers that you can use to specify the proportion of traffic to route to each one. This can be useful, for example, to do performance testing within a Region.

Usage

```
globalaccelerator()
```

Operations

create_accelerator	Create an accelerator
create_endpoint_group	Create an endpoint group for the specified listener
create_listener	Create a listener to process inbound connections from clients to an accelerator
delete_accelerator	Delete an accelerator
delete_endpoint_group	Delete an endpoint group from a listener
delete_listener	Delete a listener from an accelerator
describe_accelerator	Describe an accelerator
describe_accelerator_attributes	Describe the attributes of an accelerator
describe_endpoint_group	Describe an endpoint group
describe_listener	Describe a listener
list_accelerators	List the accelerators for an AWS account
list_endpoint_groups	List the endpoint groups that are associated with a listener
list_listeners	List the listeners for an accelerator

update_accelerator	Update an accelerator
update_accelerator_attributes	Update the attributes for an accelerator
update_endpoint_group	Update an endpoint group
update_listener	Update a listener

Examples

```
svc <- globalaccelerator()
svc$create_accelerator(
  Foo = 123
)
```

route53

Amazon Route 53

Description

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

Usage

```
route53()
```

Operations

associate_vpc_with_hosted_zone	Associates an Amazon VPC with a private hosted zone
change_resource_record_sets	Creates, changes, or deletes a resource record set, which contains authoritative DNS information
change_tags_for_resource	Adds, edits, or deletes tags for a health check or a hosted zone
create_health_check	Creates a new health check
create_hosted_zone	Creates a new public or private hosted zone
create_query_logging_config	Creates a configuration for DNS query logging
create_reusable_delegation_set	Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones
create_traffic_policy	Creates a traffic policy, which you use to create multiple DNS resource record sets
create_traffic_policy_instance	Creates resource record sets in a specified hosted zone based on the settings in a traffic policy
create_traffic_policy_version	Creates a new version of an existing traffic policy
create_vpc_association_authorization	Authorizes the AWS account that created a specified VPC to submit an AssociateVPCWithHostedZone request
delete_health_check	Deletes a health check
delete_hosted_zone	Deletes a hosted zone
delete_query_logging_config	Deletes a configuration for DNS query logging
delete_reusable_delegation_set	Deletes a reusable delegation set
delete_traffic_policy	Deletes a traffic policy
delete_traffic_policy_instance	Deletes a traffic policy instance and all of the resource record sets that Amazon Route 53 created for the instance
delete_vpc_association_authorization	Removes authorization to submit an AssociateVPCWithHostedZone request
disassociate_vpc_from_hosted_zone	Disassociates a VPC from a Amazon Route 53 private hosted zone

<code>get_account_limit</code>	Gets the specified limit for the current account, for example, the maximum number of hosted zones
<code>get_change</code>	Returns the current status of a change batch request
<code>get_checker_ip_ranges</code>	GetCheckerIpRanges still works, but we recommend that you download ip-ranges from the AWS CLI
<code>get_geo_location</code>	Gets information about whether a specified geographic location is supported for hosted zones
<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 AWS 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
<code>get_hosted_zone_count</code>	Retrieves the number of hosted zones that are associated with the current AWS account
<code>get_hosted_zone_limit</code>	Gets the specified limit for a specified hosted zone, for example, the maximum number of hosted zones
<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 number of hosted zones
<code>get_reusable_delegation_set_limit</code>	Gets the maximum number of hosted zones that you can associate with the specified reusable delegation set
<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 AWS account
<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 AWS account
<code>list_hosted_zones</code>	Retrieves a list of the public and private hosted zones that are associated with the current AWS account
<code>list_hosted_zones_by_name</code>	Retrieves a list of your hosted zones in lexicographic order
<code>list_query_logging_configs</code>	Lists the configurations for DNS query logging that are associated with the current AWS 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 AWS 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 AWS account
<code>list_traffic_policy_instances</code>	Gets information about the traffic policy instances that you created by using the AWS CLI
<code>list_traffic_policy_instances_by_hosted_zone</code>	Gets information about the traffic policy instances that you created in a specified hosted zone
<code>list_traffic_policy_instances_by_policy</code>	Gets information about the traffic policy instances that you created by using a specific traffic policy
<code>list_traffic_policy_versions</code>	Gets information about all of the versions for a specified traffic policy
<code>list_vpc_association_authorizations</code>	Gets a list of the VPCs that were created by other accounts and that can be associated with the current AWS account
<code>test_dns_answer</code>	Gets the value that Amazon Route 53 returns in response to a DNS request for a specified hosted zone
<code>update_health_check</code>	Updates an existing health check
<code>update_hosted_zone_comment</code>	Updates the comment for a specified hosted zone
<code>update_traffic_policy_comment</code>	Updates the comment for a specified traffic policy version
<code>update_traffic_policy_instance</code>	Updates the resource record sets in a specified hosted zone that were created by the current AWS account

Examples

```
# The following example associates the VPC with ID vpc-1a2b3c4d with the
# hosted zone with ID Z3M3LMPEXAMPLE.
svc <- route53()
svc$associate_vpc_with_hosted_zone(
  Comment = "",
  HostedZoneId = "Z3M3LMPEXAMPLE",
  VPC = list(
```

```

        VPCId = "vpc-1a2b3c4d",
        VPCRegion = "us-east-2"
    )
)

```

route53domains

Amazon Route 53 Domains

Description

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

Usage

```
route53domains()
```

Operations

check_domain_availability	This operation checks the availability of one domain name
check_domain_transferability	Checks whether a domain name can be transferred to Amazon Route 53
delete_tags_for_domain	This operation deletes the specified tags for a domain
disable_domain_auto_renew	This operation disables automatic renewal of domain registration for the specified domain
disable_domain_transfer_lock	This operation removes the transfer lock on the domain (specifically the clientTransferProhibit
enable_domain_auto_renew	This operation configures Amazon Route 53 to automatically renew the specified domain
enable_domain_transfer_lock	This operation sets the transfer lock on the domain (specifically the clientTransferProhibit
get_contact_reachability_status	For operations that require confirmation that the email address for the registrant contact i
get_domain_detail	This operation returns detailed information about a specified domain that is associated w
get_domain_suggestions	The GetDomainSuggestions operation returns a list of suggested domain names given a s
get_operation_detail	This operation returns the current status of an operation that is not completed
list_domains	This operation returns all the domain names registered with Amazon Route 53 for the cu
list_operations	This operation returns the operation IDs of operations that are not yet complete
list_tags_for_domain	This operation returns all of the tags that are associated with the specified domain
register_domain	This operation registers a domain
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 registrant contact i
retrieve_domain_auth_code	This operation returns the AuthCode for the domain
transfer_domain	This operation transfers a domain from another registrar to Amazon Route 53
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 with the specified
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 AWS account for a specifie

Examples

```
svc <- route53domains()
svc$check_domain_availability(
  Foo = 123
)
```

route53resolver	<i>Amazon Route 53 Resolver</i>
-----------------	---------------------------------

Description

Here's how you set up to query an Amazon Route 53 private hosted zone from your network:

1. Connect your network to a VPC using AWS Direct Connect or a VPN.
2. Run the following AWS CLI command to create a Resolver endpoint:

```
create-resolver-endpoint --name \[endpoint_name\] --direction INBOUND --creator-request-id \[unique_string\] --security-group-ids \[security_group_with_inbound_rules\] --ip-addresses SubnetId=\[subnet_id\] SubnetId=\[subnet_id_in_different_AZ\]
```

Note the resolver endpoint ID that appears in the response. You'll use it in step 3.
3. Get the IP addresses for the Resolver endpoints:

```
get-resolver-endpoint --resolver-endpoint-id \[resolver_endpoint_id\]
```
4. In your network configuration, define the IP addresses that you got in step 3 as DNS servers. You can now query instance names in your VPCs and the names of records in your private hosted zone.

You can also perform the following operations using the AWS CLI:

- `list-resolver-endpoints`: List all endpoints. The syntax includes options for pagination and filtering.
- `update-resolver-endpoints`: Add IP addresses to an endpoint or remove IP addresses from an endpoint.

To delete an endpoint, use the following AWS CLI command:

```
delete-resolver-endpoint --resolver-endpoint-id \[resolver_endpoint_id\]
```

Usage

```
route53resolver()
```

Operations

<code>associate_resolver_endpoint_ip_address</code>	Adds IP addresses to an inbound or an outbound resolver endpoint
<code>associate_resolver_rule</code>	Associates a resolver rule with a VPC
<code>create_resolver_endpoint</code>	Creates a resolver endpoint
<code>create_resolver_rule</code>	For DNS queries that originate in your VPCs, specifies which resolver endpoint
<code>delete_resolver_endpoint</code>	Deletes a resolver endpoint
<code>delete_resolver_rule</code>	Deletes a resolver rule
<code>disassociate_resolver_endpoint_ip_address</code>	Removes IP addresses from an inbound or an outbound resolver endpoint
<code>disassociate_resolver_rule</code>	Removes the association between a specified resolver rule and a specified VPC
<code>get_resolver_endpoint</code>	Gets information about a specified resolver endpoint, such as whether it's an inbound or outbound
<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 a resolver rule policy
<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 AWS account
<code>list_resolver_rule_associations</code>	Lists the associations that were created between resolver rules and VPCs using the current AWS account
<code>list_resolver_rules</code>	Lists the resolver rules that were created using the current AWS account
<code>list_tags_for_resource</code>	Lists the tags that you associated with the specified resource
<code>put_resolver_rule_policy</code>	Specifies the Resolver operations and resources that you want to allow another AWS account to perform
<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_resolver_endpoint</code>	Updates the name of an inbound or an outbound resolver endpoint
<code>update_resolver_rule</code>	Updates settings for a specified resolver rule

Examples

```
svc <- route53resolver()
svc$associate_resolver_endpoint_ip_address(
  Foo = 123
)
```

servicediscovery

AWS Cloud Map

Description

AWS Cloud Map lets you configure public DNS, private DNS, or HTTP namespaces that your microservice applications run in. When an instance of the service becomes available, you can call the AWS Cloud Map API to register the instance with AWS Cloud Map. For public or private DNS namespaces, AWS 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()
```

Operations

create_http_namespace	Creates an HTTP namespace
create_private_dns_namespace	Creates a private namespace based on DNS, which will be visible only inside a specified VPC
create_public_dns_namespace	Creates a public namespace based on DNS, which will be visible on the internet
create_service	Creates a service, which defines the configuration for the following entities: - For public namespaces, creates a health check and a Route 53 record. - For private namespaces, creates a health check and a Route 53 record. - For HTTP namespaces, creates a health check and a Route 53 record.
delete_namespace	Deletes a namespace from the current account
delete_service	Deletes a specified service
deregister_instance	Deletes the Amazon Route 53 DNS records and health check, if any, that AWS CloudMap created for the specified instance
discover_instances	Discovers registered instances for a specified namespace and service
get_instance	Gets information about a specified instance
get_instances_health_status	Gets the current health status (Healthy, Unhealthy, or Unknown) of one or more instances
get_namespace	Gets information about a namespace
get_operation	Gets information about any operation that returns an operation ID in the response, such as create_namespace or delete_namespace
get_service	Gets the settings for a specified service
list_instances	Lists summary information about the instances that you registered by using a specified namespace and service
list_namespaces	Lists summary information about the namespaces that were created by the current account
list_operations	Lists operations that match the criteria that you specify
list_services	Lists summary information for all the services that are associated with one or more namespaces
register_instance	Creates or updates one or more records and, optionally, creates a health check based on the specified settings
update_instance_custom_health_status	Submits a request to change the health status of a custom health check to healthy or unhealthy
update_service	Submits a request to perform the following operations: - Add or delete DnsRecords

Examples

```
svc <- servicediscovery()
svc$create_http_namespace(
  Foo = 123
)
```

Index

accept_direct_connect_gateway_association_proposal, [11](#)
add_listener_certificates, [15](#)
add_tags, [13](#), [15](#)
allocate_connection_on_interconnect, [11](#)
allocate_hosted_connection, [11](#)
allocate_private_virtual_interface, [11](#)
allocate_public_virtual_interface, [11](#)
allocate_transit_virtual_interface, [11](#)
apigateway, [2](#)
apigatewaymanagementapi, [5](#)
apigatewayv2, [6](#)
apply_security_groups_to_load_balancer, [13](#)
apmesh, [8](#)
associate_connection_with_lag, [11](#)
associate_hosted_connection, [11](#)
associate_resolver_endpoint_ip_address, [22](#)
associate_resolver_rule, [22](#)
associate_virtual_interface, [11](#)
associate_vpc_with_hosted_zone, [18](#)
attach_load_balancer_to_subnets, [13](#)

change_resource_record_sets, [18](#)
change_tags_for_resource, [18](#)
check_domain_availability, [20](#)
check_domain_transferability, [20](#)
cloudfront, [9](#)
configure_health_check, [13](#)
confirm_connection, [11](#)
confirm_private_virtual_interface, [11](#)
confirm_public_virtual_interface, [11](#)
confirm_transit_virtual_interface, [11](#)
create_accelerator, [17](#)
create_api, [6](#)
create_api_key, [2](#)
create_api_mapping, [6](#)
create_app_cookie_stickiness_policy, [13](#)
create_authorizer, [2](#), [6](#)
create_base_path_mapping, [2](#)
create_bgp_peer, [11](#)
create_cloud_front_origin_access_identity, [9](#)
create_connection, [11](#)
create_deployment, [2](#), [6](#)
create_direct_connect_gateway, [11](#)
create_direct_connect_gateway_association, [11](#)
create_direct_connect_gateway_association_proposal, [11](#)
create_distribution, [9](#)
create_distribution_with_tags, [9](#)
create_documentation_part, [2](#)
create_documentation_version, [3](#)
create_domain_name, [3](#), [6](#)
create_endpoint_group, [17](#)
create_field_level_encryption_config, [9](#)
create_field_level_encryption_profile, [9](#)
create_health_check, [18](#)
create_hosted_zone, [18](#)
create_http_namespace, [23](#)
create_integration, [6](#)
create_integration_response, [6](#)
create_interconnect, [11](#)
create_invalidation, [9](#)
create_lag, [11](#)
create_lb_cookie_stickiness_policy, [13](#)
create_listener, [15](#), [17](#)
create_load_balancer, [13](#), [15](#)
create_load_balancer_listeners, [13](#)
create_load_balancer_policy, [13](#)
create_mesh, [8](#)
create_model, [3](#), [6](#)

- create_private_dns_namespace, 23
- create_private_virtual_interface, 11
- create_public_dns_namespace, 23
- create_public_key, 9
- create_public_virtual_interface, 11
- create_query_logging_config, 18
- create_request_validator, 3
- create_resolver_endpoint, 22
- create_resolver_rule, 22
- create_resource, 3
- create_rest_api, 3
- create_reusable_delegation_set, 18
- create_route, 6, 8
- create_route_response, 6
- create_rule, 15
- create_service, 23
- create_stage, 3, 6
- create_streaming_distribution, 9
- create_streaming_distribution_with_tags, 9
- create_target_group, 15
- create_traffic_policy, 18
- create_traffic_policy_instance, 18
- create_traffic_policy_version, 18
- create_transit_virtual_interface, 11
- create_usage_plan, 3
- create_usage_plan_key, 3
- create_virtual_node, 8
- create_virtual_router, 8
- create_virtual_service, 8
- create_vpc_association_authorization, 18
- create_vpc_link, 3
- delete_accelerator, 17
- delete_api, 6
- delete_api_key, 3
- delete_api_mapping, 6
- delete_authorizer, 3, 6
- delete_base_path_mapping, 3
- delete_bgp_peer, 11
- delete_client_certificate, 3
- delete_cloud_front_origin_access_identity, 9
- delete_connection, 11
- delete_deployment, 3, 6
- delete_direct_connect_gateway, 11
- delete_direct_connect_gateway_association, 11
- delete_direct_connect_gateway_association_proposal, 11
- delete_distribution, 9
- delete_documentation_part, 3
- delete_documentation_version, 3
- delete_domain_name, 3, 6
- delete_endpoint_group, 17
- delete_field_level_encryption_config, 9
- delete_field_level_encryption_profile, 9
- delete_gateway_response, 3
- delete_health_check, 18
- delete_hosted_zone, 18
- delete_integration, 3, 6
- delete_integration_response, 3, 6
- delete_interconnect, 11
- delete_lag, 11
- delete_listener, 15, 17
- delete_load_balancer, 13, 15
- delete_load_balancer_listeners, 13
- delete_load_balancer_policy, 13
- delete_mesh, 8
- delete_method, 3
- delete_method_response, 3
- delete_model, 3, 6
- delete_namespace, 23
- delete_public_key, 9
- delete_query_logging_config, 18
- delete_request_validator, 3
- delete_resolver_endpoint, 22
- delete_resolver_rule, 22
- delete_resource, 3
- delete_rest_api, 3
- delete_reusable_delegation_set, 18
- delete_route, 6, 8
- delete_route_response, 6
- delete_rule, 15
- delete_service, 23
- delete_stage, 3, 6
- delete_streaming_distribution, 9
- delete_tags_for_domain, 20
- delete_target_group, 15
- delete_traffic_policy, 18
- delete_traffic_policy_instance, 18
- delete_usage_plan, 3
- delete_usage_plan_key, 3
- delete_virtual_interface, 11

- delete_virtual_node, [8](#)
- delete_virtual_router, [8](#)
- delete_virtual_service, [8](#)
- delete_vpc_association_authorization, [18](#)
- delete_vpc_link, [3](#)
- deregister_instance, [23](#)
- deregister_instances_from_load_balancer, [13](#)
- deregister_targets, [15](#)
- describe_accelerator, [17](#)
- describe_accelerator_attributes, [17](#)
- describe_account_limits, [13, 15](#)
- describe_connection_loa, [11](#)
- describe_connections, [11](#)
- describe_connections_on_interconnect, [11](#)
- describe_direct_connect_gateway_association_proposals, [11](#)
- describe_direct_connect_gateway_associations, [11](#)
- describe_direct_connect_gateway_attachments, [11](#)
- describe_direct_connect_gateways, [12](#)
- describe_endpoint_group, [17](#)
- describe_hosted_connections, [12](#)
- describe_instance_health, [13](#)
- describe_interconnect_loa, [12](#)
- describe_interconnects, [12](#)
- describe_lags, [12](#)
- describe_listener, [17](#)
- describe_listener_certificates, [15](#)
- describe_listeners, [15](#)
- describe_loa, [12](#)
- describe_load_balancer_attributes, [13, 15](#)
- describe_load_balancer_policies, [13](#)
- describe_load_balancer_policy_types, [13](#)
- describe_load_balancers, [13, 15](#)
- describe_locations, [12](#)
- describe_mesh, [8](#)
- describe_route, [8](#)
- describe_rules, [15](#)
- describe_ssl_policies, [15](#)
- describe_tags, [12, 13, 15](#)
- describe_target_group_attributes, [15](#)
- describe_target_groups, [15](#)
- describe_target_health, [15](#)
- describe_virtual_gateways, [12](#)
- describe_virtual_interfaces, [12](#)
- describe_virtual_node, [8](#)
- describe_virtual_router, [8](#)
- describe_virtual_service, [8](#)
- detach_load_balancer_from_subnets, [13](#)
- directconnect, [10](#)
- disable_availability_zones_for_load_balancer, [13](#)
- disable_domain_auto_renew, [20](#)
- disable_domain_transfer_lock, [20](#)
- disassociate_connection_from_lag, [12](#)
- disassociate_resolver_endpoint_ip_address, [22](#)
- disassociate_resolver_rule, [22](#)
- disassociate_vpc_from_hosted_zone, [18](#)
- dispose_instances, [23](#)
- elb, [12](#)
- elbv2, [14](#)
- enable_availability_zones_for_load_balancer, [13](#)
- enable_domain_auto_renew, [20](#)
- enable_domain_transfer_lock, [20](#)
- flush_stage_authorizers_cache, [3](#)
- flush_stage_cache, [3](#)
- generate_client_certificate, [3](#)
- get_account, [3](#)
- get_account_limit, [19](#)
- get_api, [7](#)
- get_api_key, [3](#)
- get_api_keys, [3](#)
- get_api_mapping, [7](#)
- get_api_mappings, [7](#)
- get_apis, [7](#)
- get_authorizer, [3, 7](#)
- get_authorizers, [3, 7](#)
- get_base_path_mapping, [3](#)
- get_base_path_mappings, [3](#)
- get_change, [19](#)
- get_checker_ip_ranges, [19](#)
- get_client_certificate, [3](#)
- get_client_certificates, [3](#)
- get_cloud_front_origin_access_identity, [9](#)

get_cloud_front_origin_access_identity_config, 9
get_contact_reachability_status, 20
get_deployment, 3, 7
get_deployments, 3, 7
get_distribution, 10
get_distribution_config, 10
get_documentation_part, 3
get_documentation_parts, 3
get_documentation_version, 3
get_documentation_versions, 4
get_domain_detail, 20
get_domain_name, 4, 7
get_domain_names, 4, 7
get_domain_suggestions, 20
get_export, 4
get_field_level_encryption, 10
get_field_level_encryption_config, 10
get_field_level_encryption_profile, 10
get_field_level_encryption_profile_config, 10
get_gateway_response, 4
get_gateway_responses, 4
get_geo_location, 19
get_health_check, 19
get_health_check_count, 19
get_health_check_last_failure_reason, 19
get_health_check_status, 19
get_hosted_zone, 19
get_hosted_zone_count, 19
get_hosted_zone_limit, 19
get_instance, 23
get_instances_health_status, 23
get_integration, 4, 7
get_integration_response, 4, 7
get_integration_responses, 7
get_integrations, 7
get_invalidation, 10
get_method, 4
get_method_response, 4
get_model, 4, 7
get_model_template, 4, 7
get_models, 4, 7
get_namespace, 23
get_operation, 23
get_operation_detail, 20
get_public_key, 10
get_public_key_config, 10
get_query_logging_config, 19
get_request_validator, 4
get_request_validators, 4
get_resolver_endpoint, 22
get_resolver_rule, 22
get_resolver_rule_association, 22
get_resolver_rule_policy, 22
get_resource, 4
get_resources, 4
get_rest_api, 4
get_rest_apis, 4
get_reusable_delegation_set, 19
get_reusable_delegation_set_limit, 19
get_route, 7
get_route_response, 7
get_route_responses, 7
get_routes, 7
get_sdk, 4
get_sdk_type, 4
get_sdk_types, 4
get_service, 23
get_stage, 4, 7
get_stages, 4, 7
get_streaming_distribution, 10
get_streaming_distribution_config, 10
get_tags, 4, 7
get_traffic_policy, 19
get_traffic_policy_instance, 19
get_traffic_policy_instance_count, 19
get_usage, 4
get_usage_plan, 4
get_usage_plan_key, 4
get_usage_plan_keys, 4
get_usage_plans, 4
get_vpc_link, 4
get_vpc_links, 4
globalaccelerator, 16
import_api_keys, 4
import_documentation_parts, 4
import_rest_api, 4
list_accelerators, 17
list_cloud_front_origin_access_identities, 10
list_distributions, 10
list_distributions_by_web_acl_id, 10
list_domains, 20

- list_endpoint_groups, [17](#)
- list_field_level_encryption_configs, [10](#)
- list_field_level_encryption_profiles, [10](#)
- list_geo_locations, [19](#)
- list_health_checks, [19](#)
- list_hosted_zones, [19](#)
- list_hosted_zones_by_name, [19](#)
- list_instances, [23](#)
- list_invalidations, [10](#)
- list_listeners, [17](#)
- list_meshes, [8](#)
- list_namespaces, [23](#)
- list_operations, [20, 23](#)
- list_public_keys, [10](#)
- list_query_logging_configs, [19](#)
- list_resolver_endpoint_ip_addresses, [22](#)
- list_resolver_endpoints, [22](#)
- list_resolver_rule_associations, [22](#)
- list_resolver_rules, [22](#)
- list_resource_record_sets, [19](#)
- list_reusable_delegation_sets, [19](#)
- list_routes, [8](#)
- list_services, [23](#)
- list_streaming_distributions, [10](#)
- list_tags_for_domain, [20](#)
- list_tags_for_resource, [8, 10, 19, 22](#)
- list_tags_for_resources, [19](#)
- list_traffic_policies, [19](#)
- list_traffic_policy_instances, [19](#)
- list_traffic_policy_instances_by_hosted_zone, [19](#)
- list_traffic_policy_instances_by_policy, [19](#)
- list_traffic_policy_versions, [19](#)
- list_virtual_nodes, [8](#)
- list_virtual_routers, [8](#)
- list_virtual_services, [8](#)
- list_vpc_association_authorizations, [19](#)

- modify_listener, [15](#)
- modify_load_balancer_attributes, [13, 15](#)
- modify_rule, [15](#)
- modify_target_group, [15](#)
- modify_target_group_attributes, [15](#)

- post_to_connection, [6](#)
- put_gateway_response, [4](#)
- put_integration, [4](#)
- put_integration_response, [4](#)
- put_method, [4](#)
- put_method_response, [4](#)
- put_resolver_rule_policy, [22](#)
- put_rest_api, [4](#)

- register_domain, [20](#)
- register_instance, [23](#)
- register_instances_with_load_balancer, [13](#)
- register_targets, [15](#)
- remove_listener_certificates, [15](#)
- remove_tags, [13, 15](#)
- renew_domain, [20](#)
- resend_contact_reachability_email, [20](#)
- retrieve_domain_auth_code, [20](#)
- route53, [18](#)
- route53domains, [20](#)
- route53resolver, [21](#)

- servicediscovery, [22](#)
- set_ip_address_type, [15](#)
- set_load_balancer_listener_ssl_certificate, [13](#)
- set_load_balancer_policies_for_backend_server, [13](#)
- set_load_balancer_policies_of_listener, [13](#)
- set_rule_priorities, [15](#)
- set_security_groups, [15](#)
- set_subnets, [15](#)

- tag_resource, [4, 7, 8, 10, 12, 22](#)
- test_dns_answer, [19](#)
- test_invoke_authorizer, [4](#)
- test_invoke_method, [4](#)
- transfer_domain, [20](#)

- untag_resource, [4, 7, 8, 10, 12, 22](#)
- update_accelerator, [18](#)
- update_accelerator_attributes, [18](#)
- update_account, [4](#)
- update_api, [7](#)
- update_api_key, [4](#)
- update_api_mapping, [7](#)
- update_authorizer, [4, 7](#)

update_base_path_mapping, [5](#)
update_client_certificate, [5](#)
update_cloud_front_origin_access_identity,
[10](#)
update_deployment, [5, 7](#)
update_direct_connect_gateway_association,
[12](#)
update_distribution, [10](#)
update_documentation_part, [5](#)
update_documentation_version, [5](#)
update_domain_contact, [20](#)
update_domain_contact_privacy, [20](#)
update_domain_name, [5, 7](#)
update_domain_nameservers, [20](#)
update_endpoint_group, [18](#)
update_field_level_encryption_config,
[10](#)
update_field_level_encryption_profile,
[10](#)
update_gateway_response, [5](#)
update_health_check, [19](#)
update_hosted_zone_comment, [19](#)
update_instance_custom_health_status,
[23](#)
update_integration, [5, 7](#)
update_integration_response, [5, 7](#)
update_lag, [12](#)
update_listener, [18](#)
update_mesh, [8](#)
update_method, [5](#)
update_method_response, [5](#)
update_model, [5, 7](#)
update_public_key, [10](#)
update_request_validator, [5](#)
update_resolver_endpoint, [22](#)
update_resolver_rule, [22](#)
update_resource, [5](#)
update_rest_api, [5](#)
update_route, [7, 9](#)
update_route_response, [7](#)
update_service, [23](#)
update_stage, [5, 7](#)
update_streaming_distribution, [10](#)
update_tags_for_domain, [20](#)
update_traffic_policy_comment, [19](#)
update_traffic_policy_instance, [19](#)
update_usage, [5](#)
update_usage_plan, [5](#)
update_virtual_interface_attributes,
[12](#)
update_virtual_node, [9](#)
update_virtual_router, [9](#)
update_virtual_service, [9](#)
update_vpc_link, [5](#)
view_billing, [20](#)