

Package ‘paws.networking’

October 20, 2019

Title Amazon Web Services Networking & Content Delivery APIs

Version 0.1.5

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 (>= 0.2.4)

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 2019-10-20 04:30:02 UTC

R topics documented:

apigateway	2
apigatewaymanagementapi	6
apigatewayv2	7
appmesh	9
cloudfront	11
directconnect	13
elb	15
elbv2	17
globalaccelerator	19
route53	22
route53domains	24
route53resolver	26
servicediscovery	28
Index	30

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(config = list())
```

Arguments

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

Service syntax

```

svc <- apigateway(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

Operations

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

<code>delete_resource</code>	Deletes a Resource resource
<code>delete_rest_api</code>	Deletes the specified API
<code>delete_stage</code>	Deletes a Stage resource
<code>delete_usage_plan</code>	Deletes a usage plan of a given plan Id
<code>delete_usage_plan_key</code>	Deletes a usage plan key and remove the underlying API key from the associated usage plan
<code>delete_vpc_link</code>	Deletes an existing VpcLink of a specified identifier
<code>flush_stage_authorizers_cache</code>	Flushes all authorizer cache entries on a stage
<code>flush_stage_cache</code>	Flushes a stage's cache
<code>generate_client_certificate</code>	Generates a ClientCertificate resource
<code>get_account</code>	Gets information about the current Account resource
<code>get_api_key</code>	Gets information about the current ApiKey resource
<code>get_api_keys</code>	Gets information about the current ApiKeys resource
<code>get_authorizer</code>	Describe an existing Authorizer resource
<code>get_authorizers</code>	Describe an existing Authorizers resource
<code>get_base_path_mapping</code>	Describe a BasePathMapping resource
<code>get_base_path_mappings</code>	Represents a collection of BasePathMapping resources
<code>get_client_certificate</code>	Gets information about the current ClientCertificate resource
<code>get_client_certificates</code>	Gets a collection of ClientCertificate resources
<code>get_deployment</code>	Gets information about a Deployment resource
<code>get_deployments</code>	Gets information about a Deployments collection
<code>get_documentation_part</code>	Get documentation part
<code>get_documentation_parts</code>	Get documentation parts
<code>get_documentation_version</code>	Get documentation version
<code>get_documentation_versions</code>	Get documentation versions
<code>get_domain_name</code>	Represents a domain name that is contained in a simpler, more intuitive URL that can be called
<code>get_domain_names</code>	Represents a collection of DomainName resources
<code>get_export</code>	Exports a deployed version of a RestApi in a specified format
<code>get_gateway_response</code>	Gets a GatewayResponse of a specified response type on the given RestApi
<code>get_gateway_responses</code>	Gets the GatewayResponses collection on the given RestApi
<code>get_integration</code>	Get the integration settings
<code>get_integration_response</code>	Represents a get integration response
<code>get_method</code>	Describe an existing Method resource
<code>get_method_response</code>	Describes a MethodResponse resource
<code>get_model</code>	Describes an existing model defined for a RestApi resource
<code>get_model_template</code>	Generates a sample mapping template that can be used to transform a payload into the structure
<code>get_models</code>	Describes existing Models defined for a RestApi resource
<code>get_request_validator</code>	Gets a RequestValidator of a given RestApi
<code>get_request_validators</code>	Gets the RequestValidators collection of a given RestApi
<code>get_resource</code>	Lists information about a resource
<code>get_resources</code>	Lists information about a collection of Resource resources
<code>get_rest_api</code>	Lists the RestApi resource in the collection
<code>get_rest_apis</code>	Lists the RestApis resources for your collection
<code>get_sdk</code>	Generates a client SDK for a RestApi and Stage
<code>get_sdk_type</code>	Get sdk type
<code>get_sdk_types</code>	Get sdk types
<code>get_stage</code>	Gets information about a Stage resource
<code>get_stages</code>	Gets information about one or more Stage resources
<code>get_tags</code>	Gets the Tags collection for a given resource

<code>get_usage</code>	Gets the usage data of a usage plan in a specified time interval
<code>get_usage_plan</code>	Gets a usage plan of a given plan identifier
<code>get_usage_plan_key</code>	Gets a usage plan key of a given key identifier
<code>get_usage_plan_keys</code>	Gets all the usage plan keys representing the API keys added to a specified usage plan
<code>get_usage_plans</code>	Gets all the usage plans of the caller's account
<code>get_vpc_link</code>	Gets a specified VPC link under the caller's account in a region
<code>get_vpc_links</code>	Gets the VpcLinks collection under the caller's account in a selected region
<code>import_api_keys</code>	Import API keys from an external source, such as a CSV-formatted file
<code>import_documentation_parts</code>	Import documentation parts
<code>import_rest_api</code>	A feature of the API Gateway control service for creating a new API from an external API d
<code>put_gateway_response</code>	Creates a customization of a GatewayResponse of a specified response type and status code
<code>put_integration</code>	Sets up a method's integration
<code>put_integration_response</code>	Represents a put integration
<code>put_method</code>	Add a method to an existing Resource resource
<code>put_method_response</code>	Adds a MethodResponse to an existing Method resource
<code>put_rest_api</code>	A feature of the API Gateway control service for updating an existing API with an input of c
<code>tag_resource</code>	Adds or updates a tag on a given resource
<code>test_invoke_authorizer</code>	Simulate the execution of an Authorizer in your RestApi with headers, parameters, and an in
<code>test_invoke_method</code>	Simulate the execution of a Method in your RestApi with headers, parameters, and an incom
<code>untag_resource</code>	Removes a tag from a given resource
<code>update_account</code>	Changes information about the current Account resource
<code>update_api_key</code>	Changes information about an ApiKey resource
<code>update_authorizer</code>	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(config = list())
```

Arguments

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

Service syntax

```

svc <- apigatewaymanagementapi(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

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(config = list())
```

Arguments

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

Service syntax

```

svc <- apigatewayv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    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

<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

update_integration_response	Updates an IntegrationResponses
update_model	Updates a Model
update_route	Updates a Route
update_route_response	Updates a RouteResponse
update_stage	Updates a Stage

Examples

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

apptmesh

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

```
apptmesh(config = list())
```

Arguments

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

Service syntax

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

```

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

```

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(config = list())
```

Arguments

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

Service syntax

```
svc <- cloudfront(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

[create_cloud_front_origin_access_identity](#)
[create_distribution](#)
[create_distribution_with_tags](#)
[create_field_level_encryption_config](#)
[create_field_level_encryption_profile](#)
[create_invalidation](#)
[create_public_key](#)
[create_streaming_distribution](#)

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

Examples

```

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

```

directconnect	<i>AWS Direct Connect</i>
---------------	---------------------------

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(config = list())
```

Arguments

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

Service syntax

```
svc <- directconnect(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    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_connection_with_lag](#)

Accepts a proposal request to attach a virtual private gateway or transit virtual gateway to a connection.
 Deprecated
 Creates a hosted connection on the specified interconnect or a link aggregation group.
 Provisions a private virtual interface to be owned by the specified Amazon account.
 Provisions a public virtual interface to be owned by the specified Amazon account.
 Provisions a transit virtual interface to be owned by the specified Amazon account.
 Associates an existing connection with a link aggregation group (LAG).

<code>associate_hosted_connection</code>	Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG)
<code>associate_virtual_interface</code>	Associates a virtual interface with a specified link aggregation group (LAG)
<code>confirm_connection</code>	Confirms the creation of the specified hosted connection on an interconnect
<code>confirm_private_virtual_interface</code>	Accepts ownership of a private virtual interface created by another AWS account
<code>confirm_public_virtual_interface</code>	Accepts ownership of a public virtual interface created by another AWS account
<code>confirm_transit_virtual_interface</code>	Accepts ownership of a transit virtual interface created by another AWS account
<code>create_bgp_peer</code>	Creates a BGP peer on the specified virtual interface
<code>create_connection</code>	Creates a connection between a customer network and a specific AWS Region
<code>create_direct_connect_gateway</code>	Creates a Direct Connect gateway, which is an intermediate object between a customer network and a virtual private gateway
<code>create_direct_connect_gateway_association</code>	Creates an association between a Direct Connect gateway and a virtual private gateway
<code>create_direct_connect_gateway_association_proposal</code>	Creates a proposal to associate the specified virtual private gateway with the specified Direct Connect gateway
<code>create_interconnect</code>	Creates an interconnect between an AWS Direct Connect Partner's network and an AWS Region
<code>create_lag</code>	Creates a link aggregation group (LAG) with the specified number of virtual interfaces
<code>create_private_virtual_interface</code>	Creates a private virtual interface
<code>create_public_virtual_interface</code>	Creates a public virtual interface
<code>create_transit_virtual_interface</code>	Creates a transit virtual interface
<code>delete_bgp_peer</code>	Deletes the specified BGP peer on the specified virtual interface
<code>delete_connection</code>	Deletes the specified connection
<code>delete_direct_connect_gateway</code>	Deletes the specified Direct Connect gateway
<code>delete_direct_connect_gateway_association</code>	Deletes the association between the specified Direct Connect gateway and virtual private gateway
<code>delete_direct_connect_gateway_association_proposal</code>	Deletes the association proposal request between the specified Direct Connect gateway and virtual private gateway
<code>delete_interconnect</code>	Deletes the specified interconnect
<code>delete_lag</code>	Deletes the specified link aggregation group (LAG)
<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_direct_connect_gateway_association_proposals</code>	Describes one or more association proposals for connection between a Direct Connect gateway and a virtual private gateway
<code>describe_direct_connect_gateway_associations</code>	Lists the associations between your Direct Connect gateways and virtual private gateways
<code>describe_direct_connect_gateway_attachments</code>	Lists the attachments between your Direct Connect gateways and virtual private gateways
<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 interconnect
<code>describe_interconnect_loa</code>	Deprecated
<code>describe_interconnects</code>	Lists the interconnects owned by the AWS account or only the specified interconnect
<code>describe_lags</code>	Describes all your link aggregation groups (LAG) or the specified link aggregation group (LAG)
<code>describe_loa</code>	Gets the LOA-CFA for a connection, interconnect, or link aggregation group (LAG)
<code>describe_locations</code>	Lists the AWS Direct Connect locations in the current AWS Region
<code>describe_tags</code>	Describes the tags associated with the specified AWS Direct Connect resource
<code>describe_virtual_gateways</code>	Lists the virtual private gateways owned by the AWS account
<code>describe_virtual_interfaces</code>	Displays all virtual interfaces for an AWS account
<code>disassociate_connection_from_lag</code>	Disassociates a connection from a link aggregation group (LAG)
<code>tag_resource</code>	Adds the specified tags to the specified AWS Direct Connect resource
<code>untag_resource</code>	Removes one or more tags from the specified AWS Direct Connect resource
<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_virtual_interface_attributes</code>	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

```
elb(config = list())
```

Arguments

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

Service syntax

```
svc <- elb(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      )
    )
  ),
```

```

        profile = "string"
    ),
    endpoint = "string",
    region = "string"
)
)

```

Operations

add_tags	Adds the specified tags to the specified load balancer
apply_security_groups_to_load_balancer	Associates one or more security groups with your load balancer in a virtual
attach_load_balancer_to_subnets	Adds one or more subnets to the set of configured subnets for the specified
configure_health_check	Specifies the health check settings to use when evaluating the health state of
create_app_cookie_stickiness_policy	Generates a stickiness policy with sticky session lifetimes that follow that of
create_lb_cookie_stickiness_policy	Generates a stickiness policy with sticky session lifetimes controlled by the
create_load_balancer	Creates a Classic Load Balancer
create_load_balancer_listeners	Creates one or more listeners for the specified load balancer
create_load_balancer_policy	Creates a policy with the specified attributes for the specified load balancer
delete_load_balancer	Deletes the specified load balancer
delete_load_balancer_listeners	Deletes the specified listeners from the specified load balancer
delete_load_balancer_policy	Deletes the specified policy from the specified load balancer
deregister_instances_from_load_balancer	Deregisters the specified instances from the specified load balancer
describe_account_limits	Describes the current Elastic Load Balancing resource limits for your AWS
describe_instance_health	Describes the state of the specified instances with respect to the specified lo
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 polic
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 lo
disable_availability_zones_for_load_balancer	Removes the specified Availability Zones from the set of Availability Zones
enable_availability_zones_for_load_balancer	Adds the specified Availability Zones to the set of Availability Zones for th
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
set_load_balancer_policies_of_listener	Replaces the current set of policies for the specified load balancer port with

Examples

```

# This example adds two tags to the specified load balancer.
svc <- elb()
svc$add_tags(
  LoadBalancerNames = list(
    "my-load-balancer"
  ),
)

```



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

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(config = list())
```

Arguments

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

Service syntax

```
svc <- elbv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

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

<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 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>modify_listener</code>	Modifies the specified properties of the specified listener
<code>modify_load_balancer_attributes</code>	Modifies the specified attributes of the specified Application Load Balancer or Network Load Balancer
<code>modify_rule</code>	Modifies 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>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 listener
<code>remove_tags</code>	Removes the specified tags from the specified Elastic Load Balancing resource
<code>set_ip_address_type</code>	Sets the type of IP addresses used by the subnets of the specified Application Load Balancer or Network 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
<code>set_subnets</code>	Enables the Availability Zone for the specified public subnets for the specified Application 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"
    )
  )
)
```

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(config = list())
```

Arguments

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

Service syntax

```
svc <- globalaccelerator(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

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

<code>list_accelerators</code>	List the accelerators for an AWS account
<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>update_accelerator</code>	Update an accelerator
<code>update_accelerator_attributes</code>	Update the attributes for an accelerator
<code>update_endpoint_group</code>	Update an endpoint group
<code>update_listener</code>	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(config = list())
```

Arguments

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

Service syntax

```
svc <- route53(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

<code>associate_vpc_with_hosted_zone</code>	Associates an Amazon VPC with a private hosted zone
<code>change_resource_record_sets</code>	Creates, changes, or deletes a resource record set, which contains authoritative DNS information for a hosted zone
<code>change_tags_for_resource</code>	Adds, edits, or deletes tags for a health check or a hosted zone
<code>create_health_check</code>	Creates a new health check
<code>create_hosted_zone</code>	Creates a new public or private hosted zone
<code>create_query_logging_config</code>	Creates a configuration for DNS query logging
<code>create_reusable_delegation_set</code>	Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones
<code>create_traffic_policy</code>	Creates a traffic policy, which you use to create multiple DNS resource record sets in a specified hosted zone based on the settings in the traffic policy
<code>create_traffic_policy_instance</code>	Creates resource record sets in a specified hosted zone based on the settings in a 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 AWS account that created a specified VPC to submit an AssociateVPCWithHostedZone request to the current AWS account
<code>delete_health_check</code>	Deletes a health check
<code>delete_hosted_zone</code>	Deletes a hosted zone
<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 creates for the instance
<code>delete_vpc_association_authorization</code>	Removes authorization to submit an AssociateVPCWithHostedZone request to the current AWS account
<code>disassociate_vpc_from_hosted_zone</code>	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 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>	GetCheckerIpRanges still works, but we recommend that you download ip-ranges from https://ip-ranges.amazonaws.com/ip-ranges.json
<code>get_geo_location</code>	Gets information about whether a specified geographic location is supported for a 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 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 that are associated with the zone
<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 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 four name servers that are associated with the set
<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

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 ass
test_dns_answer	Gets the value that Amazon Route 53 returns in response to a DNS request for
update_health_check	Updates an existing health check
update_hosted_zone_comment	Updates the comment for a specified hosted zone
update_traffic_policy_comment	Updates the comment for a specified traffic policy version
update_traffic_policy_instance	Updates the resource record sets in a specified hosted zone that were created b

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(config = list())
```

Arguments

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

Service syntax

```

svc <- route53domains(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

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
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(config = list())
```

Arguments

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

Service syntax

```

svc <- route53resolver(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

Operations

associate_resolver_endpoint_ip_address	Adds IP addresses to an inbound or an outbound resolver endpoint
associate_resolver_rule	Associates a resolver rule with a VPC
create_resolver_endpoint	Creates a resolver endpoint
create_resolver_rule	For DNS queries that originate in your VPCs, specifies which resolver endpoint
delete_resolver_endpoint	Deletes a resolver endpoint
delete_resolver_rule	Deletes a resolver rule
disassociate_resolver_endpoint_ip_address	Removes IP addresses from an inbound or an outbound resolver endpoint
disassociate_resolver_rule	Removes the association between a specified resolver rule and a specified VPC
get_resolver_endpoint	Gets information about a specified resolver endpoint, such as whether it's an inbound or outbound endpoint
get_resolver_rule	Gets information about a specified resolver rule, such as the domain name that the rule applies to
get_resolver_rule_association	Gets information about an association between a specified resolver rule and a VPC
get_resolver_rule_policy	Gets information about a resolver rule policy
list_resolver_endpoint_ip_addresses	Gets the IP addresses for a specified resolver endpoint
list_resolver_endpoints	Lists all the resolver endpoints that were created using the current AWS account
list_resolver_rule_associations	Lists the associations that were created between resolver rules and VPCs using the current AWS account
list_resolver_rules	Lists the resolver rules that were created using the current AWS account
list_tags_for_resource	Lists the tags that you associated with the specified resource
put_resolver_rule_policy	Specifies the Resolver operations and resources that you want to allow another AWS account to perform
tag_resource	Adds one or more tags to a specified resource
untag_resource	Removes one or more tags from a specified resource
update_resolver_endpoint	Updates the name of an inbound or an outbound resolver endpoint
update_resolver_rule	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(config = list())
```

Arguments

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

Service syntax

```
svc <- servicediscovery(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

[create_http_namespace](#)
[create_private_dns_namespace](#)
[create_public_dns_namespace](#)
[create_service](#)
[delete_namespace](#)

Creates an HTTP namespace
 Creates a private namespace based on DNS, which will be visible only inside a speci
 Creates a public namespace based on DNS, which will be visible on the internet
 Creates a service, which defines the configuration for the following entities: - For pu
 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 Cloud
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 insta
get_namespace	Gets information about a namespace
get_operation	Gets information about any operation that returns an operation ID in the response, su
get_service	Gets the settings for a specified service
list_instances	Lists summary information about the instances that you registered by using a specific
list_namespaces	Lists summary information about the namespaces that were created by the current A
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 s
register_instance	Creates or updates one or more records and, optionally, creates a health check based
update_instance_custom_health_status	Submits a request to change the health status of a custom health check to healthy or r
update_service	Submits a request to perform the following operations: - Add or delete DnsRecords o

Examples

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

Index

accept_direct_connect_gateway_association_proposal, [13](#)
add_listener_certificates, [18](#)
add_tags, [16](#), [18](#)
allocate_connection_on_interconnect, [13](#)
allocate_hosted_connection, [13](#)
allocate_private_virtual_interface, [13](#)
allocate_public_virtual_interface, [13](#)
allocate_transit_virtual_interface, [13](#)
apigateway, [2](#)
apigatewaymanagementapi, [6](#)
apigatewayv2, [7](#)
apply_security_groups_to_load_balancer, [16](#)
apmesh, [9](#)
associate_connection_with_lag, [13](#)
associate_hosted_connection, [14](#)
associate_resolver_endpoint_ip_address, [27](#)
associate_resolver_rule, [27](#)
associate_virtual_interface, [14](#)
associate_vpc_with_hosted_zone, [23](#)
attach_load_balancer_to_subnets, [16](#)

change_resource_record_sets, [23](#)
change_tags_for_resource, [23](#)
check_domain_availability, [25](#)
check_domain_transferability, [25](#)
cloudfront, [11](#)
configure_health_check, [16](#)
confirm_connection, [14](#)
confirm_private_virtual_interface, [14](#)
confirm_public_virtual_interface, [14](#)
confirm_transit_virtual_interface, [14](#)
create_accelerator, [21](#)
create_api, [7](#)
create_api_key, [3](#)
create_api_mapping, [7](#)
create_app_cookie_stickiness_policy, [16](#)
create_authorizer, [3](#), [7](#)
create_base_path_mapping, [3](#)
create_bgp_peer, [14](#)
create_cloud_front_origin_access_identity, [11](#)
create_connection, [14](#)
create_deployment, [3](#), [7](#)
create_direct_connect_gateway, [14](#)
create_direct_connect_gateway_association, [14](#)
create_direct_connect_gateway_association_proposal, [14](#)
create_distribution, [11](#)
create_distribution_with_tags, [11](#)
create_documentation_part, [3](#)
create_documentation_version, [3](#)
create_domain_name, [3](#), [7](#)
create_endpoint_group, [21](#)
create_field_level_encryption_config, [11](#)
create_field_level_encryption_profile, [11](#)
create_health_check, [23](#)
create_hosted_zone, [23](#)
create_http_namespace, [28](#)
create_integration, [7](#)
create_integration_response, [8](#)
create_interconnect, [14](#)
create_invalidation, [11](#)
create_lag, [14](#)
create_lb_cookie_stickiness_policy, [16](#)
create_listener, [18](#), [21](#)
create_load_balancer, [16](#), [18](#)
create_load_balancer_listeners, [16](#)
create_load_balancer_policy, [16](#)
create_mesh, [10](#)
create_model, [3](#), [8](#)

- create_private_dns_namespace, 28
- create_private_virtual_interface, 14
- create_public_dns_namespace, 28
- create_public_key, 11
- create_public_virtual_interface, 14
- create_query_logging_config, 23
- create_request_validator, 3
- create_resolver_endpoint, 27
- create_resolver_rule, 27
- create_resource, 3
- create_rest_api, 3
- create_reusable_delegation_set, 23
- create_route, 8, 10
- create_route_response, 8
- create_rule, 18
- create_service, 28
- create_stage, 3, 8
- create_streaming_distribution, 11
- create_streaming_distribution_with_tags, 12
- create_target_group, 18
- create_traffic_policy, 23
- create_traffic_policy_instance, 23
- create_traffic_policy_version, 23
- create_transit_virtual_interface, 14
- create_usage_plan, 3
- create_usage_plan_key, 3
- create_virtual_node, 10
- create_virtual_router, 10
- create_virtual_service, 10
- create_vpc_association_authorization, 23
- create_vpc_link, 3

- delete_accelerator, 21
- delete_api, 8
- delete_api_key, 3
- delete_api_mapping, 8
- delete_authorizer, 3, 8
- delete_base_path_mapping, 3
- delete_bgp_peer, 14
- delete_client_certificate, 3
- delete_cloud_front_origin_access_identity, 12
- delete_connection, 14
- delete_deployment, 3, 8
- delete_direct_connect_gateway, 14
- delete_direct_connect_gateway_association, 14

- delete_direct_connect_gateway_association_proposal, 14
- delete_distribution, 12
- delete_documentation_part, 3
- delete_documentation_version, 3
- delete_domain_name, 3, 8
- delete_endpoint_group, 21
- delete_field_level_encryption_config, 12
- delete_field_level_encryption_profile, 12
- delete_gateway_response, 3
- delete_health_check, 23
- delete_hosted_zone, 23
- delete_integration, 3, 8
- delete_integration_response, 3, 8
- delete_interconnect, 14
- delete_lag, 14
- delete_listener, 18, 21
- delete_load_balancer, 16, 18
- delete_load_balancer_listeners, 16
- delete_load_balancer_policy, 16
- delete_mesh, 10
- delete_method, 3
- delete_method_response, 3
- delete_model, 3, 8
- delete_namespace, 28
- delete_public_key, 12
- delete_query_logging_config, 23
- delete_request_validator, 3
- delete_resolver_endpoint, 27
- delete_resolver_rule, 27
- delete_resource, 4
- delete_rest_api, 4
- delete_reusable_delegation_set, 23
- delete_route, 8, 10
- delete_route_response, 8
- delete_rule, 18
- delete_service, 29
- delete_stage, 4, 8
- delete_streaming_distribution, 12
- delete_tags_for_domain, 25
- delete_target_group, 18
- delete_traffic_policy, 23
- delete_traffic_policy_instance, 23
- delete_usage_plan, 4
- delete_usage_plan_key, 4
- delete_virtual_interface, 14

- delete_virtual_node, [10](#)
- delete_virtual_router, [10](#)
- delete_virtual_service, [10](#)
- delete_vpc_association_authorization, [23](#)
- delete_vpc_link, [4](#)
- deregister_instance, [29](#)
- deregister_instances_from_load_balancer, [16](#)
- deregister_targets, [18](#)
- describe_accelerator, [21](#)
- describe_accelerator_attributes, [21](#)
- describe_account_limits, [16, 18](#)
- describe_connection_loa, [14](#)
- describe_connections, [14](#)
- describe_connections_on_interconnect, [14](#)
- describe_direct_connect_gateway_association_proposals, [14](#)
- describe_direct_connect_gateway_associations, [14](#)
- describe_direct_connect_gateway_attachments, [14](#)
- describe_direct_connect_gateways, [14](#)
- describe_endpoint_group, [21](#)
- describe_hosted_connections, [14](#)
- describe_instance_health, [16](#)
- describe_interconnect_loa, [14](#)
- describe_interconnects, [14](#)
- describe_lags, [14](#)
- describe_listener, [21](#)
- describe_listener_certificates, [18](#)
- describe_listeners, [18](#)
- describe_loa, [14](#)
- describe_load_balancer_attributes, [16, 18](#)
- describe_load_balancer_policies, [16](#)
- describe_load_balancer_policy_types, [16](#)
- describe_load_balancers, [16, 18](#)
- describe_locations, [14](#)
- describe_mesh, [10](#)
- describe_route, [10](#)
- describe_rules, [19](#)
- describe_ssl_policies, [19](#)
- describe_tags, [14, 16, 19](#)
- describe_target_group_attributes, [19](#)
- describe_target_groups, [19](#)
- describe_target_health, [19](#)
- describe_virtual_gateways, [14](#)
- describe_virtual_interfaces, [14](#)
- describe_virtual_node, [10](#)
- describe_virtual_router, [10](#)
- describe_virtual_service, [10](#)
- detach_load_balancer_from_subnets, [16](#)
- directconnect, [13](#)
- disable_availability_zones_for_load_balancer, [16](#)
- disable_domain_auto_renew, [25](#)
- disable_domain_transfer_lock, [25](#)
- disassociate_connection_from_lag, [14](#)
- disassociate_resolver_endpoint_ip_address, [27](#)
- disassociate_resolver_rule, [27](#)
- disassociate_vpc_from_hosted_zone, [23](#)
- dispose_instances, [29](#)
- elb, [15](#)
- elbv2, [17](#)
- enable_availability_zones_for_load_balancer, [16](#)
- enable_domain_auto_renew, [25](#)
- enable_domain_transfer_lock, [25](#)
- flush_stage_authorizers_cache, [4](#)
- flush_stage_cache, [4](#)
- generate_client_certificate, [4](#)
- get_account, [4](#)
- get_account_limit, [23](#)
- get_api, [8](#)
- get_api_key, [4](#)
- get_api_keys, [4](#)
- get_api_mapping, [8](#)
- get_api_mappings, [8](#)
- get_apis, [8](#)
- get_authorizer, [4, 8](#)
- get_authorizers, [4, 8](#)
- get_base_path_mapping, [4](#)
- get_base_path_mappings, [4](#)
- get_change, [23](#)
- get_checker_ip_ranges, [23](#)
- get_client_certificate, [4](#)
- get_client_certificates, [4](#)
- get_cloud_front_origin_access_identity, [12](#)

get_cloud_front_origin_access_identity_config, 12
12
get_contact_reachability_status, 25
get_deployment, 4, 8
get_deployments, 4, 8
get_distribution, 12
get_distribution_config, 12
get_documentation_part, 4
get_documentation_parts, 4
get_documentation_version, 4
get_documentation_versions, 4
get_domain_detail, 25
get_domain_name, 4, 8
get_domain_names, 4, 8
get_domain_suggestions, 25
get_export, 4
get_field_level_encryption, 12
get_field_level_encryption_config, 12
get_field_level_encryption_profile, 12
get_field_level_encryption_profile_config, 12
12
get_gateway_response, 4
get_gateway_responses, 4
get_geo_location, 23
get_health_check, 23
get_health_check_count, 23
get_health_check_last_failure_reason, 23
get_health_check_status, 23
get_hosted_zone, 23
get_hosted_zone_count, 23
get_hosted_zone_limit, 23
get_instance, 29
get_instances_health_status, 29
get_integration, 4, 8
get_integration_response, 4, 8
get_integration_responses, 8
get_integrations, 8
get_invalidation, 12
get_method, 4
get_method_response, 4
get_model, 4, 8
get_model_template, 4, 8
get_models, 4, 8
get_namespace, 29
get_operation, 29
get_operation_detail, 25
get_public_key, 12
get_public_key_config, 12
get_query_logging_config, 23
get_request_validator, 4
get_request_validators, 4
get_resolver_endpoint, 27
get_resolver_rule, 27
get_resolver_rule_association, 27
get_resolver_rule_policy, 27
get_resource, 4
get_resources, 4
get_rest_api, 4
get_rest_apis, 4
get_reusable_delegation_set, 23
get_reusable_delegation_set_limit, 23
get_route, 8
get_route_response, 8
get_route_responses, 8
get_routes, 8
get_sdk, 4
get_sdk_type, 4
get_sdk_types, 4
get_service, 29
get_stage, 4, 8
get_stages, 4, 8
get_streaming_distribution, 12
get_streaming_distribution_config, 12
get_tags, 4, 8
get_traffic_policy, 23
get_traffic_policy_instance, 23
get_traffic_policy_instance_count, 23
get_usage, 5
get_usage_plan, 5
get_usage_plan_key, 5
get_usage_plan_keys, 5
get_usage_plans, 5
get_vpc_link, 5
get_vpc_links, 5
globalaccelerator, 19

import_api_keys, 5
import_documentation_parts, 5
import_rest_api, 5

list_accelerators, 22
list_cloud_front_origin_access_identities, 12
list_distributions, 12
list_distributions_by_web_acl_id, 12
list_domains, 25

- list_endpoint_groups, 22
- list_field_level_encryption_configs, 12
- list_field_level_encryption_profiles, 12
- list_geo_locations, 23
- list_health_checks, 23
- list_hosted_zones, 23
- list_hosted_zones_by_name, 23
- list_instances, 29
- list_invalidations, 12
- list_listeners, 22
- list_meshes, 10
- list_namespaces, 29
- list_operations, 25, 29
- list_public_keys, 12
- list_query_logging_configs, 23
- list_resolver_endpoint_ip_addresses, 27
- list_resolver_endpoints, 27
- list_resolver_rule_associations, 27
- list_resolver_rules, 27
- list_resource_record_sets, 23
- list_reusable_delegation_sets, 23
- list_routes, 10
- list_services, 29
- list_streaming_distributions, 12
- list_tags_for_domain, 25
- list_tags_for_resource, 10, 12, 23, 27
- list_tags_for_resources, 23
- list_traffic_policies, 23
- list_traffic_policy_instances, 24
- list_traffic_policy_instances_by_hosted_zone, 24
- list_traffic_policy_instances_by_policy, 24
- list_traffic_policy_versions, 24
- list_virtual_nodes, 10
- list_virtual_routers, 10
- list_virtual_services, 10
- list_vpc_association_authorizations, 24

- modify_listener, 19
- modify_load_balancer_attributes, 16, 19
- modify_rule, 19
- modify_target_group, 19
- modify_target_group_attributes, 19

- post_to_connection, 6
- put_gateway_response, 5
- put_integration, 5
- put_integration_response, 5
- put_method, 5
- put_method_response, 5
- put_resolver_rule_policy, 27
- put_rest_api, 5

- register_domain, 25
- register_instance, 29
- register_instances_with_load_balancer, 16
- register_targets, 19
- remove_listener_certificates, 19
- remove_tags, 16, 19
- renew_domain, 25
- resend_contact_reachability_email, 25
- retrieve_domain_auth_code, 25
- route53, 22
- route53domains, 24
- route53resolver, 26

- servicediscovery, 28
- set_ip_address_type, 19
- set_load_balancer_listener_ssl_certificate, 16
- set_load_balancer_policies_for_backend_server, 16
- set_load_balancer_policies_of_listener, 16
- set_rule_priorities, 19
- set_security_groups, 19
- set_subnets, 19

- tag_resource, 5, 8, 10, 12, 14, 27
- test_dns_answer, 24
- test_invoke_authorizer, 5
- test_invoke_method, 5
- transfer_domain, 25

- untag_resource, 5, 8, 10, 12, 14, 27
- update_accelerator, 22
- update_accelerator_attributes, 22
- update_account, 5
- update_api, 8
- update_api_key, 5
- update_api_mapping, 8
- update_authorizer, 5, 8

update_base_path_mapping, [5](#)
update_client_certificate, [5](#)
update_cloud_front_origin_access_identity,
[12](#)
update_deployment, [5, 8](#)
update_direct_connect_gateway_association,
[14](#)
update_distribution, [12](#)
update_documentation_part, [5](#)
update_documentation_version, [5](#)
update_domain_contact, [25](#)
update_domain_contact_privacy, [25](#)
update_domain_name, [5, 8](#)
update_domain_nameservers, [25](#)
update_endpoint_group, [22](#)
update_field_level_encryption_config,
[12](#)
update_field_level_encryption_profile,
[12](#)
update_gateway_response, [5](#)
update_health_check, [24](#)
update_hosted_zone_comment, [24](#)
update_instance_custom_health_status,
[29](#)
update_integration, [5, 8](#)
update_integration_response, [5, 9](#)
update_lag, [14](#)
update_listener, [22](#)
update_mesh, [10](#)
update_method, [5](#)
update_method_response, [5](#)
update_model, [5, 9](#)
update_public_key, [12](#)
update_request_validator, [5](#)
update_resolver_endpoint, [27](#)
update_resolver_rule, [27](#)
update_resource, [5](#)
update_rest_api, [5](#)
update_route, [9, 10](#)
update_route_response, [9](#)
update_service, [29](#)
update_stage, [5, 9](#)
update_streaming_distribution, [12](#)
update_tags_for_domain, [25](#)
update_traffic_policy_comment, [24](#)
update_traffic_policy_instance, [24](#)
update_usage, [5](#)
update_usage_plan, [5](#)
update_virtual_interface_attributes,
[14](#)
update_virtual_node, [10](#)
update_virtual_router, [10](#)
update_virtual_service, [10](#)
update_vpc_link, [5](#)
view_billing, [25](#)