Package ‘paws’

March 31, 2020

Title Amazon Web Services Software Development Kit

Version 0.1.7

Description Interface to Amazon Web Services <https://aws.amazon.com>, including storage, database, and compute services, such as 'Simple Storage Service' ('S3'), 'DynamoDB' 'NoSQL' database, and 'Lambda' functions-as-a-service.

License Apache License (>= 2.0)

Imports paws.compute (>= 0.1.7), paws.storage (>= 0.1.7), paws.database (>= 0.1.7), paws.networking (>= 0.1.7), paws.management (>= 0.1.7), paws.machine.learning (>= 0.1.7), paws.analytics (>= 0.1.7), paws.security.identity (>= 0.1.7), paws.application.integration (>= 0.1.7), paws.cost.management (>= 0.1.7), paws.customer.engagement (>= 0.1.7)

Suggests testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

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 2020-03-31 15:30:06 UTC

R topics documented:

- acm ........................................... 4
- acmpca ....................................... 6
- apigateway .................................. 7
- apigatewaymanagementapi ............... 11
topics documented:

- apigatewayv2 .................................................. 12
- applicationautoscaling ...................................... 14
- applicationinsights .......................................... 16
- appmesh ....................................................... 18
- athena .......................................................... 20
- autoscaling .................................................... 21
- autoscalingplans .............................................. 23
- backup .......................................................... 25
- batch ............................................................ 27
- budgets ........................................................ 29
- clouddirectory .................................................. 31
- cloudformation ................................................ 34
- cloudfront ..................................................... 36
- cloudhsm ....................................................... 38
- cloudhsmv2 ..................................................... 39
- cloudsearch .................................................... 41
- cloudsearchdomain .......................................... 42
- cloudtrail ...................................................... 44
- cloudwatch ..................................................... 45
- cloudwatchevents ............................................ 47
- cloudwatchlogs ............................................... 49
- cognitoidentity ................................................ 51
- cognitoidentityprovider ..................................... 53
- cognitosync .................................................... 56
- comprehend .................................................... 58
- comprehendmedical .......................................... 60
- configservice .................................................. 61
- connect ........................................................ 64
- costandusagereportservice .................................. 65
- costexplorer .................................................... 67
- datapipeline .................................................... 68
- dax ............................................................... 70
- directconnect .................................................. 71
- directoryservice .............................................. 74
- dlm .............................................................. 76
- docdb ........................................................... 77
- dynamodb ........................................................ 79
- dynamodbstreams ............................................. 82
- ec2 .............................................................. 83
- ec2instanceconnect .......................................... 92
- ecr .............................................................. 93
- ecs ............................................................. 95
- efs .............................................................. 97
- eks .............................................................. 99
- elasticache ..................................................... 101
- elasticbeanstalk ............................................. 103
- elasticsearchservice ......................................... 105
- elb .............................................................. 107
### Topics Documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>elbv2</td>
<td>109</td>
</tr>
<tr>
<td>emr</td>
<td>111</td>
</tr>
<tr>
<td>eventbridge</td>
<td>113</td>
</tr>
<tr>
<td>firehose</td>
<td>115</td>
</tr>
<tr>
<td>fms</td>
<td>116</td>
</tr>
<tr>
<td>fsx</td>
<td>117</td>
</tr>
<tr>
<td>glacier</td>
<td>119</td>
</tr>
<tr>
<td>globalaccelerator</td>
<td>121</td>
</tr>
<tr>
<td>glue</td>
<td>123</td>
</tr>
<tr>
<td>guardduty</td>
<td>127</td>
</tr>
<tr>
<td>health</td>
<td>129</td>
</tr>
<tr>
<td>iam</td>
<td>131</td>
</tr>
<tr>
<td>inspector</td>
<td>136</td>
</tr>
<tr>
<td>kafka</td>
<td>138</td>
</tr>
<tr>
<td>kinesis</td>
<td>140</td>
</tr>
<tr>
<td>kinesisanalytics</td>
<td>141</td>
</tr>
<tr>
<td>kinesisanalyticsv2</td>
<td>143</td>
</tr>
<tr>
<td>kms</td>
<td>144</td>
</tr>
<tr>
<td>lambda</td>
<td>147</td>
</tr>
<tr>
<td>lexmodelbuildingsservice</td>
<td>149</td>
</tr>
<tr>
<td>lexruntimeservice</td>
<td>151</td>
</tr>
<tr>
<td>licensemanager</td>
<td>152</td>
</tr>
<tr>
<td>lightsail</td>
<td>154</td>
</tr>
<tr>
<td>machinelearning</td>
<td>157</td>
</tr>
<tr>
<td>macie</td>
<td>159</td>
</tr>
<tr>
<td>marketplacecommerceanalytics</td>
<td>160</td>
</tr>
<tr>
<td>marketplaceentitlementservice</td>
<td>161</td>
</tr>
<tr>
<td>marketplacemetering</td>
<td>162</td>
</tr>
<tr>
<td>mq</td>
<td>163</td>
</tr>
<tr>
<td>mturk</td>
<td>165</td>
</tr>
<tr>
<td>neptune</td>
<td>167</td>
</tr>
<tr>
<td>opsworks</td>
<td>169</td>
</tr>
<tr>
<td>opsworkscm</td>
<td>173</td>
</tr>
<tr>
<td>organizations</td>
<td>175</td>
</tr>
<tr>
<td>personalize</td>
<td>177</td>
</tr>
<tr>
<td>personalizeevents</td>
<td>179</td>
</tr>
<tr>
<td>personalizeruntime</td>
<td>180</td>
</tr>
<tr>
<td>pi</td>
<td>181</td>
</tr>
<tr>
<td>pinpoint</td>
<td>183</td>
</tr>
<tr>
<td>pinpointemail</td>
<td>186</td>
</tr>
<tr>
<td>pinpointsmsvoice</td>
<td>188</td>
</tr>
<tr>
<td>polly</td>
<td>190</td>
</tr>
<tr>
<td>pricing</td>
<td>191</td>
</tr>
<tr>
<td>quicksight</td>
<td>192</td>
</tr>
<tr>
<td>rds</td>
<td>195</td>
</tr>
<tr>
<td>rdsdataservice</td>
<td>196</td>
</tr>
<tr>
<td>redshift</td>
<td>201</td>
</tr>
<tr>
<td></td>
<td>202</td>
</tr>
</tbody>
</table>
acm

AWS Certificate Manager

Description

Welcome to the AWS Certificate Manager (ACM) API documentation.

You can use ACM to manage SSL/TLS certificates for your AWS-based websites and applications. For general information about using ACM, see the AWS Certificate Manager User Guide.

Usage

acm(config = list())
Arguments

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

Service syntax

```r
svc <- acm(
  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_to_certificate` Adds one or more tags to an ACM certificate
- `delete_certificate` Deletes a certificate and its associated private key
- `describe_certificate` Returns detailed metadata about the specified ACM certificate
- `export_certificate` Exports a private certificate issued by a private certificate authority (CA) for use anywhere
- `get_certificate` Retrieves a certificate specified by an ARN and its certificate chain
- `import_certificate` Imports a certificate into AWS Certificate Manager (ACM) to use with services that are integrated
- `list_certificates` Retrieves a list of certificate ARNs and domain names
- `list_tags_for_certificate` Lists the tags that have been applied to the ACM certificate
- `remove_tags_from_certificate` Remove one or more tags from an ACM certificate
- `renew_certificate` Renews an eligible ACM certificate
- `request_certificate` Requests an ACM certificate for use with other AWS services
- `resend_validation_email` Resends the email that requests domain ownership validation
- `update_certificate_options` Updates a certificate

Examples

```r
## Not run: svc <- acm()
svc$add_tags_to_certificate(
  Foo = 123
)
## End(Not run)
```
Description

This is the ACM Private CA API Reference. It provides descriptions, syntax, and usage examples for each of the actions and data types involved in creating and managing private certificate authorities (CA) for your organization.

The documentation for each action shows the Query API request parameters and the XML response. Alternatively, you can use one of the AWS SDKs to access an API that’s tailored to the programming language or platform that you’re using. For more information, see AWS SDKs.

Each ACM Private CA API action has a throttling limit which determines the number of times the action can be called per second. For more information, see API Rate Limits in ACM Private CA in the ACM Private CA user guide.

Usage

acmpca(config = list())

Arguments

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

Service syntax

svc <- acmpca(
  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_certificate_authority Creates a root or subordinate private certificate authority (CA)
create_certificate_authority_audit_report Creates an audit report that lists every time that your CA private key is used
create_permission Assigns permissions from a private CA to a designated AWS service
delete_certificate_authority Deletes a private certificate authority (CA)
delete_permission, describe_certificate_authority, describe_certificate_authority_audit_report, get_certificate, get_certificate_authority_certificate, get_certificate_authority_csr, import_certificate_authority_certificate, issue_certificate, list_certificateAuthorities, list_permissions, list_tags, restore_certificateAuthority, revoke_certificate, tag_certificateAuthority, untag_certificateAuthority, update_certificateAuthority

Revises permissions that a private CA assigned to a designated AWS service
Lists information about your private certificate authority (CA)
Lists information about a specific audit report created by calling the CreateCertificateAuthorityAuditReport action
Retrieves a certificate from your private CA
Retrieves the certificate and certificate chain for your private certificate authority
Retrieves the certificate signing request (CSR) for your private certificate authority
Imports a signed private CA certificate into ACM Private CA
Uses your private certificate authority (CA) to issue a client certificate
Lists the private certificate authorities that you created by using the CreateCertificateAuthority action
Lists all the permissions, if any, that have been assigned by a private CA
Lists the tags, if any, that are associated with your private CA
Restores a certificate authority (CA) that is in the DELETED state
Revises a certificate that was issued inside ACM Private CA
Adds one or more tags to your private CA
Remove one or more tags from your private CA
Updates the status or configuration of a private certificate authority (CA)

Examples

```r
## Not run: svc <- acmpca()
svc$create_certificateAuthority(
  Foo = 123
)
## End(Not run)
```

---

**apigateway**

**Amazon API Gateway**

**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**

```r
apigateway(config = list())
```

**Arguments**

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

```r
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, to an existing usage plan
- `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
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
generate_client_certificate  Generates a ClientCertificate resource
get_base_path_mapping   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_models              Describes existing Models defined for a RestApi resource
get_model_template      Generates a sample mapping template that can be used to transform a payload into the structure
get_request_validator   Gets a RequestValidator of a given RestApi
get_request_validators  Gets the RequestValidators collection of a given RestApi
get_resource            Lists information about a resource
get_resources           Lists information about a collection of Resource resources
get_rest_api            Lists the RestApi resource in the collection
get_rest_apis           Lists the RestApis resources for your collection
get_sdk                 Generates a client SDK for a RestApi and Stage
get_sdk_type            Get sdk type
get_sdk_types           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
get_usage_plan
get_usage_plan_key
get_usage_plan_keys
get_usage_plans
get_vpc_link
get_vpc_links
import_api_keys
import_documentation_parts
import_rest_api
put_gateway_response
put_integration
put_integration_response
put_method
put_method_response
put_rest_api
tag_resource
test_invoke_authorizer
test_invoke_method
untag_resource
update_account
update_api_key
update_authorizer
update_base_path_mapping
update_client_certificate
update_deployment
update_documentation_part
update_documentation_version
update_domain_name
update_gateway_response
update_integration
update_integration_response
update_method
update_method_response
update_model
update_request_validator
update_resource
update_rest_api
update_stage
update_usage
update_usage_plan
update_vpc_link

- Gets the usage data of a usage plan in a specified time interval
- Gets a usage plan of a given plan identifier
- Gets a usage plan key of a given key identifier
- Gets all the usage plan keys representing the API keys added to a specified usage plan
- Gets all the usage plans of the caller's account
- Gets a specified VPC link under the caller's account in a region
- Gets the VpcLinks collection under the caller's account in a selected region
- Import API keys from an external source, such as a CSV-formatted file
- Import documentation parts
- A feature of the API Gateway control service for creating a new API from an external API definition file
- Creates a customization of a GatewayResponse of a specified response type and status code
- Sets up a method's integration
- Represents a put integration
- Add a method to an existing Resource resource
- Adds a MethodResponse to an existing Method resource
- A feature of the API Gateway control service for updating an existing API with an input of external API definitions
- Updates an existing Authorizer resource
- Changes information about the BasePathMapping resource
- Updates an existing ClientCertificate resource
- Changes information about a Deployment resource
- Update documentation part
- Update documentation version
- Changes information about the DomainName resource
- Updates a GatewayResponse of a specified response type on the given RestApi
- Represents an update integration
- Updates an existing Method resource
- Updates an existing MethodResponse resource
- Changes information about a model
- Updates a RequestValidator of a given RestApi
- Changes information about a Resource resource
- Changes information about the specified API
- Changes information about a Stage resource
- Grants a temporary extension to the remaining quota of a usage plan associated with a specified API key
- Updates a usage plan of a given plan Id
- Updates an existing VpcLink of a specified identifier

### Examples

```r
## Not run: svc <- apigateway()
svc$create_api_key()
```
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

- delete_connection Delete the connection with the provided id
- get_connection Get information about the connection with the provided id
- post_to_connection Sends the provided data to the specified connection
Examples

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

**Description**

Amazon API Gateway V2

**Usage**

```r
apigatewayv2(config = list())
```

**Arguments**

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

**Service syntax**

```r
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
- `create_integration_response` Creates an IntegrationResponses
- `create_model` Creates a Model for an API
- `create_route` Creates a Route for an API
- `create_route_response` Creates a RouteResponse for a Route
- `create_stage` Creates a Stage for an API
- `delete_api` Deletes an Api resource
- `delete_api_mapping` Deletes an API mapping
- `delete_authorizer` Deletes an Authorizer
- `delete_cors_configuration` Deletes a CORS configuration
- `delete_deployment` Deletes a Deployment
- `delete_domain_name` Deletes a domain name
- `delete_integration` Deletes an Integration
- `delete_integration_response` Deletes an IntegrationResponses
- `delete_model` Deletes a Model
- `delete_route` Deletes a Route
- `delete_route_response` Deletes a RouteResponse
- `delete_route_response` Deletes a RouteResponse for a Route
- `delete_route_response` Deletes a RouteResponse
- `delete_route_response` Deletes a RouteResponse
- `delete_route_settings` Deletes the RouteSettings for a stage
- `delete_stage` Deletes a Stage
- `get_api` Gets an Api resource
- `get_api_mapping` Gets an API mapping
- `get_api_mappings` Gets API mappings
- `get_apis` Gets a collection of Api resources
- `get_authorizer` Gets an Authorizer
- `get_authorizers` Gets the Authorizers for an API
- `get_deployment` Gets a Deployment
- `get_deployments` Gets the Deployments for an API
- `get_domain_name` Gets a domain name
- `get_domain_names` Gets the domain names for an AWS account
- `get_integration` Gets an Integration
- `get_integration_response` Gets an IntegrationResponses
- `get_integration_responses` Gets the IntegrationResponses for an Integration
- `get_integrations` Gets the Integrations for an API
- `get_model` Gets a Model
- `get_models` Gets the Models for an API
- `get_model_template` Gets a model template
- `get_route` Gets a Route
- `get_route_response` Gets a RouteResponse
- `get_route_responses` Gets the RouteResponses for a Route
- `get_routes` Gets the Routes for an API
- `get_stage` Gets a Stage
get_stages  Gets the Stages for an API
get_tags    Gets a collection of Tag resources
import_api  Imports an API
reimport_api Puts an Api resource
tag_resource Creates a new Tag resource to represent a tag
untag_resource Deletes a Tag
update_api   Updates an Api resource
update_api_mapping The API mapping
update_authorizer Updates an Authorizer
update_deployment Updates a Deployment
update_domain_name Updates a domain name
update_integration 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

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

---

**Application Auto Scaling**

**Description**

With Application Auto Scaling, you can configure automatic scaling for the following resources:

- Amazon ECS services
- Amazon EC2 Spot Fleet requests
- Amazon EMR clusters
- Amazon AppStream 2.0 fleets
- Amazon DynamoDB tables and global secondary indexes throughput capacity
- Amazon Aurora Replicas
- Amazon SageMaker endpoint variants
- Custom resources provided by your own applications or services
• Amazon Comprehend document classification endpoints
• AWS Lambda function provisioned concurrency

API Summary
The Application Auto Scaling service API includes three key sets of actions:
• Register and manage scalable targets - Register AWS or custom resources as scalable targets (a resource that Application Auto Scaling can scale), set minimum and maximum capacity limits, and retrieve information on existing scalable targets.
• Configure and manage automatic scaling - Define scaling policies to dynamically scale your resources in response to CloudWatch alarms, schedule one-time or recurring scaling actions, and retrieve your recent scaling activity history.
• Suspend and resume scaling - Temporarily suspend and later resume automatic scaling by calling the RegisterScalableTarget action for any Application Auto Scaling scalable target. You can suspend and resume, individually or in combination, scale-out activities triggered by a scaling policy, scale-in activities triggered by a scaling policy, and scheduled scaling.

To learn more about Application Auto Scaling, including information about granting IAM users required permissions for Application Auto Scaling actions, see the Application Auto Scaling User Guide.

Usage
applicationautoscaling(config = list())

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

Service syntax
svc <- applicationautoscaling(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
## Operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete_scaling_policy</td>
<td>Deletes the specified scaling policy for an Application Auto Scaling scalable target</td>
</tr>
<tr>
<td>delete_scheduled_action</td>
<td>Deletes the specified scheduled action for an Application Auto Scaling scalable target</td>
</tr>
<tr>
<td>deregister_scalable_target</td>
<td>Deregisters an Application Auto Scaling scalable target</td>
</tr>
<tr>
<td>describe_scalable_targets</td>
<td>Gets information about the scalable targets in the specified namespace</td>
</tr>
<tr>
<td>describe_scaling_activities</td>
<td>Provides descriptive information about the scaling activities in the specified namespace from the previous six weeks.</td>
</tr>
<tr>
<td>describe_scaling_policies</td>
<td>Describes the Application Auto Scaling scaling policies for the specified service namespace.</td>
</tr>
<tr>
<td>describe_scheduled_actions</td>
<td>Describes the Application Auto Scaling scheduled actions for the specified service namespace.</td>
</tr>
<tr>
<td>put_scaling_policy</td>
<td>Creates or updates a policy for an Application Auto Scaling scalable target</td>
</tr>
<tr>
<td>put_scheduled_action</td>
<td>Creates or updates a scheduled action for an Application Auto Scaling scalable target</td>
</tr>
<tr>
<td>register_scalable_target</td>
<td>Registers or updates a scalable target</td>
</tr>
</tbody>
</table>

## Examples

```r
# This example deletes a scaling policy for the Amazon ECS service called web-app, which is running in the default cluster.
# Not run: svc <- applicationautoscaling()
svc$delete_scaling_policy(
  PolicyName = "web-app-cpu-lt-25",
  ResourceId = "service/default/web-app",
  ScalableDimension = "ecs:service:DesiredCount",
  ServiceNamespace = "ecs"
)
# End(Not run)
```

---

### Amazon CloudWatch Application Insights

Amazon CloudWatch Application Insights for .NET and SQL Server

Amazon CloudWatch Application Insights for .NET and SQL Server is a service that helps you detect common problems with your .NET and SQL Server-based applications. It enables you to pinpoint the source of issues in your applications (built with technologies such as Microsoft IIS, .NET, and Microsoft SQL Server), by providing key insights into detected problems.

After you onboard your application, CloudWatch Application Insights for .NET and SQL Server identifies, recommends, and sets up metrics and logs. It continuously analyzes and correlates your metrics and logs for unusual behavior to surface actionable problems with your application. For example, if your application is slow and unresponsive and leading to HTTP 500 errors in your Application Load Balancer (ALB), Application Insights informs you that a memory pressure problem with your SQL Server database is occurring. It bases this analysis on impactful metrics and log errors.
applicationinsights

Usage

applicationinsights(config = list())

Arguments

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

Service syntax

svc <- applicationinsights(
  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_application  Adds an application that is created from a resource group
create_component  Creates a custom component by grouping similar standalone instances
create_log_pattern  Adds a log pattern to a LogPatternSet
delete_application  Removes the specified application from monitoring
delete_component  Ungroups a custom component
delete_log_pattern  Removes the specified log pattern from a LogPatternSet
describe_application  Describes the application
describe_component  Describes a component and lists the resources that are grouped together
describe_component_configuration  Describes the monitoring configuration of the component
describe_component_configuration_recommendation  Describes the recommended monitoring configuration of the component
describe_log_pattern  Describe a specific log pattern from a LogPatternSet
describe_observation  Describes an anomaly or error with the application
describe_problem  Describes an application problem
describe_problem_observations  Describes the anomalies or errors associated with the problem
list_applications  Lists the IDs of the applications that you are monitoring
list_components  Lists the auto-grouped, standalone, and custom components of the application
list_log_patterns  Lists the log patterns in the specific log LogPatternSet
list_log_pattern_sets  Lists the log pattern sets in the specific application
list_problems  Lists the problems with your application
list_tags_for_resource  Retrieve a list of the tags (keys and values) that are associated with a specified application
tag_resource  Add one or more tags (keys and values) to a specified application
untag_resource  Remove one or more tags (keys and values) from a specified application
update_application
update_component
update_component_configuration
update_log_pattern

Updates the application
Updates the custom component name and/or the list of resources that make up the component
Updates the monitoring configurations for the component
Adds a log pattern to a LogPatternSet

Examples

```r
## Not run: svc <- applicationinsights()
svc$create_application(
  Foo = 123
)
## End(Not run)
```

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 core-dns are supported. For more information, see DNS for Services and Pods in the Kubernetes documentation.

**Usage**

```r
appmesh(config = list())
```

**Arguments**

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

**Service syntax**

```r
svc <- appmesh(
  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_mesh` Creates a service mesh
- `create_route` Creates a route that is associated with a virtual router
- `create_virtual_node` Creates a virtual node within a service mesh
- `create_virtual_router` Creates a virtual router within a service mesh
- `create_virtual_service` Creates a virtual service within a service mesh
- `delete_mesh` Deletes an existing service mesh
- `delete_route` Deletes an existing route
- `delete_virtual_node` Deletes an existing virtual node
- `delete_virtual_router` Deletes an existing virtual router
- `delete_virtual_service` Deletes an existing virtual service
- `describe_mesh` Describes an existing service mesh
- `describe_route` Describes an existing route
- `describe_virtual_node` Describes an existing virtual node
- `describe_virtual_router` Describes an existing virtual router
- `describe_virtual_service` Describes an existing virtual service
- `list_meshes` Returns a list of existing service meshes
- `list_routes` Returns a list of existing routes in a service mesh
- `list_tags_for_resource` List the tags for an App Mesh resource
- `list_virtual_nodes` Returns a list of existing virtual nodes
- `list_virtual_routers` Returns a list of existing virtual routers in a service mesh
- `list_virtual_services` Returns a list of existing virtual services in a service mesh
- `tag_resource` Associates the specified tags to a resource with the specified resourceArn
- `untag_resource` Deletes specified tags from a resource
- `update_mesh` Updates an existing service mesh
- `update_route` Updates an existing route for a specified service mesh and virtual router
- `update_virtual_node` Updates an existing virtual node in a specified service mesh
- `update_virtual_router` Updates an existing virtual router in a specified service mesh
- `update_virtual_service` Updates an existing virtual service in a specified service mesh

**Examples**

```r
## Not run: svc <- appmesh()
svc$screate_mesh(
  Foo = 123
)```
## Description

Amazon Athena is an interactive query service that lets you use standard SQL to analyze data directly in Amazon S3. You can point Athena at your data in Amazon S3 and run ad-hoc queries and get results in seconds. Athena is serverless, so there is no infrastructure to set up or manage. You pay only for the queries you run. Athena scales automatically—executing queries in parallel—so results are fast, even with large datasets and complex queries. For more information, see What is Amazon Athena in the Amazon Athena User Guide.

If you connect to Athena using the JDBC driver, use version 1.1.0 of the driver or later with the Amazon Athena API. Earlier version drivers do not support the API. For more information and to download the driver, see Accessing Amazon Athena with JDBC.

For code samples using the AWS SDK for Java, see Examples and Code Samples in the Amazon Athena User Guide.

## Usage

```r
athena(config = list())
```

### Arguments

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

### Service syntax

```r
csvc <- athena(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

### Operations
batch_get_named_query Returns the details of a single named query or a list of up to 50 queries, which you provide as an array of query ID strings.

batch_get_query_execution Returns the details of a single query execution or a list of up to 50 query executions, which you provide as an array of query execution ID strings.

create_named_query Creates a named query in the specified workgroup.

create_work_group Creates a workgroup with the specified name.

delete_named_query Deletes the named query if you have access to the workgroup in which the query was saved.

delete_work_group Deletes the workgroup with the specified name.

get_named_query Returns information about a single query.

get_query_execution Returns information about a single execution of a query if you have access to the workgroup in which the query ran.

get_query_results Streams the results of a single query execution specified by QueryExecutionId from the Athena query results location in Amazon S3.

get_work_group Returns information about the workgroup with the specified name.

list_named_queries Provides a list of available query IDs only for queries saved in the specified workgroup.

list_query_executions Provides a list of available query execution IDs for the queries in the specified workgroup.

list_tags_for_resource Lists the tags associated with this workgroup.

list_work_groups Lists available workgroups for the account.

start_query_execution Runs the SQL query statements contained in the Query.

stop_query_execution Stops a query execution.

tag_resource Adds one or more tags to the resource, such as a workgroup.

untag_resource Removes one or more tags from the workgroup resource.

update_work_group Updates the workgroup with the specified name.

Examples

```r
## Not run: svc <- athena()
svc$batch_get_named_query(
  Foo = 123
)
## End(Not run)
```

autoscaling

Auto Scaling

Description

Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling is designed to automatically launch or terminate EC2 instances based on user-defined scaling policies, scheduled actions, and health checks. Use this service with AWS Auto Scaling, Amazon CloudWatch, and Elastic Load Balancing.

For more information, including information about granting IAM users required permissions for Amazon EC2 Auto Scaling actions, see the Amazon EC2 Auto Scaling User Guide.

Usage

```r
autoscaling(config = list())
```
Arguments
config Optional configuration of credentials, endpoint, and/or region.

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

Operations
attach_instances Attaches one or more EC2 instances to the specified Auto Scaling group
attach_load_balancers Attaches one or more Classic Load Balancers to the specified Auto Scaling group
attach_load_balancer_target_groups Attaches one or more target groups to the specified Auto Scaling group
batch_delete_scheduled_action Deletes one or more scheduled actions for the specified Auto Scaling group
batch_put_scheduled_update_group_action Creates or updates one or more scheduled scaling actions for an Auto Scaling group
complete_lifecycle_action Completes the lifecycle action for the specified token or instance with the specified result
create_auto_scaling_group Creates an Auto Scaling group with the specified name and attributes
create_launch_configuration Creates a launch configuration
create_or_update_tags Creates or updates tags for the specified Auto Scaling group
delete_auto_scaling_group Deletes the specified Auto Scaling group
delete_launch_configuration Deletes the specified launch configuration
delete_lifecycle_hook Deletes the specified lifecycle hook
delete_notification_configuration Deletes the specified notification
delete_policy Deletes the specified scaling policy
delete_scheduled_action Deletes the specified scheduled action
delete_tags Deletes the specified tags
describe_account_limits Describes the current Amazon EC2 Auto Scaling resource limits for your AWS account
describe_adjustment_types Describes the policy adjustment types for use with PutScalingPolicy
describe_auto_scaling_groups Describes one or more Auto Scaling groups
describe_auto_scaling_instances Describes one or more Auto Scaling instances
describe_auto_scaling_notification_types Describes the notification types that are supported by Amazon EC2 Auto Scaling
describe_launch_configurations Describes one or more launch configurations
describe_lifecycle_hooks Describes the lifecycle hooks for the specified Auto Scaling group
describe_lifecycle_hook_types Describes the available types of lifecycle hooks
describe_load_balancers Describes the target groups for the specified Auto Scaling group
describe_load_balancer_target_groups
describe_metric_collection_types Describes the available CloudWatch metrics for Amazon EC2 Auto Scaling
describe_notification_configurations Describes the notification actions associated with the specified Auto Scaling group
describe_policies Describes the policies for the specified Auto Scaling group
describe_scaling_activities Describes one or more scaling activities for the specified Auto Scaling group
describe_scheduled_actions Describes the actions scheduled for your Auto Scaling group that haven’t run or run too recently
describe_tags Describes the specified tags
describe_termination_policy_types Describes the termination policies supported by Amazon EC2 Auto Scaling
detach_instances Removes one or more instances from the specified Auto Scaling group
detach_load_balancers Detaches one or more Classic Load Balancers from the specified Auto Scaling group
detach_load_balancer_target_groups Detaches one or more target groups from the specified Auto Scaling group
disable_metrics_collection Disables group metrics for the specified Auto Scaling group
enable_metrics_collection Enables group metrics for the specified Auto Scaling group
enter_standby Moves the specified instances into the standby state
execute_policy Executes the specified policy
exit_standby Moves the specified instances out of the standby state
put_lifecycle_hook Creates or updates a lifecycle hook for the specified Auto Scaling group
put_notification_configuration Configures an Auto Scaling group to send notifications when specified events take place
put_scaling_policy Creates or updates a scaling policy for an Auto Scaling group
put_scaled_update_group_action Creates or updates a scheduled scaling action for an Auto Scaling group
record_lifecycle_action_heartbeat Records a heartbeat for the lifecycle action associated with the specified token or instance
resume_processes Resumes the specified suspended automatic scaling processes, or all suspended process, for the specified Auto Scaling group
set_desired_capacity Sets the size of the specified Auto Scaling group
set_instance_health Sets the health status of the specified instance
set_instance_protection Updates the instance protection settings of the specified instances
suspend_processes Suspends the specified automatic scaling processes, or all processes, for the specified Auto Scaling group
terminate_instance_in_auto_scaling_group Terminates the specified instance and optionally adjusts the desired group size
update_auto_scaling_group Updates the configuration for the specified Auto Scaling group

Examples

```
# This example attaches the specified instance to the specified Auto
# Scaling group.
## Not run: svc <- autoscaling()
svc$attach_instances(
    AutoScalingGroupName = "my-auto-scaling-group",
    InstanceIds = list(
        "i-93633f9b"
    )
)
## End(Not run)
```

---

**autoscalingplans**

AWS Auto Scaling Plans
Description

AWS Auto Scaling

Use AWS Auto Scaling to quickly discover all the scalable AWS resources for your application and configure dynamic scaling and predictive scaling for your resources using scaling plans. Use this service in conjunction with the Amazon EC2 Auto Scaling, Application Auto Scaling, Amazon CloudWatch, and AWS CloudFormation services.

Currently, predictive scaling is only available for Amazon EC2 Auto Scaling groups.

For more information about AWS Auto Scaling, including information about granting IAM users required permissions for AWS Auto Scaling actions, see the AWS Auto Scaling User Guide.

Usage

autoscalingplans(config = list())

Arguments

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

Service syntax

svc <- autoscalingplans(
  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_scaling_plan Creates a scaling plan
delete_scaling_plan Deletes the specified scaling plan
describe_scaling_plan_resources Describes the scalable resources in the specified scaling plan
describe_scaling_plans Describes one or more of your scaling plans
get_scaling_plan_resource_forecast_data Retrieves the forecast data for a scalable resource
update_scaling_plan Updates the specified scaling plan
### AWS Backup

**Description**

AWS Backup is a unified backup service designed to protect AWS services and their associated data. AWS Backup simplifies the creation, migration, restoration, and deletion of backups, while also providing reporting and auditing.

**Usage**

```r
backup(config = list())
```

**Arguments**

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

**Service syntax**

```r
csvc <- backup(
  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_backup_plan` Backup plans are documents that contain information that AWS Backup uses to schedule backup tasks.
- `create_backup_selection` Creates a JSON document that specifies a set of resources to assign to a backup plan.
- `create_backup_vault` Creates a logical container where backups are stored.
delete_backup_plan
delete_backup_selection
delete_backup_vault
delete_backup_vault_access_policy
delete_backup_vault_notifications
delete_recovery_point
describe_backup_job
describe_backup_vault
describe_protected_resource
describe_recovery_point
describe_supported_resource_types
get_backup_plan
get_backup_plan_from_json
get_backup_plan_from_template
get_backup_selection
get_backup_vault_access_policy
get_backup_vault_notifications
get_recovery_point_restore_metadata
get_supported_resource_types
list_backup_jobs
list_backup_plans
list_backup_plan_templates
list_backup_plan_versions
list_backup_selections
list_backup_vaults
list_protected_resources
list_recovery_points_by_backup_vault
list_recovery_points_by_resource
list_restore_jobs
list_tags
put_backup_vault_access_policy
put_backup_vault_notifications
start_backup_job
start_restore_job
stop_backup_job
tag_resource
tag_resource
update_backup_plan
update_recovery_point_lifecycle

Deletes a backup plan
Deletes the resource selection associated with a backup plan that is specified by the SelectionId
Deletes the backup vault identified by its name
Deletes the policy document that manages permissions on a backup vault
Deletes event notifications for the specified backup vault
Deletes the recovery point specified by a recovery point ID
Returns metadata associated with creating a backup of a resource
Returns information about a backup vault specified by its name
Returns metadata associated with a saved resource, including the last time it was backed-up
Returns metadata associated with a recovery point, including ID, status, encryption, and storage
Returns metadata associated with a restore job that is specified by a job ID
Returns the backup plan that is specified by the plan ID as a backup template
Returns the body of a backup plan in JSON format, in addition to plan metadata
Returns a valid JSON document specifying a backup plan or an error
Returns the template specified by its templateId as a backup plan
Returns selection metadata and a document in JSON format that specifies a list of resources
Returns the access policy document that is associated with the named backup vault
Returns event notifications for the specified backup vault
Returns two sets of metadata key-value pairs
Returns the AWS resource types supported by AWS Backup
Returns metadata about your backup jobs
Returns metadata of your saved backup plans, including Amazon Resource Names (ARNs)
Returns metadata of your saved backup plan templates, including the template ID, name, and description
Returns version metadata of your backup plans, including Amazon Resource Names
Returns an array containing metadata of the resources associated with the target backup
Returns a list of recovery point storage containers along with information about them
Returns an array of resources successfully backed up by AWS Backup, including the ARN
Returns detailed information about the recovery points stored in a backup vault
Returns detailed information about recovery points of the type specified by a resource
Returns a list of jobs that AWS Backup initiated to restore a saved resource, including the ARN
Returns a list of key-value pairs assigned to a target recovery point, backup plan, or backup vault
Sets a resource-based policy that is used to manage access permissions on the target backup vault or vault
Turns on notifications on a backup vault for the specified topic and events
Starts a job to create a one-time backup of the specified resource
Recovers the saved resource identified by an Amazon Resource Name (ARN)
Attempts to cancel a job to create a one-time backup of a resource
Assigns a set of key-value pairs to a recovery point, backup plan, or backup vault identified by an ARN
Removes a set of key-value pairs from a recovery point, backup plan, or backup vault identified by an ARN
Replaces the body of a saved backup plan identified by its backupPlanId with the input JSON document
Sets the transition lifecycle of a recovery point

Examples
## Not run: svc <- backup()
svc$create_backup_plan(
  Foo = 123
)
## Description

AWS Batch enables you to run batch computing workloads on the AWS Cloud. Batch computing is a common way for developers, scientists, and engineers to access large amounts of compute resources, and AWS Batch removes the undifferentiated heavy lifting of configuring and managing the required infrastructure. AWS Batch will be familiar to users of traditional batch computing software. This service can efficiently provision resources in response to jobs submitted in order to eliminate capacity constraints, reduce compute costs, and deliver results quickly.

As a fully managed service, AWS Batch enables developers, scientists, and engineers to run batch computing workloads of any scale. AWS Batch automatically provisions compute resources and optimizes the workload distribution based on the quantity and scale of the workloads. With AWS Batch, there is no need to install or manage batch computing software, which allows you to focus on analyzing results and solving problems. AWS Batch reduces operational complexities, saves time, and reduces costs, which makes it easy for developers, scientists, and engineers to run their batch jobs in the AWS Cloud.

## Usage

```r
batch(config = list())
```

## Arguments

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

## Service syntax

```r
cvc <- batch(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```
Operations
cancel_job
create_compute_environment
create_job_queue
delete_compute_environment
delete_job_queue
deregister_job_definition
describe_compute_environments
describe_job_definitions
describe_job_queues
describe_jobs
list_jobs
register_job_definition
submit_job
terminate_job
update_compute_environment
update_job_queue

Cancels a job in an AWS Batch job queue
Creates an AWS Batch compute environment
Creates an AWS Batch job queue
Deletes an AWS Batch compute environment
Deletes the specified job queue
Deregisters an AWS Batch job definition
Describes one or more of your compute environments
Describes a list of job definitions
Describes one or more of your job queues
Describes a list of AWS Batch jobs
Returns a list of AWS Batch jobs
Registers an AWS Batch job definition
Submits an AWS Batch job from a job definition
Terminates a job in a job queue
Updates an AWS Batch compute environment
Updates a job queue

Examples

# This example cancels a job with the specified job ID.
## Not run: svc <- batch()
svc$cancel_job(
  jobId = "1d828f65-7a4d-42e8-996d-3b900ed59dc4",
  reason = " Cancelling job."
)
## End(Not run)

budgets  AWS Budgets

Description

The AWS Budgets API enables you to use AWS Budgets to plan your service usage, service costs, and instance reservations. The API reference provides descriptions, syntax, and usage examples for each of the actions and data types for AWS Budgets.

Budgets provide you with a way to see the following information:

- How close your plan is to your budgeted amount or to the free tier limits
- Your usage-to-date, including how much you've used of your Reserved Instances (RIs)
- Your current estimated charges from AWS, and how much your predicted usage will accrue in charges by the end of the month
- How much of your budget has been used
AWS updates your budget status several times a day. Budgets track your unblended costs, subscriptions, refunds, and RIs. You can create the following types of budgets:

- **Cost budgets** - Plan how much you want to spend on a service.
- **Usage budgets** - Plan how much you want to use one or more services.
- **RI utilization budgets** - Define a utilization threshold, and receive alerts when your RI usage falls below that threshold. This lets you see if your RIs are unused or under-utilized.
- **RI coverage budgets** - Define a coverage threshold, and receive alerts when the number of your instance hours that are covered by RIs fall below that threshold. This lets you see how much of your instance usage is covered by a reservation.

### Service Endpoint

The AWS Budgets API provides the following endpoint:

- https://budgets.amazonaws.com

For information about costs that are associated with the AWS Budgets API, see [AWS Cost Management Pricing](https://aws.amazon.com/cost-management/pricing).

### Usage

```python
buckets(config = list())
```

### Arguments

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

### Service syntax

```python
svc <- budgets(
  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_budget**
  
  Creates a budget and, if included, notifications and subscribers

- **create_notification**
  
  Creates a notification

- **create_subscriber**
  
  Creates a subscriber
Amazon Cloud Directory

Amazon Cloud Directory is a component of the AWS Directory Service that simplifies the development and management of cloud-scale web, mobile, and IoT applications. This guide describes the Cloud Directory operations that you can call programmatically and includes detailed information on data types and errors. For information about Cloud Directory features, see AWS Directory Service and the Amazon Cloud Directory Developer Guide.

Usage

clouddirectory(config = list())

Arguments

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

```r
csvc <- clouddirectory(
    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_facet_to_object**: Adds a new Facet to an object
- **apply_schema**: Copies the input published schema, at the specified version, into the Directory with the same name and version as that of the published schema
- **attach_object**: Attaches an existing object to another object
- **attach_policy**: Attaches a policy object to a regular object
- **attach_to_index**: Attaches the specified object to the specified index
- **attach_typed_link**: Attaches a typed link to a specified source and target object
- **batch_read**: Performs all the read operations in a batch
- **batch_write**: Performs all the write operations in a batch
- **create_directory**: Creates a Directory by copying the published schema into the directory
- **create_facet**: Creates a new Facet in a schema
- **create_index**: Creates an index object
- **create_object**: Creates an object in a Directory
- **create_schema**: Creates a new schema in a development state
- **create_typed_link_facet**: Creates a TypedLinkFacet
- **delete_directory**: Deletes a directory
- **delete_facet**: Deletes a given Facet
- **delete_object**: Deletes an object and its associated attributes
- **delete_schema**: Deletes a given schema
- **delete_typed_link_facet**: Deletes a TypedLinkFacet
- **detach_from_index**: Detaches the specified object from the specified index
- **detach_object**: Detaches a given object from the parent object
- **detach_policy**: Detaches a policy from an object
- **detach_typed_link**: Detaches a typed link from a specified source and target object
- **disable_directory**: Disables the specified directory
- **enable_directory**: Enables the specified directory
- **get_applied_schema_version**: Returns current applied schema version ARN, including the minor version in use
- **get_directory**: Retrieves metadata about a directory
- **get_facet**: Gets details of the Facet, such as facet name, attributes, Rules, or ObjectType
- **get_link_attributes**: Retrieves attributes that are associated with a typed link
- **get_object_attributes**: Retrieves attributes within a facet that are associated with an object
get_object_information  Retrieves metadata about an object
get_schema_as_json  Retrieves a JSON representation of the schema
get_typed_link_facet_information  Returns the identity attribute order for a specific TypedLinkFacet
list_applied_schema_arns  Lists schema major versions applied to a directory
list_attached_indices  Lists indices attached to the specified object
list_development_schema_arns  Retrieves each Amazon Resource Name (ARN) of schemas in the development state
list_directories  Lists directories created within an account
list_facet_attributes  Retrieves attributes attached to the facet
list_facet_names  Retrieves the names of facets that exist in a schema
list_index  Lists objects attached to the specified index
list_managed_schema_arns  Lists the major version families of each managed schema
list_object_attributes  Lists all attributes that are associated with an object
list_object_children  Retrieves a paginated list of child objects that are associated with a given object
list_object_parent_paths  Retrieves all available parent paths for any object type such as node, leaf node, policy node
list_object_parents  Lists parent objects that are associated with a given object in pagination fashion
list_object_policies  Returns policies attached to an object in pagination fashion
list_outgoing_typed_links  Returns a paginated list of all the outgoing TypedLinkSpecifier information for an object
list_policy_attachments  Returns all of the ObjectIdentifiers to which a given policy is attached
list_published_schema_arns  Lists the major version families of each published schema
list_tags_for_resource  Returns tags for a resource
list_typed_link_facet_attributes  Returns a paginated list of all attribute definitions for a particular TypedLinkFacet
list_typed_link_facet_names  Returns a paginated list of TypedLink facet names for a particular schema
lookup_policy  Publishes a development schema with a major version and a recommended minor version
put_schema_from_json  Allows a schema to be updated using JSON upload
remove_facet_from_object  Removes the specified facet from the specified object
tag_resource  An API operation for adding tags to a resource
untag_resource  An API operation for removing tags from a resource
update_facet  Does the following: 1
update_link_attributes  Updates a given typed link’s attributes
update_object_attributes  Updates a given object’s attributes
update_schema  Updates the schema name with a new name
update_typed_link_facet  Updates a TypedLinkFacet
upgrade_applied_schema  Upgrades a single directory in-place using the PublishedSchemaArn with schema updates
upgrade_published_schema  Upgrades a published schema under a new minor version revision using the current context

Examples

```r
# Not run: svc <- clouddirectory()
svc$add_facet_to_object(
  Foo = 123
)
# End(Not run)
```
Description

AWS CloudFormation allows you to create and manage AWS infrastructure deployments predictably and repeatedly. You can use AWS CloudFormation to leverage AWS products, such as Amazon Elastic Compute Cloud, Amazon Elastic Block Store, Amazon Simple Notification Service, Elastic Load Balancing, and Auto Scaling to build highly-reliable, highly scalable, cost-effective applications without creating or configuring the underlying AWS infrastructure.

With AWS CloudFormation, you declare all of your resources and dependencies in a template file. The template defines a collection of resources as a single unit called a stack. AWS CloudFormation creates and deletes all member resources of the stack together and manages all dependencies between the resources for you.

For more information about AWS CloudFormation, see the AWS CloudFormation Product Page.

Amazon CloudFormation makes use of other AWS products. If you need additional technical information about a specific AWS product, you can find the product’s technical documentation at docs.aws.amazon.com.

Usage

cloudformation(config = list())

Arguments

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

Service syntax

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

Operations
cancel_update_stack
continue_update_rollback
create_change_set
create_stack
create_stack_instances
create_stack_set
delete_change_set
delete_stack
delete_stack_instances
delete_stack_set
deregister_type
describe_account_limits
describe_change_set
describe_stack_drift_detection_status
describe_stack_events
describe_stack_instance
describe_stack_resource
describe_stack_resource_drifts
describe_stack_resources
describe_stacks
describe_stack_set
describe_stack_set_operation
describe_type
describe_type_registration
detect_stack_drift
detect_stack_resource_drift
detect_stack_set_drift
estimate_template_cost
execute_change_set
get_stack_policy
get_template
get_template_summary
list_change_sets
list_exports
list_imports
list_stack_instances
list_stack_resources
list_stacks
list_stack_set_operation_results
list_stack_set_operations
list_stack_sets
list_typeRegistrations
list_types
list_type_versions
record_handler_progress
register_type
set_stack_policy
set_type_default_version

cancels an update on the specified stack.
For a specified stack that is in the UPDATE_ROLLBACK_FAILED state, continues rolling
back the stack to the UPDATE_ROLLBACK_COMPLETE state.
Creates a change set that will be applied to a stack so that you can review the changes
before executing them.
Creates a stack as specified in the template.
Creates stack instances for the specified accounts, within the specified regions.
Creates a stack set.
Deletes the specified change set.
Deletes a specified stack.
Deletes stack instances for the specified accounts, in the specified regions.
Deletes a stack set.
Removes a type or type version from active use in the CloudFormation registry.
Retrieves your account’s AWS CloudFormation limits, such as the maximum number of
stacks that can be created in your account.
Returns the inputs for the change set and a list of changes that AWS CloudFormation
will make if you execute the change set.
Returns information about a stack drift detection operation.
Returns all stack related events for a specified stack in reverse chronological order.
Returns the stack instance that’s associated with the specified stack set, AWS account,
and region.
Returns a description of the specified resource in the specified stack.
Returns drift information for the resources that have been checked for drift in the specified
stack set.
Returns AWS resource descriptions for running and deleted stacks.
Returns the description for the specified stack; if no stack name was specified, then it
returns the description of the specified stack set.
Returns the description of the specified stack set operation.
Returns detailed information about a type that has been registered.
Returns information about a type’s registration, including its current status and type and
version identifiers.
Detects whether a stack’s actual configuration differs, or has drifted, from it’s expected
configuration, as defined in the stack template and any values specified as template
parameters.
Detects whether a resource’s actual configuration differs, or has drifted, from it’s expected
configuration, as defined in the stack template and any values specified as template
parameters.
Detects drift on a stack set.
Returns the estimated monthly cost of a template.
Updates a stack using the input information that was provided when the specified change
set was created.
Returns the stack policy for a specified stack.
Returns the template body for a specified stack.
Returns information about a new or existing template.
Returns the ID and status of each active change set for a stack.
Lists all exported output values in the account and region in which you call this action.
Lists all stacks that are importing an exported output value.
Returns summary information about stack instances that are associated with the specified
stack.
Returns descriptions of all resources of the specified stack.
Returns the summary information for stacks whose status matches the specified Stack
Status.
Returns summary information about the results of a stack set operation.
Returns summary information about operations performed on a stack set.
Returns summary information about stack sets that are associated with the user.
Returns a list of registration tokens for the specified type.
Returns summary information about types that have been registered with CloudFormation.
Returns summary information about the versions of a type.
Reports progress of a resource handler to CloudFormation.
Registers a type with the CloudFormation service.
Sets a stack policy for a specified stack.
Specify the default version of a type.
signal_resource Sends a signal to the specified resource with a success or failure status
stop_stack_set_operation Stops an in-progress operation on a stack set and its associated stack instances
update_stack Updates a stack as specified in the template
update_stack_instances Updates the parameter values for stack instances for the specified accounts, within the stack set
update_stack_set Updates the stack set, and associated stack instances in the specified accounts and regions
update_termination_protection Updates termination protection for the specified stack
validate_template Validates a specified template

Examples

```r
## Not run: svc <- cloudformation()
svc$cancel_update_stack(
  Foo = 123
)
## End(Not run)
```

---

**cloudfront**

*Amazon CloudFront*

**Description**

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

**Usage**

```r
cloudfront(config = list())
```

**Arguments**

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

**Service syntax**

```r
cvc <- cloudfront(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"),
      profile = "string"
    ),
```
Operations

- `create_cloud_front_origin_access_identity` creates a new origin access identity.
- `create_distribution` creates a new web distribution.
- `create_distribution_with_tags` creates a new distribution with tags.
- `create_field_level_encryption_config` creates a new field-level encryption configuration.
- `create_field_level_encryption_profile` creates a field-level encryption profile.
- `create_invalidation` creates a new invalidation.
- `create_public_key` adds a new public key to CloudFront to use, for example, for field-level encryption.
- `create_streaming_distribution` creates a new RTMP distribution.
- `create_streaming_distribution_with_tags` creates a new streaming distribution with tags.
- `delete_cloud_front_origin_access_identity` deletes an origin access identity.
- `delete_distribution` deletes a distribution.
- `delete_field_level_encryption_config` removes a field-level encryption configuration.
- `delete_field_level_encryption_profile` removes a field-level encryption profile.
- `delete_public_key` removes a public key you previously added to CloudFront.
- `delete_streaming_distribution` deletes a streaming distribution.
- `get_cloud_front_origin_access_identity` gets the information about an origin access identity.
- `get_cloud_front_origin_access_identity_config` gets the configuration information about an origin access identity.
- `get_distribution` gets the information about a distribution.
- `get_distribution_config` gets the configuration information about a distribution.
- `get_field_level_encryption` gets the field-level encryption configuration information.
- `get_field_level_encryption_config` gets the field-level encryption configuration information.
- `get_field_level_encryption_profile` gets the field-level encryption profile information.
- `get_field_level_encryption_profile_config` gets the field-level encryption profile configuration information.
- `get_invalidation` gets the information about an invalidation.
- `get_public_key` returns the public key information.
- `get_public_key_config` returns public key configuration information.
- `get_streaming_distribution` gets information about a specified RTMP distribution, including the distribution configuration.
- `get_streaming_distribution_config` gets the configuration information about a streaming distribution.
- `list_cloud_front_origin_access_identities` lists origin access identities.
- `list_distributions` lists CloudFront distributions.
- `list_distributions_by_web_acl_id` lists the distributions that are associated with a specified AWS WAF web ACL.
- `list_field_level_encryption_configs` lists all field-level encryption configurations that have been created in CloudFront.
- `list_field_level_encryption_profiles` requests a list of field-level encryption profiles that have been created in CloudFront.
- `list_invalidations` lists invalidation batches.
- `list_public_keys` lists all public keys that have been added to CloudFront for this account.
- `list_streaming_distributions` lists streaming distributions.
- `list_tags_for_resource` lists tags for a CloudFront resource.
- `tag_resource` adds tags to a CloudFront resource.
- `untag_resource` removes tags from a CloudFront resource.
- `update_cloud_front_origin_access_identity` updates 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

```r
## Not run: svc <- cloudfront()  
svc$create_cloud_front_origin_access_identity(  
  Foo = 123  
)  
## End(Not run)
```

---

**cloudhsm**  
*Amazon CloudHSM*

**Description**

AWS CloudHSM Service

This is documentation for **AWS CloudHSM Classic**. For more information, see **AWS CloudHSM Classic FAQs**, the **AWS CloudHSM Classic User Guide**, and the **AWS CloudHSM Classic API Reference**.

**For information about the current version of AWS CloudHSM**, see **AWS CloudHSM**, the **AWS CloudHSM User Guide**, and the **AWS CloudHSM API Reference**.

**Usage**

```r
cloudhsm(config = list())
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

**Service syntax**

```r
svc <- cloudhsm(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string"  
    ),  
    profile = "string"
  )
)```
cloudhsmv2

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

Operations

add_tags_to_resource  This is documentation for AWS CLOUDHSM CLASSIC
create_hapg           This is documentation for AWS CLOUDHSM CLASSIC
create_hsm            This is documentation for AWS CLOUDHSM CLASSIC
create_luna_client    This is documentation for AWS CLOUDHSM CLASSIC
delete_hapg           This is documentation for AWS CLOUDHSM CLASSIC
delete_hsm            This is documentation for AWS CLOUDHSM CLASSIC
delete_luna_client    This is documentation for AWS CLOUDHSM CLASSIC
describe_hapg         This is documentation for AWS CLOUDHSM CLASSIC
describe_hsm          This is documentation for AWS CLOUDHSM CLASSIC
describe_luna_client  This is documentation for AWS CLOUDHSM CLASSIC
get_config            This is documentation for AWS CLOUDHSM CLASSIC
list_available_zones  This is documentation for AWS CLOUDHSM CLASSIC
list_hapgs            This is documentation for AWS CLOUDHSM CLASSIC
list_hsms             This is documentation for AWS CLOUDHSM CLASSIC
list_luna_clients     This is documentation for AWS CLOUDHSM CLASSIC
list_tags_for_resource This is documentation for AWS CLOUDHSM CLASSIC
modify_hapg           This is documentation for AWS CLOUDHSM CLASSIC
modify_hsm            This is documentation for AWS CLOUDHSM CLASSIC
modify_luna_client    This is documentation for AWS CLOUDHSM CLASSIC
remove_tags_from_resource This is documentation for AWS CLOUDHSM CLASSIC

Examples

```r
## Not run: svc <- cloudhsm()
svc$add_tags_to_resource(
    Foo = 123
)
## End(Not run)
```

Description

For more information about AWS CloudHSM, see AWS CloudHSM and the AWS CloudHSM User Guide.
Usage
cloudhsmv2(config = list())

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

Service syntax

svc <- cloudhsmv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

Operations

copy_backup_to_region Copy an AWS CloudHSM cluster backup to a different region
create_cluster Creates a new AWS CloudHSM cluster
create_hsm Creates a new hardware security module (HSM) in the specified AWS CloudHSM cluster
delete_backup Deletes a specified AWS CloudHSM backup
delete_cluster Deletes the specified AWS CloudHSM cluster
delete_hsm Deletes the specified HSM
describe_backups Gets information about backups of AWS CloudHSM clusters
describe_clusters Gets information about AWS CloudHSM clusters
initialize_cluster Claims an AWS CloudHSM cluster by submitting the cluster certificate issued by your issuing certific
describe_tags Gets a list of tags for the specified AWS CloudHSM cluster
restore_backup Restores a specified AWS CloudHSM backup that is in the PENDING_DELETION state
tag_resource Adds or overwrites one or more tags for the specified AWS CloudHSM cluster
untag_resource Removes the specified tag or tags from the specified AWS CloudHSM cluster

Examples

## Not run: svc <- cloudhsmv2()
svc$copy_backup_to_region(
  Foo = 123
)

### Description

Amazon CloudSearch Configuration Service

You use the Amazon CloudSearch configuration service to create, configure, and manage search domains. Configuration service requests are submitted using the AWS Query protocol. AWS Query requests are HTTP or HTTPS requests submitted via HTTP GET or POST with a query parameter named Action.

The endpoint for configuration service requests is region-specific: cloudsearch.{region}.amazonaws.com. For example, cloudsearch.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

### Usage

```r
cloudsearch(config = list())
```

### Arguments

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

### Service syntax

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

Description

You use the AmazonCloudSearch2013 API to upload documents to a search domain and search those documents.
The endpoints for submitting `UploadDocuments`, `Search`, and `Suggest` requests are domain-specific. To get the endpoints for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. The domain endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console. You submit suggest requests to the search endpoint.

For more information, see the Amazon CloudSearch Developer Guide.

**Usage**

```r
cloudsearchdomain(config = list())
```

**Arguments**

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

**Service syntax**

```r
csvc <- cloudsearchdomain(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

**Operations**

- `search` Retrieves a list of documents that match the specified search criteria
- `suggest` Retrieves autocomplete suggestions for a partial query string
- `upload_documents` Posts a batch of documents to a search domain for indexing

**Examples**

```r
## Not run: svc <- cloudsearchdomain()
svc$search(
  Foo = 123
)
## End(Not run)
```
**cloudtrail**

*AWS CloudTrail*

---

**Description**

This is the CloudTrail API Reference. It provides descriptions of actions, data types, common parameters, and common errors for CloudTrail.

CloudTrail is a web service that records AWS API calls for your AWS account and delivers log files to an Amazon S3 bucket. The recorded information includes the identity of the user, the start time of the AWS API call, the source IP address, the request parameters, and the response elements returned by the service.

As an alternative to the API, you can use one of the AWS SDKs, which consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWSCloudTrail. For example, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For information about the AWS SDKs, including how to download and install them, see the [Tools for Amazon Web Services page](https://aws.amazon.com/tools/). See the AWS CloudTrail User Guide for information about the data that is included with each AWS API call listed in the log files.

**Usage**

```python
cloudtrail(config = list())
```

**Arguments**

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

**Service syntax**

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

**Operations**
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_tags</td>
<td>Adds one or more tags to a trail, up to a limit of 50</td>
</tr>
<tr>
<td>create_trail</td>
<td>Creates a trail that specifies the settings for delivery of log data to an Amazon S3 bucket</td>
</tr>
<tr>
<td>delete_trail</td>
<td>Deletes a trail</td>
</tr>
<tr>
<td>describe_trails</td>
<td>Retrieves settings for one or more trails associated with the current region for your account</td>
</tr>
<tr>
<td>get_event_selectors</td>
<td>Describes the settings for the event selectors that you configured for your trail</td>
</tr>
<tr>
<td>get_insight_selectors</td>
<td>Describes the settings for the Insights event selectors that you configured for your trail</td>
</tr>
<tr>
<td>get_trail</td>
<td>Returns settings information for a specified trail</td>
</tr>
<tr>
<td>get_trail_status</td>
<td>Returns a JSON-formatted list of information about the specified trail</td>
</tr>
<tr>
<td>list_public_keys</td>
<td>Returns all public keys whose private keys were used to sign the digest files within the specified time range</td>
</tr>
<tr>
<td>list_tags</td>
<td>Lists the tags for the trail in the current region</td>
</tr>
<tr>
<td>list_trails</td>
<td>Lists trails that are in the current account</td>
</tr>
<tr>
<td>lookup_events</td>
<td>Looks up management events or CloudTrail Insights events that are captured by CloudTrail</td>
</tr>
<tr>
<td>put_event_selectors</td>
<td>Configures an event selector for your trail</td>
</tr>
<tr>
<td>put_insight_selectors</td>
<td>Lets you enable Insights event logging by specifying the Insights selectors that you want to enable on an existing trail</td>
</tr>
<tr>
<td>remove_tags</td>
<td>Removes the specified tags from a trail</td>
</tr>
<tr>
<td>start_logging</td>
<td>Starts the recording of AWS API calls and log file delivery for a trail</td>
</tr>
<tr>
<td>stop_logging</td>
<td>Suspends the recording of AWS API calls and log file delivery for the specified trail</td>
</tr>
<tr>
<td>update_trail</td>
<td>Updates the settings that specify delivery of log files</td>
</tr>
</tbody>
</table>

**Examples**

```r
## Not run: svc <- cloudtrail()
svc$add_tags(
  Foo = 123
)
## End(Not run)
```

---

**Description**

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are the variables you want to measure for your resources and applications.

CloudWatch alarms send notifications or automatically change the resources you are monitoring based on rules that you define. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances. Then, use this data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop under-used instances to save money.

In addition to monitoring the built-in metrics that come with AWS, you can monitor your own custom metrics. With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.
Usage

cloudwatch(config = list())

Arguments

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

Service syntax

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

Operations

delete_alarms Deletes the specified alarms
delete_anomaly_detector Deletes the specified anomaly detection model from your account
delete_dashboards Deletes all dashboards that you specify
delete_insight_rules Permanently deletes the specified Contributor Insights rules
describe_alarm_history Retrieves the history for the specified alarm
describe_alarms Retrieves the specified alarms
describe_alarms_for_metric Retrieves the alarms for the specified metric
describe_anomaly_detectors Lists the anomaly detection models that you have created in your account
describe_insight_rules Returns a list of all the Contributor Insights rules in your account
disable_alarm_actions Disables the actions for the specified alarms
disable_insight_rules Disables the specified Contributor Insights rules
enable_alarm_actions Enables the actions for the specified alarms
enable_insight_rules Enables the specified Contributor Insights rules
get_dashboard Displays the details of the dashboard that you specify
get_insight_rule_report This operation returns the time series data collected by a Contributor Insights rule
get_metric_data You can use the GetMetricData API to retrieve as many as 100 different metrics in a single request
get_metric_statistics Gets statistics for the specified metric
get_metric_widget_image You can use the GetMetricWidgetImage API to retrieve a snapshot graph of one or more Amazon CloudWatch metrics
list_dashboards Returns a list of the dashboards for your account
list_metrics List the specified metrics
list_tags_for_resource Displays the tags associated with a CloudWatch resource
put_anomaly_detector Creates an anomaly detection model for a CloudWatch metric
cloudwatchevents

put_dashboard
put_insight_rule
put_metric_alarm
put_metric_data
set_alarm_state
tag_resource
untag_resource

Creates a dashboard if it does not already exist, or updates an existing dashboard
Creates a Contributor Insights rule
Creates or updates an alarm and associates it with the specified metric, metric math expression, or anomaly detection model
Publishes metric data points to Amazon CloudWatch
Temporarily sets the state of an alarm for testing purposes
Assigns one or more tags (key-value pairs) to the specified CloudWatch resource
Removes one or more tags from the specified resource

Examples

```
## Not run: svc <- cloudwatch()
svc$delete_alarms(
  Foo = 123
)
## End(Not run)
```

cloudwatchevents Amazon CloudWatch Events

Description

Amazon EventBridge helps you to respond to state changes in your AWS resources. When your resources change state, they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a predetermined schedule. For example, you can configure rules to:

- Automatically invoke an AWS Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the running state
- Direct specific API records from AWS CloudTrail to an Amazon Kinesis data stream for detailed analysis of potential security or availability risks
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume

For more information about the features of Amazon EventBridge, see the Amazon EventBridge User Guide.

Usage

```
cloudwatchevents(config = list())
```

Arguments

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

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

Operations

- **activate_event_source**: Activates a partner event source that has been deactivated
- **create_event_bus**: Creates a new event bus within your account
- **create_partner_event_source**: Called by an SaaS partner to create a partner event source
- **deactivate_event_source**: An AWS customer uses this operation to temporarily stop receiving events from the specified partner event source
- **delete_event_bus**: Deletes the specified custom event bus or partner event bus
- **delete_partner_event_source**: This operation is used by SaaS partners to delete a partner event source
- **delete_rule**: Deletes the specified rule
- **describe_event_bus**: Displays details about an event bus in your account
- **describe_event_source**: This operation lists details about a partner event source that is shared with your account
- **describe_partner_event_source**: An SaaS partner can use this operation to list details about a partner event source that they have created
- **describe_rule**: Describes the specified rule
- **disable_rule**: Disables the specified rule
- **enable_rule**: Enables the specified rule
- **list_event_buses**: Lists all the event buses in your account, including the default event bus, custom event buses, and partner event buses
- **list_event_sources**: You can use this to see all the partner event sources that have been shared with your AWS account
- **list_partner_event_source_accounts**: An SaaS partner can use this operation to display the AWS account ID that a particular partner event source is associated with
- **list_partner_event_sources**: An SaaS partner can use this operation to list all the partner event source names that they have created
- **list_rule_names_by_target**: Lists the rules for the specified target
- **list_rules**: Lists your EventBridge rules
- **list_tags_for_resource**: Displays the tags associated with an EventBridge resource
- **list_targets_by_rule**: Lists the targets assigned to the specified rule
- **put_events**: Sends custom events to EventBridge so that they can be matched to rules
- **put_partner_events**: This is used by SaaS partners to write events to a customer’s partner event bus
- **put_permission**: Running PutPermission permits the specified AWS account or AWS organization to put events to the specified event bus
- **put_rule**: Creates or updates the specified rule
- **put_targets**: Adds the specified targets to the specified rule, or updates the targets if they’re already associated
- **remove_permission**: Revokes the permission of another AWS account to be able to put events to the specified event bus
- **remove_targets**: Removes the specified targets from the specified rule
- **tag_resource**: Assigns one or more tags (key-value pairs) to the specified EventBridge resource
- **test_event_pattern**: Tests whether the specified event pattern matches the provided event
untag_resource

Removes one or more tags from the specified EventBridge resource

Examples

```r
## Not run: svc <- cloudwatchevents()
svc$activate_event_source(
  Foo = 123
)
## End(Not run)
```

cloudwatchlogs

Amazon CloudWatch Logs

Description

You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. You can then retrieve the associated log data from CloudWatch Logs using the CloudWatch console, CloudWatch Logs commands in the AWS CLI, CloudWatch Logs API, or CloudWatch Logs SDK.

You can use CloudWatch Logs to:

- **Monitor logs from EC2 instances in real-time**: You can use CloudWatch Logs to monitor applications and systems using log data. For example, CloudWatch Logs can track the number of errors that occur in your application logs and send you a notification whenever the rate of errors exceeds a threshold that you specify. CloudWatch Logs uses your log data for monitoring; so, no code changes are required. For example, you can monitor application logs for specific literal terms (such as "NullReferenceException") or count the number of occurrences of a literal term at a particular position in log data (such as "404" status codes in an Apache access log). When the term you are searching for is found, CloudWatch Logs reports the data to a CloudWatch metric that you specify.

- **Monitor AWS CloudTrail logged events**: You can create alarms in CloudWatch and receive notifications of particular API activity as captured by CloudTrail and use the notification to perform troubleshooting.

- **Archive log data**: You can use CloudWatch Logs to store your log data in highly durable storage. You can change the log retention setting so that any log events older than this setting are automatically deleted. The CloudWatch Logs agent makes it easy to quickly send both rotated and non-rotated log data off of a host and into the log service. You can then access the raw log data when you need it.

Usage

```r
cloudwatchlogs(config = list())
```
Arguments

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

Service syntax

```r
svc <- cloudwatchlogs(
  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_kms_key`: Associates the specified AWS Key Management Service (AWS KMS) customer master key (CMK) with the specified log group.
- `cancel_export_task`: Cancels the specified export task.
- `create_export_task`: Creates an export task, which allows you to efficiently export data from a log group to an Amazon S3 bucket.
- `create_log_group`: Creates a log group with the specified name.
- `create_log_stream`: Creates a log stream for the specified log group.
- `delete_destination`: Deletes the specified destination, and eventually disables all the subscription filters that publish to it.
- `delete_log_group`: Deletes the specified log group and permanently deletes all the archived log events associated with the log group.
- `delete_log_stream`: Deletes the specified log stream and permanently deletes all the archived log events associated with the log stream.
- `delete_metric_filter`: Deletes the specified metric filter.
- `delete_resource_policy`: Deletes a resource policy from this account.
- `delete_retention_policy`: Deletes the specified retention policy.
- `delete_subscription_filter`: Deletes the specified subscription filter.
- `describe_destinations`: Lists all your destinations.
- `describe_export_tasks`: Lists the specified export tasks.
- `describe_log_groups`: Lists the specified log groups.
- `describe_log_streams`: Lists the log streams for the specified log group.
- `describe_metric_filters`: Lists the specified metric filters.
- `describe_queries`: Returns a list of CloudWatch Logs Insights queries that are scheduled, executing, or have been executed recently in this account.
- `describe_resource_policies`: Lists the resource policies in this account.
- `describe_subscription_filters`: Lists the subscription filters for the specified log group.
- `disassociate_kms_key`: Disassociates the associated AWS Key Management Service (AWS KMS) customer master key (CMK) from the specified log group.
- `filter_log_events`: Lists log events from the specified log group.
- `get_log_events`: Lists log events from the specified log stream.
- `get_log_group_fields`: Returns a list of the fields that are included in log events in the specified log group, along with the percentage of log events that contain each field.
- `get_log_record`: Retrieves all the fields and values of a single log event.
- `get_query_results`: Returns the results from the specified query.
**list_tags_log_group**  Lists the tags for the specified log group
**put_destination**  Creates or updates a destination
**put_destination_policy**  Creates or updates an access policy associated with an existing destination
**put_log_events**  Uploads a batch of log events to the specified log stream
**put_metric_filter**  Creates or updates a metric filter and associates it with the specified log group
**put_resource_policy**  Creates or updates a resource policy allowing other AWS services to put log events to this account
**put_retention_policy**  Sets the retention of the specified log group
**put_subscription_filter**  Creates or updates a subscription filter and associates it with the specified log group
**start_query**  Schedules a query of a log group using CloudWatch Logs Insights
**stop_query**  Stops a CloudWatch Logs Insights query that is in progress
**tag_log_group**  Adds or updates the specified tags for the specified log group
**test_metric_filter**  Tests the filter pattern of a metric filter against a sample of log event messages
**untag_log_group**  Removes the specified tags from the specified log group

---

### Description

Amazon Cognito Federated Identities

Amazon Cognito Federated Identities is a web service that delivers scoped temporary credentials to mobile devices and other untrusted environments. It uniquely identifies a device and supplies the user with a consistent identity over the lifetime of an application.

Using Amazon Cognito Federated Identities, you can enable authentication with one or more third-party identity providers (Facebook, Google, or Login with Amazon) or an Amazon Cognito user pool, and you can also choose to support unauthenticated access from your app. Cognito delivers a unique identifier for each user and acts as an OpenID token provider trusted by AWS Security Token Service (STS) to access temporary, limited-privilege AWS credentials.

For a description of the authentication flow from the Amazon Cognito Developer Guide see [Authentication Flow](#).

For more information see [Amazon Cognito Federated Identities](#).

### Usage

```r
cognitoidentity(config = list())
```
Arguments
config Optional configuration of credentials, endpoint, and/or region.

Service syntax
```
svc <- cognitoidentity(
  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_identity_pool` Creates a new identity pool
- `delete_identities` Deletes identities from an identity pool
- `delete_identity_pool` Deletes an identity pool
- `describe_identity` Returns metadata related to the given identity, including when the identity was created and any associated linked logins
- `describe_identity_pool` Gets details about a particular identity pool, including the pool name, ID description, creation date, and current number of users
- `get_credentials_for_identity` Returns credentials for the provided identity ID
- `get_id` Generates (or retrieves) a Cognito ID
- `get_identity_pool_roles` Gets the roles for an identity pool
- `get_open_id_token` Gets an OpenID token, using a known Cognito ID
- `get_open_id_token_for_developer_identity` Registers (or retrieves) a Cognito IdentityId and an OpenID Connect token for a user authenticated by your backend authentication process
- `list_identities` Lists the identities in an identity pool
- `list_identity_pools` Lists all of the Cognito identity pools registered for your account
- `list_tags_for_resource` Lists the tags that are assigned to an Amazon Cognito identity pool
- `lookup_developer_identity` Retrieves the IdentityID associated with a DeveloperUserIdentifier or the list of DeveloperUserIdentifier values associated with an IdentityId for an existing identity
- `merge_developer_identities` Merges two users having different IdentityIds, existing in the same identity pool
- `set_identity_pool_roles` Sets the roles for an identity pool
- `tag_resource` Assigns a set of tags to an Amazon Cognito identity pool
- `unlink_developer_identity` Unlinks a DeveloperUserIdentifier from an existing identity
- `unlink_identity` Unlinks a federated identity from an existing account
- `untag_resource` Removes the specified tags from an Amazon Cognito identity pool
- `update_identity_pool` Updates an identity pool
Examples

```r
## Not run: svc <- cognitoidentity()
svc$create_identity_pool(
  Foo = 123
)
## End(Not run)
```

---

**cognitoidentityprovider**

*Amazon Cognito Identity Provider*

**Description**

Using the Amazon Cognito User Pools API, you can create a user pool to manage directories and users. You can authenticate a user to obtain tokens related to user identity and access policies.

This API reference provides information about user pools in Amazon Cognito User Pools. For more information, see the Amazon Cognito Documentation.

**Usage**

```r
cognitoidentityprovider(config = list())
```

**Arguments**

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

**Service syntax**

```r
svc <- cognitoidentityprovider(
  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_custom_attributes
admin_add_user_to_group
admin_confirm_sign_up
admin_create_user
admin_delete_user
admin_delete_user_attributes
admin_disable_provider_for_user
admin_disable_user
admin_enable_user
admin_forget_device
admin_get_device
admin_get_user
admin_initiate_auth
admin_link_provider_for_user
admin_list_devices
admin_list_groups_for_user
admin_list_user_auth_events
admin_remove_user_from_group
admin_reset_user_password
admin_respond_to_auth_challenge
admin_set_user_mfa_preference
admin_set_user_password
admin_set_user_settings
admin_update.Auth_event_feedback
admin_update_device_status
admin_update_user_attributes
associate_software_token
change_password
confirm_device
confirm_forgot_password
confirm_sign_up
create_group
create_identity_provider
create_resource_server
create_user_import_job
create_user
create_user_pool
create_user_pool_client
create_user_pool_domain
delete_group
delete_identity_provider
delete_resource_server
delete_user
delete_user_attributes
delete_user_pool
delete_user_pool_client

Adds additional user attributes to the user pool schema
Adds the specified user to the specified group
Confirms user registration as an admin without using a confirmation code
Creates a new user in the specified user pool
Deletes a user as an administrator
Deletes the user attributes in a user pool as an administrator
Disables the user from signing in with the specified external (SAML or social) identity provider
Disables the specified user
Enables the specified user as an administrator
Forgets the device, as an administrator
Gets the device, as an administrator
Gets the specified user by user name in a user pool as an administrator
Initiates the authentication flow, as an administrator
Links an existing user account in a user pool (DestinationUser) to an identity from an external identity provider
Lists the groups that the user belongs to
Lists a history of user activity and any risks detected as part of Amazon Cognito advanced security
Removes the specified user from the specified group
Resets the specified user’s password in a user pool as an administrator
Responds to an authentication challenge, as an administrator
Sets the user’s multi-factor authentication (MFA) preference, including which MFA options are enabled and if any are preferred
Sets the specified user’s password in a user pool as an administrator
This action is no longer supported
Provides feedback for an authentication event as to whether it was from a valid user
Updates the device status as an administrator
Updates the specified user’s attributes, including developer attributes, as an administrator
Signs out users from all devices, as an administrator
Returns a unique generated shared secret key code for the user account
Changes the password for a specified user in a user pool
Confirms tracking of the device
Allows a user to enter a confirmation code to reset a forgotten password
Confirms registration of a user and handles the existing alias from a previous user
Creates a new group in the specified user pool
Creates an identity provider for a user pool
Creates a new OAuth2
Creates the user import job
Creates a new Amazon Cognito user pool and sets the password policy for the pool
Creates the user pool client
Creates a new domain for a user pool
Deletes a group
Deletes an identity provider for a user pool
Deletes a resource server
Allows a user to delete himself or herself
Deletes the attributes for a user
Deletes the specified Amazon Cognito user pool
Allows the developer to delete the user pool client
delete_user_pool_domain
get_csv_header
get_device
get_group
get_identity_provider_by_identifier
get_signing_certificate
get_ui_customization
get_user
get_user_attribute_verification_code
get_user_pool_mfa_config
global_sign_out
initiate_auth
list_devices
list_groups
list_identity_providers
list_resource_servers
list_tags_for_resource
list_user_import_jobs
list_user_pool_clients
list_user_pools
list_users
list_users_in_group
resend_confirmation_code
respond_to_auth_challenge
set_risk_configuration
set_ui_customization
set_user_mfa_preference
set_user_pool_mfa_config
set_user_settings
sign_up
start_user_import_job
stop_user_import_job
tag_resource
untag_resource
update_auth_event_feedback
update_device_status
update_group
update_identity_provider
update_resource_server

Deletes a domain for a user pool
Gets information about a specific identity provider
Describes a resource server
Describes the risk configuration
Describes the user import job
Returns the configuration information and metadata of the specified user pool
Client method for returning the configuration information and metadata of the specified user pool
Gets information about a domain
Forgets the specified device
Calling this API causes a message to be sent to the end user with a confirmation code that
gets the header information for the
gets the device
gets a group
gets the specified identity provider
This method takes a user pool ID, and returns the signing certificate
gets the user attributes and metadata for a user
gets the user attribute verification code for the specified attribute name
gets the user pool multi-factor authentication (MFA) configuration
starts the authentication flow
lists the devices
lists the groups associated with a user pool
lists information about all identity providers for a user pool
lists the resource servers for a user pool
lists the tags that are assigned to an Amazon Cognito user pool
lists the user import jobs
lists the clients that have been created for the specified user pool
lists the user pools associated with an AWS account
lists the users in the Amazon Cognito user pool
lists the users in the specified group
resends the confirmation (for confirmation of registration) to a specific user in the user pool
responds to the authentication challenge
configures actions on detected risks
sets the UI customization information for a user pool’s built-in app UI
set the user’s multi-factor authentication (MFA) method preference, including which MFA factors are enabled and if any are preferred
set the user pool multi-factor authentication (MFA) configuration
set the user pool imported job
starts the user import
stops the user import job
assigns a set of tags to an Amazon Cognito user pool
removes the specified tags from an Amazon Cognito user pool
provides the feedback for an authentication event whether it was from a valid user or not
updates the device status
updates the specified group with the specified attributes
updates identity provider information for a user pool
updates the name and scopes of resource server
cognitosync

Description

Amazon Cognito Sync provides an AWS service and client library that enable cross-device syncing of application-related user data. High-level client libraries are available for both iOS and Android. You can use these libraries to persist data locally so that it’s available even if the device is offline. Developer credentials don’t need to be stored on the mobile device to access the service. You can use Amazon Cognito to obtain a normalized user ID and credentials. User data is persisted in a dataset that can store up to 1 MB of key-value pairs, and you can have up to 20 datasets per user identity.

With Amazon Cognito Sync, the data stored for each identity is accessible only to credentials assigned to that identity. In order to use the Cognito Sync service, you need to make API calls using credentials retrieved with Amazon Cognito Identity service.

If you want to use Cognito Sync in an Android or iOS application, you will probably want to make API calls via the AWS Mobile SDK. To learn more, see the Developer Guide for Android and the Developer Guide for iOS.

Usage

```r
cognitosync(config = list())
```

Arguments

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

Examples

```r
## Not run: svc <- cognitoidentityprovider()
svc$add_custom_attributes(
  Foo = 123
)
## End(Not run)
```
Service syntax

```r
cvc <- cognitosync(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

- `bulk_publish`: Initiates a bulk publish of all existing datasets for an Identity Pool to the configured stream.
- `delete_dataset`: Deletes the specific dataset.
- `describe_dataset`: Gets meta data about a dataset by identity and dataset name.
- `describe_identity_pool_usage`: Gets usage details (for example, data storage) about a particular identity pool.
- `describe_identity_usage`: Gets usage information for an identity, including number of datasets and data usage.
- `get_bulk_publish_details`: Get the status of the last BulkPublish operation for an identity pool.
- `get_cognito_events`: Gets the events and the corresponding Lambda functions associated with an identity pool.
- `get_identity_pool_configuration`: Gets the configuration settings of an identity pool.
- `list_datasets`: Lists datasets for an identity.
- `list_identity_pool_usage`: Gets a list of identity pools registered with Cognito.
- `list_records`: Gets paginated records, optionally changed after a particular sync count for a dataset and identity.
- `register_device`: Registers a device to receive push sync notifications.
- `set_cognito_events`: Sets the AWS Lambda function for a given event type for an identity pool.
- `set_identity_pool_configuration`: Sets the necessary configuration for push sync.
- `subscribe_to_dataset`: Subscribes to receive notifications when a dataset is modified by another device.
- `unsubscribe_from_dataset`: Unsubscribes from receiving notifications when a dataset is modified by another device.
- `update_records`: Posts updates to records and adds and deletes records for a dataset and user.

Examples

```r
## Not run: svc <- cognitosync()
svc$bulk_publish(
  Foo = 123
)
## End(Not run)
```
comprehend  Amazon Comprehend

Description

Amazon Comprehend is an AWS service for gaining insight into the content of documents. Use these actions to determine the topics contained in your documents, the topics they discuss, the predominant sentiment expressed in them, the predominant language used, and more.

Usage

comprehend(config = list())

Arguments

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

Service syntax

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

Operations

batch_detect_dominant_language Determines the dominant language of the input text for a batch of documents
batch_detect_entities Inspects the text of a batch of documents for named entities and returns information about them
batch_detect_key_phrases Detects the key noun phrases found in a batch of documents
batch_detect_sentiment Inspects a batch of documents and returns an inference of the prevailing sentiment
batch_detect_syntax Inspects the text of a batch of documents for the syntax and part of speech of the words in the document
classify_document Creates a new document classification request to analyze a single document in real-time
create_document_classifier Creates a new document classifier that you can use to categorize documents
create_endpoint Creates a model-specific endpoint for synchronous inference for a previously trained custom model
create_entity_recognizer Creates an entity recognizer using submitted files
delete_document_classifier Deletes a previously created document classifier Only those classifiers that are in terminated states (IN_ERROR, TRAINED) will be deleted
delete_endpoint Deletes a model-specific endpoint for a previously-trained custom model
comprehend

delete_entity_recognizer Deletes an entity recognizer
describe_document_classification_job Gets the properties associated with a document classification job
describe_document_classifier Gets the properties associated with a document classifier
describe_dominant_language_detection_job Gets the properties associated with a dominant language detection job
describe_endpoint Gets the properties associated with a specific endpoint
describe_entities_detection_job Gets the properties associated with an entities detection job
describe_entity_recognizer Provides details about an entity recognizer including status, S3 buckets containing training data, recognizer metadata, metrics, and so on
describe_key_phrases_detection_job Gets the properties associated with a key phrases detection job
describe_sentiment_detection_job Gets the properties associated with a sentiment detection job
describe_topic_detection_job Determines the dominant language of the input text
detect_dominant_language Inspects text for named entities, and returns information about them
detect_entities Detects the key noun phrases found in the text
detect_key_phrases Detects the key phrases in the text
detect_sentiment Determines the sentiment of the input text
detect_syntax Inspects text for syntax and the part of speech of words in the document
detect_topic Determines the topic of the input text
list_document_classification_jobs Gets a list of the documentation classification jobs that you have submitted
list_document_classifiers Gets a list of the document classifiers that you have created
list_dominant_language_detection_jobs Gets a list of the dominant language detection jobs that you have submitted
list_endpoints Gets a list of all existing endpoints that you've created
list_entities_detection_jobs Gets a list of the entity detection jobs that you have submitted
list_entity_recognizers Gets a list of the properties of all entity recognizers that you created, including
list_key_phrases_detection_jobs Get a list of key phrase detection jobs that you have submitted
list_sentiment_detection_jobs Gets a list of sentiment detection jobs that you have submitted
list_tags_for_resource Lists all tags associated with a given Amazon Comprehend resource
list_topics_detection_jobs Gets a list of the topic detection jobs that you have submitted
start_document_classification_job Starts an asynchronous document classification job
start_dominant_language_detection_job Starts an asynchronous dominant language detection job for a collection of documents
start_entities_detection_job Starts an asynchronous entity detection job for a collection of documents
start_key_phrases_detection_job Starts an asynchronous key phrase detection job for a collection of documents
start_sentiment_detection_job Starts an asynchronous sentiment detection job for a collection of documents
start_topic_detection_job Starts an asynchronous topic detection job
stop_dominant_language_detection_job Stops a dominant language detection job in progress
stop_entities_detection_job Stops an entities detection job in progress
stop_key_phrases_detection_job Stops a key phrases detection job in progress
stop_sentiment_detection_job Stops a sentiment detection job in progress
stop_training_document_classifier Stops a document classifier training job while in progress
stop_training_entity_recognizer Stops an entity recognizer training job while in progress
untag_resource Associates a specific tag with an Amazon Comprehend resource
untag_resource Removes a specific tag associated with an Amazon Comprehend resource
update_endpoint Updates information about the specified endpoint

Examples

## Not run: svc <- comprehend()
svc$batch_detect_dominant_language(
  Foo = 123
)
comprehendmedical

AWS Comprehend Medical

Description
Amazon Comprehend Medical extracts structured information from unstructured clinical text. Use these actions to gain insight in your documents.

Usage
comprehendmedical(config = list())

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

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

Operations
describe_entities_detection_v2_job Gets the properties associated with a medical entities detection job
describe_phi_detection_job Gets the properties associated with a protected health information (PHI) detection job
detect_entities The DetectEntities operation is deprecated
Inspects the clinical text for a variety of medical entities and returns specific information about them such as entity category, location, and confidence score on that information
detect_entities_v2
infer_icd10cm InferICD10CM detects medical conditions as entities listed in a patient record and links those entities to normalized concept identifiers in the ICD-10-CM knowledge base from the Centers for Disease Control
infer_rx_norm InferRxNorm detects medications as entities listed in a patient record and links to the normalized concept identifiers in the RxNorm database from the National Library of Medicine
list_entities_detection_v2_jobs Gets a list of medical entity detection jobs that you have submitted
list_phi_detection_jobs Gets a list of protected health information (PHI) detection jobs that you have submitted
### start_entities_detection_v2_job
Starts an asynchronous medical entity detection job for a collection of documents

### start_phi_detection_job
Starts an asynchronous job to detect protected health information (PHI)

### stop_entities_detection_v2_job
Stops a medical entities detection job in progress

### stop_phi_detection_job
Stops a protected health information (PHI) detection job in progress

---

### Examples

```r
## Not run: svc <- comprehendmedical()
svc$describe_entities_detection_v2_job(
  Foo = 123
)
## End(Not run)
```

---

### configservice(config = list())

<table>
<thead>
<tr>
<th>configservice</th>
<th>AWS Config</th>
</tr>
</thead>
</table>

### Description

AWS Config provides a way to keep track of the configurations of all the AWS resources associated with your AWS account. You can use AWS Config to get the current and historical configurations of each AWS resource and also to get information about the relationship between the resources. An AWS resource can be an Amazon Compute Cloud (Amazon EC2) instance, an Elastic Block Store (EBS) volume, an elastic network Interface (ENI), or a security group. For a complete list of resources currently supported by AWS Config, see [Supported AWS Resources](#).

You can access and manage AWS Config through the AWS Management Console, the AWS Command Line Interface (AWS CLI), the AWS Config API, or the AWS SDKs for AWS Config. This reference guide contains documentation for the AWS Config API and the AWS CLI commands that you can use to manage AWS Config. The AWS Config API uses the Signature Version 4 protocol for signing requests. For more information about how to sign a request with this protocol, see [Signature Version 4 Signing Process](#). For detailed information about AWS Config features and their associated actions or commands, as well as how to work with AWS Management Console, see [What Is AWS Config](#) in the [AWS Config Developer Guide](#).

### Usage

```r
cfgservice(config = list())
```

### Arguments

<table>
<thead>
<tr>
<th>config</th>
<th>Optional configuration of credentials, endpoint, and/or region.</th>
</tr>
</thead>
</table>
Service syntax

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

Operations

- `batch_get_aggregate_resource_config`: Returns the current configuration items for resources that are present in your AWS Config aggregator.
- `batch_get_resource_config`: Returns the current configuration for one or more requested resources.
- `delete_aggregation_authorization`: Deletes the authorization granted to the specified configuration aggregator account and region.
- `delete_config_rule`: Deletes the specified AWS Config rule and all of its evaluation results.
- `delete_configuration_aggregator`: Deletes the specified configuration aggregator and the aggregated data associated with the aggregator.
- `delete_configuration_recorder`: Deletes the configuration recorder.
- `delete_conformance_pack`: Deletes the specified conformance pack and all the AWS Config rules, remediation actions, and all evaluation results in that conformance pack.
- `delete_delivery_channel`: Deletes the delivery channel.
- `delete_evaluation_results`: Deletes the evaluation results for the specified AWS Config rule.
- `delete_organization_config_rule`: Deletes the specified organization config rule and all of its evaluation results from all member accounts in the organization.
- `delete_organization_conformance_pack`: Deletes the specified organization conformance pack and all the config rules and remediation actions from all member accounts in that organization.
- `delete_pending_aggregation_request`: Deletes pending authorization requests for a specified aggregator account and region.
- `delete_remediation_configuration`: Deletes the remediation configuration.
- `delete_remediation_exceptions`: Deletes one or more remediation exceptions mentioned in the resource keys.
- `delete_resource_config`: Records the configuration state for a custom resource that has been deleted.
- `delete_retention_configuration`: Deletes the retention configuration.
- `deliver_config_snapshot`: Schedules delivery of a configuration snapshot to the Amazon S3 bucket.
- `describe_aggregate_compliance_by_config_rules`: Returns a list of compliant and noncompliant rules with the number of resources for each rule.
- `describe_aggregation_authorizations`: Returns a list of authorizations granted to various aggregator accounts.
- `describe_compliance_by_config_rule`: Indicates whether the specified AWS Config rules are compliant.
- `describe_compliance_by_resource`: Indicates whether the specified AWS resources are compliant.
- `describe_config_rule_evaluation_status`: Returns status information for each of your AWS managed Config rules.
- `describe_config_rules`: Returns details about your AWS Config rules.
- `describe_configuration_aggregators`: Returns the details of one or more configuration aggregators.
- `describe_configuration_aggregator_sources_status`: Returns status information for sources within an aggregator.
- `describe_configuration_recorders`: Returns the details for the specified configuration recorders.
- `describe_configuration_recorder_status`: Returns the current status of the specified configuration recorder.
- `describe_conformance_pack_compliance`: Returns compliance details for each rule in that conformance pack.
- `describe_conformance_packs`: Returns a list of one or more conformance packs.
- `describe_conformance_pack_status`: Provides one or more conformance packs deployment status.
describe_delivery_channels
describe_delivery_channel_status
describe_organization_config_rules
describe_organization_config_rule_statuses
describe_organization_conformance_packs
describe_organization_conformance_pack_statuses
describe_pending_aggregation_requests
describe_remediation_configurations
describe_remediation_exception
describe_remediation_execution_status
describe_retention_configurations
get_aggregate_compliance_details_by_config_rule
get_aggregate_config_rule_compliance_summary
get_aggregate_discovered_resource_counts
get_aggregate_resource_config
get_compliance_details_by_config_rule
get_compliance_details_by_resource
get_compliance_summary_by_config_rule
get_compliance_summary_by_resource_type
get_conformance_pack_compliance_details
get_conformance_pack_compliance_summary
get_discovered_resource_counts
get_organization_config_rule_detailed_status
get_organization_conformance_pack_detailed_status
get_resource_config_history
list_aggregate_discovered_resources
list_discovered_resources
list_tags_for_resource
put_aggregation_authorization
put_config_rule
put_configuration_aggregator
put_configuration_recorder
put_conformance_pack
put_delivery_channel
put_evaluations
put_organization_config_rule
put_organization_conformance_pack
put_remediation_configurations
put_remediation_exceptions
put_resource_config
put_retention_configuration
select_resource_config
start_config_rules_evaluation
start_configuration_recorder
start_remediation_execution
stop_configuration_recorder
tag_resource
untag_resource

Returns details about the specified delivery channel
Returns the current status of the specified delivery channel
Returns a list of organization config rules
Provides organization config rule deployment status for an organization
Returns a list of organization conformance packs
Provides organization conformance pack deployment status for an organization
Returns the details of one or more remediation configurations
Returns the details of one or more remediation exceptions
Provides a detailed view of a Remediation Execution for a set of resources
Returns the details of one or more retention configurations
Returns the evaluation results for the specified AWS Config rule for a specific resource
Returns the number of compliant and noncompliant rules for one or more config rules
Returns the resource counts across accounts and regions that are present in your AWS Config aggregator
Returns configuration item that is aggregated for your specific resource
Returns the evaluation results for the specified AWS Config resource
Returns the number of AWS Config rules that are compliant and noncompliant
Returns the number of resources that are compliant and the number that are noncompliant
Returns compliance details of a conformance pack for all AWS resources
Returns compliance details for the conformance pack based on the current state of the resources
Returns the resource types, the number of each resource type, and the number of resources
Returns detailed status for each member account within an organization
Returns detailed status for each member account within an organization
Returns a list of configuration items for the specified resource
Accepts a resource type and returns a list of resource identifiers that are aggregated
Accepts a resource type and returns a list of resource identifiers for the specific account
List the tags for AWS Config resource
Authors the aggregator account and region to collect data from the specified source
Adds or updates an AWS Config rule for evaluating whether your AWS resources comply with your desired configurations
Creates and updates the configuration aggregator with the selected source accounts and regions
Creates a new configuration recorder to record the selected resource configurations
Creates or updates a conformance pack
Creates a delivery channel object to deliver configuration information to a delivery channel
Used by an AWS Lambda function to deliver evaluation results to AWS Config
Add or update organization config rule for your entire organization or for a specific AWS Config rule
Deploys conformance packs across member accounts in an AWS Organization
Add or update the remediation configuration with a specific AWS Config rule
A remediation exception is when a specific resource is no longer considered for auto-remediation
Records the configuration state for the resource provided in the request
Accepts a structured query language (SQL) SELECT command, performs the corresponding search, and returns resource configurations matching the properties
Runs an on-demand evaluation for the specified AWS Config rules against the last known remediation configuration
Starts recording configurations of the AWS resources you have selected
Runs an on-demand remediation for the specified AWS Config rules against the last known configuration state of the resources
Stop recording configurations of the AWS resources you have selected
Associates the specified tags to a resource with the specified resourceArn
Deletes specified tags from a resource
Examples

## Not run: svc <- configservice()
svc$batch_get_aggregate_resource_config(
  Foo = 123
)
## End(Not run)

---

**connect**

**Amazon Connect Service**

**Description**

Amazon Connect is a cloud-based contact center solution that makes it easy to set up and manage a customer contact center and provide reliable customer engagement at any scale.

Amazon Connect provides rich metrics and real-time reporting that allow you to optimize contact routing. You can also resolve customer issues more efficiently by putting customers in touch with the right agents.

There are limits to the number of Amazon Connect resources that you can create and limits to the number of requests that you can make per second. For more information, see Amazon Connect Service Limits in the Amazon Connect Administrator Guide.

**Usage**

```r
connect(config = list())
```

**Arguments**

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

**Service syntax**

```r
svc <- connect(
  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_user: Creates a user account for the specified Amazon Connect instance
- delete_user: Deletes a user account from the specified Amazon Connect instance
- describe_user: Describes the specified user account
- describe_user_hierarchy_group: Describes the specified hierarchy group
- describe_user_hierarchy_structure: Describes the hierarchy structure of the specified Amazon Connect instance
- get_contact_attributes: Retrieves the contact attributes for the specified contact
- get_current_metric_data: Gets the real-time metric data from the specified Amazon Connect instance
- get_federation_token: Retrieves a token for federation
- get_metric_data: Gets historical metric data from the specified Amazon Connect instance
- list_contact_flows: Provides information about the contact flows for the specified Amazon Connect instance
- list_hours_of_operations: Provides information about the hours of operation for the specified Amazon Connect instance
- list_phone_numbers: Provides information about the phone numbers for the specified Amazon Connect instance
- list_queues: Provides information about the queues for the specified Amazon Connect instance
- list_routing_profiles: Provides summary information about the routing profiles for the specified Amazon Connect instance
- list_security_profiles: Provides summary information about the security profiles for the specified Amazon Connect instance
- list_tags_for_resource: Lists the tags for the specified resource
- list_user_hierarchy_groups: Provides summary information about the hierarchy groups for the specified Amazon Connect instance
- list_users: Provides summary information about the users for the specified Amazon Connect instance
- start_chat_contact: Initiates a contact flow to start a new chat for the customer
- start_outbound_voice_contact: Initiates a contact flow to place an outbound call to a customer
- stop_contact: Ends the specified contact
- tag_resource: Adds the specified tags to the specified resource
- untag_resource: Removes the specified tags from the specified resource
- update_contact_attributes: Creates or updates the contact attributes associated with the specified contact
- update_user_hierarchy: Assigns the specified hierarchy group to the specified user
- update_user_identity_info: Updates the identity information for the specified user
- update_user_phone_config: Updates the phone configuration settings for the specified user
- update_user_routing_profile: Assigns the specified routing profile to the specified user
- update_user_security_profiles: Assigns the specified security profiles to the specified user

Examples

```r
## Not run: svc <- connect()
svc$create_user(
  Foo = 123
)
## End(Not run)
```

---

AWS Cost and Usage Report Service
Description

The AWS Cost and Usage Report API enables you to programmatically create, query, and delete AWS Cost and Usage report definitions.

AWS Cost and Usage reports track the monthly AWS costs and usage associated with your AWS account. The report contains line items for each unique combination of AWS product, usage type, and operation that your AWS account uses. You can configure the AWS Cost and Usage report to show only the data that you want, using the AWS Cost and Usage API.

Service Endpoint

The AWS Cost and Usage Report API provides the following endpoint:

- cur.us-east-1.amazonaws.com

Usage

costandusagereportservice(config = list())

Arguments

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

Service syntax

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

Operations

describe_report_definitions Lists the AWS Cost and Usage reports available to this account
modify_report_definition Allows you to programatically update your report preferences
put_report_definition Creates a new report using the description that you provide
delete_report_definition Deletes the specified report
Examples

```r
# The following example deletes the AWS Cost and Usage report named
# ExampleReport.
## Not run: svc <- costandusagereportservice()
svc$delete_report_definition(
  ReportName = "ExampleReport"
)
## End(Not run)
```

Description

The Cost Explorer API enables you to programmatically query your cost and usage data. You can query for aggregated data such as total monthly costs or total daily usage. You can also query for granular data, such as the number of daily write operations for Amazon DynamoDB database tables in your production environment.

Service Endpoint

The Cost Explorer API provides the following endpoint:

- https://ce.us-east-1.amazonaws.com

For information about costs associated with the Cost Explorer API, see AWS Cost Management Pricing.

Usage

```r
costexplorer(config = list())
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

Service syntax

```r
csvc <- costexplorer(
  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_cost_category_definition
delete_cost_category_definition
describe_cost_category_definition
get_cost_and_usage
get_cost_and_usage_with_resources
get_cost_forecast
get_dimension_values
get_reservation_coverage
get_reservation_purchase_recommendation
get_reservation_utilization
get_rightsizing_recommendation
get_savings_plans_coverage
get_savings_plans_purchase_recommendation
get_savings_plans_utilization
get_savings_plans_utilization_details
get_tags
get_usage_forecast
list_cost_category_definitions
update_cost_category_definition

Examples

## Not run: svc <- costexplorer()
svc$create_cost_category_definition(
  Foo = 123
)
## End(Not run)

### Description

AWS Data Pipeline configures and manages a data-driven workflow called a pipeline. AWS Data Pipeline handles the details of scheduling and ensuring that data dependencies are met so that your application can focus on processing the data.

AWS Data Pipeline provides a JAR implementation of a task runner called AWS Data Pipeline Task Runner. AWS Data Pipeline Task Runner provides logic for common data management scenarios,
AWS Data Pipeline implements two main sets of functionality. Use the first set to create a pipeline and define data sources, schedules, dependencies, and the transforms to be performed on the data. Use the second set in your task runner application to receive the next task ready for processing. The logic for performing the task, such as querying the data, running data analysis, or converting the data from one format to another, is contained within the task runner. The task runner performs the task assigned to it by the web service, reporting progress to the web service as it does so. When the task is done, the task runner reports the final success or failure of the task to the web service.

**Usage**

```r
datapipeline(config = list())
```

**Arguments**

- `config` (optional) configuration of credentials, endpoint, and/or region.

**Service syntax**

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

**Operations**

- `activate_pipeline` Validates the specified pipeline and starts processing pipeline tasks
- `add_tags` Adds or modifies tags for the specified pipeline
- `create_pipeline` Creates a new, empty pipeline
- `deactivate_pipeline` Deactivates the specified running pipeline
- `delete_pipeline` Deletes a pipeline, its pipeline definition, and its run history
- `describe_objects` Gets the object definitions for a set of objects associated with the pipeline
- `describe_pipelines` Retrieves metadata about one or more pipelines
- `evaluate_expression` Task runners call EvaluateExpression to evaluate a string in the context of the specified object
- `get_pipeline_definition` Gets the definition of the specified pipeline
- `list_pipelines` Lists the pipeline identifiers for all active pipelines that you have permission to access
- `poll_for_task` Task runners call PollForTask to receive a task to perform from AWS Data Pipeline
Examples

## Not run: svc <- datapipeline()
svc$activate_pipeline(
  Foo = 123
)
## End(Not run)

---

Amazon DynamoDB Accelerator (DAX)

Description

DAX is a managed caching service engineered for Amazon DynamoDB. DAX dramatically speeds up database reads by caching frequently-accessed data from DynamoDB, so applications can access that data with sub-millisecond latency. You can create a DAX cluster easily, using the AWS Management Console. With a few simple modifications to your code, your application can begin taking advantage of the DAX cluster and realize significant improvements in read performance.

Usage

dax(config = list())

Arguments

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

Service syntax

svc <- dax(
  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_cluster            Creates a DAX cluster
create_parameter_group   Creates a new parameter group
create_subnet_group      Creates a new subnet group
decrease_replication_factor Removes one or more nodes from a DAX cluster
delete_cluster           Deletes a previously provisioned DAX cluster
delete_parameter_group   Deletes the specified parameter group
delete_subnet_group      Deletes a subnet group
describe_clusters        Returns information about all provisioned DAX clusters if no cluster identifier is specified, or about a specific DAX cluster if a cluster identifier is supplied
describe_default_parameters Returns the default system parameter information for the DAX caching software
describe_events          Returns events related to DAX clusters and parameter groups
describe_parameter_groups Returns a list of parameter group descriptions
describe_parameters      Returns the detailed parameter list for a particular parameter group
describe_subnet_groups   Returns a list of subnet group descriptions
increase_replication_factor Adds one or more nodes to a DAX cluster
list_tags                 List all of the tags for a DAX cluster
reboot_node               Reboots a single node of a DAX cluster
tag_resource              Associates a set of tags with a DAX resource
untag_resource            Removes the association of tags from a DAX resource
update_cluster            Modifies the settings for a DAX cluster
update_parameter_group    Modifies the parameters of a parameter group
update_subnet_group       Modifies an existing subnet group

Examples

## Not run: svc <- dax()
svc$create_cluster(
    Foo = 123
)
## End(Not run)
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
associate_hosted_connection
associate_virtual_interface
confirm_connection
confirm_private_virtual_interface
confirm_public_virtual_interface

Accepts a proposal request to attach a virtual private gateway or transit virtual interface to a Direct Connect gateway

Deprecated

Creates a hosted connection on the specified interconnect or a link aggregation group (LAG)

Provisions a private virtual interface to be owned by the specified AWS account

Provisions a public virtual interface to be owned by the specified AWS account

Provisions a transit virtual interface to be owned by the specified AWS account

Associates an existing connection with a link aggregation group (LAG)

Associates a hosted connection and its virtual interfaces with a link aggregation group

Associates a virtual interface with a specified link aggregation group

Confirms the creation of the specified hosted connection on an interconnect

Accepts ownership of a private virtual interface created by another AWS account

Accepts ownership of a public virtual interface created by another AWS account
### Examples

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

AWS Directory Service

Description

AWS Directory Service is a web service that makes it easy for you to setup and run directories in the AWS cloud, or connect your AWS resources with an existing on-premises Microsoft Active Directory. This guide provides detailed information about AWS Directory Service operations, data types, parameters, and errors. For information about AWS Directory Services features, see AWS Directory Service and the AWS Directory Service Administration Guide.

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWS Directory Service and other AWS services. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

Usage

directoryservice(config = list())

Arguments

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

Service syntax

svc <- directoryservice(  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_shared_directory
add_ip_routes
add_tags_to_resource
cancel_schema_extension
connect_directory
create_alias
create_computer
create_conditional_forwarder
create_directory
create_log_subscription
create_microsoft_ad
create_snapshot
create_trust
delete_conditional_forwarder
delete_directory
delete_log_subscription
delete_snapshot
delete_trust
deregister_certificate
deregister_event_topic
describe_certificate
describe_conditional_forwarders
describe_directories
describe_domain_controllers
describe_event_topics
describe_ldaps_settings
describe_shared_directories
describe_snapshots
describe_trusts
disable_ldaps
disable_radius
disable_sso
enable_ldaps
enable_radius
enable_sso
get_directory_limits
get_snapshot_limits
list_certificates
list_ip_routes
list_log_subscriptions
list_schema_extensions
list_tags_for_resource
register_certificate
register_event_topic
reject_shared_directory
remove_ip_routes
remove_tags_from_resource
reset_user_password

Accepts a directory sharing request that was sent from the directory owner account.
If the DNS server for your on-premises domain uses a publicly addressable IP address,
adds or overwrites one or more tags for the specified directory.
Cancels an in-progress schema extension to a Microsoft AD directory.
Creates an AD Connector to connect to an on-premises directory.
Creates an alias for a directory and assigns the alias to the directory.
Creates a computer account in the specified directory, and joins the computer to the directory.
Creates a conditional forwarder associated with your AWS directory.
Creates a Simple AD directory.
Creates a subscription to forward real-time Directory Service domain controller security logs.
Creates a Microsoft AD directory in the AWS Cloud.
Creates a snapshot of a Simple AD or Microsoft AD directory in the AWS Cloud.
AWS Directory Service for Microsoft Active Directory allows you to configure trust relationships.
Deletes a conditional forwarder that has been set up for your AWS directory.
Deletes an AWS Directory Service directory.
Deletes the specified log subscription.
Deletes a directory snapshot.
Deletes an existing trust relationship between your AWS Managed Microsoft AD directory and an external directory.
Deletes from the system the certificate that was registered for a secured LDAP connection.
Removes the specified directory as a publisher to the specified SNS topic.
Displays information about the certificate registered for a secured LDAP connection.
Obtains information about the conditional forwarders for this account.
Obtains information about the directories that belong to this account.
Provides information about any domain controllers in your directory.
Obtains information about which SNS topics receive status messages from the specified directory.
Describes the status of LDAP security for the specified directory.
Returns the shared directories in your account.
Obtains information about the directory snapshots that belong to this account.
Obtains information about the trust relationships for this account.
Deactivates LDAP secure calls for the specified directory.
Disables multi-factor authentication (MFA) with the Remote Authentication Dial In User Service (RADIUS) for an AD Connector or Microsoft AD directory.
Disables single-sign on for a directory.
Activates the switch for the specific directory to always use LDAP secure calls.
Enables multi-factor authentication (MFA) with the Remote Authentication Dial In User Service (RADIUS) for an AD Connector or Microsoft AD directory.
Enables single-sign on for a directory.
Obtains directory limit information for the current Region.
Obtains the manual snapshot limits for a directory.
For the specified directory, lists all the certificates registered for a secured LDAP connection.
Lists the address blocks that you have added to a directory.
Lists the active log subscriptions for the AWS account.
Lists all schema extensions applied to a Microsoft AD Directory.
Lists all tags on a directory.
Registers a certificate for secured LDAP connection.
Associates a directory with an SNS topic.
Rejects a directory sharing request that was sent from the directory owner account.
Removes IP address blocks from a directory.
Removes tags from a directory.
Resets the password for any user in your AWS Managed Microsoft AD or Simple AD directory.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>restore_from_snapshot</td>
<td>Restores a directory using an existing directory snapshot</td>
</tr>
<tr>
<td>share_directory</td>
<td>Shares a specified directory (DirectoryId) in your AWS account (directory owner) with another AWS account (directory consumer)</td>
</tr>
<tr>
<td>start_schema_extension</td>
<td>Applies a schema extension to a Microsoft AD directory</td>
</tr>
<tr>
<td>unshare_directory</td>
<td>Stops the directory sharing between the directory owner and consumer accounts</td>
</tr>
<tr>
<td>update_conditional_forwarder</td>
<td>Updates a conditional forwarder that has been set up for your AWS directory</td>
</tr>
<tr>
<td>update_number_of_domain_controllers</td>
<td>Adds or removes domain controllers to or from the directory</td>
</tr>
<tr>
<td>update_radius</td>
<td>Updates the Remote Authentication Dial In User Service (RADIUS) server information for an AD Connector or Microsoft AD directory</td>
</tr>
<tr>
<td>update_trust</td>
<td>Updates the trust that has been set up between your AWS Managed Microsoft AD directory and an on-premises Active Directory</td>
</tr>
<tr>
<td>verify_trust</td>
<td>Checks the trust that has been set up between your AWS Managed Microsoft AD directory and an on-premises Active Directory</td>
</tr>
</tbody>
</table>

Examples

```
## Not run: svc <- directoryservice()
svc$accept_shared_directory(
  Foo = 123
)
## End(Not run)
```

---

**dlm**  

*Amazon Data Lifecycle Manager*

---

**Description**

With Amazon Data Lifecycle Manager, you can manage the lifecycle of your AWS resources. You create lifecycle policies, which are used to automate operations on the specified resources.

Amazon DLM supports Amazon EBS volumes and snapshots. For information about using Amazon DLM with Amazon EBS, see Automating the Amazon EBS Snapshot Lifecycle in the Amazon EC2 User Guide.

**Usage**

```r
dlm(config = list())
```

**Arguments**

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

**Service syntax**

```r
svc <- dlm(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```
Operations

create_lifecycle_policy  Creates a policy to manage the lifecycle of the specified AWS resources
delete_lifecycle_policy Deletes the specified lifecycle policy and halts the automated operations that the policy specified
get_lifecycle_policies  Gets summary information about all or the specified data lifecycle policies
get_lifecycle_policy    Gets detailed information about the specified lifecycle policy
list_tags_for_resource Lists the tags for the specified resource
tag_resource            Adds the specified tags to the specified resource
untag_resource          Removes the specified tags from the specified resource
update_lifecycle_policy Updates the specified lifecycle policy

Examples

```r
## Not run: svc <- dlm()
svc$create_lifecycle_policy(
  Foo = 123
)
## End(Not run)
```

Description

Amazon DocumentDB API documentation

Usage

docdb(config = list())

Arguments

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

```r
csvc <- docdb(
  config = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

- `add_tags_to_resource`: Adds metadata tags to an Amazon DocumentDB resource
- `apply_pending_maintenance_action`: Applies a pending maintenance action to a resource (for example, to a DB instance)
- `copy_db_cluster_parameter_group`: Copies the specified DB cluster parameter group
- `copy_db_cluster_snapshot`: Copies a snapshot of a DB cluster
- `create_db_cluster`: Creates a new Amazon DocumentDB DB cluster
- `create_db_cluster_parameter_group`: Creates a new DB cluster parameter group
- `create_db_cluster_snapshot`: Creates a snapshot of a DB cluster
- `create_db_instance`: Creates a new DB instance
- `create_db_subnet_group`: Creates a new DB subnet group
- `delete_db_cluster`: Deletes a previously provisioned DB cluster
- `delete_db_cluster_parameter_group`: Deletes a specified DB cluster parameter group
- `delete_db_cluster_snapshot`: Deletes a DB cluster snapshot
- `delete_db_instance`: Deletes a previously provisioned DB instance
- `delete_db_subnet_group`: Deletes a DB subnet group
- `describe_certificates`: Returns a list of certificate authority (CA) certificates provided by Amazon RDS for this AWS account
- `describe_db_cluster_parameter_groups`: Returns a list of DBClusterParameterGroup descriptions
- `describe_db_cluster_parameters`: Returns the detailed parameter list for a particular DB cluster parameter group
- `describe_db_clusters`: Returns information about provisioned Amazon DocumentDB DB clusters
- `describe_db_cluster_snapshot_attributes`: Returns a list of DB cluster snapshot attribute names and values for a manual DB cluster snapshot
- `describe_db_cluster_snapshots`: Returns information about DB cluster snapshots
- `describe_db_engine_versions`: Returns a list of the available DB engines
- `describe_db_instances`: Returns a list of the available DB instances
- `describe_db_subnet_groups`: Returns information about provisioned Amazon DocumentDB instances
- `describe_engine_default_cluster_parameters`: Returns the default engine and system parameter information for the cluster database engine
- `describe_event_categories`: Displays a list of categories for all event source types, or, if specified, for a specific source type
- `describe_events`: Returns events related to DB instances, DB security groups, and DB snapshots, and displays a list of categories for all event source types, or, if specified, for a specific source type
- `describe_orderable_db_instance_options`: Returns a list of orderable DB instance options for the specified engine
- `describe_pending_maintenance_actions`: Returns a list of resources (for example, DB instances) that have at least one pending maintenance action
- `failover_db_cluster`: Forces a failover for a DB cluster
- `list_tags_for_resource`: Lists all tags on an Amazon DocumentDB resource
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don’t have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic. You can scale up or scale down your tables’ throughput capacity without downtime or performance degradation, and use the AWS Management Console to monitor resource utilization and performance metrics.

DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements, while maintaining consistent and fast performance. All of your data is stored on solid state disks (SSDs) and automatically replicated across multiple Availability Zones in an AWS region, providing built-in high availability and data durability.

Usage

```r
dynamodb(config = list())
```
Arguments

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

Service syntax

dynamodb

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

Operations

batch_get_item The BatchGetItem operation returns the attributes of one or more items from one or more tables
batch_write_item The BatchWriteItem operation puts or deletes multiple items in one or more tables
create_backup Creates a backup for an existing table
create_global_table Creates a global table from an existing table
create_table The CreateTable operation adds a new table to your account
delete_backup Deletes an existing backup of a table
delete_item Deletes a single item in a table by primary key
delete_table The DeleteTable operation deletes a table and all of its items
delete_table_replica_auto_scaling Deletes an existing backup of a table
describe_backup Checks the status of continuous backups and point in time recovery on the specified table
describe_contributor_insights Returns information about contributor insights, for a given table or global secondary index
describe_continuous_backups Returns the regional endpoint information
describe_endpoints Returns information about the specified global table
describe_global_table Returns information about the table, including the current status of the table, when it was created, the primary key schema, and any indexes on the table
describe_global_table_settings Describes Region-specific settings for a global table
describe_limits Describes an existing backup of a table
describe_table Describes auto scaling settings across replicas of the global table at once
describe_table_replica_auto_scaling Gives a description of the Time to Live (TTL) status on the specified table
describe_time_to_live The GetItem operation returns a set of attributes for the item with the given primary key
describe_table_replica_auto_scaling List backups associated with an AWS account
describe_table_replica_auto_scaling List backups associated with an AWS account
desc_list_tags_of_resource List all global tables that have a replica in the specified Region
describe_table_replica_auto_scaling Returns a list of ContributorInsightsSummary for a table and all its global secondary indexes
describe_table_replica_auto_scaling Returns an array of table names associated with the current account and endpoint
describe_table_replica_auto_scaling Creates a new item, or replaces an old item with a new item
dynamodb

- `restore_table_from_backup`: Creates a new table from an existing backup
- `restore_table_to_point_in_time`: Restores the specified table to the specified point in time within EarliestRestorableDateTime and LatestRestorableDateTime
- `scan`: The Scan operation returns one or more items and item attributes by accessing every item in a table or a secondary index
- `tag_resource`: Associate a set of tags with an Amazon DynamoDB resource
- `transact_get_items`: TransactGetItems is a synchronous operation that atomically retrieves multiple items from DynamoDB.
- `transact_write_items`: TransactWriteItems is a synchronous write operation that groups up to 25 action requests.
- `untag_resource`: Removes the association of tags from an Amazon DynamoDB resource
- `update_continuous_backups`: UpdateContinuousBackups enables or disables point in time recovery for the specified table.
- `update_contributor_insights`: Updates the status for contributor insights for a specific table or index
- `update_global_table`: Adds or removes replicas in the specified global table
- `update_global_table_settings`: Updates settings for a global table
- `update_item`: Edits an existing item’s attributes, or adds a new item to the table if it does not already exist.
- `update_table`: Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams settings for a given table.
- `update_table_replica_auto_scaling`: Updates auto scaling settings on your global tables at once
- `update_time_to_live`: The UpdateTimeToLive method enables or disables Time to Live (TTL) for the specified table.

Examples

```r
# This example reads multiple items from the Music table using a batch of three GetItem requests. Only the AlbumTitle attribute is returned.
# Not run: svc <- dynamodb()
svc$batch_get_item(
  RequestItems = list(
    Music = list(
      Keys = list(
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Call Me Today"
          )
        ),
        list(
          Artist = list(
            S = "Acme Band"
          ),
          SongTitle = list(
            S = "Happy Day"
          )
        ),
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Scared of My Shadow"
          )
        )
      )
    )
  )
)
```

```
dynamodbstreams
Amazon DynamoDB Streams

Description
Amazon DynamoDB Streams provide API actions for accessing streams and processing stream records. To learn more about application development with Streams, see Capturing Table Activity with DynamoDB Streams in the Amazon DynamoDB Developer Guide.

Usage
dynamodbstreams(config = list())

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

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

Operations
describe_stream Returns information about a stream, including the current status of the stream, its Amazon Resource Name (ARN), and its corresponding DynamoDB table
get_records Retrieves the stream records from a given shard
get_shard_iterator Returns a shard iterator
list_streams Returns an array of stream ARNs associated with the current account and endpoint
Examples

# The following example describes a stream with a given stream ARN.
## Not run: svc <- dynamodbstreams()
svc$describe_stream(
)
## End(Not run)

ec2

Amazon Elastic Compute Cloud

Description

Amazon Elastic Compute Cloud (Amazon EC2) provides secure and resizable computing capacity in the AWS cloud. Using Amazon EC2 eliminates the need to invest in hardware up front, so you can develop and deploy applications faster.

To learn more, see the following resources:

- Amazon EC2: AmazonEC2 product page, Amazon EC2 documentation
- Amazon EBS: Amazon EBS product page, Amazon EBS documentation
- Amazon VPC: Amazon VPC product page, Amazon VPC documentation
- AWS VPN: AWS VPN product page, AWS VPN documentation

Usage

ec2(config = list())

Arguments

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

Service syntax

svc <- ec2(
    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_reserved_instances_exchange_quote
accept_transit_gateway_peering_attachment
accept_transit_gateway_vpc_attachment
accept_vpc_endpoint_connections
accept_vpc_peering_connection
advertise_byoip_cidr
allocate_address
allocate_hosts
apply_security_groups_to_client_vpn_target_network
assign_ipv6_addresses
assign_private_ip_addresses
associate_address
associate_client_vpn_target_network
associate_dhcp_options
associate_iam_instance_profile
associate_route_table
associate_subnet_cidr_block
associate_transit_gateway_multicast_domain
associate_transit_gateway_route_table
associate_vpc_cidr_block
attach_classic_link_vpc
attach_internet_gateway
attach_network_interface
attach_volume
attach_vpn_gateway
authorize_client_vpn_ingress
authorize_security_group_egress
authorize_security_group_ingress
bundle_instance
cancel_bundle_task
cancel_capacity_reservation
cancel_conversion_task
cancel_export_task
cancel_import_task
cancel_reserved_instances_listing
cancel_spot_fleet_requests
cancel_spot_instance_requests
confirm_product_instance
copy_fpga_image
copy_image
copy_snapshot

Accepts the Convertible Reserved Instance exchange quote.
Accepts a transit gateway peering attachment request.
Accepts a request to attach a VPC to a transit gateway.
Accepts one or more VPC endpoint connections.
Accept a VPC peering connection request.
Advertises an IPv4 address range that is provisioned for use with your AWS resources.
Allocates an Elastic IP address to your AWS account.
Allocates a Dedicated Host to your account.
Applies a security group to the association between an instance.
Assigns one or more IPv6 addresses to the specified subnet.
Assigns one or more secondary private IP addresses to an instance.
Associates an Elastic IP address with an instance in your VPC.
Associates a target network with a Client VPN endpoint.
Associates a set of DHCP options (that you’ve previously created) with a VPC.
Associates an IAM instance profile with a running or stopped EC2 instance.
Associates a subnet in your VPC or an internet gateway.
Associates a CIDR block with your subnet.
Associates the specified subnets and transit gateways.
Associates the specified attachment with the specified VPC.
Links an EC2-Classic instance to a ClassicLink-enabled VPC.
Attaches an internet gateway or a virtual private gateway.
Attaches a network interface to an instance.
Attaches an EBS volume to a running or stopped instance.
Attaches a virtual private gateway to a VPC.
Adds an ingress authorization rule to a Client VPN endpoint.
[VPC only] Adds the specified egress rules to a security group.
Bundles an Amazon instance store-backed Window.
Cancels a bundling operation for an instance store.
Cancels the specified Capacity Reservation, releasing the resources.
Cancels an active conversion task.
Cancels an active export task.
Cancels an in-process import virtual machine or snapshot.
Cancels the specified Reserved Instance listing in the Marketplace.
Cancels the specified Spot Fleet requests.
Cancels one or more Spot Instance requests.
Determines whether a product code is associated with an instance.
Copies the specified Amazon FPGA Image (AFI).
Initiates the copy of an AMI from the specified source.
Copies a point-in-time snapshot of an EBS volume.
create_capacity_reservation
create_client_vpn_endpoint
create_client_vpn_route
create_customer_gateway
create_default_subnet
create_default_vpc
create_dhcp_options
create_egress_only_internet_gateway
create_fleet
create_flow_logs
create_fpga_image
create_image
create_instance_export_task
create_internet_gateway
create_key_pair
create_launch_template
create_launch_template_version
create_local_gateway_route
create_local_gateway_route_table_vpc_association
create_nat_gateway
create_network_acl
create_network_acl_entry
create_network_interface
create_network_interface_permission
create_placement_group
create_reserved_instances_listings
create_route
create_route_table
create_security_group
create_snapshot
create_snapshots
create_spot_datafeed_subscription
create_subnet
create_tags
create_traffic_mirror_filter
create_traffic_mirror_filter_rule
create_traffic_mirror_session
create_traffic_mirror_target
create_transit_gateway
create_transit_gateway_multicast_domain
create_transit_gateway_peering_attachment
create_transit_gateway_route
create_transit_gateway_route_table
create_transit_gateway_vpc_attachment
create_volume
create_vpc
create_vpc_endpoint
create_vpc_endpoint_connection_notification

Creates a new Capacity Reservation with the specified attributes.
Creates a Client VPN endpoint.
Adds a route to a network to a Client VPN endpoint.
Provides information to AWS about your VPN customer gateway device.
Creates a default subnet with a size /20 IPv4 CIDR block.
Creates a default VPC with a size /16 IPv4 CIDR block.
Creates a set of DHCP options for your VPC.
[IPv6 only] Creates an egress-only internet gateway.
Launches an EC2 Fleet.
Creates one or more flow logs to capture information.
Creates an Amazon FPGA Image (AFI) from the specified design.
Creates an Amazon EBS-backed AMI from an Amazon EC2 instance.
Exports a running or stopped instance to an S3 bucket.
Creates an internet gateway for use with a VPC.
Creates a 2048-bit RSA key pair with the specified name.
Creates a launch template.
Creates a new version for a launch template.
Creates a static route for the specified local gateway route table.
Associates the specified VPC with the specified local gateway route table.
Creates a NAT gateway in the specified public subnet.
Creates a network ACL in a VPC.
Creates an entry (a rule) in a network ACL with the specified properties.
Creates a network interface in the specified subnet.
Grants an AWS-authorized account permission to launch instances in a VPC.
Creates a placement group in which to launch instances.
Creates a listing for Amazon EC2 Standard Reserved Instances.
Creates a route in a route table within a VPC.
Creates a route table for the specified VPC.
Creates a security group.
Creates a snapshot of an EBS volume and stores it in Amazon S3.
Creates crash-consistent snapshots of multiple EBS volumes.
Creates a data feed for Spot Instances, enabling you to view Spot Instance usage logs.
Creates a subnet in an existing VPC.
Adds or overwrites the specified tags for the specified Amazon EC2 resource.
Creates a Traffic Mirror filter.
Creates a Traffic Mirror filter rule.
Creates a Traffic Mirror session.
Creates a target for your Traffic Mirror session.
Creates a transit gateway.
Creates a multicast domain using the specified transport protocol.
Requests a transit gateway peering attachment between two VPCs.
Creates a static route for the specified transit gateway.
Creates a route table for the specified transit gateway.
Attaches the specified VPC to the specified transit gateway.
Creates an EBS volume that can be attached to an instance.
Creates a VPC with the specified IPv4 CIDR block.
Creates a VPC endpoint for a specified service.
Creates a connection notification for a specified VPC endpoint.
create_vpc_endpoint_service_configuration
create_vpc_peering_connection
create_vpn_connection
create_vpn_connection_route
create_vpn_gateway
delete_client_vpn_endpoint
delete_customer_gateway
delete_dhcp_options
delete_egress_only_internet_gateway
delete_fleets
delete_flow_logs
delete_fpga_image
delete_internet_gateway
delete_key_pair
delete_launch_template
delete_launch_template_versions
delete_local_gateway_route
delete_local_gateway_route_table_vpc_association
delete_nat_gateway
delete_network_acl
delete_network_acl_entry
delete_network_interface
delete_network_interface_permission
delete_placement_group
delete_queued_reserved_instances
delete_route
delete_route_table
delete_security_group
delete_snapshot
delete_spot_datafeed_subscription
delete_subnet
delete_tags
delete_traffic_mirror_filter
delete_traffic_mirror_filter_rule
delete_traffic_mirror_session
delete_traffic_mirror_target
delete_transit_gateway
delete_transit_gateway_multicast_domain
delete_transit_gateway_peering_attachment
delete_transit_gateway_route
delete_transit_gateway_route_table
delete_transit_gateway_vpc_attachment
delete_volume
delete_vpc
delete_vpc_endpoint_connection_notifications
delete_vpc_endpoints
delete_vpc_endpoint_service_configurations

Creates a VPC endpoint service configuration to which service consumers (AWS accounts, IAM users, and IAM roles) can connect.
Requests a VPC peering connection between two VPCs.
Creates a VPN connection between an existing VPC and a VPN customer gateway.
Creates a static route associated with a VPN connection.
Creates a virtual private gateway.
Deletes the specified Client VPN endpoint.
Deletes a route from a Client VPN endpoint.
Deletes the specified Customer Gateway.
Deletes the specified set of DHCP options.
Deletes an egress-only internet gateway.
Deletes the specified EC2 Fleet.
Deletes one or more flow logs.
Deletes the specified Amazon FPGA Image (AFI).
Deletes the specified internet gateway.
Deletes the specified key pair, by removing the public key from Amazon EC2.
Deletes a launch template.
Deletes one or more versions of a launch template.
Deletes the specified route from the specified local gateway route table.
Deletes the specified association between a VPC endpoint and a route table.
Deletes the specified NAT gateway.
Deletes the specified network ACL.
Deletes the specified ingress or egress entry (rule).
Deletes the specified network interface.
Deletes a permission for a network interface.
Deletes the specified placement group.
Deletes the queued purchases for the specified Reserved Instances.
Deletes the specified route from the specified route table.
Deletes a security group.
Deletes the specified snapshot.
Deletes the data feed for Spot Instances.
Deletes the specified subnet.
Deletes the specified set of tags from the specified set of resources.
Deletes the specified Traffic Mirror filter.
Deletes the specified Traffic Mirror rule.
Deletes the specified Traffic Mirror session.
Deletes the specified Traffic Mirror target.
Deletes the specified transit gateway.
Deletes the specified transit gateway multicast domain.
Deletes a transit gateway peering attachment.
Deletes the specified route from the specified transit gateway route table.
Deletes the specified VPC attachment.
Deletes the specified EBS volume.
Deletes the specified VPC.
Deletes one or more VPC endpoint connection notifications.
Deletes one or more specified VPC endpoints.
Deletes one or more VPC endpoint service configurations.
delete_vpc_peering_connection
delete_vpn_connection
delete_vpn_connection_route
delete_vpn_gateway
deprovision_byoip_cidr
deregister_image
deregister_transit_gateway_multicast_group_members
deregister_transit_gateway_multicast_group_sources
describe_account_attributes
describe_addresses
describe_aggregate_id_format
describe_availability_zones
describe_bundle_tasks
describe_byoip_cidrs
describe_capacity_reservations
describe_classic_link_instances
describe_client_vpn_authorization_rules
describe_client_vpn_connections
describe_client_vpn_endpoints
describe_client_vpn_routes
describe_client_vpn_target_networks
describe_coip_pools
describe_conversion_tasks
describe_customer_gateways
describe_dhcp_options
describe_egress_only_internet_gateways
describe_elastic_gpus
describe_export_image_tasks
describe_export_tasks
describe_fast_snapshot_restores
describe_fleet_history
describe_fleet_instances
describe_fleets
describe_flow_logs
describe_fpga_image_attribute
describe_fpga_images
describe_hostReservation_offering
describe_host_reservations
describe_hosts
describe_iam_instance_profile_associations
describe_identity_id_format
describe_id_format
describe_image_attribute
describe_images
describe_import_image_tasks
describe_import_snapshot_tasks
describe_instance_attribute
describe_instance_credit_specifications

Deletes a VPC peering connection
Deletes the specified VPN connection
Deletes the specified static route associated with a VPC peering connection
Deletes the specified virtual private gateway
Releases the specified address range that you provisioned for use with your AWS resources through bring your own IP addresses (BYOIP)
Deregisters the specified AMI
Deregisters the specified members (network interfaces) from the transit gateway multicast group
Deregisters the specified sources (network interfaces) from the transit gateway multicast group
Describes attributes of your AWS account
Describes the specified Elastic IP addresses or all Elastic IP addresses
Describes the longer ID format settings for all resources in a Region
Describes the Availability Zones and Local Zones
Describes the specified bundle tasks or all of your bundle tasks
Describes the IP address ranges that were specifically reserved for use with your EC2 Classic instances
Describes one or more of your Capacity Reservations
Describes one or more of your linked EC2-Classic instances
Describes the authorization rules for a specified Client VPN endpoint
Describes active client connections and connection failures for a specified Client VPN endpoint
Describes one or more Client VPN endpoints in the account
Describes the routes for the specified Client VPN endpoint
Describes the target networks associated with the specified Client VPN endpoints
Describes the specified customer-owned address pool
Describes the specified conversion tasks or all of your conversion tasks
Describes one or more of your VPN customer gateways
Describes one or more of your DHCP options settings
Describes one or more of your egress-only internet gateways
Describes the Elastic Graphics accelerator associated with your instance
Describes the specified export image tasks or all of your export image tasks
Describes the state of export image tasks or all export image tasks
Describes the state of fast snapshot restores for your instances
Describes the events for the specified EC2 Fleet during the specified time
Describes the running instances for the specified EC2 Fleet
Describes the specified EC2 Fleets or all of your EC2 Fleets
Describes one or more flow logs
Describes the specified attribute of the specified Amazon FPGA Image (AFI)
Describes the Amazon FPGA Images (AFIs) available to you
Describes the Dedicated Host reservations that are available to you
Describes reservations that are associated with Dedicated Hosts
Describes the specified Dedicated Hosts or all of your Dedicated Hosts
Describes your IAM instance profile associations
Describes the ID format settings for resources for a specified IAM user
Describes the ID format settings for your resource groups
Describes the specified attribute of the specified instance
Describes the specified images (AMIs, AKIs, and RIs)
Displays details about an import virtual machine task
Displays details about import snapshots
Displays details about your import snapshot tasks
Displays the import snapshot task information
Displays details about the import virtual machine task
Displays details about your import virtual machine tasks
Displays details about your import virtual machine state transitions
Displays information about a virtual machine
Displays the state of virtual machines in your account
Displays information about your virtual machine state transitions
Displays information about your virtual machine state transitions
describe_instances
describe_instance_status
describe_instance_type_offerings
describe_instance_types
describe.internet_gateways
describe_key_pairs
describe_launch_templates
describe_launch_template_versions
describe_local_gateway_route_tables
describe_local_gateway_route_table_virtual_interface_group_associations
describe_local_gateway_route_table_vpc_associations
describe_local_gateways
describe_local_gateway_virtual_interface_groups
describe_local_gateway_virtual_interfaces
describe_moving_addresses
describe_nat_gateways
describe_network_acls
describe_network_interface_attribute
describe_network_interface_permissions
describe_network_interfaces
describe_placement_groups
describe_prefix_lists
describe_principal_id_format
describe_public_ipv_4_pools
describe_regions
describe_reserved_instances
describe_reserved_instances_listings
describe_reserved_instances_modifications
describe_reserved_instances_offering
describe_route_tables
describe_scheduled_instance_availability
describe_scheduled_instances
describe_security_group_references
describe_security_groups
describe_snapshot_attribute
describe_snapshots
describe_spot_datafeed_subscription
describe_spot_fleet_instances
describe_spot_fleet_request_history
describe_spot_fleet_requests
describe_spot_instance_requests
describe_spot_price_history
describe_stale_security_groups
describe_subnets
describe_tags
describe_traffic_mirror_filters
describe_traffic_mirror_sessions
describe_traffic_mirror_targets

Describes the specified instances or all of AWS account’s instances
Describes the status of the specified instances or all of the instances in your AWS account
Returns a list of all instance types offered
Returns a list of all instance types offered in your current AWS Region
Describes one or more of your internet gateways
Describes the specified key pairs or all of your key pairs
Describes one or more launch templates
Describes one or more versions of a specified launch template
Describes one or more local gateway route tables
Describes the associations between virtual interface groups and local gateway route tables
Describes the specified associations between VPCs and local gateway route tables
Describes your Elastic IP addresses that are being used in your VPCs
Describes one or more of your NAT gateways
Describes one or more of your network ACLs
Describes a network interface attribute
Describes the permissions for your network interfaces
Describes one or more of your network interfaces
Describes the specified placement groups or all of your placement groups
Describes available AWS services in a prefix list format, which includes the prefix list name and prefix list ID of the service and the IP address range for the service
Describes the ID format settings for the root user and all IAM roles and IAM users that have explicitly specified a longer ID preference
Describes the specified IPv4 address pools
Describes the Regions that are enabled for your account
Describes one or more of the Reserved Instance offerings
Describes your account’s Reserved Instance listings
Describes the modifications made to your Reserved Instance offerings that are available for purchase
Describes Reserved Instance offerings that are available for purchase
Describes one or more of your route tables
Finds available schedules that meet the specified criteria
Describes the specified Scheduled Instances or all of your Scheduled Instances
[VPC only] Describes the VPCs on the other side of a VPC peering connection that are referencing the security groups you’ve specified in this request
Describes the specified security groups or all of your security groups
Describes the specified attribute of the specified security group
Describes the specified EBS snapshots available for restore
Describes the data feed for Spot Instances
Describes the running instances for the specified Spot Fleet
Describes the events for the specified Spot Fleet
Describes your Spot Fleet requests
Describes the specified Spot Instance requests
Describes the Spot price history
[VPC only] Describes the stale security group rules
Describes one or more of your subnets
Describes the specified tags for your EC2 resources
Describes one or more Traffic Mirror filters
Describes one or more Traffic Mirror sessions
Information about one or more Traffic Mirror targets
describe_transit_gateway_attachments
  Describes one or more attachments between resources and transit gateways

describe_transit_gateway_multicast_domains
  Describes one or more transit gateway multicast domains

describe_transit_gateway_peering_attachments
  Describes your transit gateway peering attachments

describe_transit_gateway_route_tables
  Describes one or more transit gateway route tables

describe_transit_gateways
  Describes one or more transit gateways

describe_transit_gateway_vpc_attachments
  Describes one or more VPC attachments

describe_volume_attribute
  Describes the specified attribute of the specified volume

describe_volumes
  Describes the specified EBS volumes or all of your EBS volumes

describe_volumes_modifications
  Reports the current modification status of EBS volumes

describe_volume_status
  Describes the status of the specified volumes

describe_vpc_attribute
  Describes the specified attribute of the specified VPC

describe_vpc_classic_link
  Describes the ClassicLink status of one or more VPCs

describe_vpc_classic_link_dns_support
  Describes the ClassicLink DNS support status of one or more VPCs

describe_vpc_endpoint_connection_notifications
  Describes the connection notifications for VPC endpoints

describe_vpc_endpoint_connections
  Describes the VPC endpoint connections to your VPC services and the VPC endpoints to your services

describe_vpc_endpoint_service_configurations
  Describes the VPC endpoint service configurations in your account (your services)

describe_vpc_endpoint_service_permissions
  Describes the principals (service consumers) that are permitted to discover your VPC endpoint service

describe_vpc_endpoint_services
  Describes available services to which you can create VPC endpoints

describe_vpc_endpoints
  Describes one or more of your VPC endpoints

describe_vpc_peering_connections
  Describes one or more of your VPC peering connections

describe_vpcs
  Describes one or more of your VPCs

describe_vpn_connections
  Describes one or more of your VPN connections

describe_vpn_gateways
  Describes one or more of your virtual private gateways

detach_classic_link_vpc
  Unlinks (detaches) a linked EC2-Classic instance from a VPC

detach_internet_gateway
  Detaches an internet gateway from a VPC, disabling connectivity between the internet and the VPC

detach_network_interface
  Detaches a network interface from an instance

detach_volume
  Detaches an EBS volume from an instance

detach_vpn_gateway
  Detaches a virtual private gateway from a VPC

disable_ebs_encryption_by_default
  Disables EBS encryption by default for your account

disable_fast_snapshot_restores
  Disables fast snapshot restores for the specified snapshots

disable_transit_gateway_route_table_propagation
  Disables the specified attachment from propagating routes to the specified propagation route table

disable_vgw_route_propagation
  Disables a virtual private gateway (VGW) from propagating routes to a specified route table of a VPC

disable_vpc_classic_link
  Disables ClassicLink for a VPC

disable_vpc_classic_link_dns_support
  Disables ClassicLink DNS support for a VPC

disassociate_address
  Disassociates an Elastic IP address from the instance or network interface it's associated with

disassociate_client_vpn_target_network
  Disassociates a target network from the specified Client VPN endpoint

disassociate_iam_instance_profile
  Disassociates an IAM instance profile from a running or stopped instance

disassociate_route_table
  Disassociates a subnet from a route table

disassociate_subnet_cidr_block
  Disassociates the specified subnets from the specified VPC

disassociate_transit_gateway_multicast_domain
  Disassociates a resource attachment from a transit gateway multicast domain

disassociate_transit_gateway_route_table
  Disassociates a CIDR block from a route table

disassociate_vpc_cidr_block
  Disassociates a CIDR block from a VPC

disable_volume_io
  Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent

enable_ebs_encryption_by_default
  Enables EBS encryption by default for your account

enable_fast_snapshot_restores
  Enables fast snapshot restores for the specified snapshots

enable_transit_gateway_route_table_propagation
  Enables the specified attachment to propagate routes to the specified propagation route table

enable_vgw_route_propagation
  Enables a virtual private gateway (VGW) to propagate routes to a specified route table of a VPC

enable_vpc_classic_link
  Enables a VPC for ClassicLink
enable_vpc_classic_link_dns_support
export_client_vpn_client_certificate_revocation_list
export_client_vpn_client_configuration
export_image
export_transit_gateway_routes
get_capacity_reservation_usage
get_coip_pool_usage
get_console_output
get_console_screenshot
get_default_credit_specification
get_ebs_default_kms_key_id
get_ebs_encryption_by_default
get_host_reservation_purchase_preview
get_launch_template_data
get_password_data
get_reserved_instances_exchange_quote
get_transit_gateway_attachment_propagations
get_transit_gateway_multicast_domain_associations
get_transit_gateway_route_table_associations
get_transit_gateway_route_table_propagations
import_client_vpn_client_certificate_revocation_list
import_image
import_instance
import_key_pair
import_snapshot
import_volume
modify_capacity_reservation
modify_client_vpn_endpoint
modify_default_credit_specification
modify_ebs_default_kms_key_id
modify_fleet
modify_fpga_image_attribute
modify_hosts
modify_identity_id_format
modify_id_format
modify_image_attribute
modify_instance_attribute
modify_instance_capacity_reservation_attributes
modify_instance_credit_specification
modify_instance_event_start_time
modify_instance_metadata_options
modify_instance_placement
modify_launch_template
modify_network_interface_attribute
modify_reserved_instances
modify_snapshot_attribute
modify_spot_fleet_request
modify_subnet_attribute
modify_transit_gateway_multicast_domain_attributes
modify_transit_gateway_multicast_domain_associations
modify_transit_gateway_route_table_associations
modify_transit_gateway_route_table_propagations
modify_vpc_classic_link_dns_support
modify_vpc_dns_hostnames
modify_vpc_route_table
modify_vpc_network_acl
modify_vpc_peering_connection
modify_vpc_peering_connection_status
modify_vpc_security_group
modify_vpc_subnet
modify_vpc_vpn_connection
modify_vpc_vpn_connection_status
modify_vpc_vpn_gateways
modify_vpc_group
modify_vpc

ec2

Enables a VPC to support DNS hostname resolution for ClassicLink
Downloads the client certificate revocation list for the specified Client VPN endpoint
Downloads the contents of the Client VPN endpoint configuration file for the specified Client VPN endpoint
Exports an Amazon Machine Image (AMI) to a local file or S3 bucket
Exports routes from the specified transit gateway route table to the specified S3 bucket
Gets usage information about a Capacity Reservation
Describes the allocations from the specified customer-owned address pool
Gets the console output for the specified instance
Retrieve a JPG-format screenshot of a running instance
Describes the default credit option for CPU usage at launch
Describes the default customer master key (CMK) for EBS encryption by default for your account in this Region
Preview a reservation purchase with configurations that match those of your Dedicated Host
Gets information about the associations for the specified VPC
Lists the route tables to which the specified resource attaches
Gets information about the associations for the specified VPC
Gets information about the route table propagations for the specified VPC
Uploads a client certificate revocation list to the specified Client VPN endpoint
Import single or multi-volume disk images or EBS snapshots into an Amazon Machine Image (AMI)
Creates an import instance task using metadata from the specified disk image
Imports the public key from an RSA key pair that you created with a third-party tool
Imports a disk into an EBS snapshot
Creates an import volume task using metadata from the specified disk image
Modifies a Capacity Reservation’s capacity and terms
Modifies the specified VPC endpoint
Modifies the default credit option for CPU usage at launch
Changes the default customer master key (CMK) for EBS encryption by default for your account in this Region
Modifies the specified EC2 Fleet
Modifies the specified attribute of the specified AMI
Modify the auto-placement setting of a Dedicated Host
Modifies the ID format of a resource for a specified Region
Modifies the ID format for the specified resource
Modifies the ID format for the specified resource
Modifies the ID format for the specified resource
Modifies the specified attribute of the specified AMI
Modifies the specified attribute of the specified resource
Modifies the Capacity Reservation settings for a specified instance
Modifies the credit option for CPU usage on a running instance
Modifies the start time for a scheduled Amazon EC2 instance
Modifies the instance metadata parameters on a running instance
Modifies the placement attributes for a specified instance
Modifies a launch template
Modifies the specified network interface attribute
Modifies the Availability Zone, instance count, instance types, and network platform
Adds or removes permission settings for the specified Spot Fleet
Modifies the specified Spot Fleet request
Modifies a subnet attribute
modify_traffic_mirror_filter_network_services
modify_traffic_mirror_filter_rule
modify_traffic_mirror_session
modify_transit_gateway_vpc_attachment
modify_volume
modify_volume_attribute
modify_vpc_attribute
modify_vpc_endpoint
modify_vpc_endpoint_connection_notification
modify_vpc_endpoint_service_configuration
modify_vpc_endpoint_service_permissions
modify_vpc_peering_connection_options
modify_vpc_tenancy
modify_vpn_connection
modify_vpn_tunnel_certificate
modify_vpn_tunnel_options
monitor_instances
move_address_to_vpc
provision_byoip_cidr
purchase_host_reservation
purchase_reserved_instances_offering
purchase_scheduled_instances
reboot_instances
register_image
register_transit_gateway_multicast_group_members
register_transit_gateway_multicast_group_sources
reject_transit_gateway_peering_attachment
reject_transit_gateway_vpc_attachment
reject_vpc_endpoint_connections
reject_vpc_peering_connection
release_address
release_hosts
replace_iam_instance_profile_association
replace_network_acl_association
replace_network_acl_entry
replace_route
replace_route_table_association
replace_transit_gateway_route
report_instance_status
request_spot_fleet
request_spot_instances
reset_ebs_default_kms_key_id
reset_fpga_image_attribute
reset_image_attribute
reset_instance_attribute
reset_network_interface_attribute
reset_snapshot_attribute
restore_address_to_classic

Allows or restricts mirroring network services
Modifies the specified Traffic Mirror rule
Modifies a Traffic Mirror session
Modifies the specified VPC attachment
You can modify several parameters of an existing EBS volume
Modifies a volume attribute
Modifies the specified attribute of the specified VPC
Modifies attributes of a specified VPC endpoint
Modifies a connection notification for VPC endpoint
Modifies the attributes of your VPC endpoint service
Modifies the permissions for your VPC endpoint
Modifies the VPC peering connection options on
Modifies the instance tenancy attribute of the specified
Modifies the target gateway of an AWS Site-to-Site
Modifies the VPN tunnel endpoint certificate
Modifies the options for a VPN tunnel in an AWS
Enables detailed monitoring for a running instance
Moves an Elastic IP address from the EC2-Classic
Provisions an address range for use with your AWS
Purchase a reservation with configurations that match
Purchases a Reserved Instance for use with your
Purchases the Scheduled Instances with the specified
Requests a reboot of the specified instances
Registers an AMI
Registers members (network interfaces) with the specified
Registers sources (network interfaces) with the specified
Rejects a transit gateway peering attachment request
Rejects a request to attach a VPC to a transit gateway
Rejects one or more VPC endpoint connection requests
Rejects a VPC peering connection request
Releases the specified Elastic IP address
When you no longer want to use an On-Demand
Replaces an IAM instance profile for the specified
Changes which network ACL a subnet is associated with
Replaces an entry (rule) in a network ACL
Replaces an existing route within a route table in
Changes the route table associated with a given subnet
Replaces the specified route in the specified
Submits feedback about the status of an instance
Creates a Spot Fleet request
Creates a Spot Instance request
Resets the default customer master key (CMK) for
Resets the specified attribute of the specified AMI
Resets an attribute of an AMI to its default value
Resets an attribute of an instance to its default value
Resets a network interface attribute
Resets permission settings for the specified snapshot
Restores an Elastic IP address that was previously
ec2instanceconnect

AWS EC2 Instance Connect

Description

AWS EC2 Connect Service is a service that enables system administrators to publish temporary SSH keys to their EC2 instances in order to establish connections to their instances without leaving a permanent authentication option.

Usage

ec2instanceconnect(config = list())
Arguments

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

Service syntax

```r
csvc <- ec2instanceconnect(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

- **send_ssh_public_key**  Pushes an SSH public key to a particular OS user on a given EC2 instance for 60 seconds

Examples

```r
# The following example pushes a sample SSH public key to the EC2 instance
# i-abcd1234 in AZ us-west-2b for use by the instance OS user ec2-user.
## Not run: svc <- ec2instanceconnect()
svc$send_ssh_public_key(
  AvailabilityZone = "us-west-2a",
  InstanceId = "i-abcd1234",
  InstanceOSUser = "ec2-user",
  SSHPublicKey = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQ3FlHqj2eqCdrGHuA6dRjfZXQ4X51XEIRHa..."
)
## End(Not run)
```

---

**ecr**  
Amazon EC2 Container Registry

Description

Amazon Elastic Container Registry
Amazon Elastic Container Registry (Amazon ECR) is a managed Docker registry service. Customers can use the familiar Docker CLI to push, pull, and manage images. Amazon ECR provides a secure, scalable, and reliable registry. Amazon ECR supports private Docker repositories with resource-based permissions using IAM so that specific users or Amazon EC2 instances can access repositories and images. Developers can use the Docker CLI to author and manage images.

Usage

ecr(config = list())

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

Service syntax

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

Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch_check_layer_availability</td>
<td>Check the availability of multiple image layers in a specified registry and repository</td>
</tr>
<tr>
<td>batch_delete_image</td>
<td>Deletes a list of specified images within a specified repository</td>
</tr>
<tr>
<td>batch_get_image</td>
<td>Gets detailed information for specified images within a specified repository</td>
</tr>
<tr>
<td>complete_layer_upload</td>
<td>Informs Amazon ECR that the image layer upload has completed for a specified registry</td>
</tr>
<tr>
<td>create_repository</td>
<td>Creates an Amazon Elastic Container Registry (Amazon ECR) repository, where users can push and pull Docker images</td>
</tr>
<tr>
<td>delete_lifecycle_policy</td>
<td>Deletes the specified lifecycle policy</td>
</tr>
<tr>
<td>delete_repository</td>
<td>Deletes an existing image repository</td>
</tr>
<tr>
<td>delete_repository_policy</td>
<td>Deletes the repository policy from a specified repository</td>
</tr>
<tr>
<td>describe_images</td>
<td>Returns metadata about the images in a repository, including image size, image tags, and creation date</td>
</tr>
<tr>
<td>describe_image_scan_findings</td>
<td>Describes the image scan findings for the specified image</td>
</tr>
<tr>
<td>describe_repositories</td>
<td>Describes image repositories in a registry</td>
</tr>
<tr>
<td>get_authorization_token</td>
<td>Retrieves a token that is valid for a specified registry for 12 hours</td>
</tr>
<tr>
<td>get_download_url_for_layer</td>
<td>Retrieves the pre-signed Amazon S3 download URL corresponding to an image layer</td>
</tr>
<tr>
<td>get_lifecycle_policy</td>
<td>Retrieves the specified lifecycle policy</td>
</tr>
<tr>
<td>get_lifecycle_policy_preview</td>
<td>Retrieves the results of the specified lifecycle policy preview request</td>
</tr>
<tr>
<td>get_repository_policy</td>
<td>Retrieves the repository policy for a specified repository</td>
</tr>
</tbody>
</table>
### Examples

```r
# This example deletes images with the tags precise and trusty in a
# repository called ubuntu in the default registry for an account.
## Not run: svc <- ecr()
svc$batch_delete_image(
    imageIds = list(
        list(
            imageTag = "precise"
        ),
        repositoryName = "ubuntu"
    ),
    ## End(Not run)
```

---

**Amazon EC2 Container Service**

#### Description

Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. You can host your cluster on a serverless infrastructure that is managed by Amazon ECS by launching your services or tasks using the Fargate launch type. For more control, you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that you manage by using the EC2 launch type. For more information about launch types, see [Amazon ECS Launch Types](#).

Amazon ECS lets you launch and stop container-based applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features.
You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs, isolation policies, and availability requirements. Amazon ECS eliminates the need for you to operate your own cluster management and configuration management systems or worry about scaling your management infrastructure.

Usage

```
ecs(config = list())
```

Arguments

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

Service syntax

```
svc <- ecs(
    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_capacity_provider`: Creates a new capacity provider
- `create_cluster`: Creates a new Amazon ECS cluster
- `create_service`: Runs and maintains a desired number of tasks from a specified task definition
- `create_task_set`: Create a task set in the specified cluster and service
- `delete_account_setting`: Disables an account setting for a specified IAM user, IAM role, or the root user for an account
- `delete_attributes`: Deletes one or more custom attributes from an Amazon ECS resource
- `delete_cluster`: Deletes the specified cluster
- `delete_service`: Deletes a specified service within a cluster
- `delete_task_set`: Deletes a specified task set within a service
- `deregister_container_instance`: Deregisters an Amazon ECS container instance from the specified cluster
- `deregister_task_definition`: Deregisters the specified task definition by family and revision
- `describe_capacity_providers`: Describes one or more of your capacity providers
- `describe_clusters`: Describes one or more of your clusters
- `describe_container_instances`: Describes Amazon Elastic Container Service container instances
- `describe_services`: Describes the specified services running in your cluster
- `describe_task_definition`: Describes a task definition
- `describe_tasks`: Describes a specified task or tasks
describe_task_sets
- Describes the task sets in the specified cluster and service

discover_poll_endpoint
- This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent

list_account_settings
- Lists the account settings for a specified principal

list_attributes
- Lists the attributes for Amazon ECS resources within a specified target type and cluster

list_clusters
- Returns a list of existing clusters

list_container_instances
- Returns a list of container instances in a specified cluster

list_services
- Lists the services that are running in a specified cluster

list_tags_for_resource
- Lists the tags for an Amazon ECS resource

list_task_definition_families
- Returns a list of task definition families that are registered to your account (which may include task definition families that no longer have any ACTIVE task definition revisions)

list_task_definitions
- Returns a list of tasks definitions that are registered to your account

list_tasks
- Returns a list of tasks for a specified cluster

put_account_setting
- Modifies an account setting

put_account_setting_default
- Modifies an account setting for all IAM users on an account for whom no individual account setting has been specified

put_attributes
- Create or update an attribute on an Amazon ECS resource

put_cluster_capacity_providers
- Modifies the available capacity providers and the default capacity provider strategy for a cluster

register_container_instance
- Registers a new task definition from the supplied family and containerDefinitions

register_task_definition
- Starts a new task using the specified task definition

run_task
- Starts a new task from the specified task definition on the specified container instance or instances

start_task
- Stops a running task

stop_task
- This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent

submit_attachment_state_changes
- This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent

submit_container_state_change
- This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent

submit_task_state_change
- This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent

tag_resource
- Associates the specified tags to a resource with the specified resourceArn

untag_resource
- Deletes specified tags from a resource

update_cluster_settings
- Modifies the settings to use for a cluster

update_container_agent
- Updates the Amazon ECS container agent on a specified container instance

update_container_instances_state
- Modifies the status of an Amazon ECS container instance

update_service
- Modifies the parameters of a service

update_service_primary_task_set
- Modifies which task set in a service is the primary task set

update_task_set
- Modifies a task set

---

**Examples**

```r
# This example creates a cluster in your default region.
## Not run: svc <- ecs()
svc$create_cluster(
  clusterName = "my_cluster"
)
## End(Not run)
```

---

**efs**

*Amazon Elastic File System*
Description

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it. For more information, see the User Guide.

Usage

```python
efs(config = list())
```

Arguments

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

Service syntax

```python
svc <- efs(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_file_system`: Creates a new, empty file system
- `create_mount_target`: Creates a mount target for a file system
- `create_tags`: Creates or overwrites tags associated with a file system
- `delete_file_system`: Deletes a file system, permanently severing access to its contents
- `delete_mount_target`: Deletes the specified mount target
- `delete_tags`: Deletes the specified tags from a file system
- `describe_file_systems`: Returns the description of a specific Amazon EFS file system if either the file system CreationToken or the FileSystemId is provided
- `describe_lifecycle_configuration`: Returns the current LifecycleConfiguration object for the specified Amazon EFS file system
- `describe_mount_targets`: Returns the descriptions of all the current mount targets, or a specific mount target, for a file system
- `describe_mount_target_security_groups`: Returns the security groups currently in effect for a mount target
- `describe_tags`: Returns the tags associated with a file system
- `modify_mount_target_security_groups`: Modifies the set of security groups in effect for a mount target
- `put_lifecycle_configuration`: Enables lifecycle management by creating a new LifecycleConfiguration object
- `update_file_system`: Updates the throughput mode or the amount of provisioned throughput of an existing file system
Examples

```r
# This operation creates a new file system with the default generalpurpose
# performance mode.
## Not run: svc <- efs()
svc$create_file_system(
  CreationToken = "tokenstring",
  PerformanceMode = "generalPurpose",
  Tags = list(
    list(
      Key = "Name",
      Value = "MyFileSystem"
    )
  )
)
## End(Not run)
```

---

**eks**  
*Amazon Elastic Kubernetes Service*

**Description**

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that makes it easy for you to run Kubernetes on AWS without needing to stand up or maintain your own Kubernetes control plane. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications.

Amazon EKS runs up-to-date versions of the open-source Kubernetes software, so you can use all the existing plugins and tooling from the Kubernetes community. Applications running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, whether running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification required.

**Usage**

```r
eks(config = list())
```

**Arguments**

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

**Service syntax**

```r
svc <- eks(
  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_cluster Creates an Amazon EKS control plane
create_fargate_profile Creates an AWS Fargate profile for your Amazon EKS cluster
create_nodegroup Creates a managed worker node group for an Amazon EKS cluster
delete_cluster Deletes the Amazon EKS cluster control plane
delete_fargate_profile Deletes an AWS Fargate profile
delete_nodegroup Deletes an Amazon EKS node group for a cluster
describe_cluster Returns descriptive information about an Amazon EKS cluster
describe_fargate_profile Returns descriptive information about an AWS Fargate profile
describe_nodegroup Returns descriptive information about an Amazon EKS node group
describe_update Returns descriptive information about an update against your Amazon EKS cluster or associated managed node group
list_clusters Lists the Amazon EKS clusters in your AWS account in the specified Region
list_fargate_profiles Lists the AWS Fargate profiles associated with the specified cluster in your AWS account in the specified Region
list_nodegroups Lists the Amazon EKS node groups associated with the specified cluster in your AWS account in the specified Region
list_tags_for_resource List the tags for an Amazon EKS resource
list_updates Lists the updates associated with an Amazon EKS cluster or managed node group in your AWS account in the specified Region
tag_resource Associates the specified tags to a resource with the specified resourceArn
untag_resource Deletes specified tags from a resource
update_cluster_config Updates an Amazon EKS cluster configuration
update_cluster_version Updates an Amazon EKS cluster to the specified Kubernetes version
update_nodegroup_config Updates an Amazon EKS managed node group configuration
update_nodegroup_version Updates the Kubernetes version or AMI version of an Amazon EKS managed node group

Examples

# The following example creates an Amazon EKS cluster called prod.
## Not run: svc <- eks()
svc$create_cluster(
  version = "1.10",
  name = "prod",
  clientRequestToken = "1d2129a1-3d38-460a-9756-e5b91fd4db951",
  resourcesVpcConfig = list(
    securityGroupIds = list("sg-6979fe18"),
  ),
)
elasticache

Description

Amazon ElastiCache is a web service that makes it easier to set up, operate, and scale a distributed cache in the cloud.

With ElastiCache, customers get all of the benefits of a high-performance, in-memory cache with less of the administrative burden involved in launching and managing a distributed cache. The service makes setup, scaling, and cluster failure handling much simpler than in a self-managed cache deployment.

In addition, through integration with Amazon CloudWatch, customers get enhanced visibility into the key performance statistics associated with their cache and can receive alarms if a part of their cache runs hot.

Usage

elasticache(config = list())

Arguments

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

Service syntax

```
svc <- elasticache(
  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_to_resource: Adds up to 50 cost allocation tags to the named resource.
- authorize_cache_security_group_ingress: Allows network ingress to a cache security group.
- batch_apply_update_action: Apply the service update.
- batch_stop_update_action: Stop the service update.
- complete_migration: Complete the migration of data.
- copy_snapshot: Makes a copy of an existing snapshot.
- create_cache_cluster: Creates a cluster.
- create_cache_parameter_group: Creates a new Amazon ElastiCache cache parameter group.
- create_cache_security_group: Creates a new cache security group.
- create_cache_subnet_group: Creates a new cache subnet group.
- create_replication_group: Creates a Redis (cluster mode disabled) or a Redis (cluster mode enabled) replication group.
- create_snapshot: Creates a copy of an entire cluster or replication group at a specific moment in time.
- decrease_replica_count: Dynamically decreases the number of replicas in a Redis (cluster mode disabled) replication group or the number of replica nodes in one or more node groups (shards) of a Redis (cluster mode enabled) replication group.
- delete_cache_cluster: Deletes a previously provisioned cluster.
- delete_cache_parameter_group: Deletes the specified cache parameter group.
- delete_cache_security_group: Deletes a cache security group.
- delete_cache_subnet_group: Deletes an existing replication group.
- delete_replication_group: Deletes an existing snapshot.
- delete_snapshot: Deletes an existing snapshot.
- describe_cache_clusters: Returns information about all provisioned clusters if no cluster identifier is specified, or about a specific cache cluster if a cluster identifier is supplied.
- describe_cache_engine_versions: Returns a list of the available cache engines and their versions.
- describe_cache_parameter_groups: Returns a list of cache parameter group descriptions.
- describe_cache_parameters: Returns the detailed parameter list for a particular cache parameter group.
- describe_cache_security_groups: Returns the detailed parameter list for a particular cache parameter group.
- describe_cache_subnet_groups: Returns a list of cache subnet group descriptions.
- describe_engine_default_parameters: Returns the default engine and system parameter information for the specified cache engine.
- describe_events: Returns events related to clusters, cache security groups, and cache parameter groups.
- describe_replication_groups: Returns information about a particular replication group.
- describe_reserved_cache_nodes: Returns information about reserved cache nodes for this account, or about a specified reserved cache node.
- describe_reserved_cache_nodes_offering: Lists available reserved cache node offerings.
- describe_service_updates: Returns details of the service updates.
- describe_snapshots: Returns information about cluster or replication group snapshots.
- describe_update_actions: Returns details of the update actions.
- increase_replica_count: Dynamically increases the number of replicas in a Redis (cluster mode disabled) replication group or the number of replica nodes in one or more node groups (shards) of a Redis (cluster mode enabled) replication group.
- list_allowed_node_type_modifications: Lists all available node types that you can scale your Redis cluster’s or replication group’s node type.
- list_tags_for_resource: Lists all cost allocation tags currently on the named resource.
- modify_cache_cluster: Modifies the settings for a cluster.
- modify_cache_parameter_group: Modifies the parameters of a cache parameter group.
- modify_cache_subnet_group: Modifies an existing cache subnet group.
- modify_replication_group: Modifies the settings for a replication group.
- modify_replication_group_shard_configuration: Modifies a replication group’s shards (node groups) by allowing you to add or remove shard(s).
- purchase_reserved_cache_nodes_offering: Allows you to purchase a reserved cache node offering.
- reboot_cache_cluster: Reboots some, or all, of the cache nodes within a provisioned cluster.
- remove_tags_from_resource: Removes the tags identified by the TagKeys list from the named resource.
- reset_cache_parameter_group: Modifies the parameters of a cache parameter group to the engine or system default value.
- revoke_cache_security_group_ingress: Revokes ingress from a cache security group.
### Description

AWS Elastic Beanstalk makes it easy for you to create, deploy, and manage scalable, fault-tolerant applications running on the Amazon Web Services cloud.

For more information about this product, go to the AWS Elastic Beanstalk details page. The location of the latest AWS Elastic Beanstalk WSDL is `http://elasticbeanstalk.s3.amazonaws.com/doc/2010-12-01/AWSElasticBeanstalk.wsdl`. To install the Software Development Kits (SDKs), Integrated Development Environment (IDE) Toolkits, and command line tools that enable you to access the API, go to Tools for Amazon Web Services.

#### Endpoints

For a list of region-specific endpoints that AWS Elastic Beanstalk supports, go to Regions and Endpoints in the Amazon Web Services Glossary.

### Usage

```
elasticbeanstalk(config = list())
```

### Arguments

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

### Service syntax

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

---

**Examples**

```r
## Not run: svc <- elasticache()
svc$add_tags_to_resource(
  Foo = 123
)
## End(Not run)
```

---

**Start the migration of data**

Start migration of data with the `start_migration` function. It is represented by the `test_failover` function, which represents the input of a `TestFailover` operation to test automatic failover on a specified node group (called shard in the console) in a replication group (called cluster in the console).
operations =
),
    profile = "string"
),
    endpoint = "string",
    region = "string"
)
)

Operations

abort_environment_update
apply_environment_managed_action
check_dns_availability
compose_environments
create_application
create_application_version
create_configuration_template
create_environment
create_platform_version
create_storage_location
delete_application
delete_application_version
delete_configuration_template
delete_environment_configuration
delete_platform_version
describe_account_attributes
describe_applications
describe_application_versions
describe_configuration_options
describe_configuration_settings
describe_environment_health
describe_environment_managed_action_history
describe_environment_managed_actions
describe_environment_resources
describe_environments
describe_events
describe_instances_health
describe_platform_version
list_available_solution_stacks
list_platform_versions
list_tags_for_resource
rebuild_environment
request_environment_info
restart_app_server
retrieve_environment_info
swap_environment_cnam_es
terminate_environment
update_application

Cancels in-progress environment configuration update or application version deployment.
Applies a scheduled managed action immediately.
Checks if the specified CNAME is available.
Create or update a group of environments that each run a separate component of a single application.
Creates an application that has one configuration template named default and no application versions.
Creates an application version for the specified application.
Creates a configuration template.
Launches an environment for the specified application using the specified configuration.
Create a new version of your custom platform.
Creates a bucket in Amazon S3 to store application versions, logs, and other files used by Elastic Beanstalk environments.
Deletes the specified application along with all associated versions and configurations.
Deletes the specified version from the specified application.
Deletes the specified configuration template.
Deletes the draft configuration associated with the running environment.
Deletes the specified version of a custom platform.
Returns attributes related to AWS Elastic Beanstalk that are associated with the calling AWS account.
Returns the descriptions of existing applications.
Retrieve a list of application versions.
Describes the configuration options that are used in a particular configuration template.
Returns a description of the settings for the specified configuration set, that is, either a configuration template or the configuration set associated with a running environment.
Lists an environment’s completed and failed managed actions.
Lists an environment’s upcoming and in-progress managed actions.
Returns AWS resources for this environment.
Returns descriptions for existing environments.
Returns list of event descriptions matching criteria up to the last 6 weeks.
Retrieves detailed information about the health of instances in your AWS Elastic Beanstalk environment.
Describes the version of the platform.
Returns a list of the available solution stack names, with the public version first and then in reverse chronological order.
Retrieves the tags applied to an AWS Elastic Beanstalk resource.
Deletes and recreates all of the AWS resources (for example: the Auto Scaling group, load balancer, etc.
Initiates a request to compile the specified type of information of the deployment.
Causes the environment to restart the application container server running on each Amazon EC2 instance.
Retrieves the compiled information from a RequestEnvironmentInfo request.
Swaps the CNAMEs of two environments.
Terminates the specified environment.
Updates the specified application to have the specified properties.
elasticsearchservice

update_application_resource_lifecycle
update_application_version
update_configuration_template
update_environment
update_tags_for_resource
validate_configuration_settings

Modifies lifecycle settings for an application
Updates the specified application version to have the specified properties
Updates the specified configuration template to have the specified properties
Update the environment description, deploys a new application version, updates the configuration settings to an entirely new configuration template, or updates select configuration option values in the running environment
Update the list of tags applied to an AWS Elastic Beanstalk resource
Takes a set of configuration settings and either a configuration template or environment identifier to determine whether the values are valid

Examples

# The following code aborts a running application version deployment for
# an environment named my-env:
## Not run: svc <- elasticbeanstalk()
svc$abort_environment_update(
  EnvironmentName = "my-env"
)
## End(Not run)

elasticsearchservice  Amazon Elasticsearch Service

Description

Amazon Elasticsearch Configuration Service

Use the Amazon Elasticsearch Configuration API to create, configure, and manage Elasticsearch domains.

For sample code that uses the Configuration API, see the Amazon Elasticsearch Service Developer Guide. The guide also contains sample code for sending signed HTTP requests to the Elasticsearch APIs.

The endpoint for configuration service requests is region-specific: es.region.amazonaws.com. For example, es.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

Usage

elasticsearchservice(config = list())

Arguments

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

svc <- elasticsearchservice(
  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
cancel_elasticsearch_service_software_update
create_elasticsearch_domain
delete_elasticsearch_domain
delete_elasticsearch_service_role
describe_elasticsearch_domain
describe_elasticsearch_domain_config
describe_elasticsearch_domains
describe_elasticsearch_instance_type_limits
describe_reserved_elasticsearch_instance_offering
describe_reserved_elasticsearch_instances
get_compatible_elasticsearch_versions
get_upgrade_history
get_upgrade_status
list_domain_names
list_elasticsearch_domain_types
list_elasticsearch_versions
list_tags
purchase_reserved_elasticsearch_instance_offering
remove_tags
start_elasticsearch_service_software_update
update_elasticsearch_domain_config
upgrade_elasticsearch_domain

Examples

## Not run: svc <- elasticsearchservice()
svc$add_tags()

Attaches tags to an existing Elasticsearch domain
Cancels a scheduled service software update for an Amazon ES domain
Creates a new Elasticsearch domain
Permanently deletes the specified Elasticsearch domain and all of its data
Deletes the service-linked role that Elasticsearch Service uses to manage and maintain VPC domains
Returns domain configuration information about the specified Elasticsearch domain
Returns domain configuration information about the specified Elasticsearch domain
Provides cluster configuration information about the specified Elasticsearch domain
Describe Elasticsearch Limits for a given InstanceType and ElasticsearchVersion
Lists available reserved Elasticsearch instance offerings
Returns information about reserved Elasticsearch instances for this account
Retrieves a list of upgrade compatible Elasticsearch versions
Retrieves the complete history of the last 10 upgrades that were performed
Retrieves the latest status of the last upgrade or upgrade eligibility check
Returns the name of all Elasticsearch domains owned by the current user
List all Elasticsearch instance types that are supported for given Elasticsearch version
List all supported Elasticsearch versions
Returns all tags for the given Elasticsearch domain
Allows you to purchase reserved Elasticsearch instances
Removes the specified set of tags from the specified Elasticsearch domain
Schedules a service software update for an Amazon ES domain
Modifies the cluster configuration of the specified Elasticsearch domain
Allows you to either upgrade your domain or perform an Upgrade eligibility check.
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"
    ),
    profile = "string"
  ),
  profile = "string"
)
```

Foo = 123

## End(Not run)
endpoints = "string",
region = "string"
)
)

**Operations**

- add_tags
- apply_security_groups_to_load_balancer
- attach_load_balancer_to_subnets
- configure_health_check
- create_app_cookie_stickiness_policy
- create_lb_cookie_stickiness_policy
- create_load_balancer
- create_load_balancer_listeners
- create_load_balancer_policy
- delete_load_balancer
- delete_load_balancer_listeners
- delete_load_balancer_policy
- deregister_instances_from_load_balancer
- describe_account_limits
- describe_instance_health
- describe_load_balancer_attributes
- describe_load_balancer_policies
- describe_load_balancer_policy_types
- describe_load_balancers
- describe_tags
- detach_load_balancer_from_subnets
- disable_availability_zones_for_load_balancer
- enable_availability_zones_for_load_balancer
- modify_load_balancer_attributes
- register_instances_with_load_balancer
- remove_tags
- set_load_balancer_listener_ssl_certificate
- set_load_balancer_policies_for_backend_server
- set_load_balancer_policies_of_listener

Adds the specified tags to the specified load balancer
Associates one or more security groups with your load balancer in a virtual private cloud (VPC)
Adds one or more subnets to the set of configured subnets for the specified load balancer
Specifies the health check settings to use when evaluating the health state of your EC2 instances
Generates a stickiness policy with sticky session lifetimes that follow that of an application-generated cookie
Generates a stickiness policy with sticky session lifetimes controlled by the browser
Creates a Classic Load Balancer
Creates one or more listeners for the specified load balancer
Creates a policy with the specified attributes for the specified load balancer
Deletes the specified load balancer
Deletes the specified listeners from the specified load balancer
Deletes the specified policy from the specified load balancer
Deregisters the specified instances from the specified load balancer
Describes the current Elastic Load Balancing resource limits for your AWS account
Describes the state of the specified instances with respect to the specified load balancer
Describes the attributes for the specified load balancer
Describes the specified load balancer policies
Describes the specified load balancer policy types or all load balancer policy types
Describes the specified load balancers
Describes the tags associated with the specified load balancers
Removes the specified subnets from the set of configured subnets for the load balancer
Removes the specified Availability Zones from the set of Availability Zones for the load balancer
Adds the specified Availability Zones to the set of Availability Zones for the load balancer
Modifies the attributes of the specified load balancer
Adds the specified instances to the specified load balancer
Removes one or more tags from the specified load balancer
Sets the certificate that terminates the specified listener’s SSL connections
Replaces the set of policies associated with the specified port on which the listener is configured
Replaces the current set of policies for the specified load balancer port with new policies

**Examples**

```r
# This example adds two tags to the specified load balancer.
## Not run: svc <- elb()
svc$add_tags(
  LoadBalancerNames = list(  
    "my-load-balancer"
  ),
  Tags = list(  
    list(  
```
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. This reference covers Application Load Balancers and Network 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). For more information, see the Elastic Load Balancing User Guide.

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

Usage

```python
elbv2(config = list())
```

Arguments

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

```r
svc <- elbv2(
  config = 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 or TLS 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 or TLS 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`: Replaces the specified properties of the specified listener
- `modify_load_balancer_attributes`: Replaces the specified properties of the specified Application Load Balancer or Network Load Balancer
- `modify_rule`: Modifies the specified rules of the specified Application Load Balancer or Network Load Balancer
- `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 or TLS listener
- `remove_tags`: Removes the specified tags from the specified Elastic Load Balancing resource
emr

set_ip_address_type
set_rule_priorities
set_security_groups
set_subnets

Sets the type of IP addresses used by the subnets of the specified Application Load Balancer.

Sets the priorities of the specified rules.

Associates the specified security groups with the specified Application Load Balancer.

Enables the Availability Zones for the specified public subnets for the specified load balancer.

Examples

# This example adds the specified tags to the specified load balancer.
## Not run: svc <- elbv2()
svc$add_tags(
  ResourceArns = list(
  ),
  Tags = list(
    list(
      Key = "project",
      Value = "lima"
    ),
    list(
      Key = "department",
      Value = "digital-media"
    )
  )
)
## End(Not run)

emr

Amazon Elastic MapReduce

Description

Amazon EMR is a web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several AWS products to do tasks such as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing.

Usage

emr(config = list())

Arguments

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

```r
svc <- emr(
    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_instance_fleet`: Adds an instance fleet to a running cluster
- `add_instance_groups`: Adds one or more instance groups to a running cluster
- `add_job_flow_steps`: AddJobFlowSteps adds new steps to a running cluster
- `add_tags`: Adds tags to an Amazon EMR resource
- `cancel_steps`: Cancels a pending step or steps in a running cluster
- `create_security_configuration`: Creates a security configuration, which is stored in the service and can be specified when a cluster is created
- `delete_security_configuration`: Deletes a security configuration
- `describe_cluster`: Provides cluster-level details including status, hardware and software configuration, VPC settings, and so on
- `describe_job_flows`: This API is deprecated and will eventually be removed
- `describe_security_configuration`: Provides the details of a security configuration by returning the configuration JSON
- `describe_step`: Provides more detail about the cluster step
- `get_block_public_access_configuration`: Returns the Amazon EMR block public access configuration for your AWS account
- `list_bootstrap_actions`: Provides information about the bootstrap actions associated with a cluster
- `list_clusters`: Provides the status of all clusters visible to this AWS account
- `list_instance_fleets`: Lists all available details about the instance fleets in a cluster
- `list_instance_groups`: Provides all available details about the instance groups in a cluster
- `list_instances`: Provides all available details about the instance groups in a cluster
- `list_security_configurations`: Lists all the security configurations visible to this account, providing their creation dates and times, and their names
- `list_steps`: Provides a list of steps for the cluster in reverse order unless you specify stepIds with the request or filter by StepStates
- `modify_cluster`: Modifies the number of steps that can be executed concurrently for the cluster specified
- `modify_instance_fleet`: Modifies the target On-Demand and target Spot capacities for the instance fleet with the specified InstanceFleetID within the cluster specified
- `modify_instance_groups`: ModifyInstanceGroups modifies the number of nodes and configuration settings of an instance group
- `put_auto_scaling_policy`: Creates or updates an automatic scaling policy for a core instance group or task instance group
- `put_block_public_access_configuration`: Creates or updates an Amazon EMR block public access configuration for your AWS account
- `remove_auto_scaling_policy`: Removes an automatic scaling policy from a specified instance group within an EMR cluster
- `remove_tags`: Removes tags from an Amazon EMR resource
- `run_job_flow`: RunJobFlow creates and starts running a new cluster (job flow)
- `set_termination_protection`: SetTerminationProtection locks a cluster (job flow) so the EC2 instances in the cluster cannot be terminated by user intervention, an API call, or in the event of a job-flow error
- `set_visible_to_all_users`: Sets the Cluster$VisibleToAllUsers value, which determines whether the cluster is visible to all users
- `terminate_job_flows`: TerminateJobFlows shuts a list of clusters (job flows) down
eventbridge

Example

```r
## Not run: svc <- emr()
svc$add_instance_fleet(
  Foo = 123
)
## End(Not run)
```

### Amazon EventBridge

**Description**

Amazon EventBridge helps you to respond to state changes in your AWS resources. When your resources change state, they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a predetermined schedule. For example, you can configure rules to:

- Automatically invoke an AWS Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the running state
- Direct specific API records from AWS CloudTrail to an Amazon Kinesis data stream for detailed analysis of potential security or availability risks
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume

For more information about the features of Amazon EventBridge, see the Amazon EventBridge User Guide.

**Usage**

```
eventbridge(config = list())
```

**Arguments**

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

**Service syntax**

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

Operations

activate_event_source Activates a partner event source that has been deactivated
create_event_bus Creates a new event bus within your account
create_partner_event_source Called by an SaaS partner to create a partner event source
deactivate_event_source An AWS customer uses this operation to temporarily stop receiving events from the specified custom event bus or partner event bus
delete_event_bus This operation is used by SaaS partners to delete a partner event source
delete_partner_event_source Deletes the specified rule
describe_event_bus Displays details about an event bus in your account
describe_event_source This operation lists details about a partner event source that is shared with your account
describe_partner_event_source An SaaS partner can use this operation to list details about a partner event source that they have created
describe_rule Describes the specified rule
disable_rule Disables the specified rule
enable_rule Enables the specified rule
list_event_buses Lists all the event buses in your account, including the default event bus, custom event buses, and partner event buses
list_event_sources You can use this to see all the partner event sources that have been shared with your AWS account
list_partner_event_source_accounts An SaaS partner can use this operation to display the AWS account ID that a particular partner event source is associated with
list_partner_event_sources An SaaS partner can use this operation to list all the partner event sources that they have created
list_rule_names_by_target Lists the rules for the specified target
list_rules Lists your EventBridge rules
list_tags_for_resource Displays the tags associated with an EventBridge resource
list_targets_by_rule Lists the targets assigned to the specified rule
put_events Sends custom events to EventBridge so that they can be matched to rules
put_events This is used by SaaS partners to write events to a customer’s partner event bus
put_permission Running PutPermission permits the specified AWS account or AWS organization to put events to the specified event bus
put_rule Creates or updates the specified rule
put_targets Adds the specified targets to the specified rule, or updates the targets if they’re already associated with the rule
remove_permission Revokes the permission of another AWS account to be able to put events to the specified event bus
remove_targets Removes the specified targets from the specified rule
tag_resource Assigns one or more tags (key-value pairs) to the specified EventBridge resource
untag_resource Removes one or more tags from the specified EventBridge resource
test_event_pattern Tests whether the specified event pattern matches the provided event

Examples

```r
## Not run: svc <- eventbridge()
svc$activate_event_source(
    Foo = 123
)```
Amazon Kinesis Data Firehose

Description

Amazon Kinesis Data Firehose API Reference

Amazon Kinesis Data Firehose is a fully managed service that delivers real-time streaming data to
destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Elasticsearch Service
(Amazon ES), Amazon Redshift, and Splunk.

Usage

```r
firehose(config = list())
```

Arguments

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

Service syntax

```r
svc <- firehose(
  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_delivery_stream**: Creates a Kinesis Data Firehose delivery stream
- **delete_delivery_stream**: Deletes a delivery stream and its data
- **describe_delivery_stream**: Describes the specified delivery stream and its status
- **list_delivery_streams**: Lists your delivery streams in alphabetical order of their names
- **list_tags_for_delivery_stream**: Lists the tags for the specified delivery stream
- **put_record**: Writes a single data record into an Amazon Kinesis Data Firehose delivery stream
put_record_batch
start_delivery_stream_encryption
stop_delivery_stream_encryption
tag_delivery_stream
untag_delivery_stream
update_destination

Writes multiple data records into a delivery stream in a single call, which can achieve higher throughput per producer than when writing single records
Enables server-side encryption (SSE) for the delivery stream
Disables server-side encryption (SSE) for the delivery stream
Adds or updates tags for the specified delivery stream
Removes tags from the specified delivery stream
Updates the specified destination of the specified delivery stream

Examples

```r
## Not run: svc <- firehose()
svc$create_delivery_stream(
  Foo = 123
)
## End(Not run)
```

---

### Description

AWS Firewall Manager

This is the AWS Firewall Manager API Reference. This guide is for developers who need detailed information about the AWS Firewall Manager API actions, data types, and errors. For detailed information about AWS Firewall Manager features, see the AWS Firewall Manager Developer Guide.

### Usage

```r
fms(config = list())
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

### Service syntax

```r
csvc <- fms(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    profile = "string"
  )
)
```
Operations

- **associate_admin_account**
  Sets the AWS Firewall Manager administrator account
- **delete_notification_channel**
  Deletes an AWS Firewall Manager association with the IAM role and the Amazon Simple Notification Service (SNS) topic that is used to record SNS logs
- **delete_policy**
  Permanently deletes an AWS Firewall Manager policy
- **disassociate_admin_account**
  Disassociates the account that has been set as the AWS Firewall Manager administrator account
- **get_admin_account**
  Returns the AWS Organizations master account that is associated with AWS Firewall Manager
- **get_compliance_detail**
  Returns detailed compliance information about the specified member account
- **get_notification_channel**
  Information about the Amazon Simple Notification Service (SNS) topic that is used to record AWS Firewall Manager SNS logs
- **get_policy**
  Returns information about the specified AWS Firewall Manager policy
- **get_protection_status**
  If you created a Shield Advanced policy, returns policy-level attack summary information in the event of a potential DDoS attack
- **list_compliance_status**
  Returns an array of PolicyComplianceStatus objects in the response
- **list_member_accounts**
  Returns a MemberAccounts object that lists the member accounts in the administrator's AWS organization
- **list_policies**
  Returns an array of PolicySummary objects in the response
- **list_tags_for_resource**
  Retrieves the list of tags for the specified AWS resource
- **put_notification_channel**
  Designates the IAM role and Amazon Simple Notification Service (SNS) topic that AWS Firewall Manager uses to record SNS logs
- **put_policy**
  Creates an AWS Firewall Manager policy
- **tag_resource**
  Adds one or more tags to an AWS resource
- **untag_resource**
  Removes one or more tags from an AWS resource

Examples

```r
## Not run: svc <- fms()
svc$associate_admin_account(
  Foo = 123
)
## End(Not run)
```

---

**fsx**

**Amazon FSx**

**Description**

Amazon FSx is a fully managed service that makes it easy for storage and application administrators to launch and use shared file storage.

**Usage**

```r
fsx(config = list())
```
Arguments

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

Service syntax

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

Operations

cancel_data_repository_task Cancels an existing Amazon FSx for Lustre data repository task if that task is in either the PENDING or EXECUTING state
create_backup Creates a backup of an existing Amazon FSx for Windows File Server file system
create_data_repository_task Creates an Amazon FSx for Lustre data repository task
create_file_system Creates a new, empty Amazon FSx file system
create_file_system_from_backup Creates a new Amazon FSx file system from an existing Amazon FSx for Windows File Server file system
delete_backup Deletes an Amazon FSx for Windows File Server backup, deleting its contents
delete_file_system Deletes a file system, deleting its contents
describe_backups Returns the description of specific Amazon FSx for Windows File Server backups, if a BackupIds value is provided for that backup
describe_data_repository_tasks Returns the description of specific Amazon FSx for Lustre data repository tasks, if one or more TaskIds values are provided in the request, or if filters are used in the request
describe_file_systems Returns the description of specific Amazon FSx file systems, if a FileSystemIds value is provided for that file system
describe_tags_for_resource Lists tags for an Amazon FSx file system and backups in the case of Amazon FSx for Windows File Server
tag_resource This action removes a tag from an Amazon FSx resource
untag_resource Updates a file system configuration

Examples

# This operation creates a new backup.
## Not run: svc <- fsx()
svc$create_backup(
  FileSystemId = "fs-0498eed5fe91001ec",
  Tags = list(
    list(
      Key = "Name",
```
Description

Amazon S3 Glacier (Glacier) is a storage solution for "cold data."

Glacier is an extremely low-cost storage service that provides secure, durable, and easy-to-use storage for data backup and archival. With Glacier, customers can store their data cost effectively for months, years, or decades. Glacier also enables customers to offload the administrative burdens of operating and scaling storage to AWS, so they don't have to worry about capacity planning, hardware provisioning, data replication, hardware failure and recovery, or time-consuming hardware migrations.

Glacier is a great storage choice when low storage cost is paramount and your data is rarely retrieved. If your application requires fast or frequent access to your data, consider using Amazon S3. For more information, see Amazon Simple Storage Service (Amazon S3).

You can store any kind of data in any format. There is no maximum limit on the total amount of data you can store in Glacier.

If you are a first-time user of Glacier, we recommend that you begin by reading the following sections in the Amazon S3 Glacier Developer Guide:

- **What is Amazon S3 Glacier** - This section of the Developer Guide describes the underlying data model, the operations it supports, and the AWS SDKs that you can use to interact with the service.
- **Getting Started with Amazon S3 Glacier** - The Getting Started section walks you through the process of creating a vault, uploading archives, creating jobs to download archives, retrieving the job output, and deleting archives.

Usage

```
glacier(config = list())
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>
Service syntax

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

Operations

- abort_multipart_upload
- abort_vault_lock
- add_tags_to_vault
- complete_multipart_upload
- complete_vault_lock
- create_vault
- delete_archive
- delete_vault
- delete_vault_access_policy
- delete_vault_notifications
- describe_job
- describe_vault
- get_data_retrieval_policy
- get_job_output
- get_vault_access_policy
- get_vault_lock
- get_vault_notifications
- initiate_job
- initiate_multipart_upload
- initiate_vault_lock
- list_jobs
- list_multipart_uploads
- list_parts
- list_provisioned_capacity
- list_tags_for_vault
- list_vaults
- purchase_provisioned_capacity
- remove_tags_from_vault
- set_data_retrieval_policy
- set_vault_access_policy

- This operation aborts a multipart upload identified by the upload ID
- This operation aborts the vault locking process if the vault lock is not in the Locked state
- This operation adds the specified tags to a vault
- You call this operation to inform Amazon S3 Glacier (Glacier) that all the archive parts have been uploaded
- This operation completes the vault locking process by transitioning the vault lock from the InProgress state to the Locked state
- This operation creates a new vault with the specified name
- This operation deletes an archive from a vault
- This operation deletes a vault
- This operation deletes the access policy associated with the specified vault
- This operation returns the notification configuration set for a vault
- This operation returns information about a job you previously initiated, including the job initiation date and job status code
- This operation returns information about a vault, including the vault’s Amazon Resource Name (ARN)
- This operation returns the current data retrieval policy for the account and region specified in the request
- This operation downloads the output of the job you initiated using InitiateJob
- This operation retrieves the access-policy subresource set on the vault; for more information on setting this subresource, see Set Vault Access Policy (PUT access-policy)
- This operation retrieves the following attributes from the lock-policy subresource set on the specified vault
- This operation retrieves the notification-configuration subresource of the specified vault
- This operation initiates a job of the specified type, which can be a select, an archival retrieval, or a vault retrieval
- This operation initiates a multipart upload
- This operation initiates the vault locking process by doing the following: - Installing a vault lock policy on the specified vault
- This operation lists jobs for a vault, including jobs that are in-progress and jobs that have recently completed
- This operation lists in-progress multipart uploads for the specified vault
- This operation lists the parts of an archive that have been uploaded in a specific multipart upload
- This operation lists the provisioned capacity units for the specified AWS account
- This operation lists all the tags attached to a vault
- This operation purchases a provisioned capacity unit for an AWS account
- This operation removes one or more tags from the set of tags attached to a vault
- This operation sets and then enacts a data retrieval policy in the region specified in the PUT request
- This operation configures an access policy for a vault and will overwrite an existing policy
Examples

# The example deletes an in-progress multipart upload to a vault named # my-vault:
## Not run: svc <- glacier()
svc$abort_multipart_upload(
  accountId = "-",
  uploadId = "19gaRezEXAMPLES6Ry5YydqthHOC_kGRCT03L9yetr2280UmPtBYKk-OssZtLqyFu7sY1_1R7vgFuJV...",
  vaultName = "my-vault"
)
## End(Not run)
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
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

```r
## Not run: svc <- globalaccelerator()
svc$create_accelerator(
  Foo = 123
)
## End(Not run)
```

Description

Defines the public endpoint for the AWS Glue service.
Usage

```r
glue(config = list())
```

Arguments

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

Service syntax

```r
csvc <- glue(
c  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

- `batch_create_partition`: Creates one or more partitions in a batch operation
- `batch_delete_connection`: Deletes a list of connection definitions from the Data Catalog
- `batch_delete_partition`: Deletes one or more partitions in a batch operation
- `batch_delete_table`: Deletes multiple tables at once
- `batch_delete_table_version`: Deletes a specified batch of versions of a table
- `batch_get_crawlers`: Returns a list of resource metadata for a given list of crawler names
- `batch_get_dev_endpoints`: Returns a list of resource metadata for a given list of development endpoint names
- `batch_get_jobs`: Returns a list of resource metadata for a given list of job names
- `batch_get_partition`: Retrieves partitions in a batch request
- `batch_get_triggers`: Returns a list of resource metadata for a given list of trigger names
- `batch_get_workflows`: Returns a list of resource metadata for a given list of workflow names
- `batch_stop_job_run`: Stops one or more job runs for a specified job definition
- `cancel_ml_task_run`: Cancels (stops) a task run
- `create_classifier`: Creates a classifier in the user’s account
- `create_connection`: Creates a connection definition in the Data Catalog
- `create_crawler`: Creates a new crawler with specified targets, role, configuration, and optional schedule
- `create_database`: Creates a new database in a Data Catalog
- `create_dev_endpoint`: Creates a new development endpoint
- `create_job`: Creates a new job definition
- `create_ml_transform`: Creates an AWS Glue machine learning transform
- `create_partition`: Creates a new partition
- `create_script`: Transforms a directed acyclic graph (DAG) into code
create_security_configuration
create_table
create_trigger
create_user_defined_function
create_workflow
delete_classifier
delete_connection
delete_crawler
delete_database
delete_dev_endpoint
delete_job
delete_ml_transform
delete_partition
delete_resource_policy
delete_security_configuration
delete_table
delete_table_version
delete_trigger
delete_user_defined_function
delete_workflow
get_catalog_import_status
get_classifier
get_classifiers
get_connection
get_connections
get_crawler
get_crawler_metrics
get_crawlers
get_database
get_databases
get_data_catalog_encryption_settings
get_dataflow_graph
get_dev_endpoint
get_dev_endpoints
get_job
get_job_bookmark
get_job_run
get_job_runs
get_jobs
get_mapping
get_ml_task_run
get_ml_task_runs
get_ml_transform
get_ml_transforms
get_partition
get_partitions
get_plan
get_resource_policy

Creates a new security configuration
Creates a new table definition in the Data Catalog
Creates a new trigger
Creates a new function definition in the Data Catalog
Creates a new workflow
Removes a classifier from the Data Catalog
Deletes a connection from the Data Catalog
Removes a specified crawler from the AWS Glue Data Catalog, unless the crawler state is RUNNING
Removes a specified database from a Data Catalog
Deletes a specified development endpoint
Deletes a specified job definition
Deletes an AWS Glue machine learning transform
Deletes a specified partition
Deletes a specified policy
Deletes a specified security configuration
Removes a table definition from the Data Catalog
Deletes a specified version of a table
Deletes a specified trigger
Deletes an existing function definition from the Data Catalog
Deletes a workflow
Retrieves the status of a migration operation
Retrieve a classifier by name
Lists all classifier objects in the Data Catalog
 Retrieves a connection definition from the Data Catalog
Retrieves a list of connection definitions from the Data Catalog
Retrieves metadata for a specified crawler
Retrieves metrics about specified crawlers
Retrieves metadata for all crawlers defined in the customer account
Retrieves the definition of a specified database
Retrieves all databases defined in a given Data Catalog
Retrieves the security configuration for a specified catalog
Transforms a Python script into a directed acyclic graph (DAG)
Retrieves information about a specified development endpoint
Retrieves all the development endpoints in this AWS account
Retrieves an existing job definition
Returns information on a job bookmark entry
Retrieves the metadata for a given job run
Retrieves metadata for all runs of a given job definition
Retrieves all current job definitions
Creates mappings
Gets details for a specific task run on a machine learning transform
Gets a list of runs for a machine learning transform
Gets an AWS Glue machine learning transform artifact and all its corresponding metadata
Gets a sortable, filterable list of existing AWS Glue machine learning transforms
Retrieves information about a specified partition
Retrieves information about the partitions in a table
Gets code to perform a specified plan
Retrieves a specified resource policy
get_security_configuration
get_security_configurations
get_table
get_tables
get_table_version
get_table_versions
get_tags
get_trigger
get_triggers
get_user_defined_function
get_user_defined_functions
get_workflow
get_workflow_run
get_workflow_run_properties
get_workflow_runs
import_catalog_to_glue
list_crawlers
list_dev_endpoints
list_jobs
list_triggers
list_workflows
put_data_catalog_encryption_settings
put_resource_policy
put_workflow_run_properties
reset_job_bookmark
search_tables
start_crawler
start_crawler_schedule
start_export_labels_task_run
start_import_labels_task_run
start_job_run
start_ml_evaluation_task_run
start_ml_labeling_set_generation_task_run
start_trigger
start_workflow_run
stop_crawler
stop_crawler_schedule
stop_trigger
tag_resource
untag_resource
update_classifier
update_connection
update_crawler
update_crawler_schedule
update_database
update_dev_endpoint
update_job
update_ml_transform

Retrieves a specified security configuration
Retrieves a list of all security configurations
Retrieves the Table definition in a Data Catalog for a specified table
Retrieves the definitions of some or all of the tables in a given Database
Retrieves a specified version of a table
Retrieves a list of strings that identify available versions of a specified table
Retrieves a list of tags associated with a resource
Retrieves the definition of a trigger
Gets all the triggers associated with a job
Retrieves a specified function definition from the Data Catalog
Retrieves multiple function definitions from the Data Catalog
Retrieves resource metadata for a workflow
Retrieves the metadata for a given workflow run
Retrieves the workflow run properties which were set during the run
Retrieves metadata for all runs of a given workflow
Imports an existing Amazon Athena Data Catalog to AWS Glue
Retrieves the names of all crawler resources in this AWS account, or the resources with the specified tag
Retrieves the names of all DevEndpoint resources in this AWS account, or the resources with the specified tag
Retrieves the names of all job resources in this AWS account, or the resources with the specified tag
Retrieves the names of all trigger resources in this AWS account, or the resources with the specified tag
Lists names of workflows created in the account
Sets the security configuration for a specified catalog
Sets the Data Catalog resource policy for access control
Puts the specified workflow run properties for the given workflow run
Resets a bookmark entry
Searches a set of tables based on properties in the table metadata as well as on the table's parent database
Starts a crawl using the specified crawler, regardless of what is scheduled
Changes the schedule state of the specified crawler to SCHEDULED, unless the crawler is already running or its schedule state is SCHEDULED
Begins an asynchronous task to export all labeled data for a particular transform
Enables you to provide additional labels (examples of truth) to be used to teach the machine learning transform and improve its quality
Starts a job run using a job definition
Starts a task to estimate the quality of the transform
Starts the active learning workflow for your machine learning transform to improve its quality
Starts a new run of the specified workflow
If the specified crawler is running, stops the crawl
Sets the schedule state of the specified crawler to NOT_SCHEDULED, but does not stop the crawl
Stops a specified trigger
Adds tags to a resource
Removes tags from a resource
Modifies an existing classifier (a GrokClassifier, an XMLClassifier, a JsonClassifier, or a CsvClassifier, depending on which field is present)
Updates a connection definition in the Data Catalog
Updates a crawler
Updates the schedule of a crawler using a cron expression
Updates an existing database definition in a Data Catalog
Updates a specified development endpoint
Updates an existing job definition
Updates an existing machine learning transform
update_partition Updates a partition
update_table Updates a metadata table in the Data Catalog
update_trigger Updates a trigger definition
update_user_defined_function Updates an existing function definition in the Data Catalog
update_workflow Updates an existing workflow

Examples

```r
## Not run: svc <- glue()
svc$batch_create_partition(
   Foo = 123
)
## End(Not run)
```

guardduty

Amazon GuardDuty

Description

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following data sources: VPC Flow Logs, AWS CloudTrail event logs, and DNS logs. It uses threat intelligence feeds, such as lists of malicious IPs and domains, and machine learning to identify unexpected and potentially unauthorized and malicious activity within your AWS environment. This can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IPs, URLs, or domains. For example, GuardDuty can detect compromised EC2 instances serving malware or mining bitcoin. It also monitors AWS account access behavior for signs of compromise, such as unauthorized infrastructure deployments, like instances deployed in a region that has never been used, or unusual API calls, like a password policy change to reduce password strength. GuardDuty informs you of the status of your AWS environment by producing security findings that you can view in the GuardDuty console or through Amazon CloudWatch events. For more information, see Amazon GuardDuty User Guide.

Usage

guardduty(config = list())

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>
Service syntax

```r
svc <- guardduty(
  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_invitation`: Accepts the invitation to be monitored by a master GuardDuty account
- `archive_findings`: Archives GuardDuty findings specified by the list of finding IDs
- `create_detector`: Creates a single Amazon GuardDuty detector
- `create_filter`: Creates a filter using the specified finding criteria
- `create_ip_set`: Creates a new IPSet, called Trusted IP list in the console user interface
- `create_members`: Creates member accounts of the current AWS account by specifying a list of AWS account IDs
- `create_publishing_destination`: Creates a publishing destination to send findings to
- `create_sample_findings`: Generates example findings of types specified by the list of finding types
- `create_threat_intel_set`: Create a new ThreatIntelSet
- `decline_invitations`: Declines invitations sent to the current member account by AWS accounts specified by their account IDs
- `delete_detector`: Deletes a Amazon GuardDuty detector specified by the detector ID
- `delete_filter`: Deletes the filter specified by the filter name
- `delete_invitations`: Deletes invitations sent to the current member account by AWS accounts specified by their account IDs
- `delete_ip_set`: Deletes the IPSet specified by the ipSetId
- `delete_members`: Deletes GuardDuty member accounts (to the current GuardDuty master account) specified by the account IDs
- `delete_publishing_destination`: Deletes the publishing definition with the specified destinationId
- `delete_threat_intel_set`: Deletes ThreatIntelSet specified by the ThreatIntelSet ID
- `describe_publishing_destination`: Returns information about the publishing destination specified by the provided destinationId
- `disassociate_from_master_account`: Disassociates the current GuardDuty member account from its master account
- `disassociate_members`: Disassociates GuardDuty member accounts (to the current GuardDuty master account) specified by the account IDs
- `get_detector`: Retrieves an Amazon GuardDuty detector specified by the detectorId
- `get_filter`: Returns the details of the filter specified by the filter name
- `get_findings`: Describes Amazon GuardDuty findings specified by finding IDs
- `get_findings_statistics`: Lists Amazon GuardDuty findings’ statistics for the specified detector ID
- `get_ip_set`: Retrieves the count of all GuardDuty membership invitations that were sent to the current member account
- `get_master_account`: Provides the details for the GuardDuty master account associated with the current GuardDuty member account
- `get_members`: Retrieves GuardDuty member accounts (to the current GuardDuty master account) specified by the account IDs
- `get_threat_intel_set`: Retrieves the ThreatIntelSet that is specified by the ThreatIntelSet ID
- `invite_members`: Invites other AWS accounts (created as members of the current AWS account by CreateMember)
list_detectorIds of all the existing Amazon GuardDuty detector resources

list_filters
Returns a paginated list of the current filters

list_findings
Lists Amazon GuardDuty findings for the specified detector ID

list_invitations
Lists all GuardDuty membership invitations that were sent to the current AWS account

list_ip_sets
Lists the IPSets of the GuardDuty service specified by the detector ID

list_members
Lists details about all member accounts for the current GuardDuty master account

list_publishing_destinations
Returns a list of publishing destinations associated with the specified detector ID

list_tags_for_resource
Lists tags for a resource

list_threat_intel_sets
Lists the ThreatIntelSets of the GuardDuty service specified by the detector ID

start_monitoring_members
Turns on GuardDuty monitoring of the specified member accounts

stop_monitoring_members
Stops GuardDuty monitoring for the specified member accounts

tag_resource
Adds tags to a resource

untag_resource
Removes tags from a resource

unarchive_findings
Unarchives GuardDuty findings specified by the findingIds

update_detector
Updates the Amazon GuardDuty detector specified by the detectorId

update_filter
Updates the filter specified by the filter name

update_findings_feedback
Marks the specified GuardDuty findings as useful or not useful

update_ip_set
Updates the IPSet specified by the IPSet ID

update_publishing_destination
Updates information about the publishing destination specified by the destinationId

update_threat_intel_set
Updates the ThreatIntelSet specified by ThreatIntelSet ID

Examples

```r
data <- list()
data$accept_invitation(Foo = 123)
data$update_detector(detectorId = "example")
data$unarchive_findings(findingIds = c("1", "2"))
data$untag_resource(resourceId = "example", tagKeys = c("key1", "key2"))
data$start_monitoring_members(accounts = c("account1", "account2"))
data$stop_monitoring_members(accounts = c("account1", "account2"))
data$tag_resource(resourceId = "example", tagKeys = c("key1", "key2"))
data$untag_resource(resourceId = "example", tagKeys = c("key1", "key2"))
```

Description

AWS Health

The AWS Health API provides programmatic access to the AWS Health information that is presented in the AWS Personal Health Dashboard. You can get information about events that affect your AWS resources:

- **DescribeEvents**: Summary information about events.
- **DescribeEventDetails**: Detailed information about one or more events.
- **DescribeAffectedEntities**: Information about AWS resources that are affected by one or more events.
In addition, these operations provide information about event types and summary counts of events or affected entities:

- **DescribeEventTypes**: Information about the kinds of events that AWS Health tracks.
- **DescribeEventAggregates**: A count of the number of events that meet specified criteria.
- **DescribeEntityAggregates**: A count of the number of affected entities that meet specified criteria.

AWS Health integrates with AWS Organizations to provide a centralized view of AWS Health events across all accounts in your organization.

- **DescribeEventsForOrganization**: Summary information about events across the organization.
- **DescribeAffectedAccountsForOrganization**: List of accounts in your organization impacted by an event.
- **DescribeEventDetailsForOrganization**: Detailed information about events in your organization.
- **DescribeAffectedEntitiesForOrganization**: Information about AWS resources in your organization that are affected by events.

You can use the following operations to enable or disable AWS Health from working with AWS Organizations.

- **EnableHealthServiceAccessForOrganization**: Enables AWS Health to work with AWS Organizations.
- **DisableHealthServiceAccessForOrganization**: Disables AWS Health from working with AWS Organizations.
- **DescribeHealthServiceStatusForOrganization**: Status information about enabling or disabling AWS Health from working with AWS Organizations.

The Health API requires a Business or Enterprise support plan from AWS Support. Calling the Health API from an account that does not have a Business or Enterprise support plan causes a `SubscriptionRequiredException`.

For authentication of requests, AWS Health uses the **Signature Version 4 Signing Process**.

See the [AWS Health User Guide](https://docs.aws.amazon.com/health/latest/userguide/) for information about how to use the API.

**Service Endpoint**

The HTTP endpoint for the AWS Health API is:

- `https://health.us-east-1.amazonaws.com`

**Usage**

```python
health(config = list())
```

**Arguments**

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

Service syntax

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

Operations

- `describe_affected_accounts_for_organization`: Returns a list of accounts in the organization from AWS Organizations that are affected by the provided event.
- `describe_affected_entities`: Returns a list of entities that have been affected by the specified events, based on the filter criteria.
- `describe_affected_entities_for_organization`: Returns a list of entities that have been affected by one or more events for one or more accounts in your organization in AWS Organizations, based on the filter criteria.
- `describe_entity_aggregates`: Returns the number of entities that are affected by each of the specified events.
- `describe_event_aggregates`: Returns the number of events of each event type (issue, scheduled change, and account notification).
- `describe_event_details`: Returns detailed information about one or more specified events.
- `describe_event_details_for_organization`: Returns detailed information about one or more specified events for one or more accounts in your organization.
- `describe_events`: Returns information about events that meet the specified filter criteria.
- `describe_events_for_organization`: Returns information about events across your organization in AWS Organizations that meet the specified filter criteria.
- `describe_event_types`: Returns the event types that meet the specified filter criteria.
- `describe_health_service_status_for_organization`: This operation provides status information on enabling or disabling AWS Health to work with your organization.
- `disable_health_service_access_for_organization`: Calling this operation disables Health from working with AWS Organizations.
- `enable_health_service_access_for_organization`: Calling this operation enables AWS Health to work with AWS Organizations.

Examples

```r
class(svc) <- "health"
svc$describe_affected_accounts_for_organization(
  Foo = 123
)
```

IAM

AWS Identity and Access Management
Description

AWS Identity and Access Management (IAM) is a web service that you can use to manage users and user permissions under your AWS account. This guide provides descriptions of IAM actions that you can call programmatically. For general information about IAM, see AWS Identity and Access Management (IAM). For the user guide for IAM, see Using IAM.

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to IAM and AWS. For example, the SDKs take care of tasks such as cryptographically signing requests (see below), managing errors, and retrying requests automatically. For information about the AWS SDKs, including how to download and install them, see the Tools for Amazon Web Services page.

We recommend that you use the AWS SDKs to make programmatic API calls to IAM. However, you can also use the IAM Query API to make direct calls to the IAM web service. To learn more about the IAM Query API, see Making Query Requests in the Using IAM guide. IAM supports GET and POST requests for all actions. That is, the API does not require you to use GET for some actions and POST for others. However, GET requests are subject to the limitation size of a URL. Therefore, for operations that require larger sizes, use a POST request.

Signing Requests

Requests must be signed using an access key ID and a secret access key. We strongly recommend that you do not use your AWS account access key ID and secret access key for everyday work with IAM. You can use the access key ID and secret access key for an IAM user or you can use the AWS Security Token Service to generate temporary security credentials and use those to sign requests.

To sign requests, we recommend that you use Signature Version 4. If you have an existing application that uses Signature Version 2, you do not have to update it to use Signature Version 4. However, some operations now require Signature Version 4. The documentation for operations that require version 4 indicate this requirement.

Additional Resources

For more information, see the following:

- AWS Security Credentials. This topic provides general information about the types of credentials used for accessing AWS.
- IAM Best Practices. This topic presents a list of suggestions for using the IAM service to help secure your AWS resources.
- Signing AWS API Requests. This set of topics walk you through the process of signing a request using an access key ID and secret access key.

Usage

```
iam(config = list())
```

Arguments

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

```r
svc <- iam(
  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_client_id_to_open_id_connect_provider` Adds a new client ID (also known as audience) to the list of client IDs associated with the specified IAM OpenID Connect (OIDC) provider.
- `add_role_to_instance_profile` Adds the specified IAM role to the specified instance profile.
- `add_user_to_group` Adds the specified user to the specified group.
- `attach_group_policy` Attaches the specified managed policy to the specified IAM group.
- `attach_role_policy` Attaches the specified managed policy to the specified IAM role.
- `attach_user_policy` Attaches the specified managed policy to the specified user.
- `change_password` Changes the password of the IAM user who is calling this operation.
- `create_access_key` Creates a new AWS secret access key and corresponding AWS access key ID for the specified user.
- `create_account_alias` Creates an alias for your AWS account.
- `create_group` Creates a new group.
- `create_instance_profile` Creates a new instance profile.
- `create_login_profile` Creates a password for the specified user, giving the user the ability to access AWS services through the AWS Management Console.
- `create_open_id_connect_provider` Creates an IAM entity to describe an identity provider (IdP) that supports OpenID Connect (OIDC).
- `create_policy` Creates a new managed policy for your AWS account.
- `create_policy_version` Creates a new version of the specified managed policy.
- `create_role` Creates a new role for your AWS account.
- `create_saml_provider` Creates an IAM resource that describes an identity provider (IdP) that supports SAML 2.0 authentication.
- `create_service_linked_role` Creates an IAM role that is linked to a specific AWS service.
- `create_service_specific_credential` Generates a set of credentials consisting of a user name and password that can be used to access the service specified in the request.
- `create_user` Creates a new IAM user for your AWS account.
- `create_virtual_mfa_device` Creates a new virtual MFA device for the AWS account.
- `deactivate_mfa_device` Deactivates the specified MFA device and removes it from association with the user name for which it was originally enabled.
- `delete_access_key` Deletes the access key pair associated with the specified IAM user.
- `delete_account_alias` Deletes the specified AWS account alias.
- `delete_account_password_policy` Deletes the password policy for the AWS account.
- `delete_group` Deletes the specified IAM group.
- `delete_group_policy` Deletes the specified inline policy that is embedded in the specified IAM group.
- `delete_instance_profile` Deletes the specified instance profile.
- `delete_login_profile` Deletes the password for the specified IAM user, which terminates the user's ability to access AWS services through the AWS Management Console.
- `delete_open_id_connect_provider` Deletes an OpenID Connect identity provider (IdP) resource object in IAM.
list_account_aliases
list_attached_group_policies
list_attached_role_policies
list_attached_user_policies
list_entities_for_policy
list_group_policies
list_groups
list_groups_for_user
list_instance_profiles
list_instance_profiles_for_role
list_mfa_devices
list_open_id_connect_providers
list_policies
list_policies_granting_service_access
list_policy_versions
list_role_policies
list_roles
list_role_tags
list_saml_providers
list_server_certificates
list_service_specific_credentials
list_signing_certificates
list_ssh_public_keys
list_user_policies
list_users
list_user_tags
list_virtual_mfa_devices
put_group_policy
put_role_permissions_boundary
put_role_policy
put_user_permissions_boundary
put_user_policy
remove_client_id_from_open_id_connect_provider
remove_role_from_instance_profile
remove_user_from_group
reset_service_specific_credential
reset_mfa_device
set_default_policy_version
set_security_token_service_preferences
simulate_custom_policy
simulate_principal_policy
tag_role
tag_user
untag_role
untag_user
update_access_key
update_account_password_policy
update_assume_role_policy

Lists the account alias associated with the AWS account (Note: you can have only one)
Lists all managed policies that are attached to the specified IAM group
Lists all managed policies that are attached to the specified IAM role
Lists all managed policies that are attached to the specified IAM user
Lists all IAM users, groups, and roles that the specified managed policy
Lists the names of the inline policies that are embedded in the specified group
Lists the IAM groups that have the specified path prefix
Lists the IAM groups that the specified IAM user belongs to
Lists the instance profiles that have the specified path prefix
Lists the instance profiles that have the specified associated IAM role
Lists the MFA devices for an IAM user
Lists all the managed policies that are available in your AWS account, including your own customer-defined managed policies and all AWS managed policies
Retrieves information about the IAM OpenID Connect (OIDC) provider resource objects defined in the AWS account
Retrieves a list of policies that the IAM identity (user, group, or role) can use to access each specified service
Lists information about the versions of the specified managed policy, including the version that is currently set as the policy’s default version
Lists the names of the inline policies that are embedded in the specified role
Lists the IAM roles that have the specified path prefix
Lists the IAM roles that are attached to the specified role
Lists the SAML provider resource objects defined in IAM in the account
Lists the server certificates stored in IAM that have the specified path prefix
Returns information about the service-specific credentials associated with the specified IAM user
Returns information about the signing certificates associated with the specified IAM user
Returns information about the SSH public keys associated with the specified IAM user
Lists the names of the inline policies embedded in the specified IAM user
Lists the IAM users that have the specified path prefix
Lists the IAM users that are attached to the specified user
Lists the virtual MFA devices defined in the AWS account by assignment
Adds or updates an inline policy document that is embedded in the specified group
Adds or updates the policy that is specified as the IAM role’s permission boundary
Adds or updates an inline policy document that is embedded in the specified role
Adds or updates the policy that is specified as the IAM user’s permission boundary
Adds or updates an inline policy document that is embedded in the specified role
Removes the specified client ID (also known as audience) from the list of registered audiences for the specified IAM OpenID Connect (OIDC) provider
Removes the specified IAM role from the specified EC2 instance profile
Removes the specified user from the specified group
Resets the password for a service-specific credential
Synchronizes the specified MFA device with its IAM resource object on the AWS servers
Sets the specified version of the specified policy as the policy’s default version
Sets the specified version of the global endpoint token as the token version used for the AWS account
Simulate how a set of IAM policies and optionally a resource-based policy works with a list of API operations and AWS resources to determine the policies’ effective permissions
Simulate how a set of IAM policies attached to an IAM entity works with a list of API operations and AWS resources to determine the policies’ effective permissions
Adds one or more tags to an IAM role
Adds one or more tags to an IAM user
Removes the specified tags from the role
Removes the specified tags from the user
Changes the status of the specified access key from Active to Inactive, or vice versa
Updates the password policy settings for the AWS account
Updates the policy that grants an IAM entity permission to assume a role
update_group
update_login_profile
update_open_id_connect_provider_thumbprint
update_role
update_role_description
update_saml_provider
update_server_certificate
update_service_specific_credential
update_signing_certificate
update_ssh_public_key
update_user
upload_server_certificate
upload_signing_certificate
upload_ssh_public_key

Examples

# The following add-client-id-to-open-id-connect-provider command adds the
# client ID my-application-ID to the OIDC provider named
# server.example.com:
## Not run: svc <- iam()
svc$add_client_id_to_open_id_connect_provider(
  ClientID = "my-application-ID",
  OpenIDConnectProviderArn = "arn:aws:iam::123456789012:oidc-provider/server.example.com"
)
## End(Not run)

inspector

Amazon Inspector

Description

Amazon Inspector enables you to analyze the behavior of your AWS resources and to identify potential security issues. For more information, see Amazon Inspector User Guide.

Usage

inspector(config = list())

Arguments

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

```r
svc <- inspector(
  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_attributes_to_findings`: Assigns attributes (key and value pairs) to the findings that are specified by the ARNs of the findings.
- `create_assessment_target`: Creates a new assessment target using the ARN of the resource group that is generated by CreateResourceGroup.
- `create_assessment_template`: Creates an assessment template for the assessment target that is specified by the ARN of the assessment target.
- `create_exclusions_preview`: Starts the generation of an exclusions preview for the specified assessment template.
- `create_resource_group`: Creates a resource group using the specified set of tags (key and value pairs) that are used to select the EC2 instances to be included in an Amazon Inspector assessment target.
- `delete_assessment_run`: Deletes the assessment run that is specified by the ARN of the assessment run.
- `delete_assessment_target`: Deletes the assessment target that is specified by the ARN of the assessment target.
- `delete_assessment_template`: Deletes the assessment template that is specified by the ARN of the assessment template.
- `describe_assessment_runs`: Describes the assessment runs that are specified by the ARNs of the assessment runs.
- `describe_assessment_targets`: Describes the assessment targets that are specified by the ARNs of the assessment targets.
- `describe_assessment_templates`: Describes the assessment templates that are specified by the ARNs of the assessment templates.
- `describe_cross_account_access_role`: Describes the IAM role that enables Amazon Inspector to access your AWS account.
- `describe_exclusions`: Describes the exclusions that are specified by the exclusions’ ARNs.
- `describe_findings`: Describes the findings that are specified by the ARNs of the findings.
- `describe_resource_groups`: Describes the resource groups that are specified by the ARNs of the resource groups.
- `describe_rules_packages`: Describes the rules packages that are specified by the ARNs of the rules packages.
- `get_assessment_report`: Produces an assessment report that includes detailed and comprehensive results of a specified assessment run.
- `get_exclusions_preview`: Retrieves the exclusions preview (a list of ExclusionPreview objects) specified by the preview token.
- `get_telemetry_metadata`: Information about the data that is collected for the specified assessment run.
- `list_assessment_run_agents`: Lists the agents of the assessment runs that are specified by the ARNs of the assessment runs.
- `list_assessment_runs`: Lists the assessment runs that correspond to the assessment templates that are specified by the ARNs of the assessment templates.
- `list_assessment_targets`: Lists the ARNs of the assessment targets within this AWS account.
- `list_assessment_templates`: Lists the assessment templates that correspond to the assessment targets that are specified by the ARNs of the assessment targets.
- `list_event_subscriptions`: Lists all the event subscriptions for the assessment template that is specified by the ARN of the assessment template.
- `list_exclusions`: Lists exclusions that are generated by the assessment run.
- `list_findings`: Lists findings that are generated by the assessment runs that are specified by the ARNs of the assessment run.
- `list_rules_packages`: Lists all available Amazon Inspector rules packages.
- `list_tags_for_resource`: Lists all tags associated with an assessment template.
- `preview_agents`: Prepares the agents installed on the EC2 instances that are part of the specified assessment target.
- `register_cross_account_access_role`: Registers the IAM role that grants Amazon Inspector access to AWS Services needed to perform security assessments.
remove_attributes_from_findings
set_tags_for_resource
start_assessment_run
stop_assessment_run
subscribe_to_event
unsubscribe_from_event
update_assessment_target

Removes entire attributes (key and value pairs) from the findings that are specified by the ARNs of the findings.
Sets tags (key and value pairs) to the assessment template that is specified by the ARN of the assessment template.
Starts the assessment run specified by the ARN of the assessment template.
Stops the assessment run that is specified by the ARN of the assessment run.
Enables the process of sending Amazon Simple Notification Service (SNS) notifications about a specified event to a specified SNS topic.
Disables the process of sending Amazon Simple Notification Service (SNS) notifications about a specified event to a specified SNS topic.
Updates the assessment target that is specified by the ARN of the assessment target.

Examples

```r
# Assigns attributes (key and value pairs) to the findings that are specified by the ARNs of the findings.
# Not run: svc <- inspector()
svc$add_attributes_to_findings(
  attributes = list(
    list(
      key = "Example",
      value = "example"
    )
  ),
  findingArns = list(
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z0..."
  )
)
## End(Not run)
```

---

**kafka**

*Managed Streaming for Kafka*

**Description**

Managed Streaming for Kafka

**Usage**

```r
kafka(config = list())
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>
Service syntax

```r
svc <- kafka(
  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_cluster` Creates a new MSK cluster
- `create_configuration` Creates a new MSK configuration
- `delete_cluster` Deletes the MSK cluster specified by the Amazon Resource Name (ARN) in the request
- `describe_cluster` Returns a description of the MSK cluster whose Amazon Resource Name (ARN) is specified in the request
- `describe_cluster_operation` Returns a description of the cluster operation specified by the ARN
- `describe_configuration` Returns a description of this MSK configuration
- `describe_configuration_revision` Returns a description of this revision of the configuration
- `get_bootstrap_brokers` A list of brokers that a client application can use to bootstrap
- `list_cluster_operations` Returns a list of all the operations that have been performed on the specified MSK cluster
- `list_clusters` Returns a list of all the MSK clusters in the current Region
- `list_configuration_revisions` Returns a list of all the MSK configurations in this Region
- `list_configurations` Returns a list of all the MSK configurations in this Region
- `list_nodes` Returns a list of the broker nodes in the cluster
- `list_tags_for_resource` Returns a list of the tags associated with the specified resource
- `tag_resource` Adds tags to the specified MSK resource
- `untag_resource` Removes the tags associated with the keys that are provided in the query
- `update_broker_count` Updates the number of broker nodes in the cluster
- `update_broker_storage` Updates the EBS storage associated with MSK brokers
- `update_cluster_configuration` Updates the cluster with the configuration that is specified in the request body
- `update_monitoring` Updates the monitoring settings for the cluster

Examples

```r
## Not run: svc <- kafka()
svc$create_cluster(
  Foo = 123
)
## End(Not run)
```
kinesis  
Amazon Kinesis

Description

Amazon Kinesis Data Streams Service API Reference

Amazon Kinesis Data Streams is a managed service that scales elastically for real-time processing of streaming big data.

Usage

kinesis(config = list())

Arguments

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

Service syntax

svc <- kinesis(
    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_to_stream  Adds or updates tags for the specified Kinesis data stream
create_stream  Creates a Kinesis data stream
decrease_stream_retention_period  Decreases the Kinesis data stream’s retention period, which is the length of time data records are available
delete_stream  Deletes a Kinesis data stream and all its shards and data
deregister_stream_consumer  To deregister a consumer, provide its ARN
describe_limits  Describes the shard limits and usage for the account
describe_stream  Describes the specified Kinesis data stream
describe_stream_consumer  To get the description of a registered consumer, provide the ARN of the consumer
describe_stream_summary  Provides a summarized description of the specified Kinesis data stream without the shard
disable_enhanced_monitoring | Disables enhanced monitoring
enable_enhanced_monitoring | Enables enhanced Kinesis data stream monitoring for shard-level metrics
get_records | Gets data records from a Kinesis data stream’s shard
get_shard_iterator | Gets an Amazon Kinesis shard iterator
increase_stream_retention_period | Increases the Kinesis data stream’s retention period, which is the length of time data records are accessible after they are added to the stream
list_shards | Lists the shards in a stream and provides information about each shard
list_stream_consumers | Lists the consumers registered to receive data from a stream using enhanced fan-out, and
list_streams | Lists your Kinesis data streams
list_tags_for_stream | Lists the tags for the specified Kinesis data stream
merge_shards | Merges two adjacent shards in a Kinesis data stream and combines them into a single shard
put_record | Writes a single data record into an Amazon Kinesis data stream
put_records | Writes multiple data records into a Kinesis data stream in a single call (also referred to as an PutRecords request)
register_stream_consumer | Registers a consumer with a Kinesis data stream
remove_tags_from_stream | Removes tags from the specified Kinesis data stream
split_shard | Splits a shard into two new shards in the Kinesis data stream, to increase the stream’s capacity to ingest and transport data
start_stream_encryption | Enables or updates server-side encryption using an AWS KMS key for a specified stream
stop_stream_encryption | Disables server-side encryption for a specified stream
update_shard_count | Updates the shard count of the specified stream to the specified number of shards

Examples

```r
## Not run: svc <- kinesis()
svc$add_tags_to_stream(
  Foo = 123
)
## End(Not run)
```

kinesisanalytics  Amazon Kinesis Analytics

Description

Overview

This documentation is for version 1 of the Amazon Kinesis Data Analytics API, which only supports SQL applications. Version 2 of the API supports SQL and Java applications. For more information about version 2, see Amazon Kinesis Data Analytics API V2 Documentation.

This is the Amazon Kinesis Analytics v1 API Reference. The Amazon Kinesis Analytics Developer Guide provides additional information.

Usage

```r
kinesisanalytics(config = list())
```
Arguments

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

Service syntax

```r
svc <- kinesisanalytics(
  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_application_cloud_watch_logging_option`
- `add_application_input`
- `add_application_input_processing_configuration`
- `add_application_output`
- `add_application_reference_data_source`
- `create_application`
- `delete_application`
- `delete_application_cloud_watch_logging_option`
- `delete_application_input_processing_configuration`
- `delete_application_output`
- `delete_application_reference_data_source`
- `describe_application`
- `discover_input_schema`
- `list_applications`
- `list_tags_for_resource`
- `start_application`
- `stop_application`
- `tag_resource`
- `untag_resource`
- `update_application`

**Examples**

```r
## Not run: svc <- kinesisanalytics()
```
Description
Amazon Kinesis Data Analytics is a fully managed service that you can use to process and analyze streaming data using SQL or Java. The service enables you to quickly author and run SQL or Java code against streaming sources to perform time series analytics, feed real-time dashboards, and create real-time metrics.

Usage
```
kinesisanalyticsv2(config = list())
```

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

Service syntax
```
svc <- kinesisanalyticsv2(
  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_application_cloud_watch_logging_option Adds an Amazon CloudWatch log stream to monitor application configuration errors
add_application_input Adds a streaming source to your SQL-based Amazon Kinesis Data Analytics application
add_application_input_processing_configuration Adds an InputProcessingConfiguration to an SQL-based Kinesis Data Analytics application
add_application_output Adds an external destination to your SQL-based Amazon Kinesis Data Analytics application
```
add_application_reference_data_source
add_application_vpc_configuration
create_application
create_application_snapshot
delete_application
delete_application_cloud_watch_logging_option
delete_application_input_processing_configuration
delete_application_output
delete_application_reference_data_source
delete_application_snapshot
delete_application_vpc_configuration
describe_application
describe_application_snapshot
discover_input_schema
list_applications
list_application_snapshots
list_tags_for_resource
start_application
stop_application
tag_resource
untag_resource
update_application

Add a reference data source to an existing SQL-based Amazon Kinesis Data Analytics application.
Adds a Virtual Private Cloud (VPC) configuration to the application.
Creates an Amazon Kinesis Data Analytics application.
Creates a snapshot of the application’s state data.
Deletes the specified application.
Deletes an Amazon CloudWatch log stream from an Amazon Kinesis Data Analytics application.
Deletes an InputProcessingConfiguration from an input.
Deletes the output destination configuration from your SQL-based Amazon Kinesis Data Analytics application.
Deletes a reference data source configuration from the specified SQL-based Amazon Kinesis Data Analytics application.
Deletes a snapshot of application state.
Removes a VPC configuration from a Kinesis Data Analytics application.
Returns information about a specific Amazon Kinesis Data Analytics application.
Returns information about a snapshot of application state data.
Infers a schema for an SQL-based Amazon Kinesis Data Analytics application.
Returns a list of Amazon Kinesis Data Analytics applications in your account.
Lists information about the current application snapshots.
Retrieves the list of key-value tags assigned to the application.
Starts the specified Amazon Kinesis Data Analytics application.
Stops the application from processing data.
Adds one or more key-value tags to a Kinesis Analytics application.
Removes one or more tags from a Kinesis Analytics application.
Updates an existing Amazon Kinesis Data Analytics application.

Examples

```r
## Not run: svc <- kinesisanalyticsv2()
svc$add_application_cloud_watch_logging_option(
  Foo = 123
)
## End(Not run)
```

AWS Key Management Service

AWS Key Management Service (AWS KMS) is an encryption and key management web service. This guide describes the AWS KMS operations that you can call programmatically. For general information about AWS KMS, see the AWS Key Management Service Developer Guide.

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, macOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWS KMS and other AWS services. For example, the SDKs take care of tasks such as signing requests (see below), managing errors, and retrying requests.
automatically. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

We recommend that you use the AWS SDKs to make programmatic API calls to AWS KMS. Clients must support TLS (Transport Layer Security) 1.0. We recommend TLS 1.2. Clients must also support cipher suites with Perfect Forward Secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

**Signing Requests**

Requests must be signed by using an access key ID and a secret access key. We strongly recommend that you do not use your AWS account (root) access key ID and secret key for everyday work with AWS KMS. Instead, use the access key ID and secret access key for an IAM user. You can also use the AWS Security Token Service to generate temporary security credentials that you can use to sign requests.

All AWS KMS operations require **Signature Version 4**.

**Logging API Requests**

AWS KMS supports AWS CloudTrail, a service that logs AWS API calls and related events for your AWS account and delivers them to an Amazon S3 bucket that you specify. By using the information collected by CloudTrail, you can determine what requests were made to AWS KMS, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

**Additional Resources**

For more information about credentials and request signing, see the following:

- **AWS Security Credentials** - This topic provides general information about the types of credentials used for accessing AWS.
- **Temporary Security Credentials** - This section of the IAM User Guide describes how to create and use temporary security credentials.
- **Signature Version 4 Signing Process** - This set of topics walks you through the process of signing a request using an access key ID and a secret access key.

**Commonly Used API Operations**

Of the API operations discussed in this guide, the following will prove the most useful for most applications. You will likely perform operations other than these, such as creating keys and assigning policies, by using the console.

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext

**Usage**

```python
kms(config = list())
```
Arguments

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

Service syntax

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

Operations

cancel_key_deletion  Cancels the deletion of a customer master key (CMK)
connect_custom_key_store  Connects or reconnects a custom key store to its associated AWS CloudHSM cluster
create_alias  Creates a display name for a customer managed customer master key (CMK)
create_custom_key_store  Creates a custom key store that is associated with an AWS CloudHSM cluster that you own and manage
create_grant  Adds a grant to a customer master key (CMK)
create_key  Creates a unique customer managed customer master key (CMK) in your AWS account and Region
decrypt  Decrypts ciphertext that was encrypted by a AWS KMS customer master key (CMK)
delete_alias  Deletes the specified alias
delete_custom_key_store  Deletes a custom key store
delete_imported_key_material  Deletes key material that you previously imported
describe_custom_key_stores  Provides information about custom key stores in the account and region
describe_key  Gets detailed information about a customer master key (CMK)
disable_key  Sets the state of a customer master key (CMK) to disabled, thereby preventing its use for cryptographic operations
disable_key_rotation  Disables automatic rotation of the key material for the specified symmetric customer master key (CMK)
disconnect_custom_key_store  Disconnects the custom key store from its associated AWS CloudHSM cluster
disable_key_rotation  Sets the key state of a customer master key (CMK) to enabled
enable_key  Enables automatic rotation of the key material for the specified symmetric customer master key (CMK)
encrypt  Encrypts plaintext into ciphertext by using a customer master key (CMK)
generate_data_key  Generates a unique symmetric data key
generate_data_key_pair  Generates a unique asymmetric data key pair
generate_data_key_pair_without_plaintext  Generates a unique asymmetric data key pair
generate_data_key_without_plaintext  Generates a unique symmetric data key
generate_random  Returns a random byte string that is cryptographically secure
generate_random  Gets a key policy attached to the specified customer master key (CMK)
get_key_policy  Gets a Boolean value that indicates whether automatic rotation of the key material is enabled for the specified customer master key (CMK)
generate_random  Returns the items you need to import key material into a symmetric, customer managed customer master key (CMK)
generate_random  Get the parameters for import
**lambda**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_public_key</td>
<td>Returns the public key of an asymmetric CMK</td>
</tr>
<tr>
<td>import_key_material</td>
<td>Imports key material into an existing symmetric AWS KMS customer master key</td>
</tr>
<tr>
<td>list_aliases</td>
<td>Gets a list of aliases in the caller’s AWS account and region</td>
</tr>
<tr>
<td>list_grants</td>
<td>Gets a list of all grants for the specified customer master key (CMK)</td>
</tr>
<tr>
<td>list_key_policies</td>
<td>Gets the names of the key policies that are attached to a customer master key (CMK)</td>
</tr>
<tr>
<td>list_keys</td>
<td>Gets a list of all customer master keys (CMKs) in the caller’s AWS account and region</td>
</tr>
<tr>
<td>list_resource_tags</td>
<td>Returns a list of all tags for the specified customer master key (CMK)</td>
</tr>
<tr>
<td>list_retirable_grants</td>
<td>Returns a list of all grants for which the grant’s RetiringPrincipal matches the one specified</td>
</tr>
<tr>
<td>put_key_policy</td>
<td>Attaches a key policy to the specified customer master key (CMK)</td>
</tr>
<tr>
<td>re_encrypt</td>
<td>Decrypts ciphertext and then reencrypts it entirely within AWS KMS</td>
</tr>
<tr>
<td>retire_grant</td>
<td>Retires a grant</td>
</tr>
<tr>
<td>revoke_grant</td>
<td>Revokes the specified grant for the specified customer master key (CMK)</td>
</tr>
<tr>
<td>schedule_key_deletion</td>
<td>Schedules the deletion of a customer master key (CMK)</td>
</tr>
<tr>
<td>sign</td>
<td>Creates a digital signature for a message or message digest by using the private key</td>
</tr>
<tr>
<td>tag_resource</td>
<td>Adds or edits tags for a customer master key (CMK)</td>
</tr>
<tr>
<td>untag_resource</td>
<td>Removes the specified tags from the specified customer master key (CMK)</td>
</tr>
<tr>
<td>update_alias</td>
<td>Associates an existing AWS KMS alias with a different customer master key (CMK)</td>
</tr>
<tr>
<td>update_custom_key_store</td>
<td>Changes the properties of a custom key store</td>
</tr>
<tr>
<td>update_key_description</td>
<td>Updates the description of a customer master key (CMK)</td>
</tr>
<tr>
<td>verify</td>
<td>Verifies a digital signature that was generated by the Sign operation</td>
</tr>
</tbody>
</table>

**Examples**

```r
# The following example cancels deletion of the specified CMK.
## Not run: svc <- kms()
svc$cancel_key_deletion(
  KeyId = "1234abcd-12ab-34cd-56ef-1234567890ab"
)
## End(Not run)
```

---

**Description**

**Overview**

This is the *AWS Lambda API Reference*. The AWS Lambda Developer Guide provides additional information. For the service overview, see *What is AWS Lambda*, and for information about how the service works, see *AWS Lambda: How it Works* in the *AWS Lambda Developer Guide*.

**Usage**

```r
lambda(config = list())
```
Arguments

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

Service syntax

svc <- lambda(
  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_layer_version_permission Adds permissions to the resource-based policy of a version of an AWS Lambda layer
add_permission Grants an AWS service or another account permission to use a function
create_alias Creates an alias for a Lambda function version
create_event_source_mapping Creates a mapping between an event source and an AWS Lambda function
create_function Creates a Lambda function
delete_alias Deletes a Lambda function alias
delete_event_source_mapping Deletes an event source mapping
delete_function Deletes a Lambda function
delete_function_concurrency Removes a concurrent execution limit from a function
delete_function_event_invoke_config Deletes the configuration for asynchronous invocation for a function, version, or alias
delete_layer_version Deletes a version of an AWS Lambda layer
delete_provisioned_concurrency_config Deletes the provisioned concurrency configuration for a function
get_account_settings Retrieves details about your account’s limits and usage in an AWS Region
get_alias Returns details about a Lambda function alias
get_event_source_mapping Returns details about an event source mapping
get_function Returns information about the function or function version, with a link to download the deployment package
get_function_concurrency Returns details about the concurrency configuration for a function
get_function_configuration Returns the version-specific settings of a Lambda function or version
get_function_event_invoke_config Retrieves the configuration for asynchronous invocation for a function, version, or alias
get_layer_version Returns information about a version of an AWS Lambda layer, with a link to download the deployment package
get_layer_version_by_arn Returns the permission policy for a version of an AWS Lambda layer
get_layer_version_policy Returns the resource-based IAM policy for a function, version, or alias
get_provisioned_concurrency_config Returns the provisioned concurrency configuration for a function’s alias or version
invoke Invokes a Lambda function
invoke_async For asynchronous function invocation, use Invoke
list_aliases
list_event_source_mappings
list_function_event_invoke_configs
list_functions
list_layers
list_layer_versions
list_provisioned_concurrency_configs
list_tags
list_versions_by_function
publish_layer_version
publish_version
put_function_concurrency
put_function_event_invoke_config
put_provisioned_concurrency_config
remove_layer_version_permission
remove_permission
tag_resource
untag_resource
update_alias
update_event_source_mapping
update_function_code
update_function_configuration
update_function_event_invoke_config

Returns a list of aliases for a Lambda function
Lists event source mappings
Retrieves a list of configurations for asynchronous invocation for a function
Returns a list of Lambda functions, with the version-specific configuration of each
Lists AWS Lambda layers and shows information about the latest version of each
Retrieves a list of provisioned concurrency configurations for a function
Returns a function’s tags
Returns a list of versions, with the version-specific configuration of each
Creates an AWS Lambda layer from a ZIP archive
Creates a version from the current code and configuration of a function
Configures options for asynchronous invocation on a function, version, or alias
Adds a provisioned concurrency configuration to a function’s alias or version
Removes a statement from the permissions policy for a version of an AWS Lambda function
Revokes function-use permission from an AWS service or another account
Adds tags to a function
Removes tags from a function
Updates the configuration of a Lambda function alias
Updates an event source mapping
Updates a Lambda function’s code
Modify the version-specific settings of a Lambda function
Updates the configuration for asynchronous invocation for a function, version, or alias

Examples

```r
# This example adds a permission for an S3 bucket to invoke a Lambda function.
# Not run: svc <- lambda()
svc$add_permission(
  Action = "lambda:InvokeFunction",
  FunctionName = "MyFunction",
  Principal = "s3.amazonaws.com",
  SourceAccount = "123456789012",
  SourceArn = "arn:aws:s3:::examplebucket/*",
  StatementId = "ID-1"
)
# End(Not run)
```

---

Amazon Lex Model Building Service

---

lexmodelbuildingservice
Description

Amazon Lex Build-Time Actions

Amazon Lex is an AWS service for building conversational voice and text interfaces. Use these actions to create, update, and delete conversational bots for new and existing client applications.

Usage

lexmodelbuildingservice(config = list())

Arguments

cfg (Optional configuration of credentials, endpoint, and/or region.)

Service syntax

svc <- lexmodelbuildingservice(
    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_bot_version** - Creates a new version of the bot based on the $LATEST version
- **create_intent_version** - Creates a new version of an intent based on the $LATEST version of the intent
- **create_slot_type_version** - Creates a new version of a slot type based on the $LATEST version of the specified slot type
- **delete_bot** - Deletes all versions of the bot, including the $LATEST version
- **delete_bot_alias** - Deletes an alias for the specified bot
- **delete_bot_channel_association** - Deletes the association between an Amazon Lex bot and a messaging platform
- **delete_bot_version** - Deletes a specific version of a bot
- **delete_intent** - Deletes all versions of the intent, including the $LATEST version
- **delete_intent_version** - Deletes a specific version of an intent
- **delete_slot_type** - Deletes all versions of the slot type, including the $LATEST version
- **delete_slot_type_version** - Deletes a specific version of a slot type
- **delete_utterances** - Deletes stored utterances
- **get_bot** - Returns metadata information for a specific bot
- **get_bot_alias** - Returns information about an Amazon Lex bot alias
- **get_botaliases** - Returns a list of aliases for a specified Amazon Lex bot
- **get_bot_channel_association** - Returns information about the association between an Amazon Lex bot and a messaging platform
LexRuntimeService

get_bot_channel_associations
Returns a list of all of the channels associated with the specified bot

get_bots
Returns bot information as follows: - If you provide the nameContains field, the response includes information for the $LATEST version of all bots whose name contains the specified string

get_bot_versions
Returns information about all of the versions of a bot

get_builtin_intent
Gets information about a built-in intent

get_builtin_intents
Gets a list of built-in intents that meet the specified criteria

get_builtin_slot_types
Gets a list of built-in slot types that meet the specified criteria

get_export
Exports the contents of an Amazon Lex resource in a specified format

get_import
Gets information about an import job started with the StartImport operation

get_intent
Returns information about an intent

get_intents
Returns intent information as follows: - If you specify the nameContains field, returns the $LATEST version of all intents that contain the specified string

get_intent_versions
Gets information about all of the versions of an intent

get_slot_type
Returns information about a specific version of a slot type

get_slot_types
Returns slot type information as follows: - If you specify the nameContains field, returns the $LATEST version of all slot types that contain the specified string

get_slot_type_versions
Gets information about all versions of a slot type

get_utterances_view
Use the GetUtterancesView operation to get information about the utterances that your users have made to your bot

put_bot
Creates an Amazon Lex conversational bot or replaces an existing bot

put_bot_alias
Creates an alias for the specified version of the bot or replaces an existing bot

put_intent
Creates an intent or replaces an existing intent

put_slot_type
Creates a custom slot type or replaces an existing custom slot type

start_import
Starts a job to import a resource to Amazon Lex

Examples

# This example shows how to get configuration information for a bot.
## Not run: svc <- lexmodelbuildingservice()
  svc$get_bot(
    name = "DocOrderPizza",
    versionOrAlias = "$LATEST"
  )
## End(Not run)

Description

Amazon Lex provides both build and runtime endpoints. Each endpoint provides a set of operations (API). Your conversational bot uses the runtime API to understand user utterances (user input text or voice). For example, suppose a user says "I want pizza", your bot sends this input to Amazon Lex using the runtime API. Amazon Lex recognizes that the user request is for the OrderPizza intent (one of the intents defined in the bot). Then Amazon Lex engages in user conversation on behalf of the bot to elicit required information (slot values, such as pizza size and crust type), and then performs fulfillment activity (that you configured when you created the bot). You use the build-time API to create and manage your Amazon Lex bot. For a list of build-time operations, see the build-time API.
Usage

`lexruntimeservice(config = list())`

Arguments

cfgi

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

Service syntax

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

Operations

- **delete_session**  Removes session information for a specified bot, alias, and user ID
- **get_session**     Returns session information for a specified bot, alias, and user ID
- **post_content**    Sends user input (text or speech) to Amazon Lex
- **post_text**       Sends user input to Amazon Lex
- **put_session**     Creates a new session or modifies an existing session with an Amazon Lex bot

Examples

```r
## Not run: svc <- lexruntimeservice()
svc$delete_session(
  Foo = 123
)
## End(Not run)

AWS License Manager
AWS License Manager makes it easier to manage licenses from software vendors across multiple AWS accounts and on-premises servers.

Usage

```
licensemanager(config = list())
```

Arguments

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

Service syntax

```
svc <- licensemanager(
  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_license_configuration`: Creates a license configuration
- `delete_license_configuration`: Deletes the specified license configuration
- `get_license_configuration`: Gets detailed information about the specified license configuration
- `get_service_settings`: Gets the License Manager settings for the current Region
- `list_associations_for_license_configuration`: Lists the resource associations for the specified license configuration
- `list_failures_for_license_configuration_operations`: Lists the license configuration operations that failed
- `list_license_configurations`: Lists the license configurations for your account
- `list_license_specifications_for_resource`: Describes the license configurations for the specified resource
- `list_resource_inventory`: Lists resources managed using Systems Manager inventory
- `list_tags_for_resource`: Lists the tags for the specified license configuration
- `list_usage_for_license_configuration`: Lists all license usage records for a license configuration, displaying license consumption details by resource at a selected point in time
- `tag_resource`: Adds the specified tags to the specified license configuration
- `untag_resource`: Removes the specified tags from the specified license configuration
- `update_license_configuration`: Modifies the attributes of an existing license configuration
- `update_license_specifications_for_resource`: Adds or removes the specified license configurations for the specified AWS resource
- `update_service_settings`: Updates License Manager settings for the current Region
Examples

```r
## Not run: svc <- licensemanager()
svc$create_license_configuration(
    Foo = 123
)
## End(Not run)
```

## lightsail

### Amazon Lightsail

#### Description

Amazon Lightsail is the easiest way to get started with AWS for developers who just need virtual private servers. Lightsail includes everything you need to launch your project quickly - a virtual machine, a managed database, SSD-based storage, data transfer, DNS management, and a static IP - for a low, predictable price. You manage those Lightsail servers through the Lightsail console or by using the API or command-line interface (CLI).

For more information about Lightsail concepts and tasks, see the Lightsail Dev Guide.

To use the Lightsail API or the CLI, you will need to use AWS Identity and Access Management (IAM) to generate access keys. For details about how to set this up, see the Lightsail Dev Guide.

#### Usage

```r
lightsail(config = list())
```

#### Arguments

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

#### Service syntax

```r
csv <- lightsail(
    config = list(
        credentials = list(
            creds = list(
                access_key_id = "string",
                secret_access_key = "string",
                session_token = "string"
            ),
            profile = "string"
        ),
        endpoint = "string",
        region = "string"
    )
)
```
**Operations**

- allocate_static_ip
- attach_disk
- attach_instances_to_load_balancer
- attach_load_balancer_tls_certificate
- attach_static_ip
- close_instance_public_ports
- copy_snapshot
- create_cloud_formation_stack
- create_disk
- create_disk_from_snapshot
- create_disk_snapshot
- create_domain
- create_domain_entry
- create_instances
- create_instances_from_snapshot
- create_instance_snapshot
- create_key_pair
- create_load_balancer
- create_load_balancer_tls_certificate
- create_relational_database
- create_relational_database_from_snapshot
- create_relational_database_snapshot
- delete_auto_snapshot
- delete_disk
- delete_disk_snapshot
- delete_domain
- delete_domain_entry
- delete_instance
- delete_instance_snapshot
- delete_key_pair
- delete_known_host_keys
- delete_load_balancer
- delete_load_balancer_tls_certificate
- delete_relational_database
- delete_relational_database_snapshot
- detach_disk
- detach_instances_from_load_balancer
- detach_static_ip
- disable_add_on
- download_default_key_pair
- enable_add_on
- export_snapshot
- get_active_names
- get_auto_snapshots
- get_blueprints
- get_bundles

Allocates a static IP address
Attaches a block storage disk to a running or stopped Lightsail instance and exposes it to the instance with the specified disk name
Attaches one or more Lightsail instances to a load balancer
Attaches a Transport Layer Security (TLS) certificate to your load balancer
Attaches a static IP address to a specific Amazon Lightsail instance
Closes the public ports on a specific Amazon Lightsail instance
Copies a manual snapshot of an instance or disk as another manual snapshot
Creates a domain resource associated with the domain: Add create_one_of_the_following_entry_records associated with the domain
Creates one of the following entry records associated with the domain: Address (A), canonical name (CNAME), mail exchanger (MX), name server (NS), start of authority (SOA), service locator (SRV), or text (TXT)
Creates a new database in Amazon Lightsail
Creates a new database from an existing database snapshot in Amazon Lightsail
Deletes a specific snapshot of a virtual private server, or instance
Creates an Amazon Lightsail instance
Creates a Lightsail load balancer
Creates a Lightsail load balancer TLS certificate
Deletes an automatic snapshot of an instance or disk
Deletes the specified block storage disk
Deletes the specified domain recordset and all of its domain records
Deletes a specific domain entry
Deletes an Amazon Lightsail instance
Deletes a specific snapshot of a virtual private server (or instance)
Deletes a specific SSH key pair
Deletes the known host key or certificate used by the Amazon Lightsail browser client
Deletes a Lightsail load balancer and all its associated SSL/TLS certificates
Deletes an SSL/TLS certificate associated with a Lightsail load balancer
Deletes a database in Amazon Lightsail
Deletes a database snapshot in Amazon Lightsail
Detaches a stopped block storage disk from a Lightsail instance
Detaches the specified instances from a Lightsail load balancer
Detaches a static IP from the Amazon Lightsail instance to which it is attached
Disables an add-on for an Amazon Lightsail resource
Downloads the default SSH key pair from the user’s account
Enables or modifies an add-on for an Amazon Lightsail resource
Exports an Amazon Lightsail instance or block storage disk snapshot to Amazon S3
Returns the names of all active (not deleted) resources
Returns the available automatic snapshots for an instance or disk
Returns the list of available instance images, or blueprints
Returns the list of bundles that are available for purchase
get_cloud_formation_stack_records  Returns the CloudFormation stack record created as a result of the create cloud formation stack operation
get_disk  Returns information about a specific block storage disk
get_disks  Returns information about all block storage disks in your AWS account and region
get_disk_snapshot  Returns information about a specific block storage disk snapshot
get_disk_snapshots  Returns information about all block storage disk snapshots in your AWS account and region
get_domain  Returns a list of all domains in the user’s account
get_domains  Returns the export snapshot record created as a result of the export snapshot operation
get_disks  Returns information about all block storage disks in your AWS account and region
get_disk_snapshot  Returns information about a specific block storage disk snapshot
get_disk_snapshots  Returns information about all block storage disk snapshots in your AWS account and region
get_domain  Returns a list of all domains in the user’s account
get_domains  Returns the export snapshot record created as a result of the export snapshot operation
get_instance  Returns information about a specific Amazon Lightsail instance, which is a virtual private server
get_instance_access_details  Returns temporary SSH keys you can use to connect to a specific virtual private server, or instance
get_instance_metric_data  Returns the port states for a specific virtual private server, or instance
get_instances  Returns information about all Amazon Lightsail virtual private servers, or instances
get_instance_snapshot  Returns information about a specific instance snapshot
get_instance_snapshots  Returns all instance snapshots for the user’s account
get_instance_state  Returns the state of a specific instance
get_key_pair  Returns information about a specific key pair
get_key_pairs  Returns information about all key pairs in the user’s account
get_load_balancer  Returns information about the specified Lightsail load balancer
get_load_balancer_metric_data  Returns information about health metrics for your Lightsail load balancer
get_load_balancers  Returns information about all load balancers in an account
get_load_balancer_tls_certificates  Returns information about the TLS certificates that are associated with the specified Lightsail load balancer
get_load_balancer_access_details  Returns information about the TLS certificates that are associated with the specified Lightsail load balancer
get_operations  Returns information about all operations
get_operations_for_resource  Gets operations for a specific resource (e.g., get_operations_for_resource("instance"), get_operations_for_resource("load_balancer"), etc.)
get_regions  Returns a list of all valid regions for Amazon Lightsail
get_relational_database  Returns information about a specific database in Amazon Lightsail
get_relational_database_blueprints  Returns a list of available database blueprints in Amazon Lightsail
get_relational_database_bundles  Returns the list of bundles that are available in Amazon Lightsail
get_relational_database_events  Returns a list of events for a specific database in Amazon Lightsail
get_relational_database_log_events  Returns a list of log events for a specific database in Amazon Lightsail
get_relational_database_log_streams  Returns a list of available log streams for a specific database in Amazon Lightsail
get_relational_database_master_user_password  Returns the current, previous, or pending versions of the master user password for a Lightsail database
get_relational_database_metric_data  Returns all of the runtime parameters offered by the underlying database software, or engine, for a specific database in Amazon Lightsail
get_relations_database_parameters  Returns information about all of your databases in Amazon Lightsail
get_relations_database_snapshot  Returns information about a specific database snapshot in Amazon Lightsail
get_relations_database_snapshots  Returns information about all of your database snapshots in Amazon Lightsail
get_static_ip  Returns information about a specific static IP
get_static_ips  Returns information about all static IPs in the user’s account
import_key_pair  Imports a public SSH key from a specific key pair
is_vpc_peered  Imports a public SSH key from a specific key pair
open_instance_public_ports  Adds public ports to an Amazon Lightsail instance
peer_vpc  Tries to peer the Lightsail VPC with the user’s default VPC
put_instance_public_ports  Sets the specified open ports for an Amazon Lightsail instance, and closes all ports for every protocol not included in the current request
reboot_instance  Restarts a specific instance
reboot_relational_database  Restarts a specific database in Amazon Lightsail
release_static_ip  Deletes a specific static IP from your account
Definition of the public APIs exposed by Amazon Machine Learning

Usage

machinelearning(config = list())

Arguments

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

Service syntax

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

Examples

## Not run: svc <- lightsail()
svc$allocate_static_ip(
  Foo = 123
)
## End(Not run)
profile = "string"
),
endpoint = "string",
region = "string"
)

Operations

- `add_tags` Adds one or more tags to an object, up to a limit of 10
- `create_batch_prediction` Generates predictions for a group of observations
- `create_data_source_from_rds` Creates a DataSource object from an Amazon Relational Database Service (Amazon RDS)
- `create_data_source_from_redshift` Creates a DataSource from a database hosted on an Amazon Redshift cluster
- `create_data_source_from_s3` Creates a DataSource object
- `create_evaluation` Creates a new Evaluation of an MLModel
- `create_ml_model` Creates a new MLModel using the DataSource and the recipe as information sources
- `create_realtime_endpoint` Creates a real-time endpoint for the MLModel
- `delete_batch_prediction` Assigns the DELETED status to a BatchPrediction, rendering it unusable
- `delete_data_source` Assigns the DELETED status to a DataSource, rendering it unusable
- `delete_evaluation` Assigns the DELETED status to an Evaluation, rendering it unusable
- `delete_ml_model` Assigns the DELETED status to an MLModel, rendering it unusable
- `delete_realtime_endpoint` Deletes a real time endpoint of an MLModel
- `delete_tags` Deletes the specified tags associated with an ML object
- `describe_batch_predictions` Returns a list of BatchPrediction operations that match the search criteria in the request
- `describe_data_sources` Returns a list of DataSource that match the search criteria in the request
- `describe_evaluations` Returns a list of DescribeEvaluations that match the search criteria in the request
- `describe_ml_models` Returns a list of MLModel that match the search criteria in the request
- `describe_tags` Describes one or more of the tags for your Amazon ML object
- `get_batch_prediction` Returns a BatchPrediction that includes detailed metadata, status, and data file information
- `get_data_source` Returns a DataSource that includes metadata and data file information, as well as the current status of the DataSource
- `get_evaluation` Returns an Evaluation that includes metadata as well as the current status of the Evaluation
- `get_ml_model` Returns an MLModel that includes detailed metadata, data source information, and the current status of the MLModel
- `predict` Generates a prediction for the observation using the specified ML Model
- `update_batch_prediction` Updates the BatchPredictionName of a BatchPrediction
- `update_data_source` Updates the DataSourceName of a DataSource
- `update_evaluation` Updates the EvaluationName of an Evaluation
- `update_ml_model` Updates the MLModelName and the ScoreThreshold of an MLModel

Examples

```r
## Not run: svc <- machinelearning()
svc$add_tags(
  Foo = 123
)
## End(Not run)
```
Amazon Macie

Description

Amazon Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property, and provides you with dashboards and alerts that give visibility into how this data is being accessed or moved. For more information, see the Macie User Guide.

Usage

macie(config = list())

Arguments

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

Service syntax

svc <- macie(  
  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_member_account**: Associates a specified AWS account with Amazon Macie as a member account.
- **associate_s3_resources**: Associates specified S3 resources with Amazon Macie for monitoring and data classification.
- **disassociate_member_account**: Removes the specified member account from Amazon Macie.
- **disassociate_s3_resources**: Removes specified S3 resources from being monitored by Amazon Macie.
- **list_member_accounts**: Lists all Amazon Macie member accounts for the current Amazon Macie master account.
- **list_s3_resources**: Lists all the S3 resources associated with Amazon Macie.
- **update_s3_resources**: Updates the classification types for the specified S3 resources.
## Not run:
```r
svc <- macie()
svc$associate_member_account(
  Foo = 123
)
## End(Not run)
```

### Description

Provides AWS Marketplace business intelligence data on-demand.

### Usage

```r
marketplacecommerceanalytics(config = list())
```

### Arguments

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

### Service syntax

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

### Operations

- **generate_data_set**
  - Given a data set type and data set publication date, asynchronously publishes the requested data set to the specified S3 bucket and notifies the specified SNS topic once the data is available.

- **start_support_data_export**
  - Given a data set type and a from date, asynchronously publishes the requested customer support data to the specified S3 bucket and notifies the specified SNS topic once the data is available.
marketplaceentitlementservice

## Examples

```r
## Not run: svc <- marketplacecommerceanalytics()
svc$generate_data_set(
  Foo = 123
)
## End(Not run)
```

### Description

This reference provides descriptions of the AWS Marketplace Entitlement Service API.

AWS Marketplace Entitlement Service is used to determine the entitlement of a customer to a given product. An entitlement represents capacity in a product owned by the customer. For example, a customer might own some number of users or seats in an SaaS application or some amount of data capacity in a multi-tenant database.

### Getting Entitlement Records

- `GetEntitlements`: Gets the entitlements for a Marketplace product.

### Usage

```r
marketplaceentitlementservice(config = list())
```

### Arguments

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

### Service syntax

```r
tsvc <- marketplaceentitlementservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"),
      endpoint = "string",
      region = "string"
    )
  )
)```
## Operations

**get_entitlements**  
GetEntitlements retrieves entitlement values for a given product

### Examples

```
## Not run: svc <- marketplaceentitlementservice()
svc$get_entitlements(
    Foo = 123
)
## End(Not run)
```

---

**marketplacemetering  AWSMarketplace Metering**

### Description

AWS Marketplace Metering Service

This reference provides descriptions of the low-level AWS Marketplace Metering Service API. AWS Marketplace sellers can use this API to submit usage data for custom usage dimensions.

**Submitting Metering Records**

- **MeterUsage** - Submits the metering record for a Marketplace product. MeterUsage is called from an EC2 instance or a container running on EKS or ECS.
- **BatchMeterUsage** - Submits the metering record for a set of customers. BatchMeterUsage is called from a software-as-a-service (SaaS) application.

**Accepting New Customers**

- **ResolveCustomer** - Called by a SaaS application during the registration process. When a buyer visits your website during the registration process, the buyer submits a Registration Token through the browser. The Registration Token is resolved through this API to obtain a CustomerIdentifier and Product Code.

**Entitlement and Metering for Paid Container Products**

- Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the RegisterUsage operation for software entitlement and metering. Free and BYOL products for Amazon ECS or Amazon EKS aren’t required to call RegisterUsage, but you can do so if you want to receive usage data in your seller reports. For more information on using the RegisterUsage operation, see Container-Based Products.

BatchMeterUsage API calls are captured by AWS CloudTrail. You can use Cloudtrail to verify that the SaaS metering records that you sent are accurate by searching for records with the eventName of BatchMeterUsage. You can also use CloudTrail to audit records over time. For more information, see the AWS CloudTrail User Guide.
Usage

marketplacemetering(config = list())

Arguments

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

Service syntax

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

Operations

- **batch_meter_usage**: BatchMeterUsage is called from a SaaS application listed on the AWS Marketplace to post metering records
- **meter_usage**: API to emit metering records
- **register_usage**: Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the RegisterUsage operation for software entitlement and metering
- **resolve_customer**: ResolveCustomer is called by a SaaS application during the registration process

Examples

```r
## Not run: svc <- marketplacemetering()
svc$batch_meter_usage(
  Foo = 123
)
## End(Not run)
```

mq  

AmazonMQ
Description
Amazon MQ is a managed message broker service for Apache ActiveMQ that makes it easy to set up and operate message brokers in the cloud. A message broker allows software applications and components to communicate using various programming languages, operating systems, and formal messaging protocols.

Usage
mq(config = list())

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

Service syntax
svc <- mq(
  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_broker Creates a broker
create_configuration Creates a new configuration for the specified configuration name
create_tags Add a tag to a resource
create_user Creates an ActiveMQ user
delete_broker Deletes a broker
delete_tags Removes a tag from a resource
delete_user Deletes an ActiveMQ user
describe_broker Returns information about the specified broker
describe_broker_engine_types Describe available engine types and versions
describe_broker_instance_options Describe available broker instance options
describe_configuration Returns information about the specified configuration
describe_configuration_revision Returns the specified configuration revision for the specified configuration
describe_user Returns information about an ActiveMQ user
list_brokers Returns a list of all brokers
list_configuration_revisions Returns a list of all revisions for the specified configuration
mturk

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_configurations</td>
<td>Returns a list of all configurations</td>
</tr>
<tr>
<td>list_tags</td>
<td>Lists tags for a resource</td>
</tr>
<tr>
<td>list_users</td>
<td>Returns a list of all ActiveMQ users</td>
</tr>
<tr>
<td>reboot_broker</td>
<td>Reboots a broker</td>
</tr>
<tr>
<td>update_broker</td>
<td>Adds a pending configuration change to a broker</td>
</tr>
<tr>
<td>update_configuration</td>
<td>Updates the specified configuration</td>
</tr>
<tr>
<td>update_user</td>
<td>Updates the information for an ActiveMQ user</td>
</tr>
</tbody>
</table>

Examples

```r
## Not run: svc <- mq()
svc=create_broker(
  Foo = 123
)
## End(Not run)
```

Description

Amazon Mechanical Turk API Reference

Usage

```r
mturk(config = list())
```

Arguments

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

Service syntax

```r
svc <- mturk(
  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_qualification_request`
- `approve_assignment`
- `associate_qualification_with_worker`
- `create_additional_assignments_for_hit`
- `create_hit`
- `create_hit_type`
- `create_hit_with_hit_type`
- `create_qualification_type`
- `create_worker_block`
- `delete_hit`
- `delete_qualification_type`
- `delete_worker_block`
- `disassociate_qualification_from_worker`
- `get_account_balance`
- `get_assignment`
- `get_file_upload_url`
- `get_hit`
- `get_qualification_score`
- `get_qualification_type`
- `list_assignments_for_hit`
- `list_bonus_payments`
- `list_hi_ts`
- `list_hi_ts_for_qualification_type`
- `list_qualification_requests`
- `list_qualification_types`
- `list_reviewable_hi_ts`
- `list_review_policy_results_for_hit`
- `list_worker_blocks`
- `list_workers_with_qualification_type`
- `notify_workers`
- `reject_assignment`
- `reject_qualification_request`
- `send_bonus`
- `send_test_event_notification`
- `update_expiration_for_hit`
- `update_hit_review_status`
- `update_hit_type_of_hit`
- `update_notification_settings`
- `update_qualification_type`

Examples

```r
## Not run: svc <- mturk()
svc$accept_qualification_request()
  Foo = 123
```
Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications that work with highly connected datasets. The core of Amazon Neptune is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Amazon Neptune supports popular graph models Property Graph and W3C’s RDF, and their respective query languages Apache TinkerPop Gremlin and SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets. Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security.

This interface reference for Amazon Neptune contains documentation for a programming or command line interface you can use to manage Amazon Neptune. Note that Amazon Neptune is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

Usage

```r
neptune(config = list())
```

Arguments

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

Service syntax

```r
svc <- neptune(
  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_role_to_db_cluster` Associates an Identity and Access Management (IAM) role from an Neptune DB cluster
- `add_source_identifier_to_subscription` Adds a source identifier to an existing event notification subscription
- `apply_pending_maintenance_action` Applies a pending maintenance action to a resource (for example, to a DB instance)
- `copy_db_cluster_parameter_group` Copies the specified DB cluster parameter group
- `copy_db_cluster_snapshot` Copies a snapshot of a DB cluster
- `create_db_cluster` Creates a new Amazon Neptune DB cluster
- `create_db_cluster_parameter_group` Creates a new DB cluster parameter group
- `create_db_cluster_snapshot` Creates a snapshot of a DB cluster
- `create_db_instance` Creates a new DB instance
- `create_db_parameter_group` Creates a new DB parameter group
- `create_db_subnet_group` Creates a new DB subnet group
- `create_event_subscription` Creates an event notification subscription
- `delete_db_cluster` The DeleteDBCluster action deletes a previously provisioned DB cluster
- `delete_db_cluster_parameter_group` Deletes a specified DB cluster parameter group
- `delete_db_cluster_snapshot` Deletes a DB cluster snapshot
- `delete_db_instance` The DeleteDBInstance action deletes a previously provisioned DB instance
- `delete_db_parameter_group` Deletes a specified DBParameterGroup
- `delete_db_subnet_group` Deletes a DB subnet group
- `delete_event_subscription` Deletes an event notification subscription
- `describe_db_cluster_parameter_groups` Returns a list of DBClusterParameterGroup descriptions
- `describe_db_cluster_parameters` Returns the detailed parameter list for a particular DB cluster parameter group
- `describe_db_clusters` Returns information about provisioned DB clusters
- `describe_db_cluster_snapshot_attributes` Returns a list of DB cluster snapshot attribute names and values for a manual DB cluster snapshot
- `describe_db_engine_versions` Returns a list of the available DB engines
- `describe_db_instances` Returns information about provisioned instances
- `describe_db_parameter_groups` Returns a list of DBParameterGroup descriptions
- `describe_db_parameters` Returns the detailed parameter list for a particular DB parameter group
- `describe_db_subnet_groups` Returns a list of DBSubnetGroup descriptions
- `describe_engine_default_cluster_parameters` Returns the default engine and system parameter information for the cluster database engine
- `describe_engine_default_parameters` Displays a list of categories for all event source types, or, if specified, for a specific type
- `describe_event_categories` Displays a list of categories for all event source types, or, if specified, for a specific type
- `describe_events` Displays a list of categories for all event source types, or, if specified, for a specific type
- `describe_event_subscriptions` Displays a list of categories for all event source types, or, if specified, for a specific type
- `describe_orderable_db_instance_options` Returns a list of orderable DB instance options for the specified engine
- `describe_pending_maintenance_actions` Returns a list of resources (for example, DB instances) that have at least one pending maintenance action
- `describe_valid_db_instance_modifications` Forces a failover for a DB cluster
- `failover_db_cluster` Lists all tags on an Amazon Neptune resource
- `list_tags_for_resource` Modifies a setting for a DB cluster
- `modify_db_cluster` Modifies the parameters of a DB cluster parameter group
- `modify_db_cluster_parameter_group` Adds an attribute and values to, or removes an attribute and values from, a manual DB cluster snapshot attribute
- `modify_db_cluster_snapshot_attribute` Modifies settings for a DB instance
- `modify_db_instance` Modifies the parameters of a DB parameter group
modify_db_subnet_group
modify_event_subscription
promote_read_replica_db_cluster
reboot_db_instance
remove_role_from_db_cluster
remove_source_identifier_from_subscription
remove_tags_from_resource
reset_db_cluster_parameter_group
reset_db_parameter_group
restore_db_cluster_from_snapshot
restore_db_cluster_to_point_in_time

Modifies an existing DB subnet group
Modifies an existing event notification subscription
Not supported
You might need to reboot your DB instance, usually for maintenance reasons
Disassociates an Identity and Access Management (IAM) role from a DB cluster
Removes a source identifier from an existing event notification subscription
Removes metadata tags from an Amazon Neptune resource
Modifies the parameters of a DB cluster parameter group to the default value
Modifies the parameters of a DB parameter group to the engine/system default
Creates a new DB cluster from a DB snapshot or DB cluster snapshot
Restores a DB cluster to an arbitrary point in time

Examples

```r
## Not run: svc <- neptune()
svc$add_role_to_db_cluster(
  Foo = 123
)
## End(Not run)
```

---

**Description**

Welcome to the *AWS OpsWorks Stacks API Reference*. This guide provides descriptions, syntax, and usage examples for AWS OpsWorks Stacks actions and data types, including common parameters and error codes.

AWS OpsWorks Stacks is an application management service that provides an integrated experience for overseeing the complete application lifecycle. For information about this product, go to the [AWS OpsWorks](https://aws.amazon.com/opsworks) details page.

**SDKs and CLI**

The most common way to use the AWS OpsWorks Stacks API is by using the AWS Command Line Interface (CLI) or by using one of the AWS SDKs to implement applications in your preferred language. For more information, see:

- AWS CLI
- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for PHP 2
- AWS SDK for Ruby
- AWS SDK for Node.js
• AWS SDK for Python (Boto)

Endpoints
AWS OpsWorks Stacks supports the following endpoints, all HTTPS. You must connect to one of the following endpoints. Stacks can only be accessed or managed within the endpoint in which they are created.

• opsworks.us-east-1.amazonaws.com
• opsworks.us-east-2.amazonaws.com
• opsworks.us-west-1.amazonaws.com
• opsworks.us-west-2.amazonaws.com
• opsworks.ca-central-1.amazonaws.com (API only; not available in the AWS console)
• opsworks.eu-west-1.amazonaws.com
• opsworks.eu-west-2.amazonaws.com
• opsworks.eu-west-3.amazonaws.com
• opsworks.eu-central-1.amazonaws.com
• opsworks.ap-northeast-1.amazonaws.com
• opsworks.ap-northeast-2.amazonaws.com
• opsworks.ap-south-1.amazonaws.com
• opsworks.ap-southeast-1.amazonaws.com
• opsworks.ap-southeast-2.amazonaws.com
• opsworks.sa-east-1.amazonaws.com

Chef Versions
When you call CreateStack, CloneStack, or UpdateStack we recommend you use the ConfigurationManager parameter to specify the Chef version. The recommended and default value for Linux stacks is currently 12. Windows stacks use Chef 12.2. For more information, see Chef Versions.

You can specify Chef 12, 11.10, or 11.4 for your Linux stack. We recommend migrating your existing Linux stacks to Chef 12 as soon as possible.

Usage
opsworks(config = list())

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

Service syntax
svc <- opsworks(config = list(
    credentials = list(
        creds = list(
            access_key_id = "string",
            access_key_secret = "string",
            region = "string",
            endpoint = "string"
        )
    ),
    region = "string"
))
secret_access_key = "string",
session_token = "string"
),
profile = "string"
),
endpoint = "string",
region = "string"
)
)

Operations

assign_instance
assign_volume
associate_elastic_ip
attach_elastic_load_balancer
clon_stack
create_app
create_deployment
create_instance
create_layer
create_stack
create_user_profile
delete_app
delete_instance
delete_layer
delete_stack
delete_user_profile
deregister_ecs_cluster
deregister_elastic_ip
deregister_instance
deregister_rds_db_instance
deregister_volume
describe_agent_versions
describe_apps
describe_commands
describe_deployments
describe_ecs_clusters
describe_elastic_ips
describe_elastic_load_balancers
describe_instances
describe_layers
describe_load_based_auto_scaling
describe_my_user_profile
describe_operating_systems
describe_permissions
describe_raid_arrays
describe_rds_db_instances

Assign a registered instance to a layer
Assigns one of the stack’s registered Amazon EBS volumes to a specified instance
Associates one of the stack’s registered Elastic IP addresses with a specified instance
Attaches an Elastic Load Balancing load balancer to a specified layer
Creates a clone of a specified stack
Creates an app for a specified stack
Runs deployment or stack commands
Creates an instance in a specified stack
Creates a layer
Creates a new stack
Creates a new user profile
Deletes a specified app
Deletes a specified instance, which terminates the associated Amazon EC2 instance
Deletes a specified layer
Deletes a specified stack
Deletes a user profile
Deregisters a specified Amazon ECS cluster from a stack
Deregisters a specified Elastic IP address
Deregister a registered Amazon EC2 or on-premises instance
Deregisters an Amazon RDS instance
Deregisters an Amazon EBS volume
Describes the available AWS OpsWorks Stacks agent versions
Requests a description of a specified set of apps
Describes the results of specified commands
Requests a description of a specified set of deployments
Describes Amazon ECS clusters that are registered with a stack
Describes Elastic IP addresses
Describes a stack’s Elastic Load Balancing instances
Requests a description of a set of instances
Requests a description of one or more layers in a specified stack
Describes load-based auto scaling configurations for specified layers
Describes a user’s SSH information
Describes the operating systems that are supported by AWS OpsWorks Stacks
Describes the permissions for a specified stack
Describe an instance’s RAID arrays
Describes Amazon RDS instances
describe_service_errors
describe_stack_provisioning_parameters
describe_stacks
describe_stack_summary
describe_time_based_auto_scaling
describe_user_profiles
describe_volumes
detach_elastic_load_balancer
disassociate_elastic_ip
get_hostname_suggestion
grant_access
list_tags
reboot_instance
register_ecs_cluster
register_elastic_ip
register_instance
register_rds_db_instance
register_volume
set_load_based_auto_scaling
set_permission
set_time_based_auto_scaling
start_instance
start_stack
stop_instance
stop_stack
tag_resource
unassign_instance
unassign_volume
untag_resource
update_app
update_elastic_ip
update_instance
update_layer
update_my_user_profile
update_rds_db_instance
update_stack
update_user_profile
update_volume

Describes AWS OpsWorks Stacks service errors
Requests a description of a stack’s provisioning parameters
Requests a description of one or more stacks
Describes the number of layers and apps in a specified stack, and the number of instances in each state
Describes time-based auto scaling configurations for specified instances
Describe specified users
Describes an instance’s Amazon EBS volumes
Detaches a specified Elastic Load Balancing instance from its layer
Disassociates an Elastic IP address from its instance
Gets a generated host name for the specified layer, based on the current host name theme
This action can be used only with Windows stacks
Returns a list of tags that are applied to the specified stack or layer
Reboots a specified instance
Registers a specified Amazon ECS cluster with a stack
Registers an Elastic IP address with a specified stack
Registers instances that were created outside of AWS OpsWorks Stacks with a specified stack
Registers an Amazon RDS instance with a stack
Registers an Amazon EBS volume with a specified stack
Specify the load-based auto scaling configuration for a specified layer
Specifies a user’s permissions
Specify the time-based auto scaling configuration for a specified instance
Starts a specified instance
Starts a stack’s instances
Stops a specified instance
Stops a specified stack
Apply cost-allocation tags to a specified stack or layer in AWS OpsWorks Stacks
Unassigns a registered instance from all layers that are using the instance
Unassigns an assigned Amazon EBS volume
Removes tags from a specified stack or layer
Updates a specified app
Updates a registered Elastic IP address’s name
Updates a specified instance
Updates a specified layer
Updates a user’s SSH public key
Updates an Amazon RDS instance
Updates a specified stack
Updates a specified user profile
Updates an Amazon EBS volume’s name or mount point

Examples

```r
## Not run: svc <- opsworks()
svc$assign_instance(
    Foo = 123
)
## End(Not run)
```
Description

AWS OpsWorks for configuration management (CM) is a service that runs and manages configuration management servers. You can use AWS OpsWorks CM to create and manage AWS OpsWorks for Chef Automate and AWS OpsWorks for Puppet Enterprise servers, and add or remove nodes for the servers to manage.

Glossary of terms

- **Server**: A configuration management server that can be highly-available. The configuration management server runs on an Amazon Elastic Compute Cloud (EC2) instance, and may use various other AWS services, such as Amazon Relational Database Service (RDS) and Elastic Load Balancing. A server is a generic abstraction over the configuration manager that you want to use, much like Amazon RDS. In AWS OpsWorks CM, you do not start or stop servers. After you create servers, they continue to run until they are deleted.

- **Engine**: The engine is the specific configuration manager that you want to use. Valid values in this release include Chef Automate and Puppet.

- **Backup**: This is an application-level backup of the data that the configuration manager stores. AWS OpsWorks CM creates an S3 bucket for backups when you launch the first server. A backup maintains a snapshot of a server's configuration-related attributes at the time the backup starts.

- **Events**: Events are always related to a server. Events are written during server creation, when health checks run, when backups are created, when system maintenance is performed, etc. When you delete a server, the server's events are also deleted.

- **Account attributes**: Every account has attributes that are assigned in the AWS OpsWorks CM database. These attributes store information about configuration limits (servers, backups, etc.) and your customer account.

Endpoints

AWS OpsWorks CM supports the following endpoints, all HTTPS. You must connect to one of the following endpoints. Your servers can only be accessed or managed within the endpoint in which they are created.

- opsworks-cm.us-east-1.amazonaws.com
- opsworks-cm.us-east-2.amazonaws.com
- opsworks-cm.us-west-1.amazonaws.com
- opsworks-cm.us-west-2.amazonaws.com
- opsworks-cm.ap-northeast-1.amazonaws.com
- opsworks-cm.ap-southeast-1.amazonaws.com
- opsworks-cm.ap-southeast-2.amazonaws.com
- opsworks-cm.eu-central-1.amazonaws.com
Throttling limits
All API operations allow for five requests per second with a burst of 10 requests per second.

Usage

```
opsworkscm(config = list())
```

Arguments

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

Service syntax

```
svc <- opsworkscm(
    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_node    Associates a new node with the server
create_backup     Creates an application-level backup of a server
create_server     Creates and immediately starts a new server
delete_backup     Deletes a backup
delete_server     Deletes the server and the underlying AWS CloudFormation stacks (including the server’s EC2 instance)
describe_account_attributes Describes your OpsWorks-CM account attributes
describe_backups  Describes backups
describe_events   Describes events for a specified server
describe_node_association_status Returns the current status of an existing association or disassociation request
describe_servers  Lists all configuration management servers that are identified with your account
disassociate_node Disassociates a node from an AWS OpsWorks CM server, and removes the node from the server's managed nodes
export_server_engine_attribute Exports a specified server engine attribute as a base64-encoded string
list_tags_for_resource Returns a list of tags that are applied to the specified AWS OpsWorks for Chef Automate or AWS OpsWorks for Puppet Enterprise server
restore_server    Restores a backup to a server that is in a CONNECTION_LOST, HEALTHY, RUNNING, UNHEALTHY, or TERMINATED state
start_maintenance Manually starts server maintenance
tag_resource      Applies tags to an AWS OpsWorks for Chef Automate or AWS OpsWorks for Puppet Enterprise server
untag_resource     Removes specified tags from an AWS OpsWorks-CM server or backup
```
update_server  Updates settings for a server
update_server_engine_attributes  Updates engine-specific attributes on a specified server

Examples

```r
## Not run: svc <- opsworkscm()
svc$associate_node(
  Foo = 123
)
## End(Not run)
```

organizations  AWS Organizations

Description

AWS Organizations

Usage

```r
organizations(config = list())
```

Arguments

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

Service syntax

```r
csvc <- organizations(
  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_handshake
attach_policy
cancel_handshake
create_account
create_gov_cloud_account
create_organization
create_organizational_unit
create_policy
decline_handshake
delete_organization
delete_organizational_unit
delete_policy
describe_account
describe_create_account_status
describe_effective_policy
describe_handshake
describe_organization
describe_organizational_unit
describe_policy
detach_policy
disable_aws_service_access
disable_policy_type
enable_all_features
enable_aws_service_access
enable_policy_type
invite_account_to_organization
leave_organization
list_accounts
list_accounts_for_parent
list_aws_service_access_for_organization
list_children
list_create_account_status
list_handshakes_for_account
list_handshakes_for_organization
list_organizational_units_for_parent
list_parents
list_policies
list_policies_for_target
list_roots
list_tags_for_resource
list_targets_for_policy
move_account
remove_account_from_organization
tag_resource
untag_resource
update_organizational_unit
update_policy

Sends a response to the originator of a handshake agreeing to the action proposed in the handshake request.
Attaches a policy to a root, an organizational unit (OU), or an individual account.
Cancels a handshake.
Creates an AWS account that is automatically a member of the organization whose credentials made the request.
This action is available if all of the following are true: - You’re authorized to create an AWS account
Creates an AWS organization.
Creates an organizational unit (OU) within a root or parent OU.
Creates a policy of a specified type that you can attach to a root, an organizational unit (OU), or an individual account.
Declines a handshake request.
Deletes the organization.
Deletes an organizational unit (OU) from a root or another OU.
Deletes the specified policy from your organization.
Retrieves AWS Organizations related information about the specified account.
Retrieves the current status of an asynchronous request to create an account.
Returns the contents of the effective tag policy for the account.
Retrieves information about a previously requested handshake.
Retrieves information about the organization that the user’s account belongs to.
Retrieves information about an organizational unit (OU).
Retrieves information about a policy.
Detaches a policy from a target root, organizational unit (OU), or account.
Disables the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.
Disables an organizational control policy type in a root and detaches all policies of that type from the organization root, OUs, and accounts.
Enables all features in an organization.
Enables the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.
Enables a policy type in a root.
Sends an invitation to another account to join your organization as a member account.
Removes a member account from its parent organization.
Lists all the accounts in the organization.
Lists the accounts in an organization that are contained by the specified target root or OU.
Returns a list of the AWS services that you enabled to integrate with your organization.
Lists all of the organizational units (OUs) or accounts that are contained in the specified target root or OU.
Lists the account creation requests that match the specified status that is currently being tracked for the organization.
Lists the current handshakes that are associated with the account of the requesting user.
Lists the handshakes that are associated with the organization that the requesting user is part of.
Lists the organizational units (OUs) in a parent organizational unit or root.
Lists the root or organizational units (OUs) that serve as the immediate parent of the specified target root.
Retrieves the list of all policies in an organization of a specified type.
Lists the policies that are directly attached to the specified target root, organization, or OU.
Lists the roots that are defined in the current organization.
Lists tags for the specified resource.
Lists all the roots, organizational units (OUs), and accounts that the specified policy is attached to.
Moves an account from its current source parent root or organizational unit (OU).
Removes the specified account from the organization.
Adds one or more tags to the specified resource.
Removes a tag from the specified resource.
Renames the specified organizational unit (OU).
Updates an existing policy with a new name, description, or content.
Examples

# Bill is the owner of an organization, and he invites Juan's account
# (222222222222) to join his organization. The following example shows
# Juan's account accepting the handshake and thus agreeing to the
# invitation.
## Not run: svc <- organizations()
svc$accept_handshake(
  HandshakeId = "h-examplehandshakeid111"
)
## End(Not run)

**personalize**  
*Amazon Personalize*

Description

Amazon Personalize is a machine learning service that makes it easy to add individualized recommendations to customers.

Usage

personalize(config = list())

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

Service syntax

```
svc <- personalize(
  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_batch_inference_job
create_campaign
create_dataset
create_dataset_group
create_dataset_import_job
create_event_tracker
create_schema
create_solution
create_solution_version
delete_campaign
delete_dataset
delete_dataset_group
delete_event_tracker
delete_schema
delete_solution
describe_algorithm
describe_batch_inference_job
describe_campaign
describe_dataset
describe_dataset_group
describe_dataset_import_job
describe_event_tracker
describe_feature_transformation
describe_recipe
describe_schema
describe_solution
describe_solution_version
get_solution_metrics
list_batch_inference_jobs
list_campaigns
list_dataset_groups
list_dataset_import_jobs
list_datasets
list_event_trackers
list_recipes
list_schemas
list_solutions
list_solution_versions
update_campaign

create_batch_inference_job
create_campaign
create_dataset
create_dataset_group
create_dataset_import_job
create_event_tracker
create_schema
create_solution
create_solution_version
delete_campaign
delete_dataset
delete_dataset_group
delete_event_tracker
delete_schema
delete_solution
describe_algorithm
describe_batch_inference_job
describe_campaign
describe_dataset
describe_dataset_group
describe_dataset_import_job
describe_event_tracker
describe_feature_transformation
describe_recipe
describe_schema
describe_solution
describe_solution_version
get_solution_metrics
list_batch_inference_jobs
list_campaigns
list_dataset_groups
list_dataset_import_jobs
list_datasets
list_event_trackers
list_recipes
list_schemas
list_solutions
list_solution_versions
update_campaign

create_batch_inference_job
create_campaign
create_dataset
create_dataset_group
create_dataset_import_job
create_event_tracker
create_schema
create_solution
create_solution_version
delete_campaign
delete_dataset
delete_dataset_group
delete_event_tracker
delete_schema
delete_solution
describe_algorithm
describe_batch_inference_job
describe_campaign
describe_dataset
describe_dataset_group
describe_dataset_import_job
describe_event_tracker
describe_feature_transformation
describe_recipe
describe_schema
describe_solution
describe_solution_version
get_solution_metrics
list_batch_inference_jobs
list_campaigns
list_dataset_groups
list_dataset_import_jobs
list_datasets
list_event_trackers
list_recipes
list_schemas
list_solutions
list_solution_versions
update_campaign

Examples

## Not run: svc <- personalize()
svc$create_batch_inference_job(  
  Foo = 123
)
## End(Not run)
**personalizeevents**  
*Amazon Personalize Events*

**Description**

Amazon Personalize Events

**Usage**

```
personalizeevents(config = list())
```

**Arguments**

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

**Service syntax**

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

**Operations**

- `put_events`  
  Records user interaction event data

**Examples**

```r
## Not run: svc <- personalizeevents()
svc$put_events(
  Foo = 123
)
## End(Not run)
```
**personalizeruntime**  
*Amazon Personalize Runtime*

**Description**

Amazon Personalize Runtime

**Usage**

personalizeruntime(config = list())

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

**Service syntax**

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

**Operations**

- **get_personalized_ranking**  
  Re-ranks a list of recommended items for the given user
- **get_recommendations**  
  Returns a list of recommended items

**Examples**

```r
## Not run: svc <- personalizeruntime()
svc$get_personalized_ranking(
  Foo = 123
)
```
AWS Performance Insights

Description

AWS Performance Insights enables you to monitor and explore different dimensions of database load based on data captured from a running RDS instance. The guide provides detailed information about Performance Insights data types, parameters and errors. For more information about Performance Insights capabilities see Using Amazon RDS Performance Insights in the Amazon RDS User Guide.

The AWS Performance Insights API provides visibility into the performance of your RDS instance, when Performance Insights is enabled for supported engine types. While Amazon CloudWatch provides the authoritative source for AWS service vended monitoring metrics, AWS Performance Insights offers a domain-specific view of database load measured as Average Active Sessions and provided to API consumers as a 2-dimensional time-series dataset. The time dimension of the data provides DB load data for each time point in the queried time range, and each time point decomposes overall load in relation to the requested dimensions, such as SQL, Wait-event, User or Host, measured at that time point.

Usage

```r
pi(config = list())
```

Arguments

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

Service syntax

```r
csvc <- pi(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```
Operations
**describe_dimension_keys**  
For a specific time period, retrieve the top N dimension keys for a metric

**get_resource_metrics**  
Retrieve Performance Insights metrics for a set of data sources, over a time period

---

### Examples

```r
## Not run: svc <- pi()
svc$describe_dimension_keys(
  Foo = 123
)
## End(Not run)
```

---

**Amazon Pinpoint**

---

### Description

Doc Engage API - Amazon Pinpoint API

### Usage

```r
pinpoint(config = list())
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

### Service syntax

```r
csvc <- pinpoint(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

### Operations
pinpoint

create_app
create_campaign
create_email_template
create_export_job
create_import_job
create_journey
create_push_template
create_segment
create_sms_template
create_voice_template
delete_adm_channel
delete_apns_channel
delete_apns_sandbox_channel
delete_apns_voip_channel
delete_apns_voip_sandbox_channel
delete_app
delete_baidu_channel
delete_campaign
delete_email_channel
delete_email_template
delete_endpoint
delete_event_stream
delete_gcm_channel
delete_journey
delete_push_template
delete_segment
delete_sms_channel
delete_sms_template
delete_user_endpoints
delete_voice_channel
delete_voice_template
get_adm_channel
get_apns_channel
get_apns_sandbox_channel
get_apns_voip_channel
get_apns_voip_sandbox_channel
get_app
get_application_date_range_kpi
get_application_settings
get_apps
get_baidu_channel
get_campaign
get_campaign_activities
get_campaign_date_range_kpi
get_campaigns
get_campaign_version
get_campaign_versions
get_channels

Creates an application
Creates a new campaign for an application or updates the settings of an existing campaign
Creates a message template for messages that are sent through the email channel
Creates an export job for an application
Creates an import job for an application
Creates a journey for an application
Creates a new segment for an application or updates the configuration, dimension, and other settings for an existing segment
Creates a message template for messages that are sent through the SMS channel
Creates a message template for messages that are sent through the voice channel
Disables the ADM channel for an application and deletes any existing settings for the channel
Disables the APNs channel for an application and deletes any existing settings for the channel
Disables the APNs sandbox channel for an application and deletes any existing settings for the channel
Disables the APNs VoIP channel for an application and deletes any existing settings for the channel
Disables the APNs VoIP sandbox channel for an application and deletes any existing settings for the channel
Deletes an application
Disables the Baidu channel for an application and deletes any existing settings for the channel
Deletes a campaign from an application
Disables the email channel for an application and deletes any existing settings for the channel
Deletes a message template for messages that were sent through the email channel
Deletes an endpoint from an application
Deletes the event stream for an application
Disables the GCM channel for an application and deletes any existing settings for the channel
Deletes a journey from an application
Deletes a message template for messages that were sent through a push notification channel
Deletes a segment from an application
Disables the SMS channel for an application and deletes any existing settings for the channel
Deletes a message template for messages that were sent through the SMS channel
Deletes all the endpoints that are associated with a specific user ID
Disables the voice channel for an application and deletes any existing settings for the channel
Deletes a message template for messages that were sent through the voice channel
Retrieves information about the status and settings of the ADM channel for an application
Retrieves information about the status and settings of the APNs channel for an application
Retrieves information about the status and settings of the APNs sandbox channel for an application
Retrieves information about the status and settings of the APNs VoIP channel for an application
Retrieves information about the status and settings of the APNs VoIP sandbox channel for an application
Retrieves (queries) pre-aggregated data for a standard metric that applies to an application
Retrieves information about the settings for an application
Retrieves information about all the applications that are associated with your Amazon Pinpoint account
Retrieves information about the status and settings of the Baidu channel for an application
Retrieves information about the status, configuration, and other settings for a campaign
Retrieves information about all the activities for a campaign
Retrieves (queries) pre-aggregated data for a standard metric that applies to a campaign
Retrieves information about the status, configuration, and other settings for all the campaigns associated with an application
Retrieves information about the status, configuration, and other settings for all versions of a campaign
Retrieves information about the history and status of each channel for an application
get_email_channel
get_email_template
get_endpoint
get_event_stream
get_export_job
get_export_jobs
get_gcm_channel
get_import_job
get_import_jobs
get_journey
get_journey_date_range_kpi
get_journey_execution_activity_metrics
get_journey_execution_metrics
get_push_template
get_segment
get_segment_export_jobs
get_segment_import_jobs
get_segments
get_segment_version
get_segment_versions
get_sms_channel
get_sms_template
get_user_endpoints
get_voice_channel
get_voice_template
list_journeys
list_tags_for_resource
list_templates
list_template_versions
phone_number_validate
put_event
put_event_stream
remove_attributes
send_messages
send_users_messages
tag_resource
untag_resource
update_adm_channel
update_apns_channel
update_apns_sandbox_channel
update_apns_voip_channel
update_apns_voip_sandbox_channel
update_application_settings
update_baidu_channel
update_campaign
update_email_channel
update_email_template
update_endpoint

Retrieves information about the status and settings of the email channel for an application.
Retrieves the content and settings of a message template for messages that are sent through the email channel.
Retrieves information about the settings and attributes of a specific endpoint for an application.
Retrieves information about the event stream settings for an application.
Retrieves information about the status and settings of a specific export job for an application.
Retrieves information about the status and settings of all the export jobs for an application.
Retrieves information about the status and settings of the GCM channel for an application.
Retrieves information about the status and settings of a specific import job for an application.
Retrieves information about the status and settings of all the import jobs for an application.
Retrieves information about the status, configuration, and other settings for a journey.
Retrieves (queries) pre-aggregated data for a standard engagement metric that applies to a journey.
Retrieves (queries) pre-aggregated data for a standard execution metric that applies to a journey.
Retrieves (queries) pre-aggregated data for a standard execution metric that applies to a journey activity.
Retrieves information about the configuration, dimension, and other settings for a segment.
Retrieves information about the status and settings of the export jobs for a segment.
Retrieves information about the status and settings of the import jobs for a segment.
Retrieves information about the configuration, dimension, and other settings for all the endpoints that are associated with a specific user ID.
Retrieves all the tags (keys and values) that are associated with an application, campaign, journey, message template, or segment.
Retrieves information about all the versions that are associated with your Amazon Pinpoint account.
Retrieves information about all the versions of a specific message template.
Retrieves information about a phone number.
Creates a new event to record for endpoints, or creates or updates endpoint data that already exists.
Creates a new event stream for an application or updates the settings of an existing event stream.
Removes one or more attributes, of the same attribute type, from all the endpoints that are associated with an application.
Creates and sends a direct message.
Add one or more tags (keys and values) to an application, campaign, journey, message template, or segment.
Removes one or more tags (keys and values) from an application, campaign, journey, message template, or segment.
Enables the ADM channel for an application or updates the status and settings of the ADM channel.
Enables the APNs channel for an application or updates the status and settings of the APNs channel.
Enables the APNs sandbox channel for an application or updates the status and settings of the APNs sandbox channel.
Enables the APNs VoIP channel for an application or updates the status and settings of the APNs VoIP channel.
Enables the APNs VoIP sandbox channel for an application or updates the status and settings of the APNs VoIP sandbox channel.
Updates the settings for an application.
Enables the Baidu channel for an application or updates the status and settings of the Baidu channel.
Updates the configuration and other settings for a campaign.
Enables the email channel for an application or updates the status and settings of the email channel.
Updates an existing message template for messages that are sent through the email channel.
Creates a new endpoint for an application or updates the settings and attributes of an existing endpoint.
update_endpoints_batch
update_gcm_channel
update_journey
update_journey_state
update_push_template
update_segment
update_sms_channel
update_sms_template
update_template_active_version
update_voice_channel
update_voice_template

Creates a new batch of endpoints for an application or updates the settings and attributes of a batch of existing endpoints for an application.

Enables the GCM channel for an application or updates the status and settings of the GCM channel for an application.

Updates the configuration and other settings for a journey.

Cancels (stops) an active journey.

Updates an existing message template for messages that are sent through a push notification channel.

Creates a new segment for an application or updates the configuration, dimension, and other settings for an existing segment that’s associated with an application.

Enables the SMS channel for an application or updates the status and settings of the SMS channel for an application.

Updates an existing message template for messages that are sent through the SMS channel.

Changes the status of a specific version of a message template to `active`.

Enables the voice channel for an application or updates the status and settings of the voice channel for an application.

Updates an existing message template for messages that are sent through the voice channel.

### Examples
```
## Not run: svc <- pinpoint()
svc$create_app(
  Foo = 123
)
## End(Not run)
```

---

## Amazon Pinpoint Email Service

### Description

Welcome to the **Amazon Pinpoint Email API Reference**. This guide provides information about the Amazon Pinpoint Email API (version 1.0), including supported operations, data types, parameters, and schemas.

**Amazon Pinpoint** is an AWS service that you can use to engage with your customers across multiple messaging channels. You can use Amazon Pinpoint to send email, SMS text messages, voice messages, and push notifications. The Amazon Pinpoint Email API provides programmatic access to options that are unique to the email channel and supplement the options provided by the Amazon Pinpoint API.

If you’re new to Amazon Pinpoint, you might find it helpful to also review the **Amazon Pinpoint Developer Guide**. The Amazon Pinpoint Developer Guide provides tutorials, code samples, and procedures that demonstrate how to use Amazon Pinpoint features programmatically and how to integrate Amazon Pinpoint functionality into mobile apps and other types of applications. The guide also provides information about key topics such as Amazon Pinpoint integration with other AWS services and the limits that apply to using the service.

The Amazon Pinpoint Email API is available in several AWS Regions and it provides an endpoint for each of these Regions. For a list of all the Regions and endpoints where the API is currently available, see **AWS Service Endpoints** in the Amazon Web Services General Reference. To learn
more about AWS Regions, see Managing AWS Regions in the Amazon Web Services General Reference.

In each Region, AWS maintains multiple Availability Zones. These Availability Zones are physically isolated from each other, but are united by private, low-latency, high-throughput, and highly redundant network connections. These Availability Zones enable us to provide very high levels of availability and redundancy, while also minimizing latency. To learn more about the number of Availability Zones that are available in each Region, see AWS Global Infrastructure.

Usage

pinpointemail(config = list())

Arguments

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

Service syntax

svc <- pinpointemail(
  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_configuration_set Create a configuration set
create_configuration_set_event_destination Create an event destination
create_dedicated_ip_pool Create a new pool of dedicated IP addresses
create_deliverability_test_report Create a new predictive inbox placement test
create_email_identity Verifies an email identity for use with Amazon Pinpoint
delete_configuration_set Delete an existing configuration set
delete_configuration_set_event_destination Delete an event destination
delete_dedicated_ip_pool Delete a dedicated IP pool
delete_email_identity Deletes an email identity that you previously verified for use with Amazon Pinpoint
get_account Obtain information about the email-sending status and capabilities of your Amazon Pinpoint account
get_blacklist_reports Retrieve a list of the blacklists that your dedicated IP addresses appear on
get_configuration_set Get information about an existing configuration set, including the dedicated IP addresses that are associated with it
get_configuration_set_event_destinations Retrieve a list of event destinations that are associated with a configuration set
get_dedicated_ip Get information about a dedicated IP address, including the name of the dedicated IP pool it belongs to
### Examples

```r
# Not run: svc <- pinpointemail()
svc$create_configuration_set(
  Foo = 123
)
# End(Not run)
```

---

**pinpointsmsvoice**  
*Amazon Pinpoint SMS and Voice Service*

---

**Description**

Pinpoint SMS and Voice Messaging public facing APIs
Usage

pinpointsmsvoice(config = list())

Arguments

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

Service syntax

svc <- pinpointsmsvoice(
  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_configuration_set Create a new configuration set
create_configuration_set_event_destination Create a new event destination in a configuration set
delete_configuration_set Deletes an existing configuration set
delete_configuration_set_event_destination Deletes an event destination in a configuration set
get_configuration_set_event_destinations Obtain information about an event destination, including the types of events it reports, the Amazon Resource Name (ARN) of the destination, and the name of the event destination
list_configuration_sets List all of the configuration sets associated with your Amazon Pinpoint account
send_voice_message Create a new voice message and send it to a recipient’s phone number
update_configuration_set_event_destination Update an event destination in a configuration set

Examples

## Not run: svc <- pinpointsmsvoice()
svc$create_configuration_set(
  Foo = 123
)
## End(Not run)
Amazon Polly

Description

Amazon Polly is a web service that makes it easy to synthesize speech from text.

The Amazon Polly service provides API operations for synthesizing high-quality speech from plain text and Speech Synthesis Markup Language (SSML), along with managing pronunciations lexicons that enable you to get the best results for your application domain.

Usage

`polly(config = list())`

Arguments

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

Service syntax

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

Operations

- `delete_lexicon` Deletes the specified pronunciation lexicon stored in an AWS Region
- `describe_voices` Returns the list of voices that are available for use when requesting speech synthesis
- `get_lexicon` Returns the content of the specified pronunciation lexicon stored in an AWS Region
- `get_speech_synthesis_task` Retrieves a specific SpeechSynthesisTask object based on its TaskID
- `list_lexicons` Returns a list of pronunciation lexicons stored in an AWS Region
- `list_speech_synthesis_tasks` Returns a list of SpeechSynthesisTask objects ordered by their creation date
- `put_lexicon` Stores a pronunciation lexicon in an AWS Region
- `start_speech_synthesis_task` Allows the creation of an asynchronous synthesis task, by starting a new SpeechSynthesisTask
- `synthesize_speech` Synthesizes UTF-8 input, plain text or SSML, to a stream of bytes
Examples

# Deletes a specified pronunciation lexicon stored in an AWS Region.
## Not run: svc <- polly()
svc$delete_lexicon(
  Name = "example"
)
## End(Not run)

---

**Description**

AWS Price List Service API (AWS Price List Service) is a centralized and convenient way to programmatically query Amazon Web Services for services, products, and pricing information. The AWS Price List Service uses standardized product attributes such as Location, Storage Class, and Operating System, and provides prices at the SKU level. You can use the AWS Price List Service to build cost control and scenario planning tools, reconcile billing data, forecast future spend for budgeting purposes, and provide cost benefit analysis that compare your internal workloads with AWS.

Use GetServices without a service code to retrieve the service codes for all AWS services, then GetServices with a service code to retrieve the attribute names for that service. After you have the service code and attribute names, you can use GetAttributeValueValues to see what values are available for an attribute. With the service code and an attribute name and value, you can use GetProducts to find specific products that you’re interested in, such as an AmazonEC2 instance, with a Provisioned IOPS volumeType.

**Service Endpoint**

AWS Price List Service API provides the following two endpoints:

- https://api.pricing.us-east-1.amazonaws.com
- https://api.pricing.ap-south-1.amazonaws.com

**Usage**

`pricing(config = list())`

**Arguments**

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

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

Operations

- **describe_services**
  
  Returns the metadata for one service or a list of the metadata for all services

- **get_attribute_values**
  
  Returns a list of attribute values

- **get_products**
  
  Returns a list of all products that match the filter criteria

Examples

```r
## Not run: svc <- pricing()
svc$describe_services(
  FormatVersion = "aws_v1",
  MaxResults = 1L,
  ServiceCode = "AmazonEC2"
)
## End(Not run)
```

---

**quicksight**

Amazon QuickSight API Reference

Amazon QuickSight is a fully managed, serverless business intelligence service for the AWS Cloud that makes it easy to extend data and insights to every user in your organization. This API reference contains documentation for a programming interface that you can use to manage Amazon QuickSight.
Usage

quicksight(config = list())

Arguments

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

Service syntax

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

Operations

cancel_ingestion Cancels an ongoing ingestion of data into SPICE
create_dashboard Creates a dashboard from a template
create_data_set Creates a dataset
create_data_source Creates a data source
create_group Creates an Amazon QuickSight group
create_group_membership Adds an Amazon QuickSight user to an Amazon QuickSight group
create_iam_policy_assignment Creates an assignment with one specified IAM policy, identified by its Amazon Resource Name (ARN)
create_ingestion Creates and starts a new SPICE ingestion on a dataset Any ingestions operating on tagged datasets inherit the same tags automatically for use in access control
create_template Creates a template from an existing QuickSight analysis or template
create_template_alias Creates a template alias for a template
delete_dashboard Deletes a dashboard
delete_data_set Deletes a dataset
delete_data_source Deletes the data source permanently
delete_group Removes a user group from Amazon QuickSight
delete_group_membership Removes a user from a group so that the user is no longer a member of the group
delete_iam_policy_assignment Deletes an existing IAM policy assignment
delete_template Deletes a template
delete_template_alias Deletes the item that the specified template alias points to
delete_user Deletes the Amazon QuickSight user that is associated with the identity of the AWS Identity and Access Management (IAM) user or role that's making the call
delete_user_by_principal_id Deletes a user identified by its principal ID
describe_dashboard Provides a summary for a dashboard
describe_dashboard_permissions Describes read and write permissions for a dashboard
describe_data_set  
Describes a dataset  

describe_data_set_permissions  
Describes the permissions on a dataset  

describe_data_source  
Describes a data source  

describe_data_source_permissions  
Describes the permissions for a data source  

describe_group  
Returns an Amazon QuickSight group’s description and Amazon Resource Name (ARN)  

describe_iam_policy_assignment  
Describes an existing IAM policy assignment, as specified by the assignment name  

describe_ingestion  
Describes a SPICE ingestion  

describe_template  
Describes a template’s metadata  

describe_template_alias  
Describes the template alias for a template  

describe_template_permissions  
Describes read and write permissions on a template  

describe_user  
Returns information about a user, given the user name  

get_dashboard_embed_url  
Generates a server-side embeddable URL and authorization code  

list_dashboards  
Lists dashboards in an AWS account  

list_dashboard_versions  
Lists all the versions of the dashboards in the QuickSight subscription  

list_data_sets  
Lists all of the datasets belonging to the current AWS account in an AWS Region  

list_data_sources  
Lists data sources in current AWS Region that belong to this AWS account  

list_group_memberships  
Lists member users in a group  

list_groups  
Lists all user groups in Amazon QuickSight  

list_iam_policy_assignments  
Lists IAM policy assignments in the current Amazon QuickSight account  

list_iam_policy_assignments_for_user  
Lists all the IAM policy assignments, including the Amazon Resource Names (ARNs)  

list_ingestions  
Lists the history of SPICE ingestions for a dataset  

list_tags_for_resource  
Lists the tags assigned to a resource  

list_template_aliases  
Lists all the aliases of a template  

list_templates  
Lists all the templates assigned to a resource  

list_template_versions  
Lists all the templates in the current Amazon QuickSight account  

list_user_groups  
Lists the Amazon QuickSight groups that an Amazon QuickSight user is a member of  

list_users  
Returns a list of all of the Amazon QuickSight users belonging to this account  

register_user  
Creates an Amazon QuickSight user, whose identity is associated with the AWS Identity and Access Management (IAM) identity or role specified in the request  

tag_resource  
Assigns one or more tags (key-value pairs) to the specified QuickSight resource  

untag_resource  
Removes a tag or tags from a resource  

update_dashboard  
Updates a dashboard in an AWS account  

update_dashboard_permissions  
Updates read and write permissions on a dashboard  

update_dashboard_published_version  
Updates the published version of a dashboard  

update_data_set  
Updates a dataset  

update_data_set_permissions  
Updates the permissions on a dataset  

update_data_source  
Updates a data source  

update_data_source_permissions  
Updates the permissions to a data source  

update_group  
Changes a group description  

update_iam_policy_assignment  
Updates an existing IAM policy assignment  

update_template  
Updates a template from an existing Amazon QuickSight analysis or another template  

update_template_alias  
Updates the template alias of a template  

update_template_permissions  
Updates the resource permissions for a template  

update_user  
Updates an Amazon QuickSight user  

Examples

```
## Not run: svc <- quicksight()
```
ram

svc$s_cancel_ingestion(
    Foo = 123
)
## End(Not run)

---

**AWS Resource Access Manager**

**Description**

Use AWS Resource Access Manager to share AWS resources between AWS accounts. To share a resource, you create a resource share, associate the resource with the resource share, and specify the principals that can access the resources associated with the resource share. The following principals are supported: AWS accounts, organizational units (OU) from AWS Organizations, and organizations from AWS Organizations.

For more information, see the *AWS Resource Access Manager User Guide*.

**Usage**

```r
ram(config = list())
```

**Arguments**

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

**Service syntax**

```r
csvc <- ram(  
    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_resource_share_invitation` Accepts an invitation to a resource share from another AWS account
- `associate_resource_share` Associates the specified resource share with the specified principals and resources
- `associate_resource_share_permission` Associates a permission with a resource share
- `create_resource_share` Creates a resource share
- `delete_resource_share` Deletes the specified resource share
- `disassociate_resource_share` Disassociates the specified principals or resources from the specified resource share
- `disassociate_resource_share_permission` Disassociates an AWS RAM permission from a resource share
- `enable_sharing_with_aws_organization` Enables resource sharing within your AWS Organization
- `get_permission` Gets the contents of an AWS RAM permission in JSON format
- `get_resource_policies` Gets the policies for the specified resources that you own and have shared
- `get_resource_share_associations` Gets the resources or principals for the resource shares that you own
- `get_resource_share_invitations` Gets the invitations for resource sharing that you’ve received
- `get_resource_shares` Gets the resource shares that you own or the resource shares that are shared with you
- `list_pending_invitation_resources` Lists the resources in a resource share that is shared with you but that the invitation is still pending for
- `list_permissions` Lists the AWS RAM permissions
- `list_principals` Lists the principals that you have shared resources with or that have shared resources with you
- `list_resources` Lists the resources that you added to a resource shares or the resources that are shared with you
- `list_resource_share_permissions` Lists the AWS RAM permissions that are associated with a resource share
- `promote_resource_share_created_from_policy` Resource shares that were created by attaching a policy to a resource are visible only to the resource share owner, and the resource share cannot be modified in AWS RAM
- `reject_resource_share_invitation` Rejects an invitation to a resource share from another AWS account
- `tag_resource` Adds the specified tags to the specified resource share that you own
- `untag_resource` Removes the specified tags from the specified resource share that you own
- `update_resource_share` Updates the specified resource share that you own

Examples

```r
## Not run: svc <- ram()
svc$accept_resource_share_invitation(
  Foo = 123
)
## End(Not run)
```

rds

Amazon Relational Database Service

Description

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizeable capacity for an industry-standard relational database and manages common database administration tasks, freeing up developers to focus on what makes their applications and businesses unique.
Amazon RDS gives you access to the capabilities of a MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, Oracle, or Amazon Aurora database server. These capabilities mean that the code, applications, and tools you already use today with your existing databases work with Amazon RDS without modification. Amazon RDS automatically backs up your database and maintains the database software that powers your DB instance. Amazon RDS is flexible: you can scale your DB instance’s compute resources and storage capacity to meet your application’s demand. As with all Amazon Web Services, there are no up-front investments, and you pay only for the resources you use.

This interface reference for Amazon RDS contains documentation for a programming or command line interface you can use to manage Amazon RDS. Note that Amazon RDS is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

**Amazon RDS API Reference**

- For the alphabetical list of API actions, see API Actions.
- For the alphabetical list of data types, see Data Types.
- For a list of common query parameters, see Common Parameters.
- For descriptions of the error codes, see Common Errors.

**Amazon RDS User Guide**

- For a summary of the Amazon RDS interfaces, see Available RDS Interfaces.
- For more information about how to use the Query API, see Using the Query API.

### Usage

```r
rds(config = list())
```

### Arguments

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

### Service syntax

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

Operations

add_role_to_db_cluster
add_role_to_db_instance
add_source_identifier_to_subscription
add_tags_to_resource
apply_pending_maintenance_action
authorize_db_security_group_ingress
backtrack_db_cluster
build_auth_token
copy_db_cluster_parameter_group
copy_db_cluster_snapshot
copy_db_parameter_group
copy_db_snapshot
copy_option_group
create_custom_availability_zone
create_db_cluster
create_db_cluster_endpoint
create_db_cluster_parameter_group
create_db_cluster_snapshot
create_db_instance
create_db_instance_read_replica
create_db_parameter_group
create_db_proxy
create_db_security_group
create_db_snapshot
create_db_subnet_group
create_event_subscription
create_global_cluster
create_option_group
delete_custom_availability_zone
delete_db_cluster
delete_db_cluster_endpoint
delete_db_cluster_parameter_group
delete_db_cluster_snapshot
delete_db_instance
delete_db_instance_automated_backup
delete_db_parameter_group
delete_db_proxy
delete_db_security_group
delete_db_snapshot
delete_db_subnet_group
delete_event_subscription
delete_global_cluster

region = "string"

This is prerelease documentation for the RDS Database Proxy feature in preview release.

Associates an Identity and Access Management (IAM) role from an Amazon Aurora DB cluster
Associates an AWS Identity and Access Management (IAM) role with a DB instance
Adds a source identifier to an existing RDS event notification subscription
Adds metadata tags to an Amazon RDS resource
Applies a pending maintenance action to a resource (for example, to a DB instance)
Enables ingress to a DBSecurityGroup using one of two forms of authorization
Backtracks a DB cluster to a specific time, without creating a new DB cluster
Return an authentication token for a database connection
Copies the specified DB cluster parameter group
Copies a snapshot of a DB cluster
Copies the specified DB parameter group
Copies the specified DB snapshot
Copies the specified option group
Creates a custom Availability Zone (AZ)
Creates a new Amazon Aurora DB cluster
Creates a new custom endpoint and associates it with an Amazon Aurora DB cluster
Creates a new DB cluster parameter group
Creates a snapshot of a DB cluster
Creates a new DB instance
Creates a new DB instance that acts as a Read Replica for an existing source DB instance
Creates a new DB parameter group
This is prerelease documentation for the RDS Database Proxy feature in preview release
Creates a new DB security group
Creates a DB snapshot
Creates a new DB subnet group
Creates an RDS event notification subscription
Creates an Aurora global database spread across multiple regions
Creates a new option group
Deletes a custom Availability Zone (AZ)
The DeleteDBCluster action deletes a previously provisioned DB cluster
Deletes a custom endpoint and removes it from an Amazon Aurora DB cluster
Deletes a specified DB cluster parameter group
Deletes a DB cluster snapshot
The DeleteDBInstance action deletes a previously provisioned DB instance
Deletes automated backups based on the source instance’s DbiResourceId value
Deletes a specified DB parameter group
This is prerelease documentation for the RDS Database Proxy feature in preview release
Deletes a DB security group
Deletes a DB snapshot
Deletes a DB subnet group
Deletes an RDS event notification subscription
Deletes a global database cluster
delete_installation_media
delete_option_group
deregister_db_proxy_targets
describe_account_attributes
describe_certificates
describe_custom_availability_zones
describe_db_cluster_backtracks
describe_db_cluster_endpoints
describe_db_cluster_parameter_groups
describe_db_cluster_parameters
describe_db_clusters
describe_db_cluster_snapshot_attributes
describe_db_cluster_snapshots
describe_db_engine_versions
describe_db_instance_automated_backups
describe_db_instances
describe_db_log_files
describe_db_parameter_groups
describe_db_parameters
describe_db_proxy_target_groups
describe_db_proxy_targets
describe_db_security_groups
describe_db_snapshot_attributes
describe_db_snapshots
describe_db_subnet_groups
describe_engine_default_cluster_parameters
describe_engine_default_parameters
describe_event_categories
describe_events
describe_event_subscriptions
describe_global_clusters
describe_installation_media
describe_option_group_options
describe_option_groups
describe_orderable_db_instance_options
describe_pending_maintenance_actions
describe_reserved_db_instances
describe_reserved_db_instances_offerings
describe_source_regions
describe_valid_db_instance_modifications
download_db_log_file_portion
failover_db_cluster
import_installation_media
list_tags_for_resource
modify_certificates
modify_current_db_cluster_capacity
modify_db_cluster

Deletes the installation medium for a DB engine that requires an on-premises customer provided license, such as Microsoft SQL Server
Delete an existing option group
This is prerelease documentation for the RDS Database Proxy feature in preview release
Lists all of the attributes for a customer account
Lists the set of CA certificates provided by Amazon RDS for this AWS account
Returns information about custom Availability Zones (AZs)
Returns information about backtracks for a DB cluster
Returns information about endpoints for an Amazon Aurora DB cluster
Returns a list of DBClusterParameterGroup descriptions
Returns the detailed parameter list for a particular DB cluster parameter group
Returns information about provisioned Aurora DB clusters
Returns a list of DB cluster snapshot attribute names and values for a manual DB cluster snapshot
Returns information about DB cluster snapshots
Returns a list of the available DB engines
Displays backups for both current and deleted instances
Returns information about provisioned RDS instances
Returns a list of DB log files for the DB instance
Returns a list of DBParameterGroup descriptions
Returns the detailed parameter list for a particular DB parameter group
This is prerelease documentation for the RDS Database Proxy feature in preview release
This is prerelease documentation for the RDS Database Proxy feature in preview release
This is prerelease documentation for the RDS Database Proxy feature in preview release
Returns a list of DBSecurityGroup descriptions
Returns a list of DB snapshot attribute names and values for a manual DB snapshot
Returns information about DB snapshots
Returns a list of DBSubnetGroup descriptions
Returns the default engine and system parameter information for the cluster database engine
Returns the default engine and system parameter information for the specified database engine
Displays a list of categories for all event source types, or, if specified, for a specific event source type
Returns events related to DB instances, DB security groups, DB snapshots, and DB parameter groups
Returns information about Aurora global database clusters
Describes the available installation media for a DB engine that requires an on-premises customer provided license
Describes all available options
Describes the available option groups
Returns a list of orderable DB instance options for the specified engine
Returns a list of resources (for example, DB instances) that have at least one pending maintenance action
Returns information about reserved DB instances for this account, or about a specific reserved DB instance
Lists available reserved DB instance offerings
Returns a list of the source AWS Regions where the current AWS Region can create a Read Replica or copy a DB snapshot from
You can call DescribeValidDBInstanceModifications to learn what modifications are available
Downloads all or a portion of the specified log file, up to 1 MB in size
Forces a failover for a DB cluster
Imports the installation media for a DB engine that requires an on-premises customer provided license
Lists all tags on an Amazon RDS resource
Override the system-default Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificate for Amazon RDS for new DB instances, or remove the override
Set the capacity of an Aurora Serverless DB cluster to a specific value
Modify a setting for an Amazon Aurora DB cluster
modify_db_cluster_endpoint
modify_db_cluster_parameter_group
modify_db_cluster_snapshot_attribute
modify_db_instance
modify_db_parameter_group
modify_db_proxy
modify_db_proxy_target_group
modify_db_snapshot
modify_db_snapshot_attribute
modify_db_subnet_group
modify_event_subscription
modify_global_cluster
modify_option_group
promote_read_replica
promote_read_replica_db_cluster
purchase_reserved_db_instances_offering
reboot_db_instance
register_db_proxy_targets
remove_from_global_cluster
remove_role_from_db_cluster
remove_role_from_db_instance
remove_source_identifier_from_subscription
remove_tags_from_resource
reset_db_cluster_parameter_group
reset_db_parameter_group
restore_db_cluster_from_s3
restore_db_cluster_from_snapshot
restore_db_cluster_to_point_in_time
restore_db_instance_from_db_snapshot
restore_db_instance_from_s3
restore_db_instance_to_point_in_time
revoke_db_security_group_ingress
start_activity_stream
start_db_cluster
start_db_instance
stop_activity_stream
stop_db_cluster
stop_db_instance

Modifies the properties of an endpoint in an Amazon Aurora DB cluster
Modifies the parameters of a DB cluster parameter group
Adds an attribute and values to, or removes an attribute and values from, a manual DB cluster snapshot
Modifies settings for a DB instance
Modifies the parameters of a DB parameter group
This is prerelease documentation for the RDS Database Proxy feature in preview release
This is prerelease documentation for the RDS Database Proxy feature in preview release
Updates a manual DB snapshot, which can be encrypted or not encrypted, with attribute and values
Adds an attribute and values to, or removes an attribute and values from, a manual DB snapshot
Modifies an existing DB subnet group
Modifies an existing RDS event notification subscription
Modifies a setting for an Amazon Aurora global cluster
Modifies an existing option group
Promotes a Read Replica DB instance to a standalone DB instance
Promotes a Read Replica DB cluster to a standalone DB cluster
Purchases a reserved DB instance offering
You might need to reboot your DB instance, usually for maintenance reasons
This is prerelease documentation for the RDS Database Proxy feature in preview release
Detaches an Aurora secondary cluster from an Aurora global database cluster
Disassociates an AWS Identity and Access Management (IAM) role from an Amazon Aurora DB instance
Disassociates an AWS Identity and Access Management (IAM) role from a DB cluster
Removes a source identifier from an existing RDS event notification subscription
Removes metadata tags from an Amazon RDS resource
Modifies the parameters of a DB cluster parameter group to the default value
Modifies the parameters of a DB parameter group to the engine/system default value
Creates an Amazon Aurora DB cluster from data stored in an Amazon S3 bucket
Creates a new DB cluster from a DB snapshot or DB cluster snapshot
Restores a DB cluster to an arbitrary point in time
Creates a new DB instance from a DB snapshot
Amazon Relational Database Service (Amazon RDS) supports importing MySQL databases
Restores a DB instance to an arbitrary point in time
Reveks ingress from a DBSecurityGroup for previously authorized IP ranges
Starts a database activity stream to monitor activity on the database
Starts an Amazon Aurora DB cluster that was stopped using the AWS console
Starts an Amazon RDS DB instance that was stopped using the AWS console
Stops a database activity stream that was started using the AWS console, the stop_db_cluster
Stops an Amazon Aurora DB cluster
Stops an Amazon RDS DB instance

Examples

```r
## Not run: svc <- rds()
svc$add_role_to_db_cluster($
  Foo = 123
)
## End(Not run)
```
AWS RDS Data Service

Description

Amazon RDS Data Service

Amazon RDS provides an HTTP endpoint to run SQL statements on an Amazon Aurora Serverless DB cluster. To run these statements, you work with the Data Service API.

For more information about the Data Service API, see Using the Data API for Aurora Serverless in the Amazon Aurora User Guide.

If you have questions or comments related to the Data API, send email to Rds-data-api-feedback@amazon.com.

Usage

rdsdataservice(config = list())

Arguments

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

Service syntax

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

Operations

- **batch_execute_statement**: Runs a batch SQL statement over an array of data
- **begin_transaction**: Starts a SQL transaction
- **commit_transaction**: Ends a SQL transaction started with the BeginTransaction operation and commits the changes
- **execute_sql**: Runs one or more SQL statements
- **execute_statement**: Runs a SQL statement against a database
- **rollback_transaction**: Performs a rollback of a transaction
Examples

```r
## Not run: svc <- rdsdataservice()
svc$batch_execute_statement(
  Foo = 123
)
## End(Not run)
```

---

## Description

### Overview

This is an interface reference for Amazon Redshift. It contains documentation for one of the programming or command line interfaces you can use to manage Amazon Redshift clusters. Note that Amazon Redshift is asynchronous, which means that some interfaces may require techniques, such as polling or asynchronous callback handlers, to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a change is applied immediately, on the next instance reboot, or during the next maintenance window. For a summary of the Amazon Redshift cluster management interfaces, go to Using the Amazon Redshift Management Interfaces.

Amazon Redshift manages all the work of setting up, operating, and scaling a data warehouse: provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine. You can focus on using your data to acquire new insights for your business and customers.

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the Amazon Redshift Getting Started Guide.

If you are a database developer, the Amazon Redshift Database Developer Guide explains how to design, build, query, and maintain the databases that make up your data warehouse.

### Usage

```r
redshift(config = list())
```

### Arguments

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

    svc <- redshift(
        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_reserved_node_exchange`: Exchanges a DC1 Reserved Node for a DC2 Reserved Node with no changes to the configuration (term, payment type, or number of nodes) and no additional costs.
- `authorize_cluster_security_group_ingress`: Adds an inbound (ingress) rule to an Amazon Redshift security group.
- `authorize_cluster_security_group_ingress`: Authorizes the specified AWS customer account to restore the specified snapshot.
- `authorize_snapshot_access`: Deletes a set of cluster snapshots.
- `authorize_snapshot_access`: Modifies the settings for a list of snapshots.
- `batch_delete_cluster_snapshots`: Cancels a resize operation.
- `batch_delete_cluster_snapshots`: Copies the specified automated cluster snapshot to a new manual cluster snapshot.
- `cancel_resize`: Creates a new cluster.
- `copy_cluster_snapshot`: Creates an Amazon Redshift parameter group.
- `create_cluster`: Creates a new Amazon Redshift security group.
- `create_cluster`: Creates a manual snapshot of the specified cluster.
- `create_cluster`: Creates a new Amazon Redshift subnet group.
- `create_cluster`: Creates an Amazon Redshift event notification subscription.
- `create_cluster`: Creates an HSM client certificate that an Amazon Redshift cluster will use to connect to the client's HSM in order to store and retrieve the keys used to encrypt the cluster databases.
- `create_cluster`: Creates an HSM configuration that contains the information required by an Amazon Redshift cluster to store and use database encryption keys in a Hardware Security Module (HSM).
- `create_cluster`: Create a scheduled action.
- `create_cluster`: Creates a snapshot copy grant that permits Amazon Redshift to use a customer master key (CMK) from AWS Key Management Service (AWS KMS) to encrypt copied snapshots in a destination region.
- `create_cluster`: Create a snapshot schedule.
- `create_cluster`: Adds one or more tags to a specified resource.
- `create_cluster`: Deletes a previously provisioned cluster.
- `create_cluster`: Deletes an Amazon Redshift security group.
- `create_cluster`: Deletes the specified manual snapshot.
- `create_cluster`: Deletes the specified cluster subnet group.
- `create_cluster`: Deletes an Amazon Redshift event notification subscription.
- `create_cluster`: Deletes the specified HSM client certificate.
- `create_cluster`: Deletes the specified Amazon Redshift HSM configuration.
- `create_cluster`: Deletes a scheduled action.
- `create_cluster`: Deletes the specified snapshot copy grant.
- `create_cluster`: Deletes a snapshot schedule.
Delete a tag or tags from a resource
Returns a list of attributes attached to an account
Returns an array of ClusterDbRevision objects
Returns a list of Amazon Redshift parameter groups, including parameter groups you created and the default parameter group
Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group
Returns properties of provisioned clusters including general cluster properties, cluster database properties, maintenance and backup properties, and security and access properties
Returns information about Amazon Redshift security groups
Returns one or more snapshot objects, which contain metadata about your cluster snapshots
Returns one or more cluster subnet group objects, which contain metadata about your cluster subnet groups
Returns a list of all the available maintenance tracks
Returns descriptions of the available Amazon Redshift cluster versions
Returns a list of parameter settings for the specified parameter group family
Displays a list of event categories for all event source types, or for a specified source type
Returns events related to clusters, security groups, snapshots, and parameter groups
Lists descriptions of all the Amazon Redshift event notification subscriptions for a specified account
Returns information about the specified HSM client certificate
Returns information about the specified Amazon Redshift HSM configuration
Describes whether information, such as queries and connection attempts, is being logged for the specified Amazon Redshift cluster
Returns a list of orderable cluster options
Returns a list of the available reserved node offerings by Amazon Redshift with their associated costs
Returns the descriptions of the reserved nodes
Returns information about the last resize operation for the specified cluster
Describes properties of scheduled actions
Returns a list of snapshot copy grants owned by the AWS account in the destination AWS Region
Returns a list of snapshot schedules
Returns the total amount of snapshot usage and provisioned storage in megabytes
Lists the status of one or more table restore requests made using the RestoreTableFromClusterSnapshot API action
Returns a list of tags
Stops logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster
Disables the automatic copying of snapshots from one region to another region for a specified cluster
Starts logging information, such as queries and connection attempts, for the specified Amazon Redshift cluster
Enables the automatic copy of snapshots from one region to another region for a specified cluster
Returns a database user name and temporary password with temporary authorization
Returns an array of DC2 ReservedNodeOfferings that matches the payment type, term, and usage price for the specified reserved node
Modifies the settings for a cluster
Modifies the database revision of a cluster
Modifies the list of AWS Identity and Access Management (IAM) roles that can be used by the cluster
Modifies the maintenance settings of a cluster
Modifies the parameters of a parameter group
Modifies the settings for a snapshot
Modifies a snapshot schedule for a cluster
Modifies a cluster subnet group to include the specified list of VPC subnets
Modifies an existing Amazon Redshift event notification subscription
Modify a scheduled action
Modifies the number of days to retain snapshots in the destination AWS Region and for your account
Modifies a snapshot schedule
Allows you to purchase reserved nodes
**rekognition**

- `reboot_cluster` Reboots a cluster
- `reset_cluster_parameter_group` Sets one or more parameters of the specified parameter group to their default values and sets the source values of the parameters to "engine-default"
- `resize_cluster` Changes the size of the cluster
- `restore_from_cluster_snapshot` Creates a new cluster from a snapshot
- `restore_table_from_cluster_snapshot` Creates a new table from a table in an Amazon Redshift cluster snapshot
- `revoke_cluster_security_group_ingress` Revokes an ingress rule in an Amazon Redshift security group for a previously authorized IP range or Amazon EC2 security group
- `revoke_snapshot_access` Removes the ability of the specified AWS customer account to restore the specified snapshot
- `rotate_encryption_key` Rotates the encryption keys for a cluster

**Examples**

```r
## Not run: svc <- redshift()
svc$accept_reserved_node_exchange(
  Foo = 123
)
## End(Not run)
```

---

**Description**

This is the Amazon Rekognition API reference.

**Usage**

```r
rekognition(config = list())
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>config</code></td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

**Service syntax**

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

**Operations**

- **compare_faces**: Compares a face in the source input image with each of the 100 largest faces detected in the target input image.
- **create_collection**: Creates a collection in an AWS Region.
- **create_project**: Creates a new Amazon Rekognition Custom Labels project.
- **create_project_version**: Creates a new version of a model and begins training.
- **create_stream_processor**: Creates an Amazon Rekognition stream processor that you can use to detect and recognize faces.
- **delete_collection**: Deletes the specified collection.
- **delete_faces**: Deletes faces from a collection.
- **delete_stream_processor**: Deletes the stream processor identified by Name.
- **describe_collection**: Describes the specified collection.
- **describe_projects**: Lists and gets information about your Amazon Rekognition Custom Labels projects.
- **describe_project_versions**: Lists and describes the models in an Amazon Rekognition Custom Labels project.
- **describe_stream_processor**: Provides information about a stream processor created by CreateStreamProcessor.
- **detect_custom_labels**: Detects custom labels in a supplied image by using an Amazon Rekognition Custom Labels model.
- **detect_faces**: Detects faces within an image that is provided as input.
- **detect_labels**: Detects instances of real-world entities within an image (JPEG or PNG) provided as input.
- **detect_moderation_labels**: Detects unsafe content in a specified JPEG or PNG format image.
- **detect_text**: Detects text in the input image and converts it into machine-readable text.
- **get_celebrity_info**: Gets the name and additional information about a celebrity based on his or her Amazon Rekognition ID.
- **get_celebrity_recognition**: Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartCelebrityRecognition.
- **get_content_moderation**: Gets the unsafe content analysis results for a Amazon Rekognition Video analysis started by StartContentModeration.
- **get_face_detection**: Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDetection.
- **get_face_search**: Gets face search results for a Amazon Rekognition Video face search started by StartFaceSearch.
- **get_label_detection**: Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelDetection.
- **get_person_tracking**: Gets the path tracking results of a Amazon Rekognition Video analysis started by StartPersonTracking.
- **index_faces**: Detects faces in the input image and adds them to the specified collection.
- **list_collections**: Returns list of collection IDs in your account.
- **list_faces**: Returns metadata for faces in the specified collection.
- **list_stream_processors**: Gets a list of stream processors that you have created with CreateStreamProcessor.
- **recognize_custom_labels**: Returns an array of celebrities recognized in the input image.
- **search_faces**: For a given input face ID, searches for matching faces in the collection the face belongs to.
- **search_faces_by_image**: For a given input image, first detects the largest face in the image, and then searches the specified collection for matching faces.
- **start_celebrity_recognition**: Starts asynchronous recognition of celebrities in a stored video.
- **start_content_moderation**: Starts asynchronous detection of unsafe content in a stored video.
- **start_face_detection**: Starts asynchronous detection of faces in a stored video.
- **start_face_search**: Starts the asynchronous search for faces in a collection that match the faces of persons detected in a stored video.
- **start_label_detection**: Starts asynchronous detection of labels in a stored video.
- **start_person_tracking**: Starts the asynchronous tracking of a person’s path in a stored video.
- **start_project_version**: Starts processing a stream processor.
- **stop_project_version**: Stops a running model.
- **stop_stream_processor**: Stops a running stream processor that was created by CreateStreamProcessor.
Examples

```r
# This operation compares the largest face detected in the source image
# with each face detected in the target image.
## Not run: svc <- rekognition()
svc$compare_faces(
  SimilarityThreshold = 90L,
  SourceImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mysourceimage"
    )
  ),
  TargetImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mytargetimage"
    )
  )
)
## End(Not run)
```

---

**Description**

AWS Resource Groups lets you organize AWS resources such as Amazon EC2 instances, Amazon Relational Database Service databases, and Amazon S3 buckets into groups using criteria that you define as tags. A resource group is a collection of resources that match the resource types specified in a query, and share one or more tags or portions of tags. You can create a group of resources based on their roles in your cloud infrastructure, lifecycle stages, regions, application layers, or virtually any criteria. Resource groups enable you to automate management tasks, such as those in AWS Systems Manager Automation documents, on tag-related resources in AWS Systems Manager. Groups of tagged resources also let you quickly view a custom console in AWS Systems Manager that shows AWS Config compliance and other monitoring data about member resources.

To create a resource group, build a resource query, and specify tags that identify the criteria that members of the group have in common. Tags are key-value pairs.

For more information about Resource Groups, see the AWS Resource Groups User Guide.

AWS Resource Groups uses a REST-compliant API that you can use to perform the following types of operations.

- Create, Read, Update, and Delete (CRUD) operations on resource groups and resource query entities
- Applying, editing, and removing tags from resource groups
- Resolving resource group member ARNs so they can be returned as search results
- Getting data about resources that are members of a group
- Searching AWS resources based on a resource query

**Usage**

```r
resourcegroups(config = list())
```

**Arguments**

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

**Service syntax**

```r
csvc <- resourcegroups(
  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_group` Creates a group with a specified name, description, and resource query
- `delete_group` Deletes a specified resource group
- `get_group` Returns information about a specified resource group
- `get_group_query` Returns the resource query associated with the specified resource group
- `get_tags` Returns a list of tags that are associated with a resource group, specified by an ARN
- `list_group_resources` Returns a list of ARNs of resources that are members of a specified resource group
- `list_groups` Returns a list of existing resource groups in your account
- `search_resources` Returns a list of AWS resource identifiers that matches a specified query
- `tag` Adds tags to a resource group with the specified ARN
- `untag` Deletes specified tags from a specified resource
- `update_group` Updates an existing group with a new or changed description
- `update_group_query` Updates the resource query of a group
## Description

Resource Groups Tagging API

This guide describes the API operations for the resource groups tagging.

A tag is a label that you assign to an AWS resource. A tag consists of a key and a value, both of which you define. For example, if you have two Amazon EC2 instances, you might assign both a tag key of "Stack." But the value of "Stack" might be "Testing" for one and "Production" for the other.

Tagging can help you organize your resources and enables you to simplify resource management, access management and cost allocation.

You can use the resource groups tagging API operations to complete the following tasks:

- Tag and untag supported resources located in the specified Region for the AWS account.
- Use tag-based filters to search for resources located in the specified Region for the AWS account.
- List all existing tag keys in the specified Region for the AWS account.
- List all existing values for the specified key in the specified Region for the AWS account.

To use resource groups tagging API operations, you must add the following permissions to your IAM policy:

- `tag:GetResources`
- `tag:TagResources`
- `tag:UntagResources`
- `tag:GetTagKeys`
- `tag:GetTagValues`

You'll also need permissions to access the resources of individual services so that you can tag and untag those resources.

For more information on IAM policies, see Managing IAM Policies in the IAM User Guide.

You can use the Resource Groups Tagging API to tag resources for the following AWS services.
• Alexa for Business (a4b)
• API Gateway
• Amazon AppStream
• AWS AppSync
• AWS App Mesh
• Amazon Athena
• Amazon Aurora
• AWS Backup
• AWS Certificate Manager
• AWS Certificate Manager Private CA
• Amazon Cloud Directory
• AWS CloudFormation
• Amazon CloudFront
• AWS CloudHSM
• AWS CloudTrail
• Amazon CloudWatch (alarms only)
• Amazon CloudWatch Events
• Amazon CloudWatch Logs
• AWS CodeBuild
• AWS CodeCommit
• AWS CodePipeline
• AWS CodeStar
• Amazon Cognito Identity
• Amazon Cognito User Pools
• Amazon Comprehend
• AWS Config
• AWS Data Pipeline
• AWS Database Migration Service
• AWS DataSync
• AWS Direct Connect
• AWS Directory Service
• Amazon DynamoDB
• Amazon EBS
• Amazon EC2
• Amazon ECR
• Amazon ECS
• AWS Elastic Beanstalk
• Amazon Elastic File System
• Elastic Load Balancing
• Amazon ElastiCache
• Amazon Elasticsearch Service
• AWS Elemental MediaLive
• AWS Elemental MediaPackage
• AWS Elemental MediaTailor
• Amazon EMR
• Amazon FSx
• Amazon S3 Glacier
• AWS Glue
• Amazon GuardDuty
• Amazon Inspector
• AWS IoT Analytics
• AWS IoT Core
• AWS IoT Device Defender
• AWS IoT Device Management
• AWS IoT Events
• AWS IoT Greengrass
• AWS Key Management Service
• Amazon Kinesis
• Amazon Kinesis Data Analytics
• Amazon Kinesis Data Firehose
• AWS Lambda
• AWS License Manager
• Amazon Machine Learning
• Amazon MQ
• Amazon MSK
• Amazon Neptune
• AWS OpsWorks
• AWS Organizations
• Amazon Quantum Ledger Database (QLDB)
• Amazon RDS
• Amazon Redshift
• AWS Resource Access Manager
• AWS Resource Groups
• AWS RoboMaker
• Amazon Route 53
• Amazon Route 53 Resolver
• Amazon S3 (buckets only)
• Amazon SageMaker
• AWS Secrets Manager
• AWS Security Hub
• AWS Service Catalog
• Amazon Simple Notification Service (SNS)
• Amazon Simple Queue Service (SQS)
• AWS Step Functions
• AWS Storage Gateway
• AWS Systems Manager
• AWS Transfer for SFTP
• Amazon VPC
• Amazon WorkSpaces

Usage

resourcegroupstaggingapi(config = list())

Arguments

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

Service syntax

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

- **describe_report_creation**
  Describes the status of the StartReportCreation operation

- **get_compliance_summary**
  Returns a table that shows counts of resources that are noncompliant with their tag policies

- **get_resources**
  Returns all the tagged or previously tagged resources that are located in the specified Region for the AWS account

- **get_tag_keys**
  Returns all tag keys in the specified Region for the AWS account

- **get_tag_values**
  Returns all tag values for the specified key in the specified Region for the AWS account

- **start_report_creation**
  Generates a report that lists all tagged resources in accounts across your organization and tells whether each resource is compliant with the effective tag policy

- **tag_resources**
  Applies one or more tags to the specified resources

- **untag_resources**
  Removes the specified tags from the specified resources

### Examples

```r
## Not run: svc <- resourcegroupstaggingapi()
svc$describe_report_creation(
  Foo = 123
)
## End(Not run)
```

---

### 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"
    ),
    profile = "string"
  ),
  profile = "string"
)
```
Operations

associate_vpc_with_hosted_zone
change_resource_record_sets
change_tags_for_resource
create_health_check
create_hosted_zone
create_query_logging_config
create_reusable_delegation_set
create_traffic_policy
create_traffic_policy_instance
create_traffic_policy_version
create_vpc_association_authorization
delete_health_check
delete_hosted_zone
delete_query_logging_config
delete_reusable_delegation_set
delete_traffic_policy
delete_traffic_policy_instance
disassociate_vpc_from_hosted_zone
going_to_change
get_account_limit
get_change
get_checker_ip_ranges
get_geo_location
get_health_check
get_health_check_count
get_health_check_last_failure_reason
get_health_check_status
get_hosted_zone
get_hosted_zone_count
get_hosted_zone_limit
get_query_logging_config
get_reusable_delegation_set
get_reusable_delegation_set_limit
get_traffic_policy
get_traffic_policy_instance
get_traffic_policy_instance_count
list_geo_locations
list_health_checks
list_hosted_zones
list_hosted_zones_by_name

Associates an Amazon VPC with a private hosted zone
Creates, changes, or deletes a resource record set, which contains authoritative DNS information
Adds, edits, or deletes tags for a health check or a hosted zone
Creates a new health check
Creates a new public or private hosted zone
Creates a configuration for DNS query logging
Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones
Creates a traffic policy, which you use to create multiple DNS resource record sets
Creates resource record sets in a specified hosted zone based on the settings in a specified traffic policy
Creates a new version of an existing traffic policy
Authorizes the AWS account that created a specified VPC to submit an AssociateVPCWithHostedZone request
Deletes a health check
Deletes a hosted zone
Deletes a configuration for DNS query logging
Deletes a reusable delegation set
Deletes a traffic policy
Deletes a traffic policy instance and all of the resource record sets that Amazon Route 53 created when you created the instance
Disassociates a VPC from a Route 53 private hosted zone
Gets the specified limit for the current account, for example, the maximum number of health checks
Returns the current status of a change batch request
GetCheckerIpRanges still works, but we recommend that you download ip-ranges
Gets information about whether a specified geographic location is supported for Amazon Route 53
Gets information about a specified health check
Retrieves the number of health checks that are associated with the current AWS account
Gets the reason that a specified health check failed most recently
Gets status of a specified health check
Gets information about a specified hosted zone including the four name servers assigned to the zone
Retrieves the number of hosted zones that are associated with the current AWS account
Gets the specified limit for a specified hosted zone, for example, the maximum number of resource record sets
Gets information about a specified configuration for DNS query logging
Retrieves information about a specified reusable delegation set, including the four name servers assigned to the delegation set
Gets the maximum number of hosted zones that you can associate with the specified VPC
Gets information about a specific traffic policy version
Gets information about a specified traffic policy instance
Gets the number of traffic policy instances that are associated with the current AWS account
Retrieves a list of supported geographic locations
Retrieve a list of the health checks that are associated with the current AWS account
Retrieves a list of the public and private hosted zones that are associated with the current AWS account
Retrieves a list of your hosted zones in lexicographic order
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

```r
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 clientTransferProhibited status) to allow domain transfers
- `enable_domain_auto_renew` - This operation configures Amazon Route 53 to automatically renew the specified domain before the domain registration expires
- `enable_domain_transfer_lock` - This operation sets the transfer lock on the domain (specifically the clientTransferProhibited status) to prevent domain transfers
- `get_contact_reachability_status` - For operations that require confirmation that the email address for the registrant contact is reachable, this operation returns information about whether the registrant contact has responded
- `get_domain_detail` - This operation returns detailed information about a specified domain that is associated with the current AWS account
- `get_domain_suggestions` - The GetDomainSuggestions operation returns a list of suggested domain names given a string, which can either be a domain name or simply a word or phrase (without spaces)
- `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 current AWS account
- `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
- `renew_domain` - For operations that require confirmation that the email address for the registrant contact is reachable, this operation resends the confirmation email to the current email address for the registrant contact
- `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 set of name servers
- `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 specified period

Examples

```r
## Not run: svc <- route53domains()
```
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:
   ```bash
   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:
   ```bash
   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:
```bash
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

```r
csvc <- 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 the queries pass through, one domain name that you want to forward to your network, and the IP addresses of the DNS resolvers in your network
- `delete_resolver_endpoint`: Deletes a resolver endpoint
- `delete_resolver_rule`: Deletes a resolver rule
- `disassociateResolverEndpointIpAddresses`: Removes IP addresses from an inbound or an outbound resolver endpoint
- `disassociateResolverRule`: Removes the association between a specified resolver rule and a specified VPC
- `getResolverEndpoint`: Gets information about a specified resolver endpoint, such as whether it's an inbound or an outbound resolver endpoint
- `getResolverRule`: Gets information about a specified resolver rule, such as the domain name that the rule forwards DNS queries for and the ID of the outbound resolver endpoint that the rule is associated with
- `getResolverRuleAssociation`: Gets information about an association between a specified resolver rule and a VPC
- `getResolverRulePolicy`: Gets information about a resolver rule policy
- `listResolverEndpointIpAddresses`: Lists all the resolver endpoints that were created using the current AWS account
- `listResolverEndpoints`: Lists the associations that were created between resolver rules and VPCs using the current AWS account
- `listResolverRuleAssociations`: Lists the resolver rules that were created using the current AWS account
- `listResolverRules`: Lists the tags that you associated with the specified resource
- `putResolverRulePolicy`: Specifies the Resolver operations and resources that you want to allow another AWS account to be able to use
- `tag_resource`: Adds one or more tags to a specified resource
- `untag_resource`: Removes one or more tags from a specified resource
- `updateResolverEndpoint`: Updates the name of an inbound or an outbound resolver endpoint
- `updateResolverRule`: Updates settings for a specified resolver rule

Examples

```r
## Not run: svc <- route53resolver()
svc$associate_resolver_endpoint_ip_address(  
  Foo = 123
)```
## End(Not run)

### Description

Amazon Simple Storage Service

### Usage

```r
s3(config = list())
```

### Arguments

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

### Service syntax

```r
svc <- s3(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string"  
    ),  
    endpoint = "string",  
    region = "string"  
  )
)
```

### Operations

- **abort_multipart_upload**
  - This operation aborts a multipart upload
- **complete_multipart_upload**
  - Completes a multipart upload by assembling previously uploaded parts
- **copy_object**
  - Creates a copy of an object that is already stored in Amazon S3
- **create_bucket**
  - Creates a new bucket
- **create_multipart_upload**
  - This operation initiates a multipart upload and returns an upload ID
- **delete_bucket**
  - Deletes the bucket
- **delete_bucket_analytics_configuration**
  - Deletes an analytics configuration for the bucket (specified by the analytics configuration ID)
- **delete_bucket_cors**
  - Deletes the cors configuration information set for the bucket
- **delete_bucket_encryption**
  - This implementation of the DELETE operation removes default encryption from the
delete_bucket_inventory_configuration
delete_bucket_lifecycle
delete_bucket_metrics_configuration
delete_bucket_policy
delete_bucket_replication
delete_bucket_tagging
delete_bucket_website
delete_object
delete_objects
delete_object_tagging
delete_public_access_block
get_bucket_accelerate_configuration
get_bucket_acl
get_bucket_analytics_configuration
get_bucket_cors
get_bucket_encryption
get_bucket_inventory_configuration
get_bucket_lifecycle
get_bucket_lifecycle_configuration
get_bucket_location
get_bucket_logging
get_bucket_metrics_configuration
get_bucket_notification
get_bucket_notification_configuration
get_bucket_policy
get_bucket_policy_status
get_bucket_replication
get_bucket_request_payment
get_bucket_tagging
get_bucket_versioning
get_bucket_website
get_object
get_object_acl
get_object_legal_hold
get_object_lock_configuration
get_object_retention
get_object_tagging
get_object_torrent
get_public_access_block
head_bucket
head_object
list_bucket_analytics_configurations
list_bucket_inventory_configurations
list_bucket_metrics_configurations
list_buckets
list_multipart_uploads
list_objects
list_objects_v2

Deletes an inventory configuration (identified by the inventory ID) from the bucket
Deletes the lifecycle configuration from the specified bucket
Deletes a metrics configuration for the Amazon CloudWatch request metrics (specified by the metrics configuration ID)
This implementation of the DELETE operation uses the policy subresource to delete the policy of a specified bucket
Deletes the replication configuration from the bucket
Deletes the tags from the bucket
This operation removes the website configuration for a bucket
Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object
This operation enables you to delete multiple objects from a bucket using a single HTTP request
Removes the entire tag set from the specified object
Removes the PublicAccessBlock configuration for an Amazon S3 bucket
This implementation of the GET operation uses the accelerate subresource to return the transfer acceleration state of a bucket
This implementation of the GET operation uses the acl subresource to return the access control list (ACL) of a bucket
This implementation of the GET operation returns an analytics configuration (identified by the analytics configuration ID)
Returns the cors configuration information set for the bucket
Returns the default encryption configuration for an Amazon S3 bucket
Returns an inventory configuration (identified by the inventory configuration ID) from the bucket
For an updated version of this API, see GetBucketLifecycleConfiguration
Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both
Returns the logging status of a bucket and the permissions users have to view and modify that status
Gets a metrics configuration (specified by the metrics configuration ID) from the bucket
No longer used, see GetBucketNotificationConfiguration
Returns the notification configuration of a bucket
Returns the policy of a specified bucket
Retrieves the policy status for an Amazon S3 bucket, indicating whether the bucket is public
Returns the replication configuration of a bucket
Returns the request payment configuration of a bucket
Returns the tag set associated with the bucket
Returns the versioning state of a bucket
Returns the website configuration for a bucket
Retrieves objects from Amazon S3
Returns the access control list (ACL) of an object
Gets an object’s current Legal Hold status
Gets the Object Lock configuration for a bucket
Retrieves an object’s retention settings
Returns the tag-set of an object
Retrieves torrent files from a bucket
Retrieves the PublicAccessBlock configuration for an Amazon S3 bucket
This operation is useful to determine if a bucket exists and you have permission to access it
The HEAD operation retrieves metadata from an object without returning the object itself
Lists the analytics configurations for the bucket
Returns a list of inventory configurations for the bucket
Lists the metrics configurations for the bucket
Returns a list of all buckets owned by the authenticated sender of the request
This operation lists in-progress multipart uploads
Returns some or all (up to 1,000) of the objects in a bucket
Returns some or all (up to 1,000) of the objects in a bucket
list_object_versions
list_parts
put_bucket_accelerate_configuration
put_bucket_acl
put_bucket_analytics_configuration
put_bucket_cors
put_bucket_encryption
put_bucket_inventory_configuration
put_bucket_lifecycle
put_bucket_lifecycle_configuration
put_bucket_logging
put_bucket_metrics_configuration
put_bucket_notification
put_bucket_notification_configuration
put_bucket_policy
put_bucket_replication
put_bucket_request_payment
put_bucket_tagging
put_bucket_versioning
put_bucket_website
put_object
put_object_acl
put_object_legal_hold
put_object_lock_configuration
put_object_retention
put_object_tagging
put_public_access_block
restore_object
select_object_content
upload_part
upload_part_copy

list_object_versions
Returns metadata about all of the versions of objects in a bucket
Lists the parts that have been uploaded for a specific multipart upload
Sets the accelerate configuration of an existing bucket
Sets the permissions on an existing bucket using access control lists (ACL)
Sets an analytics configuration for the bucket (specified by the analytics configuration ID)
Sets the cors configuration for your bucket
This implementation of the PUT operation uses the encryption subresource to set the encryption status
This implementation of the PUT operation adds an inventory configuration (identified by the inventory ID)
For an updated version of this API, see PutBucketLifecycleConfiguration
Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle
Set the logging parameters for a bucket and to specify permissions for who can view
Sets a metrics configuration (specified by the metrics configuration ID) for the bucket
No longer used, see the PutBucketNotificationConfiguration operation
Enables notifications of specified events for a bucket
Applies an Amazon S3 bucket policy to an Amazon S3 bucket
Creates a replication configuration or replaces an existing one
Sets the request payment configuration for a bucket
Sets the tags for a bucket
Sets the versioning state of an existing bucket
Sets the configuration of the website that is specified in the website subresource
Adds an object to a bucket
Uses the acl subresource to set the access control list (ACL) permissions for an object
Applies a Legal Hold configuration to the specified object
Places an Object Lock configuration on the specified bucket
Places an Object Retention configuration on an object
Sets the supplied tag-set to an object that already exists in a bucket A tag is a key-value pair
Creates or modifies the PublicAccessBlock configuration for an Amazon S3 bucket
Restores an archived copy of an object back into Amazon S3 This operation performs
This operation filters the contents of an Amazon S3 object based on a simple structured SQL query
Uploads a part in a multipart upload
Uploads a part by copying data from an existing object as data source

Examples

# The following example aborts a multipart upload.
## Not run: svc <- s3()
svc$abort_multpart_upload(
  Bucket = "examplebucket",
  Key = "bigobject",
  UploadId = "xadcOB_7YPB0JuOFiQ9cz4P3Pe6F1zZw04f7wN93uHsNBEw97p15eNwzExg0LAT2dUN91cOmrEQHDSp..."
)
## End(Not run)
s3control

AWS S3 Control

Description

AWS S3 Control provides access to Amazon S3 control plane operations.

Usage

s3control(config = list())

Arguments

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

Service syntax

svc <- s3control(
  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_access_point: Creates an access point and associates it with the specified bucket
create_job: Creates an Amazon S3 batch operations job
delete_access_point: Deletes the specified access point
delete_access_point_policy: Deletes the access point policy for the specified access point
delete_public_access_block: Removes the PublicAccessBlock configuration for an Amazon Web Services account
describe_job: Retrieves the configuration parameters and status for a batch operations job
describe_access_point: Returns configuration information about the specified access point
describe_access_point_policy: Returns the access point policy associated with the specified access point
describe_access_point_policy_status: Indicates whether the specified access point currently has a policy that allows public access
describe_public_access_block: Retrieves the PublicAccessBlock configuration for an Amazon Web Services account
list_access_points: Returns a list of the access points currently associated with the specified bucket
list_jobs: Lists current jobs and jobs that have ended within the last 30 days for the AWS account making the request
put_access_point_policy: Associates an access policy with the specified access point
sagemaker

put_public_access_block
update_job_priority
update_job_status

Creates or modifies the PublicAccessBlock configuration for an Amazon Web Services account.
Updates an existing job’s priority.
Updates the status for the specified job.

Examples

```r
## Not run: svc <- s3control()
svc$create_access_point(
  Foo = 123
)
## End(Not run)
```

sagemaker

Amazon SageMaker Service

Description

Provides APIs for creating and managing Amazon SageMaker resources.

Usage

```r
sagemaker(config = list())
```

Arguments

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

Service syntax

```r
svc <- sagemaker(
  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
associate_trial_component
create_algorithm
create_app
create_auto_ml_job
create_code_repository
create_compilation_job
create_domain
create_endpoint
create_endpoint_config
create_experiment
create_flow_definition
create_human_task_ui
create_hyper_parameter_tuning_job
create_labeling_job
create_model
create_model_package
create_monitoring_schedule
create_notebook_instance
create_notebook_instance_lifecycle_config
create_presigned_domain_url
create_presigned_notebook_instance_url
create_processing_job
create_training_job
create_transform_job
create_trial
create_trial_component
create_user_profile
create_workteam
delete_algorithm
delete_app
delete_code_repository
delete_domain
delete_endpoint
delete_endpoint_config
delete_experiment
delete_flow_definition
delete_model
delete_model_package
delete_monitoring_schedule
delete_notebook_instance
delete_notebook_instance_lifecycle_config
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workteam
describe_algorithm

Adds or overwrites one or more tags for the specified Amazon SageMaker resource.
Associates a trial component with a trial.
Create a machine learning algorithm that you can use in Amazon SageMaker.
Creates a running App for the specified UserProfile.
Creates an AutoPilot job.
Creates a Git repository as a resource in your Amazon SageMaker account.
Starts a model compilation job.
 Creates a Domain for Amazon SageMaker Amazon SageMaker Studio.
Creates an endpoint using the endpoint configuration specified in the request.
Creates an endpoint configuration that Amazon SageMaker hosting services will use to deploy models.
Creates an Amazon SageMaker experiment.
Creates a flow definition.
Defines the settings you will use for the human review workflow user interface.
Starts a hyperparameter tuning job.
Creates a job that uses workers to label the data objects in your input dataset.
Creates a model in Amazon SageMaker.
Creates a model package that you can use to create Amazon SageMaker models.
Creates a schedule that regularly starts Amazon SageMaker Processing Jobs.
Creates an Amazon SageMaker notebook instance.
Creates a lifecycle configuration that you can associate with a notebook.
Creates a URL that you can use to connect to the Jupyter server from a browser.
Creates a processing job.
Starts a model training job.
Starts a transform job.
Creates an Amazon SageMaker trial.
Creates a trial component, which is a stage of a machine learning trial.
Creates a new user profile.
Creates a new work team for labeling your data.
Removes the specified algorithm from your account.
Use to stop and delete an app.
Deletes the specified Git repository from your account.
Use to delete a domain.
Deletes an endpoint.
Deletes an endpoint configuration.
Deletes an Amazon SageMaker experiment.
Deletes the specified flow definition.
Deletes a model.
Deletes a model package.
Deletes a monitoring schedule.
Deletes an Amazon SageMaker notebook instance.
Deletes a notebook instance lifecycle configuration.
Deletes the specified tags from an Amazon SageMaker resource.
Deletes the specified trial.
Deletes the specified trial component.
Deletes a user profile.
Deletes an existing work team.
Returns a description of the specified algorithm that is in your account.
describe_app
describe_auto_ml_job
describe_code_repository
describe_compilation_job
describe_domain
describe_endpoint
describe_endpoint_config
describe_experiment
describe_flow_definition
describe_human_task_ui
describe_hyper_parameter_tuning_job
describe_labeling_job
describe_model
describe_model_package
describe_monitoring_schedule
describe_notebook_instance
describe_notebook_instance_lifecycle_config
describe_processing_job
describe_subscribed_workteam
describe_training_job
describe_transform_job
describe_trial
describe_trial_component
describe_user_profile
describe_workteam
disassociate_trial_component
get_search_suggestions
list_algorithms
list_apps
list_auto_ml_jobs
list_candidates_for_auto_ml_job
list_code_repositories
list_compilation_jobs
list_domains
list_endpoint_configs
list_endpoints
list_experiments
list_flow_definitions
list_human_task_uis
list_hyper_parameter_tuning_jobs
list_labeling_jobs
list_labeling_jobs_for_workteam
list_model_packages
list_models
list_monitoring_executions
list_monitoring_schedules
list_notebook_instance_lifecycle_configs
list_notebook_instances

Describes the app
Returns information about an Amazon SageMaker job
Gets details about the specified Git repository
Returns information about a model compilation job
The description of the domain
Returns the description of an endpoint
Returns the description of an endpoint configuration created using the CreateEndpointConfig API
Provides a list of an experiment’s properties
Returns information about the specified flow definition
Returns information about the requested human task user interface
Gets a description of a hyperparameter tuning job
Gets information about a labeling job
Describes a model that you created using the CreateModel API
Returns a description of the specified model package, which is used to create Amazon SageMaker models
Describes the schedule for a monitoring job
Returns information about a notebook instance
Returns a description of a notebook instance lifecycle configuration
Returns a description of a processing job
Gets information about a work team provided by a vendor
Returns information about a training job
Returns information about a transform job
Provides a list of a trial’s properties
Provides a list of a trials component’s properties
Describes the user profile
Gets information about a specific work team
Disassociates a trial component from a trial
An auto-complete API for the search functionality in the Amazon SageMaker console
Lists the machine learning algorithms that have been created
Lists apps
Request a list of jobs
List the Candidates created for the job
 Gets a list of the Git repositories in your account
Lists model compilation jobs that satisfy various filters
Lists the domains
Lists endpoint configurations
Lists endpoints
Lists all the experiments in your account
Returns information about the flow definitions in your account
Returns information about the human task user interfaces in your account
Gets a list of HyperParameterTuningJobSummary objects that describe hyperparameter tuning jobs
 Gets a list of labeling jobs
Gets a list of labeling jobs assigned to a specified work team
Lists the model packages that have been created
Lists models created with the CreateModel API
Returns list of all monitoring job executions
Returns list of all monitoring schedules
Lists notebook instance lifestyle configurations created with the CreateNotebookInstanceLifecycleConfig API
Returns a list of the Amazon SageMaker notebook instances in the required AWS Region.
list_processing_jobs
list_subscribed_workteams
list_tags
list_training_jobs
list_training_jobs_for_hyper_parameter_tuning_job
list_transform_jobs
list_trial_components
list_trials
list_user_profiles
list_workteams
render_ui_template
search
start_monitoring_schedule
start_notebook_instance
stop_auto_ml_job
stop_compilation_job
stop_hyper_parameter_tuning_job
stop_labeling_job
stop_monitoring_schedule
stop_notebook_instance
stop_processing_job
stop_training_job
stop_transform_job
update_code_repository
update_domain
update_endpoint
update_endpoint_weights_and_capacities
update_experiment
update_monitoring_schedule
update_notebook_instance
update_notebook_instance_lifecycle_config
update_trial
update_trial_component
update_user_profile
update_workteam

Lists processing jobs that satisfy various filters
Gets a list of the work teams that you are subscribed to in the AWS Marketplace
Returns the tags for the specified Amazon SageMaker resource
Lists training jobs
Gets a list of TrainingJobSummary objects that describe the training jobs
Lists transform jobs
Lists the trial components in your account
Lists the trials in your account
Lists user profiles
Gets a list of work teams that you have defined in a region
Renders the UI template so that you can preview the worker’s experience
Finds Amazon SageMaker resources that match a search query
Starts a previously stopped monitoring schedule
Launches an ML compute instance with the latest version of the libraries
A method for forcing the termination of a running job
Stops a model compilation job
Stops a running hyperparameter tuning job and all running training jobs
Stops a running labeling job
Stops a previously started monitoring schedule
Terminates the ML compute instance
Stops a processing job
Stops a training job
Stops a transform job
Updates the specified Git repository with the specified values
Updates a domain
Deploys the new EndpointConfig specified in the request, switches to using the newly created endpoint, and then deletes resources provisioned for the endpoint using the old EndpointConfig
Updates variant weight of one or more variants associated with an existing endpoint
Updates or removes the description of an experiment
Updates a previously created schedule
Updates a notebook instance
Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
Updates the display name of a trial
Updates one or more properties of a trial component
Updates a user profile
Updates an existing work team with new member definitions or descriptions

Examples

```r
## Not run: svc <- sagemaker()
svc$add_tags(
  Foo = 123
)
## End(Not run)
```
Description

The Amazon SageMaker runtime API.

Usage

sagemakerruntime(config = list())

Arguments

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

Service syntax

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

Operations

- `invoke_endpoint` After you deploy a model into production using Amazon SageMaker hosting services, your client applications use this API to get inferences from the model hosted at the specified endpoint.

Examples

```
## Not run: svc <- sagemakerruntime()
svc$invoke_endpoint(
  Foo = 123
)
## End(Not run)
```
Description

AWS Secrets Manager API Reference

AWS Secrets Manager is a web service that enables you to store, manage, and retrieve, secrets. This guide provides descriptions of the Secrets Manager API. For more information about using this service, see the AWS Secrets Manager User Guide.

API Version

This version of the Secrets Manager API Reference documents the Secrets Manager API version 2017-10-17. As an alternative to using the API directly, you can use one of the AWS SDKs, which consist of libraries and sample code for various programming languages and platforms (such as Java, Ruby, .NET, iOS, and Android). The SDKs provide a convenient way to create programmatic access to AWS Secrets Manager. For example, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

We recommend that you use the AWS SDKs to make programmatic API calls to Secrets Manager. However, you also can use the Secrets Manager HTTP Query API to make direct calls to the Secrets Manager web service. To learn more about the Secrets Manager HTTP Query API, see Making Query Requests in the AWS Secrets Manager User Guide.

Secrets Manager supports GET and POST requests for all actions. That is, the API doesn’t require you to use GET for some actions and POST for others. However, GET requests are subject to the limitation size of a URL. Therefore, for operations that require larger sizes, use a POST request.

Support and Feedback for AWS Secrets Manager

We welcome your feedback. Send your comments to awssecretsmanager-feedback@amazon.com, or post your feedback and questions in the AWS Secrets Manager Discussion Forum. For more information about the AWS Discussion Forums, see Forums Help.

How examples are presented

The JSON that AWS Secrets Manager expects as your request parameters and that the service returns as a response to HTTP query requests are single, long strings without line breaks or white space formatting. The JSON shown in the examples is formatted with both line breaks and white space to improve readability. When example input parameters would also result in long strings that extend beyond the screen, we insert line breaks to enhance readability. You should always submit the input as a single JSON text string.

Logging API Requests

AWS Secrets Manager supports AWS CloudTrail, a service that records AWS API calls for your AWS account and delivers log files to an Amazon S3 bucket. By using information that’s collected by AWS CloudTrail, you can determine which requests were successfully made to Secrets Manager, who made the request, when it was made, and so on. For more about AWS Secrets Manager and its support for AWS CloudTrail, see Logging AWS Secrets Manager Events with AWS CloudTrail in the AWS Secrets Manager User Guide. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.
Usage

secretsmanager(config = list())

Arguments

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

Service syntax

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

Operations

cancel_rotate_secret Disables automatic scheduled rotation and cancels the rotation of a secret if one is currently in
create_secret Creates a new secret
delete_resource_policy Deletes the resource-based permission policy that’s attached to the secret
delete_secret Deletes an entire secret and all of its versions
describe_secret Retrieves the details of a secret
get_random_password Generates a random password of the specified complexity
get_resource_policy Retrieves the JSON text of the resource-based policy document that’s attached to the specified secret
get_secret_value Retrieves the contents of the encrypted fields SecretString or SecretBinary from the specified version
list_secrets Lists all of the secrets that are stored by Secrets Manager in the AWS account
list_secret_version_ids Lists all of the versions attached to the specified secret
put_resource_policy Attaches the contents of the specified resource-based permission policy to a secret
put_secret_value Stores a new encrypted secret value in the specified secret
restore_secret Cancels the scheduled deletion of a secret by removing the DeletedDate time stamp
rotate_secret Configures and starts the asynchronous process of rotating this secret
tag_resource Attaches one or more tags, each consisting of a key name and a value, to the specified secret
untag_resource Removes one or more tags from the specified secret
update_secret Modifies many of the details of the specified secret
update_secret_version_stage Modifies the staging labels attached to a version of a secret
Examples

# The following example shows how to cancel rotation for a secret. The
# operation sets the RotationEnabled field to false and cancels all
# scheduled rotations. To resume scheduled rotations, you must re-enable
# rotation by calling the rotate-secret operation.
## Not run: svc <- secretsmanager()
svc$cancel_rotate_secret(
    SecretId = "MyTestDatabaseSecret"
)
## End(Not run)

Description

Security Hub provides you with a comprehensive view of the security state of your AWS environment and resources. It also provides you with the compliance status of your environment based on CIS AWS Foundations compliance checks. Security Hub collects security data from AWS accounts, services, and integrated third-party products and helps you analyze security trends in your environment to identify the highest priority security issues. For more information about Security Hub, see the AWS Security Hub User Guide.

When you use operations in the Security Hub API, the requests are executed only in the AWS Region that is currently active or in the specific AWS Region that you specify in your request. Any configuration or settings change that results from the operation is applied only to that Region. To make the same change in other Regions, execute the same command for each Region to apply the change to. For example, if your Region is set to us-west-2, when you use CreateMembers to add a member account to Security Hub, the association of the member account with the master account is created only in the us-west-2 Region. Security Hub must be enabled for the member account in the same Region that the invite was sent from.

The following throttling limits apply to using Security Hub API operations:

- GetFindings - RateLimit of 3 requests per second, and a BurstLimit of 6 requests per second.
- UpdateFindings - RateLimit of 1 request per second, and a BurstLimit of 5 requests per second.
- All other operations - RateLimit of 10 request per second, and a BurstLimit of 30 requests per second.

Usage

securityhub(config = list())

Arguments

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

```r
svc <- securityhub(
  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_invitation`: Accepts the invitation to be a member account and be monitored by the Security Hub master account
- `batch_disable_standards`: Disables the standards specified by the provided StandardsSubscriptionArns
- `batch_enable_standards`: Enables the standards specified by the provided standardsArn
- `batch_import_findings`: Imports security findings generated from an integrated third-party product into Security Hub
- `create_action_target`: Creates a custom action target in Security Hub
- `create_insight`: Creates a custom insight in Security Hub
- `create_members`: Creates a member association in Security Hub between the specified accounts and the master account
- `decline_invitations`: Declines invitations to become a member account
- `delete_action_target`: Deletes a custom action target from Security Hub
- `delete_insight`: Deletes the insight specified by the InsightArn
- `delete_invitations`: Deletes invitations received by the AWS account to become a member account
- `delete_members`: Deletes the specified member accounts from Security Hub
- `describe_action_targets`: Returns a list of the custom action targets in Security Hub in your account
- `describe_hub`: Returns details about the Hub resource in your account, including the HubArn and the time when you enabled Security Hub
- `describe_products`: Returns information about the products available that you can subscribe to and integrate with Security Hub to consolidate findings
- `disable_import_findings_for_product`: Disables the integration of the specified product with Security Hub
- `disable_security_hub`: Disables Security Hub in your account only in the current Region
- `disassociate_from_master_account`: Disassociates the current Security Hub member account from the associated master account
- `disassociate_members`: Disassociates the specified member accounts from the associated master account
- `enable_import_findings_for_product`: Enables the integration of a partner product with Security Hub
- `enable_security_hub`: Enables Security Hub for your account in the current Region or the Region you specify
- `get_enabled_standards`: Returns a list of the standards that are currently enabled
- `get_findings`: Returns a list of findings that match the specified criteria
- `get_insight_results`: Lists the results of the Security Hub insight that the insight ARN specifies
- `get_insights`: Lists and describes insights that insight ARNs specify
- `get_invitations_count`: Returns the count of all Security Hub membership invitations that were sent to the current member account
- `get_master_account`: Provides the details for the Security Hub master account to the current member account
- `get_members`: Returns the details on the Security Hub member accounts that the account IDs specify
- `invite_members`: Invites other AWS accounts to become member accounts for the Security Hub master account
- `list_enabled_products_for_import`: Lists all findings-generating solutions (products) whose findings you have subscribed to
list_invitations  Lists all Security Hub membership invitations that were sent to the current AWS account
list_members   Lists details about all member accounts for the current Security Hub master account
list_tags_for_resource Returns a list of tags associated with a resource
tag_resource   Adds one or more tags to a resource
untag_resource Removes one or more tags from a resource
update_action_target Updates the name and description of a custom action target in Security Hub
update_findings Updates the Note and RecordState of the Security Hub-aggregated findings that the filter attributes specify
update_insight Updates the Security Hub insight that the insight ARN specifies

Examples

```r
## Not run: svc <- securityhub()
svc$accept_invitation(
  Foo = 123
)
## End(Not run)
```

---

**serverlessapplicationrepository**

**AWSServerlessApplicationRepository**

**Description**

The AWS Serverless Application Repository makes it easy for developers and enterprises to quickly find and deploy serverless applications in the AWS Cloud. For more information about serverless applications, see Serverless Computing and Applications on the AWS website.

The AWS Serverless Application Repository is deeply integrated with the AWS Lambda console, so that developers of all levels can get started with serverless computing without needing to learn anything new. You can use category keywords to browse for applications such as web and mobile backends, data processing applications, or chatbots. You can also search for applications by name, publisher, or event source. To use an application, you simply choose it, configure any required fields, and deploy it with a few clicks.

You can also easily publish applications, sharing them publicly with the community at large, or privately within your team or across your organization. To publish a serverless application (or app), you can use the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS SDKs to upload the code. Along with the code, you upload a simple manifest file, also known as the AWS Serverless Application Model (AWS SAM) template. For more information about AWS SAM, see AWS Serverless Application Model (AWS SAM) on the AWS Labs GitHub repository.

The AWS Serverless Application Repository Developer Guide contains more information about the two developer experiences available:
**Consuming Applications** – Browse for applications and view information about them, including source code and readme files. Also install, configure, and deploy applications of your choosing.

**Publishing Applications** – Configure and upload applications to make them available to other developers, and publish new versions of applications.

### Usage

```python
serverlessapplicationrepository(config = list())
```

### Arguments

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

### Service syntax

```python
svc <- serverlessapplicationrepository(
    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_application**
  - Creates an application, optionally including an AWS SAM file to create the first application version.

- **create_application_version**
  - Creates an application version.

- **create_cloud_formation_change_set**
  - Creates an AWS CloudFormation change set for the given application.

- **create_cloud_formation_template**
  - Creates an AWS CloudFormation template.

- **delete_application**
  - Deletes the specified application.

- **get_application**
  - Gets the specified application.

- **get_application_policy**
  - Retrieves the policy for the application.

- **get_cloud_formation_template**
  - Gets the specified AWS CloudFormation template.

- **list_application_dependencies**
  - Retrieves the list of applications nested in the containing application.

- **list_applications**
  - Lists applications owned by the requester.

- **list_application_versions**
  - Lists versions for the specified application.

- **put_application_policy**
  - Sets the permission policy for an application.

- **update_application**
  - Updates the specified application.
## Not run: svc <- serverlessapplicationrepository()
svc$create_application(
  Foo = 123
)
## End(Not run)

### Description

AWS Service Catalog enables organizations to create and manage catalogs of IT services that are approved for use on AWS. To get the most out of this documentation, you should be familiar with the terminology discussed in AWS Service Catalog Concepts.

### Usage

servicecatalog(config = list())

### Arguments

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

### Service syntax

svc <- servicecatalog(
  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_portfolio_share`: Accepts an offer to share the specified portfolio
- `associate_budget_with_resource`: Associates the specified budget with the specified resource
- `associate_principal_with_portfolio`: Associates the specified principal ARN with the specified portfolio
associate_product_with_portfolio
associate_service_action_with_provisioning_artifact
associate_tag_option_with_resource
batch_associate_service_action_with_provisioning_artifact
batch_disassociate_service_action_from_provisioning_artifact
copy_product
create_constraint
create_portfolio
create_portfolio_share
create_product
create_provisioned_product_plan
create_provisioning_artifact
create_service_action
create_tag_option
delete_constraint
delete_portfolio
delete_portfolio_share
delete_product
delete_provisioned_product_plan
delete_provisioning_artifact
delete_service_action
delete_tag_option
describe_constraint
describe_copy_product_status
describe_portfolio
describe_portfolio_share_status
describe_product
describe_product_as_admin
describe_product_view
describe_provisioned_product
describe_provisioned_product_plan
describe_provisioning_artifact
describe_provisioning_parameters
describe_record
describe_service_action
describe_service_action_execution_parameters
describe_tag_option
disable_aws_organizations_access
disassociate_budget_from_resource
disassociate_principal_from_portfolio
disassociate_product_from_portfolio
disassociate_service_action_from_provisioning_artifact
disassociate_tag_option_from_resource
enable_aws_organizations_access
execute_provisioned_product_plan
execute_provisioned_product_service_action
get_aws_organizations_access_status
list_accepted_portfolio_shares

Associates the specified product with the specified portfolio
Associates a self-service action with a provisioning artifact
Associate the specified TagOption with the specified portfolio
Associates multiple self-service actions with provisioning artifacts
Disassociates a batch of self-service actions from the specified provisioning artifact
Copies the specified source product to the specified target product
Creates a constraint
Creates a portfolio
Shares the specified portfolio with the specified account or organization
Creates a product
Creates a plan
Creates a provisioning artifact (also known as a version) for the specified product
Creates a self-service action
Creates a TagOption
Deletes the specified constraint
Deletes the specified portfolio
Deletes the specified plan
Deletes the specified product
Disassociates the specified budget from the specified resource
Disassociates the specified product from the specified portfolio
Disassociates the specified self-service action association from the specified provisioning artifact
Disassociates the specified TagOption from the specified resource
Enables portfolio sharing through AWS Organizations feature
Executes a self-service action against a provisioned product
Get the Access Status for AWS Organization portfolio share
Lists all portfolios for which sharing was accepted by this account.
### list_budgets_for_resource
Lists all the budgets associated to the specified resource

### list_constraints_for_portfolio
Lists the constraints for the specified portfolio and product

### list_launch_paths
Lists the paths to the specified product

### list_organization_portfolio_access
Lists the organization nodes that have access to the specified portfolio

### list_portfolio_access
Lists the account IDs that have access to the specified portfolio

### list_portfolios
Lists all portfolios in the catalog

### list_portfolios_for_product
Lists all portfolios that the specified product is associated with

### list_portfolio_access
Lists all principal ARNs associated with the specified portfolio

### list_portfolios
Lists all portfolios in the catalog

### list_provisioned_product_plans
Lists all provisioning artifacts and plans associated with the specified portfolio

### list_provisioned_products
Lists all provisioned products available (not terminated)

### list_provisioning_artifacts
Lists all provisioning artifacts (also known as versions) for the specified product

### list_provisioning_artifacts_for_service_action
Lists all provisioning artifacts (also known as versions) for the specified self-service action

### list_record_history
Lists the specified requests or all performed requests

### list_resources_for_tag_option
Lists the resources associated with the specified TagOption

### list_service_actions
Lists all self-service actions

### list_service_actions_for_provisioning_artifact
Returns a paginated list of self-service actions associated with the specified Provisioning Artifact ID

### list_stack_instances_for_provisioned_product
Returns summary information about stack instances that are associated with the specified provisioned product

### list_tag_options
Lists the specified TagOptions or all TagOptions

### provision_product
Provisions the specified product

### reject_portfolio_share
Rejects an offer to share the specified portfolio

### scan_provisioned_products
Lists the provisioned products that are available (not terminated)

### search_products
Gets information about the products to which the caller has access

### search_products_as_admin
Gets information about the products for the specified portfolio or all products

### search_provisioned_products
Gets information about the provisioned products that meet the specified criteria

### terminate_provisioned_product
Terminates the specified provisioned product

### update_constraint
Updates the specified constraint

### update_portfolio
Updates the specified portfolio

### update_product
Updates the specified product

### update_provisioned_product
Requests updates to the configuration of the specified provisioned product

### update_provisioned_product_properties
Requests updates to the properties of the specified provisioned product

### update_provisioning_artifact
Updates the specified provisioning artifact (also known as a version)

### update_service_action
Updates a self-service action

### update_tag_option
Updates the specified TagOption

---

**Examples**

```r
## Not run: svc <- servicecatalog()
svc$accept_portfolio_share(
  Foo = 123
)
## End(Not run)
```

---

**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 Creates an HTTP namespace
create_private_dns_namespace Creates a private namespace based on DNS, which will be visible only inside a specified Amazon VPC
create_public_dns_namespace Creates a public namespace based on DNS, which will be visible on the internet
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 Cloud Map 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 a CreateService request
get_service Gets the settings for a specified service


Service Quotas

Description

Service Quotas is a web service that you can use to manage many of your AWS service quotas. Quotas, also referred to as limits, are the maximum values for a resource, item, or operation. This guide provide descriptions of the Service Quotas actions that you can call from an API. For the Service Quotas user guide, which explains how to use Service Quotas from the console, see What is Service Quotas.

AWS provides SDKs that consist of libraries and sample code for programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc...). The SDKs provide a convenient way to create programmatic access to Service Quotas and AWS. For information about the AWS SDKs, including how to download and install them, see the Tools for Amazon Web Services page.

Usage

servicequotas(config = list())

Arguments

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

Examples

```r
## Not run: svc <- servicediscovery()
svc$create_http_namespace(
  Foo = 123
)
## End(Not run)
```
Service syntax

```r
svc <- servicequotas(
  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_service_quota_template`: Associates the Service Quotas template with your organization so that when new accounts are created in your organization, the template submits increase requests for the specified service quotas.
- `delete_service_quota_increase_request_from_template`: Removes a service quota increase request from the Service Quotas template.
- `disassociate_service_quota_template`: Disables the Service Quotas template.
- `get_association_for_service_quota_template`: Retrieves the ServiceQuotaTemplateAssociationStatus value from the service.
- `get_aws_default_service_quota`: Retrieves the default service quotas values.
- `get_requested_service_quota_change`: Retrieves the details for a particular increase request.
- `get_service_quota`: Returns the details for the specified service quota.
- `get_service_quota_increase_request_from_template`: Returns the details of the service quota increase request in your template.
- `get_service_quotas`: Lists all service quotas for the specified AWS service.
- `get_services`: Lists the AWS services available in Service Quotas.
- `list_aws_default_service_quotas`: Lists all default service quotas for the specified AWS service or all AWS services.
- `list_requested_service_quota_change_history`: Requests a list of the changes to quotas for a service.
- `list_requested_service_quota_change_history_by_quota`: Requests a list of the changes to specific service quotas.
- `list_service_quota_increase_requests_in_template`: Returns a list of the quota increase requests in the template.
- `list_service_quotas`: Lists all service quotas for the specified AWS service.
- `list_services`: Lists the AWS services available in Service Quotas.
- `put_service_quota_increase_request_into_template`: Defines and adds a quota to the service quota template.
- `request_service_quota_increase`: Retrieves the details of a service quota increase request.

Examples

```r
## Not run: svc <- servicequotas()
svc$associate_service_quota_template(  
  Foo = 123  
)
## End(Not run)
```
Amazon Simple Email Service

Description

This document contains reference information for the Amazon Simple Email Service (Amazon SES) API, version 2010-12-01. This document is best used in conjunction with the Amazon SES Developer Guide.

For a list of Amazon SES endpoints to use in service requests, see Regions and Amazon SES in the Amazon SES Developer Guide.

Usage

`ses(config = list())`

Arguments

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

Service syntax

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

Operations

- `clone_receipt_rule_set`: Creates a receipt rule set by cloning an existing one
- `create_configuration_set`: Creates a configuration set
- `create_configuration_set_event_destination`: Creates a configuration set event destination
- `create_configuration_set_tracking_options`: Creates an association between a configuration set and a custom domain
- `create_custom_verification_email_template`: Creates a new custom verification email template
- `create_receipt_filter`: Creates a new IP address filter
- `create_receipt_rule`: Creates a receipt rule
- `create_receipt_rule_set`: Creates an empty receipt rule set
- `create_template`: Creates an email template
delete_configuration_set Deletes a configuration set
delete_configuration_set_event_destination Deletes a configuration set event destination
delete_configuration_set_tracking_options Deletes an association between a configuration set and a custom domain for open and click event tracking
delete_identity Deletes an existing custom verification email template
delete_identity Deletes the specified identity (an email address or a domain) from the verified list
delete_identity Deletes the specified sending authorization policy for the given identity
delete_identity_policy Deletes the specified IP address filter
delete_receipt_rule Deletes the specified receipt rule set and all of the receipt rules it contains
delete_receipt_rule Deletes an email template
Deprecation
Describe the metadata and receipt rules for the receipt rule set that is currently active
Describe the details of the specified configuration set
Describe the details of the specified receipt rule set
Describe the email sending status of the Amazon SES account for the current AWS Region
Returns the custom email verification template for the template name you specify
Returns the current status of Easy DKIM signing for an entity
Returns the custom MAIL FROM attributes for a list of identities (email addresses and/or domains), returns the current status of Easy DKIM signing for an entity
Returns the requested sending authorization policies for the given identity
Returns the verification status and (for domain identities) the verification token for each identity
Provides the sending limits for the Amazon SES account
Provides sending statistics for the current AWS Region
Displays the template object (which includes the Subject line, HTML part and text part) for the template you specify
Returns a list containing all of the identities (email addresses and domains) for your AWS account
Lists the IP address filters associated with your AWS account in the current AWS Region
Lists the receipt rule sets that exist under your AWS account in the current AWS Region
Lists the email templates present in your Amazon SES account in the current AWS Region
Deprecates
Add or updates the delivery options for a configuration set
Add or updates a sending authorization policy for the specified identity
Reorders the receipt rules within a receipt rule set
Generates and sends a bounce message to the sender of an email you receive
Composes an email message to multiple destinations
Adds an email address to the list of identities for your Amazon SES account
Composes an email message and immediately queues it for sending
Composes an email message and immediately queues it for sending
Sets the specified receipt rule set as the active receipt rule set
Sets an Amazon Simple Notification Service (Amazon SNS) topic to receive notifications.
set_receipt_rule_position
set_test_render_template
update_account_sending_enabled
update_configuration_set_event_destination
update_configuration_set_reputation_metrics_enabled
update_configuration_set_sending_enabled
update_configuration_set_tracking_options
update_custom_verification_email_template
update_receipt_rule
update_template
verify_domain_dkim
verify_domain_identity
verify_email_address
verify_email_identity

Examples

# The following example creates a receipt rule set by cloning an existing
# one:
## Not run:
svc <- ses()
svc$clone_receipt_rule_set(
  OriginalRuleSetName = "RuleSetToClone",
  RuleSetName = "RuleSetToCreate"
)
## End(Not run)

sfn

AWS Step Functions

Description

AWS Step Functions is a service that lets you coordinate the components of distributed applications and microservices using visual workflows.

You can use Step Functions to build applications from individual components, each of which performs a discrete function, or task, allowing you to scale and change applications quickly. Step Functions provides a console that helps visualize the components of your application as a series of steps. Step Functions automatically triggers and tracks each step, and retries steps when there are errors, so your application executes predictably and in the right order every time. Step Functions logs the state of each step, so you can quickly diagnose and debug any issues.

Step Functions manages operations and underlying infrastructure to ensure your application is available at any scale. You can run tasks on AWS, your own servers, or any system that has access to AWS. You can access and use Step Functions using the console, the AWS SDKs, or an HTTP API. For more information about Step Functions, see the AWS Step Functions Developer Guide.
Usage

sfn(config = list())

Arguments

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

Service syntax

svc <- sfn(
  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_activity          Creates an activity
  create_state_machine     Creates a state machine
  delete_activity          Deletes an activity
  delete_state_machine     Deletes a state machine
  describe_activity       Describes an activity
  describe_execution       Describes an execution
  describe_state_machine   Describes a state machine
  describe_state_machine_for_execution
                           Describes the state machine associated with a specific execution
  get_activity_task        Used by workers to retrieve a task (with the specified activity ARN) which has been
                           scheduled for execution by a running state machine
  get_execution_history    Returns the history of the specified execution as a list of events
  list_activities          Lists the existing activities
  list_executions          Lists the executions of a state machine that meet the filtering criteria
  list_state_machines      Lists the existing state machines
  list_tags_for_resource   List tags for a given resource
  send_task_failure        Used by activity workers and task states using the callback pattern to report that the task
                           identified by the taskToken failed
  send_task_heartbeat      Used by activity workers and task states using the callback pattern to report to Step Functions
                           that the task represented by the specified taskToken is still making progress
  send_task_success        Used by activity workers and task states using the callback pattern to report that the task
                           identified by the taskToken completed successfully
  start_execution          Starts a state machine execution
  stop_execution           Stops an execution
  tag_resource             Add a tag to a Step Functions resource
  untag_resource           Remove a tag from a Step Functions resource
  update_state_machine     Updates an existing state machine by modifying its definition and/or roleArn
Examples

```r
## Not run: svc <- sfn()
svc$create_activity(
  Foo = 123
)
## End(Not run)
```

Description

AWS Shield Advanced

This is the AWS Shield Advanced API Reference. This guide is for developers who need detailed information about the AWS Shield Advanced API actions, data types, and errors. For detailed information about AWS WAF and AWS Shield Advanced features and an overview of how to use the AWS WAF and AWS Shield Advanced APIs, see the AWS WAF and AWS Shield Developer Guide.

Usage

```r
shield(config = list())
```

Arguments

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

Service syntax

```r
svc <- shield(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```
associate_drt_log_bucket
associate_drt_role
create_protection
create_subscription
delete_protection
delete_subscription
describe_attack
describe_drt_access
describe_emergency_contact_settings
describe_protection
describe_subscription
disassociate_drt_log_bucket
disassociate_drt_role
get_subscription_state
list_attacks
list_protections
update_emergency_contact_settings
update_subscription

Examples

```r
## Not run: svc <- shield()
svc$associate_drt_log_bucket(
  Foo = 123
)
## End(Not run)
```

Amazon SimpleDB

Amazon SimpleDB is a web service providing the core database functions of data indexing and querying in the cloud. By offloading the time and effort associated with building and operating a web-scale database, SimpleDB provides developers the freedom to focus on application development. A traditional, clustered relational database requires a sizable upfront capital outlay, is complex to design, and often requires extensive and repetitive database administration. Amazon SimpleDB is dramatically simpler, requiring no schema, automatically indexing your data and providing a simple API for storage and access. This approach eliminates the administrative burden of data modeling, index maintenance, and performance tuning. Developers gain access to this functionality within Amazon’s proven computing environment, are able to scale instantly, and pay only for what they use. Visit [http://aws.amazon.com/simpeldb/](http://aws.amazon.com/simpeldb/) for more information.
Usage

simpledb(config = list())

Arguments

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

Service syntax

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

Operations

- **batch_delete_attributes**: Performs multiple DeleteAttributes operations in a single call, which reduces round trips and latencies.
- **batch_put_attributes**: The BatchPutAttributes operation creates or replaces attributes within one or more items.
- **create_domain**: The CreateDomain operation creates a new domain.
- **delete_attributes**: Deletes one or more attributes associated with an item.
- **delete_domain**: The DeleteDomain operation deletes a domain.
- **domain_metadata**: Returns information about the domain, including when the domain was created, the number of items, and the size of the attribute names and values.
- **get_attributes**: Returns all of the attributes associated with the specified item.
- **list_domains**: The ListDomains operation lists all domains associated with the Access Key ID.
- **put_attributes**: The PutAttributes operation creates or replaces attributes in an item.
- **select**: The Select operation returns a set of attributes for ItemNames that match the select expression.

Examples

```
## Not run: svc <- simpledb()
svc$batch_delete_attributes(
  Foo = 123
)
## End(Not run)
```
Amazon Simple Notification Service

Description

Amazon Simple Notification Service (Amazon SNS) is a web service that enables you to build distributed web-enabled applications. Applications can use Amazon SNS to easily push real-time notification messages to interested subscribers over multiple delivery protocols. For more information about this product see https://aws.amazon.com/sns. For detailed information about Amazon SNS features and their associated API calls, see the Amazon SNS Developer Guide.

We also provide SDKs that enable you to access Amazon SNS from your preferred programming language. The SDKs contain functionality that automatically takes care of tasks such as: cryptographically signing your service requests, retrying requests, and handling error responses. For a list of available SDKs, go to Tools for Amazon Web Services.

Usage

`sns(config = list())`

Arguments

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

Service syntax

```r
svc <- sns(
  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_permission` Adds a statement to a topic’s access control policy, granting access for the specified AWS accounts to the specified actions
- `check_if_phone_number_is_opted_out` Accepts a phone number and indicates whether the phone holder has opted out of receiving SMS messages from your account
- `confirm_subscription` Verifies an endpoint owner’s intent to receive messages by validating the token sent
- `create_platform_application` Creates a platform application object for one of the supported push notification services
- `create_platform_endpoint` Creates an endpoint for a device and mobile app on one of the supported push notification services
create_topic Creates a topic to which notifications can be published
delete_endpoint Deletes the endpoint for a device and mobile app from Amazon SNS
delete_platform_application Deletes a platform application object for one of the supported push notification services
delete_topic Deletes a topic and all its subscriptions
get_endpoint_attributes Retrieves the endpoint attributes for a device on one of the supported push notification services
get_platform_application_attributes Retrieves the attributes of the platform application object for the supported push notification services
get_sms_attributes Returns the settings for sending SMS messages from your account
get_subscription_attributes Returns all of the properties of a subscription
get_topic_attributes Returns all of the properties of a topic
list_endpoints_by_platform_application Lists the endpoints and endpoint attributes for devices in a supported push notification service
list_phone_numbers_opted_out Returns a list of phone numbers that are opted out, meaning you cannot send SMS messages to them
list_platform_applications Lists the platform application objects for the supported push notification services, such as APNS and FCM
list_platform_subscriptions Returns a list of the requester’s subscriptions
list_subscriptions Returns a list of the subscriptions to a specific topic
list_tags_for_resource List all tags added to the specified Amazon SNS topic
list_topics Returns a list of the requester’s topics
opt_in_phone_number Use this request to opt in a phone number that is opted out, which enables you to resume sending SMS messages to the number
publish Sends a message to an Amazon SNS topic or sends a text message (SMS message)
remove_permission Removes a statement from a topic’s access control policy
set_endpoint_attributes Sets the attributes for an endpoint for a device on one of the supported push notification services
set_platform_application_attributes Sets the attributes of the platform application object for the supported push notification services
set_sms_attributes Use this request to set the default settings for sending SMS messages and receiving daily SMS usage reports
set_subscription_attributes Allows a subscription owner to set an attribute of the subscription to a new value
set_topic_attributes Allows a topic owner to set an attribute of the topic to a new value
subscribe Prepares to subscribe an endpoint by sending the endpoint a confirmation message
subscribe_resource Add tags to the specified Amazon SNS topic
unsubscribe Deletes a subscription
untag_resource Remove tags from the specified Amazon SNS topic

Examples

```r
## Not run: svc <- sns()
svc$add_permission(
  Foo = 123
)
## End(Not run)
```

### Description

Welcome to the Amazon Simple Queue Service API Reference.
Amazon Simple Queue Service (Amazon SQS) is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components.

You can use AWS SDKs to access Amazon SQS using your favorite programming language. The SDKs perform tasks such as the following automatically:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

**Additional Information**

- [Amazon SQS Product Page](#)
- [Amazon Simple Queue Service Developer Guide](#)
  - Making API Requests
  - Amazon SQS Message Attributes
  - Amazon SQS Dead-Letter Queues
- [Amazon SQS in the AWS CLI Command Reference](#)
- [Amazon Web Services General Reference](#)
  - Regions and Endpoints

**Usage**

```sqs(config = list())```

**Arguments**

```config```

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

**Service syntax**

```sqs(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_permission  Adds a permission to a queue for a specific principal
change_message_visibility  Changes the visibility timeout of a specified message in a queue to a new value
change_message_visibility_batch  Changes the visibility timeout of multiple messages
create_queue  Creates a new standard or FIFO queue
delete_message  Deletes the specified message from the specified queue
delete_message_batch  Deletes up to ten messages from the specified queue
delete_queue  Deletes the queue specified by the QueueUrl, regardless of the queue’s contents
get_queue_attributes  Gets attributes for the specified queue
get_queue_url  Returns the URL of an existing Amazon SQS queue
list_dead_letter_source_queues  Returns a list of your queues that have the RedrivePolicy queue attribute configured with a dead-letter queue
list_queues  Returns a list of your queues
list_queue_tags  List all cost allocation tags added to the specified Amazon SQS queue
purge_queue  Deletes the messages in a queue specified by the QueueURL parameter
receive_message  Retrives one or more messages (up to 10), from the specified queue
remove_permission  Revokes any permissions in the queue policy that matches the specified Label parameter
send_message  Delivers a message to the specified queue
send_message_batch  Delivers up to ten messages to the specified queue
set_queue_attributes  Sets the value of one or more queue attributes
tag_queue  Add cost allocation tags to the specified Amazon SQS queue
untag_queue  Remove cost allocation tags from the specified Amazon SQS queue

Examples

```r
## Not run: svc <- sqs()
svc$add_permission(
    Foo = 123
)
## End(Not run)
```

Amazon Simple Systems Manager (SSM)

Description

AWS Systems Manager

AWS Systems Manager is a collection of capabilities that helps you automate management tasks such as collecting system inventory, applying operating system (OS) patches, automating the creation of Amazon Machine Images (AMIs), and configuring operating systems (OSs) and applications at scale. Systems Manager lets you remotely and securely manage the configuration of your managed instances. A managed instance is any Amazon EC2 instance or on-premises machine in your hybrid environment that has been configured for Systems Manager.

This reference is intended to be used with the AWS Systems Manager User Guide.

To get started, verify prerequisites and configure managed instances. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.
For information about other API actions you can perform on Amazon EC2 instances, see the Amazon EC2 API Reference. For information about how to use a Query API, see Making API Requests.

Usage

```r
ssm(config = list())
```

Arguments

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

Service syntax

```r
csvc <- ssm(
  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_to_resource**
  
  Adds or overwrites one or more tags for the specified resource.

- **cancel_command**
  
  Attempts to cancel the command specified by the Command ID.

- **cancel_maintenance_window_execution**
  
  Stops a maintenance window execution that is already in progress.

- **create_activation**
  
  Registers your on-premises server or virtual machine with Amazon EC2 so that you can manage these resources using Run Command.

- **create_association**
  
  Associates the specified Systems Manager document with the specified instances or targets.

- **create_association_batch**
  
  Associates the specified Systems Manager document with the specified instances or targets.

- **create_document**
  
  Creates a Systems Manager document.

- **create_maintenance_window**
  
  Creates a new maintenance window.

- **create_ops_item**
  
  Creates a new OpsItem.

- **create_patch_baseline**
  
  Creates a patch baseline.

- **create_resource_data_sync**
  
  A resource data sync helps you view data from multiple sources in a single location.

- **delete_activation**
  
  Deletes an activation.

- **delete_association**
  
  Disassociates the specified Systems Manager document from the specified instances or targets.

- **delete_document**
  
  Deletes the Systems Manager document and all instance associations to the document.

- **delete_inventory**
  
  Delete a custom inventory type, or the data associated with a custom inventory type.

- **delete_maintenance_window**
  
  Deletes a maintenance window.

- **delete_parameter**
  
  Delete a parameter from the system.

- **delete_parameters**
  
  Delete a list of parameters.

- **delete_patch_baseline**
  
  Deletes a patch baseline.
delete_resource_data_sync

delete_resource_data_sync

deregister_managed_instance

deregister_patch_baseline_for_patch_group

deregister_target_from_maintenance_window

deregister_task_from_maintenance_window

describe_activations

describe_association

describe_association_executions

describe_association_execution_targets

describe_automation_executions

describe_automation_step_executions

describe_available_patches

describe_document

describe_document_permission

describe_effective_instance_associations

describe_effective_patches_for_patch_baseline

describe_instance_associations_status

describe_instance_information

describe_instance_patches

describe_instance_patch_states

describe_instance_patch_states_for_patch_group

describe_inventory_deletions

describe_maintenance_window_executions

describe_maintenance_window_execution_task_invocations

describe_maintenance_window_execution_tasks

describe_maintenance_windows

describe_maintenance_window_schedule

describe_maintenance_windows_for_target

describe_maintenance_window_targets

describe_maintenance_window_tasks

describe_ops_items

describe_parameters

describe_patch_baselines

describe_patch_groups

describe_patch_group_state

describe_patch_properties

describe_sessions

get_automation_execution

get_calendar_state

get_command_invocation

get_connection_status

get_default_patch_baseline

get_deployable_patch_snapshot_for_instance

get_document

get_inventory

get_inventory_schema

get_maintenance_window

get_maintenance_window_execution

Deletes a Resource Data Sync configuration

Removes the server or virtual machine from the list of registered servers

Removes a patch group from a patch baseline

Removes a target from a maintenance window

Removes a task from a maintenance window

Describes details about the activation, such as the date and time it was created, the IAM role assigned to the instances in the activation, and the number of instances registered by using this activation

Describes the association for the specified target or instance

Use this API action to view all executions for a specific association ID

Use this API action to view information about a specific execution of a maintenance window

Provides details about all active and terminated Automation executions

Information about all active and terminated step executions in a maintenance window execution

Lists all patches eligible to be included in a patch baseline

Describes the specified Systems Manager document

Describes the permissions for a Systems Manager document

All associations for the instance(s)

Retrieves the current effective patches (the patch and the approval state)

The status of the associations for the instance(s)

Describes one or more of your instances

Retrieves information about the patches on the specified instance

Retrieves the high-level patch state of one or more instances

Retrieves the high-level patch state for the instances in the specified patch group

Describes a specific delete inventory operation

Lists the executions of a maintenance window

Retrieves the individual task executions (one per target) for a particular task run as part of a maintenance window execution

For a given maintenance window execution, lists the tasks that were run

Retrieves the maintenance windows in an AWS account

Retrieves information about upcoming executions of a maintenance window

Retrieves information about the maintenance window targets or tasks that an instance is associated with

Lists the targets registered with the maintenance window

Lists the tasks in a maintenance window

Query a set of OpsItems

Get information about a parameter

Lists the patch baselines in your AWS account

Lists all patch groups that have been registered with patch baselines

Retrieves high-level aggregated patch compliance state for a patch baseline

Lists the properties of available patches organized by product, product family, classification, severity, and other properties

Get detailed information about a particular Automation execution

Gets the state of the AWS Systems Manager Change Calendar

Returns detailed information about command execution for an instance

Retrieves the Session Manager connection status for an instance

Retrieves the default patch baseline

Retrieves the current snapshot for the patch baseline the instance uses

Gets the contents of the specified Systems Manager document

Query inventory information

Return a list of inventory type names for the account, or return inventory information from a query

Retrieves a maintenance window

Retrieves details about a specific a maintenance window execution
get_maintenance_window_execution_task
get_maintenance_window_execution_task_invocation
get_maintenance_window_task
get_ops_item
get_ops_summary
get_parameter
get_parameter_history
get_parameters
get_parameters_by_path
get_patch_baseline
get_patch_baseline_for_patch_group
get_service_setting
label_parameter_version
list_associations
list_association_versions
list_command_invocations
list_commands
list_compliance_items
list_compliance_summaries
list_documents
list_document_versions
list_inventory_entries
list_resource_compliance_summaries
list_resource_data_sync
list_tags_for_resource
modify_document_permission
put_compliance_items
put_inventory
put_parameter
register_default_patch_baseline
register_patch_baseline_for_patch_group
register_target_with_maintenance_window
register_task_with_maintenance_window
remove_tags_from_resource
reset_service_setting
resume_session
send_automation_signal
send_command
start_associations_once
start_automation_execution
start_session
stop_automation_execution
terminate_session
update_association
update_association_status
update_document
update_document_default_version
update_maintenance_window

Retrieves the details about a specific task run as part of a maintenance window
Retrieves information about a specific task running on a specific target
Lists the tasks in a maintenance window
Get information about an OpsItem by using the ID
View a summary of OpsItems based on specified filters and aggregation
Get information about a parameter by using the parameter name
Query a list of all parameters used by the AWS account
Get details of a parameter
Retrieve information about one or more parameters in a specific hierarchy
Retrieves information about a patch baseline
Retrieves the patch baseline that should be used for the specified patch group
ServiceSetting is an account-level setting for an AWS service
A parameter label is a user-defined alias to help you manage different versions of a parameter
Lists the associations for the specified Systems Manager document
Retrieves all versions of an association for a specific association ID
An invocation is copy of a command sent to a specific instance
Lists the commands requested by users of the AWS account
For a specified resource ID, this API action returns a list of compliance items
Returns a summary count of compliant and non-compliant resources
Describes one or more of your Systems Manager documents
List all versions for a document
A list of inventory items returned by the request
Returns a resource-level summary count
Lists your resource data sync configurations
Returns a list of the tags assigned to the specified resource
Shares a Systems Manager document publicly or privately
Registers a compliance type and other compliance details on a designated resource
Bulk update custom inventory items on one more instance
Add a parameter to the system
Defines the default patch baseline for the relevant operating system
Registers a patch baseline for a patch group
Registers a target with a maintenance window
Adds a new task to a maintenance window
Removes tag keys from the specified resource
ServiceSetting is an account-level setting for an AWS service
Reconnects a session to an instance after it has been disconnected
Sends a signal to an Automation execution to change the current behavior or status
Runs commands on one or more managed instances
Use this API action to run an association immediately and only once
Initiates execution of an Automation document
Initiates a connection to a target (for example, an instance) for a running Automation that is currently running
Permanently ends a session and closes the data connection between the Session Manager client and SSM Agent on the instance
Updates an association
Updates the status of the Systems Manager document associated with a resource
Updates one or more values for an SSM document
Set the default version of a document
Updates an existing maintenance window
storagegateway

update_maintenance_window_target
update_maintenance_window_task
update_managed_instance_role
update_ops_item
update_patch_baseline
update_resource_data_sync
update_service_setting

Modifies the target of an existing maintenance window
Modifies a task assigned to a maintenance window
Assigns or changes an Amazon Identity and Access Management (IAM) role for the managed instance
Edit or change an OpsItem
Modifies an existing patch baseline
Update a resource data sync
ServiceSetting is an account-level setting for an AWS service

Examples

```r
## Not run: svc <- ssm()
svc$add_tags_to_resource(
    Foo = 123
)
## End(Not run)
```

---

storagegateway  AWS Storage Gateway

---

Description

AWS Storage Gateway Service

AWS Storage Gateway is the service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization’s on-premises IT environment and the AWS storage infrastructure. The service enables you to securely upload data to the AWS cloud for cost effective backup and rapid disaster recovery.

Use the following links to get started using the AWS Storage Gateway Service API Reference:

- **AWS Storage Gateway Required Request Headers**: Describes the required headers that you must send with every POST request to AWS Storage Gateway.
- **Signing Requests**: AWS Storage Gateway requires that you authenticate every request you send; this topic describes how sign such a request.
- **Error Responses**: Provides reference information about AWS Storage Gateway errors.
- **Operations in AWS Storage Gateway**: Contains detailed descriptions of all AWS Storage Gateway operations, their request parameters, response elements, possible errors, and examples of requests and responses.
- **AWS Storage Gateway Regions and Endpoints**: Provides a list of each AWS Region and the endpoints available for use with AWS Storage Gateway.

AWS Storage Gateway resource IDs are in uppercase. When you use these resource IDs with the Amazon EC2 API, EC2 expects resource IDs in lowercase. You must change your resource ID to lowercase to use it with the EC2 API. For example, in Storage Gateway the ID for a volume might
be `vol-aa22bb012345daf670`. When you use this ID with the EC2 API, you must change it to `vol-aa22bb012345daf670`. Otherwise, the EC2 API might not behave as expected.

IDs for Storage Gateway volumes and Amazon EBS snapshots created from gateway volumes are changing to a longer format. Starting in December 2016, all new volumes and snapshots will be created with a 17-character string. Starting in April 2016, you will be able to use these longer IDs so you can test your systems with the new format. For more information, see Longer EC2 and EBS Resource IDs.

For example, a volume Amazon Resource Name (ARN) with the longer volume ID format looks like the following:

```
```

A snapshot ID with the longer ID format looks like the following: `snap-78e22663445566ee`.

For more information, see Announcement: Heads-up – Longer AWS Storage Gateway volume and snapshot IDs coming in 2016.

**Usage**

```
storagegateway(config = list())
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

**Service syntax**

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

**Operations**

- **activate_gateway**
  - Activates the gateway you previously deployed on your host
- **add_cache**
  - Configures one or more gateway local disks as cache for a gateway
- **add_tags_to_resource**
  - Adds one or more tags to the specified resource
- **add_upload_buffer**
  - Configures one or more gateway local disks as upload buffer for a specified gateway
- **add_working_storage**
  - Configures one or more gateway local disks as working storage for a gateway
- **assign_tape_pool**
  - Assigns a tape to a tape pool for archiving
- **attach_volume**
  - Connects a volume to an iSCSI connection and then attaches the volume to the gateway
cancel_archival
cancel_retrieval
create_cachedi_scsi_volume
create_nfs_file_share
create_smb_file_share
create_snapshot
create_snapshot_from_volume_recovery_point
create_storedi_scsi_volume
create_tapes
create_tape_with_barcode
delete_bandwidth_rate_limit
delete_chap_credentials
delete_file_share
delete_gateway
delete_snapshot_schedule
delete_tape
delete_tape_archive
delete_volume
describe_availability_monitor_test
describe_bandwidth_rate_limit
describe_cache
describe_cachedi_scsi_volumes
describe_chap_credentials
describe_gateway_information
describe_maintenance_start_time
describe_nfs_file_shares
describe_smb_file_shares
describe_smb_settings
describe_snapshot_schedule
describe_storedi_scsi_volumes
describe_tape_archives
describe_tape_recovery_points
describe_tapes
describe_upload_buffer
describe_vtl_devices
describe_working_storage
detach_volume
disable_gateway
join_domain
list_file_shares
list_gateways
list_local_disks
list_tags_for_resource
list_tapes
list_volume_initiators
list_volume_recovery_points
list_volumes
notify_when_uploaded

Cancels archiving of a virtual tape to the virtual tape shelf (VTS) after the archiving process has started.
Cancels retrieval of a virtual tape from the virtual tape shelf (VTS) to a gateway after the retrieval process has started.
Creates a cached volume on a specified cached volume gateway.
Creates a Network File System (NFS) file share on an existing file gateway.
Creates a Server Message Block (SMB) file share on an existing file gateway.
Initiates a snapshot of a volume.
Initiates a snapshot of a gateway from a volume recovery point.
Creates a volume on a specified gateway.
Creates one or more virtual tapes.
Creates a virtual tape by using your own barcode.
Deletes the bandwidth rate limits of a gateway.
Deletes Challenge-Handshake Authentication Protocol (CHAP) credentials for a specified iSCSI target and initiator pair.
Deletes a file share from a file gateway.
Deletes a gateway.
Deletes a snapshot of a volume.
Deletes the specified virtual tape.
Deletes the specified virtual tape from the virtual tape shelf (VTS).
Deletes the specified storage volume that you previously created using the CreateCachediSCSIV olume or CreateStor ediSCSIV olume API.
Returns information about the most recent High Availability monitoring test that was performed on the host in a cluster.
Returns the bandwidth rate limits of a gateway.
Returns information about the cache of a gateway.
Returns a description of the gateway volumes specified in the request.
Returns an array of Challenge-Handshake Authentication Protocol (CHAP) credentials information for a specified iSCSI target, one for each target-initiator pair.
Returns metadata about a gateway such as its name, network interfaces, configured time zone, and the state (whether the gateway is running or not).
Returns your gateway’s weekly maintenance start time including the day and time of the week.
Gets a description for one or more Network File System (NFS) file shares from a file gateway.
Gets a description for one or more Server Message Block (SMB) file shares from a file gateway.
Gets a description of a Server Message Block (SMB) file share settings from a file gateway.
Describes the snapshot schedule for the specified gateway volume.
Returns the description of the gateway volumes specified in the request.
Returns a description of specified virtual tapes in the virtual tape shelf (VTS).
Returns a list of virtual tape recovery points that are available for the specified virtual tape.
Returns a description of the specified Amazon Resource Name (ARN) of virtual tape.
Returns information about the upload buffer of a gateway.
Returns a description of virtual tape library (VTL) devices for the specified tape gateway.
Returns information about the working storage of a gateway.
Disconnects a volume from an iSCSI connection and then detaches the volume from the specified gateway.
Disables a tape gateway when the gateway is no longer functioning.
Adds a file gateway to an Active Directory domain.
Gets a list of the file shares for a specific file gateway, or the list of file shares that belong to the calling user account.
Lists gateways owned by an AWS account in an AWS Region specified in the request.
Returns a list of the gateway’s local disks.
Lists the tags that have been added to the specified resource.
Lists virtual tapes in your virtual tape library (VTL) and your virtual tape shelf.
Lists iSCSI initiators that are connected to a volume.
Lists the recovery points for a specified gateway.
Lists the iSCSI stored volumes of a gateway.
Sends you notification through CloudWatch Events when all files written to your file share have been uploaded to Amazon S3.
refresh_cache
remove_tags_from_resource
reset_cache
retrieve_tape_archive
retrieve_tape_recovery_point
set_local_console_password
set_smb_guest_password
shutdown_gateway
start_availability_monitor_test
start_gateway
update_bandwidth_rate_limit
update_chap_credentials
update_gateway_information
update_gateway_software_now
update_maintenance_start_time
update_nfs_file_share
update_smb_file_share
update_smb_security_strategy
update_snapshot_schedule
update_vtl_device_type

Refreshes the cache for the specified file share
Removes one or more tags from the specified resource
Resets all cache disks that have encountered an error and makes the disks available for reconfiguration
Retrieves an archived virtual tape from the virtual tape shelf (VTS) to a tape gateway
Retrieves the recovery point for the specified virtual tape
Sets the password for your VM local console
Sets the password for the guest user smbguest
Shuts down a gateway
Start a test that verifies that the specified gateway is configured for High Availability
Starts a gateway that you previously shut down (see ShutdownGateway)
Updates the bandwidth rate limits of a gateway
Updates the Challenge-Handshake Authentication Protocol (CHAP) credentials for a specified iSCSI target
Updates a gateway’s metadata, which includes the gateway’s name and time zone
Updates the gateway virtual machine (VM) software
Updates a gateway’s weekly maintenance start time information, including day and time
Updates a Network File System (NFS) file share
Updates a Server Message Block (SMB) file share
Updates the SMB security strategy on a file gateway
Updates a snapshot schedule configured for a gateway volume
Updates the type of medium changer in a tape gateway

Examples

# Activates the gateway you previously deployed on your host.
## Not run: svc <- storagegateway()
svc$activate_gateway(
  ActivationKey = "29AV1-3OFV9-VVIUB-NKT0I-LRO6V",
  GatewayName = "My_Gateway",
  GatewayRegion = "us-east-1",
  GatewayTimezone = "GMT-12:00",
  GatewayType = "STORED",
  MediumChangerType = "AWS-Gateway-VTL",
  TapeDriveType = "IBM-ULT3580-TD5"
)
## End(Not run)

---

**sts**

**AWS Security Token Service**

**Description**

The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users). This guide provides descriptions of the STS API. For more detailed information about using this service, go to Temporary Security Credentials.
For information about setting up signatures and authorization through the API, go to Signing AWS API Requests in the AWS General Reference. For general information about the Query API, go to Making Query Requests in Using IAM. For information about using security tokens with other AWS products, go to AWS Services That Work with IAM in the IAM User Guide.

If you're new to AWS and need additional technical information about a specific AWS product, you can find the product's technical documentation at http://aws.amazon.com/documentation/.

Endpoints

By default, AWS Security Token Service (STS) is available as a global service, and all AWS STS requests go to a single endpoint at https://sts.amazonaws.com. Global requests map to the US East (N. Virginia) region. AWS recommends using Regional AWS STS endpoints instead of the global endpoint to reduce latency, build in redundancy, and increase session token validity. For more information, see Managing AWS STS in an AWS Region in the IAM User Guide.

Most AWS Regions are enabled for operations in all AWS services by default. Those Regions are automatically activated for use with AWS STS. Some Regions, such as Asia Pacific (Hong Kong), must be manually enabled. To learn more about enabling and disabling AWS Regions, see Managing AWS Regions in the AWS General Reference. When you enable these AWS Regions, they are automatically activated for use with AWS STS. You cannot activate the STS endpoint for a Region that is disabled. Tokens that are valid in all AWS Regions are longer than tokens that are valid in Regions that are enabled by default. Changing this setting might affect existing systems where you temporarily store tokens. For more information, see Managing Global Endpoint Session Tokens in the IAM User Guide.

After you activate a Region for use with AWS STS, you can direct AWS STS API calls to that Region. AWS STS recommends that you provide both the Region and endpoint when you make calls to a Regional endpoint. You can provide the Region alone for manually enabled Regions, such as Asia Pacific (Hong Kong). In this case, the calls are directed to the STS Regional endpoint. However, if you provide the Region alone for Regions enabled by default, the calls are directed to the global endpoint of https://sts.amazonaws.com.

To view the list of AWS STS endpoints and whether they are active by default, see Writing Code to Use AWS STS Regions in the IAM User Guide.

Recording API requests

STS supports AWS CloudTrail, which is a service that records AWS calls for your AWS account and delivers log files to an Amazon S3 bucket. By using information collected by CloudTrail, you can determine what requests were successfully made to STS, who made the request, when it was made, and so on.

If you activate AWS STS endpoints in Regions other than the default global endpoint, then you must also turn on CloudTrail logging in those Regions. This is necessary to record any AWS STS API calls that are made in those Regions. For more information, see Turning On CloudTrail in Additional Regions in the AWS CloudTrail User Guide.

AWS Security Token Service (STS) is a global service with a single endpoint at https://sts.amazonaws.com. Calls to this endpoint are logged as calls to a global service. However, because this endpoint is physically located in the US East (N. Virginia) Region, your logs list us-east-1 as the event Region. CloudTrail does not write these logs to the US East (Ohio) Region unless you choose to include global service logs in that Region. CloudTrail writes calls to all Regional endpoints to their respective Regions. For example, calls to sts.us-east-2.amazonaws.com are published to the US East (Ohio) Region and calls to sts.eu-central-1.amazonaws.com are published to the EU (Frankfurt) Region.
To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

Usage

```r
sts(config = list())
```

Arguments

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

Service syntax

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

Operations

- `assume_role`: Returns a set of temporary security credentials that you can use to access AWS resources that you might not normally have access to.
- `assume_role_with_saml`: Returns a set of temporary security credentials for users who have been authenticated via a SAML authentication response.
- `assume_role_with_web_identity`: Returns a set of temporary security credentials for users who have been authenticated in a mobile or web application with a web identity provider.
- `decode_authorization_message`: Decodes additional information about the authorization status of a request from an encoded message returned in response to an AWS request.
- `get_access_key_info`: Returns the account identifier for the specified access key ID.
- `get_caller_identity`: Returns details about the IAM user or role whose credentials are used to call the operation.
- `get_federation_token`: Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user.
- `get_session_token`: Returns a set of temporary credentials for an AWS account or IAM user.

Examples

```r
# Not run: svc <- sts()
svc$assume_role(
  ExternalId = "123ABC",
  Policy = "{"Version":"2012-10-17","Statement": [{"Sid":"Stmt1","Effect":"..."},
  RoleArn = "arn:aws:iam::123456789012:role/demo",
```

```r
```
RoleSessionName = "testAssumeRoleSession", 
Tags = list( 
   list( 
      Key = "Project", 
      Value = "Unicorn" 
   ),
   list( 
      Key = "Team", 
      Value = "Automation" 
   ), 
   list( 
      Key = "Cost-Center", 
      Value = "12345" 
   ) 
 ),
TransitiveTagKeys = list( 
   "Project", 
   "Cost-Center" 
)
## End(Not run)

---

**Description**

The AWS Support API reference is intended for programmers who need detailed information about the AWS Support operations and data types. This service enables you to manage your AWS Support cases programmatically. It uses HTTP methods that return results in JSON format.

The AWS Support service also exposes a set of Trusted Advisor features. You can retrieve a list of checks and their descriptions, get check results, specify checks to refresh, and get the refresh status of checks.

The following list describes the AWS Support case management operations:

- **Service names, issue categories, and available severity levels.** The DescribeServices and DescribeSeverityLevels operations return AWS service names, service codes, service categories, and problem severity levels. You use these values when you call the CreateCase operation.

- **Case creation, case details, and case resolution.** The CreateCase, DescribeCases, DescribeAttachment, and ResolveCase operations create AWS Support cases, retrieve information about cases, and resolve cases.

- **Case communication.** The DescribeCommunications, AddCommunicationToCase, and AddAttachmentsToSet operations retrieve and add communications and attachments to AWS Support cases.

The following list describes the operations available from the AWS Support service for Trusted Advisor:
- DescribeTrustedAdvisorChecks returns the list of checks that run against your AWS resources.
- Using the checkId for a specific check returned by DescribeTrustedAdvisorChecks, you can call DescribeTrustedAdvisorCheckResult to obtain the results for the check you specified.
- DescribeTrustedAdvisorCheckSummaries returns summarized results for one or more Trusted Advisor checks.
- RefreshTrustedAdvisorCheck requests that Trusted Advisor rerun a specified check.
- DescribeTrustedAdvisorCheckRefreshStatuses reports the refresh status of one or more checks.

For authentication of requests, AWS Support uses Signature Version 4 Signing Process. See About the AWS Support API in the AWS Support User Guide for information about how to use this service to create and manage your support cases, and how to call Trusted Advisor for results of checks on your resources.

Usage

```r
support(config = list())
```

Arguments

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

Service syntax

```r
svc <- support(
  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_attachments_to_set Adds one or more attachments to an attachment set
- add_communication_to_case Adds additional customer communication to an AWS Support case
- create_case Creates a new case in the AWS Support Center
- describe_attachment Returns the attachment that has the specified ID
- describe_cases Returns a list of cases that you specify by passing one or more case IDs
- describe_communications Returns communications (and attachments) for one or more support cases
- describe_services Returns the current list of AWS services and a list of service categories that can be applied to cases
- describe_severity_levels Returns the list of severity levels that you can assign to an AWS Support case
### Examples

```r
## Not run: svc <- support()
svc$add_attachments_to_set(
  Foo = 123
)
## End(Not run)
```

## Amazon Simple Workflow Service

### Description

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that use Amazon's cloud to coordinate work across distributed components. In Amazon SWF, a task represents a logical unit of work that is performed by a component of your workflow. Coordinating tasks in a workflow involves managing intertask dependencies, scheduling, and concurrency in accordance with the logical flow of the application.

Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

This documentation serves as reference only. For a broader overview of the Amazon SWF programming model, see the Amazon SWF Developer Guide.

### Usage

```r
swf(config = list())
```

### Arguments

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

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

Operations

- `count_closed_workflow_executions` Returns the number of closed workflow executions within the given domain that meet the specified filtering criteria.
- `count_open_workflow_executions` Returns the number of open workflow executions within the given domain that meet the specified filtering criteria.
- `count_pending_activity_tasks` Returns the estimated number of activity tasks in the specified task list.
- `count_pending_decision_tasks` Returns the estimated number of decision tasks in the specified task list.
- `deprecate_activity_type` Deprecates the specified activity type.
- `deprecate_domain` Deprecates the specified domain.
- `deprecate_workflow_type` Deprecates the specified workflow type.
- `describe_activity_type` Returns information about the specified activity type.
- `describe_domain` Returns information about the specified domain, including description and status.
- `describe_workflow_execution` Returns information about the specified workflow execution including its type and some statistics.
- `describe_workflow_type` Returns information about the specified workflow type.
- `get_workflow_execution_history` Returns the history of the specified workflow execution.
- `list_activity_types` Returns information about all activities registered in the specified domain that match the specified name and registration status.
- `list_closed_workflow_executions` Returns a list of closed workflow executions in the specified domain that meet the filtering criteria.
- `list_domains` Returns the list of domains registered in the account.
- `list_open_workflow_executions` Returns a list of open workflow executions in the specified domain that meet the filtering criteria.
- `list_tags_for_resource` List tags for a given domain.
- `list_workflow_types` Returns information about workflow types in the specified domain.
- `poll_for_activity_task` Used by workers to get an ActivityTask from the specified activity taskList.
- `poll_for_decision_task` Used by deciders to get a DecisionTask from the specified decision taskList.
- `record_activity_task_heartbeat` Used by activity workers to report to the service that the ActivityTask represented by the taskToken is still making progress.
- `register_activity_type` Registers a new activity type along with its configuration settings in the specified domain.
- `register_domain` Registers a new domain.
- `register_workflow_type` Registers a new workflow type and its configuration settings in the specified domain.
- `request_cancel_workflow_execution` Records a WorkflowExecutionCancelRequested event in the currently running workflow execution.
- `respond_activity_task_canceled` Used by workers to tell the service that the ActivityTask identified by the taskToken was canceled.
- `respond_activity_task_completed` Used by workers to tell the service that the ActivityTask identified by the taskToken completed successfully with a result (if provided).
- `respond_activity_task_failed` Used by workers to tell the service that the ActivityTask identified by the taskToken has failed with reason (if specified).
- `respond_decision_task_completed` Used by deciders to tell the service that the DecisionTask identified by the taskToken has completed.
- `signal_workflow_execution` Records a WorkflowExecutionSignaled event in the workflow execution history and creates a decision task for the workflow execution identified by the given domain, workflowId and runId.
**textract**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_workflow_execution</td>
<td>Starts an execution of the workflow type in the specified domain using the provided workflowId and input data.</td>
</tr>
<tr>
<td>tag_resource</td>
<td>Add a tag to a Amazon SWF domain.</td>
</tr>
<tr>
<td>terminate_workflow_execution</td>
<td>Records a WorkflowExecutionTerminated event and forces closure of the workflow execution.</td>
</tr>
<tr>
<td>undeprecate_activity_type</td>
<td>Undeprecates a previously deprecated activity type.</td>
</tr>
<tr>
<td>undeprecate_domain</td>
<td>Undeprecates a previously deprecated domain.</td>
</tr>
<tr>
<td>undeprecate_workflow_type</td>
<td>Undeprecates a previously deprecated workflow type.</td>
</tr>
<tr>
<td>untag_resource</td>
<td>Remove a tag from a Amazon SWF domain.</td>
</tr>
</tbody>
</table>

**Examples**

```r
## Not run: svc <- swf()
svc$count_closed_workflow_executions(
  Foo = 123
)
## End(Not run)
```

---

### Description

Amazon Textract detects and analyzes text in documents and converts it into machine-readable text. This is the API reference documentation for Amazon Textract.

### Usage

`textract(config = list())`

#### Arguments

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

#### Service syntax

```r
csvc <- textract(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
  )
)```
transcribeservice

region = "string"
)
)

Operations

analyze_document  Analyzes an input document for relationships between detected items
detect_document_text  Detects text in the input document
get_document_analysis  Gets the results for an Amazon Textract asynchronous operation that analyzes text in a document
get_document_text_detection  Gets the results for an Amazon Textract asynchronous operation that detects text in a document
start_document_analysis  Starts the asynchronous analysis of an input document for relationships between detected items
start_document_text_detection  Starts the asynchronous detection of text in a document

Examples

## Not run: svc <- textract()
svc$analyze_document(
  Foo = 123
)
## End(Not run)

transcribeservice  Amazon Transcribe Service

Description

Operations and objects for transcribing speech to text.

Usage

transcribeservice(config = list())

Arguments

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

Service syntax

svc <- transcribeservice(  
  config = list(    
    credentials = list(      
      creds = list(        
        access_key_id = "string",
        secret_access_key = "string",
      
Examples

## Not run: svc <- textract()
svc$analyze_document(
  Foo = 123
)
## End(Not run)
session_token = "string"
),
profile = "string"
),
endpoint = "string",
region = "string"
)

## Operations

- **create_vocabulary**
  - Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription.

- **create_vocabulary_filter**
  - Creates a new vocabulary filter that you can use to filter words, such as profane words, from the output of a transcription job.

- **delete_transcription_job**
  - Deletes a previously submitted transcription job along with any other generated results such as the transcription, models, and so on.

- **delete_vocabulary**
  - Deletes a vocabulary from Amazon Transcribe.

- **delete_vocabulary_filter**
  - Removes a vocabulary filter.

- **get_transcription_job**
  - Returns information about a transcription job.

- **get_vocabulary**
  - Gets information about a vocabulary.

- **get_vocabulary_filter**
  - Returns information about a vocabulary filter.

- **list_transcription_jobs**
  - Lists transcription jobs with the specified status.

- **list_vocabulary**
  - Returns a list of vocabularies that match the specified criteria.

- **list_vocabulary_filter**
  - Gets information about vocabulary filters.

- **start_transcription_job**
  - Starts an asynchronous job to transcribe speech to text.

- **update_vocabulary**
  - Updates an existing vocabulary with new values.

- **update_vocabulary_filter**
  - Updates a vocabulary filter with a new list of filtered words.

## Examples

```r
## Not run: svc <- transcribeservice()
svc$create_vocabulary(
  Foo = 123
)
## End(Not run)
```

---

## translate

### Amazon Translate

#### Description

Provides translation between one source language and another of the same set of languages.

#### Usage

```r
translate(config = list())
```
Arguments
config Optional configuration of credentials, endpoint, and/or region.

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

Operations
- `delete_terminology` A synchronous action that deletes a custom terminology
- `describe_text_translation_job` Gets the properties associated with an asynchronous batch translation job including name, ID, input, and output
- `get_terminology` Retrieves a custom terminology
- `import_terminology` Creates or updates a custom terminology, depending on whether or not one already exists for the given terminology name
- `list_terminologies` Provides a list of custom terminologies associated with your account
- `list_text_translation_jobs` Gets a list of the batch translation jobs that you have submitted
- `start_text_translation_job` Starts an asynchronous batch translation job
- `stop_text_translation_job` Stops an asynchronous batch translation job that is in progress
- `translate_text` Translates input text from the source language to the target language

Examples
```
## Not run: svc <- translate()
svc$delete_terminology(
  Foo = 123
)
## End(Not run)
```
Description

This is the AWS WAF API Reference for using AWS WAF with Amazon CloudFront. The AWS WAF actions and data types listed in the reference are available for protecting Amazon CloudFront distributions. You can use these actions and data types via the endpoint waf.amazonaws.com. This guide is for developers who need detailed information about the AWS WAF API actions, data types, and errors. For detailed information about AWS WAF features and an overview of how to use the AWS WAF API, see the AWS WAF Developer Guide.

Usage

```r
waf(config = list())
```

Arguments

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

Service syntax

```r
svc <- waf(
  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_byte_match_set**: Creates a ByteMatchSet
- **create_geo_match_set**: Creates a GeoMatchSet, which you use to specify which web requests you want to allow or block based on the country that the requests originate from
- **create_ip_set**: Creates an IPSet, which you use to specify which web requests that you want to allow or block based on the IP addresses that the requests originate from
- **create_rate_based_rule**: Creates a RateBasedRule
- **create_regex_match_set**: Creates a RegexMatchSet
- **create_regex_pattern_set**: Creates a RegexPatternSet
- **create_rule**: Creates a Rule, which contains the IPSet objects, ByteMatchSet objects, and other predicates that identify the requests that you want to block
- **create_rule_group**: Creates a RuleGroup
- **create_size_constraint_set**: Creates a SizeConstraintSet
- **create_sql_injection_match_set**: Creates a SqlInjectionMatchSet, which you use to allow, block, or count requests that contain snippets of SQL code in a specified part of web requests
- **create_web_acl**: Creates a WebACL, which contains the Rules that identify the CloudFront web requests that you want to allow, block, or count
- **create_xss_match_set**: Creates an XssMatchSet, which you use to allow, block, or count requests that contain cross-site scripting attacks in the specified part of web requests
- **delete_byte_match_set**: Permanently deletes a ByteMatchSet
delete_geo_match_set
delete_ip_set
delete_logging_configuration
delete_permission_policy
delete_rate_based_rule
delete_regex_match_set
delete_regex_pattern_set
delete_rule
delete_rule_group
delete_size_constraint_set
delete_sql_injection_match_set
delete_web_acl
delete_xss_match_set
get_byte_match_set
get_change_token
get_change_token_status
get_geo_match_set
get_ip_set
get_logging_configuration
get_permission_policy
get_rate_based_rule
get_rate_based_rule_managed_keys
get_regex_match_set
get_regex_pattern_set
get_rule
get_rule_group
get_sampled_requests
get_size_constraint_set
get_sql_injection_match_set
get_web_acl
get_xss_match_set
list_activated_rules_in_rule_group
list_byte_match_sets
list_geo_match_sets
list_ip_sets
list_logging_configurations
list_rate_based_rules
list_regex_match_sets
list_regex_pattern_sets
list_rule_groups
list_rules
list_size_constraint_sets
list_sql_injection_match_sets
list_subscribed_rule_groups
list_tags_for_resource
list_web_ac_ls
list_xss_match_sets
put_logging_configuration

Permanently deletes a GeoMatchSet
Permanently deletes an IPSet
Permanently deletes the LoggingConfiguration from the specified web ACL
Permanently deletes an IAM policy from the specified RuleGroup
Permanently deletes a RateBasedRule
Permanently deletes a RegexMatchSet
Permanently deletes a RegexPatternSet
Permanently deletes a Rule
Permanently deletes a RuleGroup
Permanently deletes a SizeConstraintSet
Permanently deletes a SqlInjectionMatchSet
Permanently deletes a WebACL
Permanently deletes an XssMatchSet
Returns the ByteMatchSet specified by ByteMatchSetId
When you want to create, update, or delete AWS WAF objects, get a change token and include the change token in the create, update, or delete request
Returns the status of a ChangeToken that you got by calling GetChangeToken
Returns the GeoMatchSet that is specified by GeoMatchSetId
Returns the IPSet that is specified by IPSetId
Returns the LoggingConfiguration for the specified web ACL
Returns the IAM policy attached to the RuleGroup
Returns the RateBasedRule that is specified by the RuleId that you included in the GetRateBasedRule request
Returns an array of IP addresses currently being blocked by the RateBasedRule that is specified by the RuleId
Returns the RegexMatchSet specified by RegexMatchSetId
Returns the RegexPatternSet specified by RegexPatternSetId
Returns the Rule that is specified by the RuleId that you included in the GetRule request
Returns the RuleGroup that is specified by the RuleGroupId that you included in the GetRuleGroup request
Gets detailed information about a specified number of requests—a sample—that AWS WAF selected for analysis
Returns the SizeConstraintSet specified by SizeConstraintSetId
Returns the SqlInjectionMatchSet that is specified by SqlInjectionMatchSetId
Returns the WebACL that is specified by WebACLId
Returns the XssMatchSet that is specified by XssMatchSetId
Returns an array of ActivatedRule objects
Returns an array of ByteMatchSetSummary objects
Returns an array of GeoMatchSetSummary objects in the response
Returns an array of IPSetsSummary objects in the response
Returns an array of LoggingConfiguration objects
Returns an array of RuleSummary objects
Returns an array of RegexMatchSetSummary objects
Returns an array of RegexPatternSetSummary objects
Returns an array of RuleGroup objects
Returns an array of RuleSummary objects
Returns an array of SizeConstraintSetSummary objects
Returns an array of SqlInjectionMatchSet objects
Returns an array of RuleGroup objects that you are subscribed to
List tags for resource
Returns an array of WebACLSummary objects in the response
Returns an array of XssMatchSet objects
Associates a LoggingConfiguration with a specified web ACL
### put_permission_policy
Attaches an IAM policy to the specified resource

### tag_resource
Tag resource

### untag_resource
Untag resource

### update_byte_match_set
Inserts or deletes ByteMatchTuple objects (filters) in a ByteMatchSet

### update_geo_match_set
Inserts or deletes GeoMatchConstraint objects in a GeoMatchSet

### update_ip_set
Inserts or deletes IPSetDescriptor objects in an IPSet

### update_rate_based_rule
Inserts or deletes Predicate objects in a rule and updates the RateLimit in the rule

### update_regex_match_set
Inserts or deletes RegexMatchTuple objects (filters) in a RegexMatchSet

### update_regex_pattern_set
Inserts or deletes RegexPatternString objects in a RegexPatternSet

### update_rule
Inserts or deletes Predicate objects in a Rule

### update_rule_group
Inserts or deletes ActivatedRule objects in a RuleGroup

### update_size_constraint_set
Inserts or deletes SizeConstraint objects (filters) in a SizeConstraintSet

### update_sql_injection_match_set
Inserts or deletes SqlInjectionMatchTuple objects (filters) in a SqlInjectionMatchSet

### update_web_acl
Inserts or deletes ActivatedRule objects in a WebACL

### update_xss_match_set
Inserts or deletes XssMatchTuple objects (filters) in an XssMatchSet

---

### Examples

```r
# The following example creates an IP match set named MyIPSetFriendlyName.
## Not run: svc <- waf()
svc$create_ip_set(
    ChangeToken = "abcd12f2-46da-4f6b-b8d5-fbd4c466928f",
    Name = "MyIPSetFriendlyName"
)
## End(Not run)
```

---

### Description

This is the **AWS WAF Regional API Reference** for using AWS WAF with Elastic Load Balancing (ELB) Application Load Balancers. The AWS WAF actions and data types listed in the reference are available for protecting Application Load Balancers. You can use these actions and data types by means of the endpoints listed in **AWS Regions and Endpoints**. This guide is for developers who need detailed information about the AWS WAF API actions, data types, and errors. For detailed information about AWS WAF features and an overview of how to use the AWS WAF API, see the **AWS WAF Developer Guide**.

### Usage

```r
wafregional(config = list())
```

### Arguments

- **config**
  Optional configuration of credentials, endpoint, and/or region.
Service syntax

```r
csvc <- wafregional(
  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_web_acl`: Associates a web ACL with a resource, either an application load balancer or Amazon API Gateway stage.
- `create_byte_match_set`: Creates a ByteMatchSet.
- `create_geo_match_set`: Creates an GeoMatchSet, which you use to specify which web requests you want to allow or block based on the country that the requests originate from.
- `create_ip_set`: Creates an IPSet, which you use to specify which web requests that you want to allow or block based on the IP addresses that the requests originate from.
- `create_rate_based_rule`: Creates a RateBasedRule.
- `create_regex_match_set`: Creates a RegexMatchSet.
- `create_regex_pattern_set`: Creates a RegexPatternSet.
- `create_rule`: Creates a Rule, which contains the IPSet objects, ByteMatchSet objects, and other predicates that identify the requests that you want to block.
- `create_rule_group`: Creates a RuleGroup.
- `create_size_constraint_set`: Creates a SizeConstraintSet.
- `create_sql_injection_match_set`: Creates a SqlInjectionMatchSet, which you use to allow, block, or count requests that contain snippets of SQL code in a specified part of web requests.
- `create_web_acl`: Creates a WebACL, which contains the Rules that identify the CloudFront web requests that you want to allow, block, or count.
- `create_xss_match_set`: Creates an XssMatchSet, which you use to allow, block, or count requests that contain cross-site scripting attacks in the specified part of web requests.
- `delete_byte_match_set`: Permanently deletes a ByteMatchSet.
- `delete_geo_match_set`: Permanently deletes a GeoMatchSet.
- `delete_ip_set`: Permanently deletes an IPSet.
- `delete_logging_configuration`: Permanently deletes the LoggingConfiguration from the specified web ACL.
- `delete_permission_policy`: Permanently deletes an IAM policy from the specified RuleGroup.
- `delete_rate_based_rule`: Permanently deletes a RateBasedRule.
- `delete_regex_match_set`: Permanently deletes a RegexMatchSet.
- `delete_regex_pattern_set`: Permanently deletes a RegexPatternSet.
- `delete_rule`: Permanently deletes a Rule.
- `delete_rule_group`: Permanently deletes a RuleGroup.
- `delete_size_constraint_set`: Permanently deletes a SizeConstraintSet.
- `delete_sql_injection_match_set`: Permanently deletes a SqlInjectionMatchSet.
- `delete_web_acl`: Permanently deletes a WebACL.
- `delete_xss_match_set`: Permanently deletes an XssMatchSet.
- `disassociate_web_acl`: Removes a web ACL from the specified resource, either an application load balancer or Amazon API Gateway stage.
- `get_byte_match_set`: Returns the ByteMatchSet specified by ByteMatchSetId.
- `get_change_token`: When you want to create, update, or delete AWS WAF objects, get a change token and include the change token in the create, update, or delete request.
get_change_token_status
get_geo_match_set
get_ip_set
get_logging_configuration
get_permission_policy
get_rate_based_rule
get_rate_based_rule_managed_keys
get_regex_match_set
get_regex_pattern_set
get_rule
get_rule_group
get_sampled_requests
get_size_constraint_set
get_web_acl
get_web_acl_for_resource
get_xss_match_set
list_activated_rules_in_rule_group
list_byte_match_sets
list_geo_match_sets
list_ip_sets
list_logging_configurations
list_rate_based_rules
list_regex_match_sets
list_regex_pattern_sets
list_resources_for_web_acl
list_rule_groups
list_rules
list_size_constraint_sets
list_sql_injection_match_sets
list_subscribed_rule_groups
list_tags_for_resource
list_web_acls
list_xss_match_sets
put_logging_configuration
put_permission_policy
tag_resource
unbind_resource
update_byte_match_set
update_geo_match_set
update_ip_set
update_rate_based_rule
update_regex_match_set
update_regex_pattern_set
update_rule
update_rule_group
update_size_constraint_set
update_sql_injection_match_set

Returns the status of a ChangeToken that you got by calling GetChangeToken
Returns the GeoMatchSet that is specified by GeoMatchSetId
Returns the IPSet that is specified by IPSetId
Returns the LoggingConfiguration for the specified web ACL
Returns the IAM policy attached to the RuleGroup
Returns the RateBasedRule that is specified by the RuleId that you included in the GetRateBasedRule request
Returns an array of IP addresses currently being blocked by the RateBasedRule that is specified by the RuleId
Returns the RegexMatchSet specified by RegexMatchSetId
Returns the RegexPatternSet specified by RegexPatternSetId
Returns the Rule that is specified by the RuleId that you included in the GetRule request
Returns the RuleGroup that is specified by the RuleGroupId that you included in the GetRuleGroup request
Gets detailed information about a specified number of requests—a sample— that AWS WAF records in the specified time period
Returns the SizeConstraintSet specified by SizeConstraintSetId
Returns the SqlInjectionMatchSet that is specified by SqlInjectionMatchSetId
Returns the WebACL that is specified by WebACLId
Returns the web ACL for the specified resource, either an application load balancer or an API Gateway stage
Returns the XssMatchSet that is specified by XssMatchSetId
Returns an array of ActivatedRule objects
Returns an array of ByteMatchSetSummary objects
Returns an array of GeoMatchSetSummary objects in the response
Returns an array of IPSetSummary objects in the response
Returns an array of LoggingConfiguration objects
Returns an array of RuleSummary objects
Returns an array of RegexMatchSetSummary objects
Returns an array of RegexPatternSetSummary objects
Returns an array of resources associated with the specified web ACL
Returns an array of RuleGroup objects
Returns an array of RuleSummary objects
Returns an array of SizeConstraintSetSummary objects
Returns an array of SqlInjectionMatchSet objects
Returns an array of RuleGroup objects that you are subscribed to
Returns an array of WebACLSummary objects in the response
Returns an array of XssMatchSet objects
Associates a LoggingConfiguration with a specified web ACL
Attaches an IAM policy to the specified resource
Tag resource
Untag resource
Inserts or deletes ByteMatchTuple objects (filters) in a ByteMatchSet
Inserts or deletes GeoMatchConstraint objects in a GeoMatchSet
Inserts or deletes IPSetDescriptor objects in an IPSet
Inserts or deletes Predicate objects in a rule and updates the RateLimit in the rule
Inserts or deletes RegexMatchTuple objects (filters) in a RegexMatchSet
Inserts or deletes RegexPatternString objects in a RegexPatternSet
Inserts or deletes Predicate objects in a Rule
Inserts or deletes ActivatedRule objects in a RuleGroup
Inserts or deletes SizeConstraint objects (filters) in a SizeConstraintSet
Inserts or deletes SqlInjectionMatchTuple objects (filters) in a SqlInjectionMatchSet
update_web_acl Inserts or deletes ActivatedRule objects in a WebACL
update_xss_match_set Inserts or deletes XssMatchTuple objects (filters) in an XssMatchSet

Examples

# The following example creates an IP match set named MyIPSetFriendlyName.
## Not run: svc <- wafregional()
svc$create_ip_set(
  ChangeToken = "abcd12f2-46da-4f6b-b8d5-fbd4c466928f",
  Name = "MyIPSetFriendlyName"
)
## End(Not run)
Index

abort_environment_update, 104
abort_multipart_upload, 120, 219
abort_vault_lock, 120
accept_direct_connect_gateway_association_protocol_negotiation, 72
accept_handshake, 176
accept_invitation, 128, 231
accept_portfolio_share, 234
accept_resource_share_invitation, 196
accept_shared_directory, 75
accept_transit_gateway_peering_attachment, 84
accept_transit_gateway_vpc_attachment, 84
accept_vpc_endpoint_connections, 84
accept_vpc_peering_connection, 84
acm, 4
acmpca, 6
activate_event_source, 48, 114
activate_gateway, 256
activate_pipeline, 69
add_application_cloud_watch_logging_option, 142, 143
add_application_input, 142, 143
add_application_input_processing_configuration, 142, 143
add_application_output, 142, 143
add_application_reference_data_source, 142, 144
add_application_vpc_configuration, 144
add_attachments_to_set, 262
add_attributes_to_findings, 137
add_cache, 256
add_client_id_to_open_id_connect_provider, 133
add_communication_to_case, 262
add_custom_attributes, 54
add_facet_to_object, 32
add_instance_fleet, 112
add_instance_groups, 112
add_ip_routes, 75
add_job_flow_steps, 112
add_layer_version_permission, 148
add_listener_certificates, 110
add_permission, 148, 248, 251
add_role_to_db_cluster, 168, 198
add_role_to_db_instance, 198
add_role_to_instance_profile, 133
add_source_identifier_to_subscription, 168, 198
add_tags, 45, 69, 106, 108, 110, 112, 158, 224
add_tags_to_certificate, 5
add_tags_to_resource, 39, 75, 78, 102, 168, 198, 252, 256
add_tags_to_stream, 140
add_tags_to_vault, 120
add_upload_buffer, 256
add_user_to_group, 133
add_working_storage, 256
admin_add_user_to_group, 54
admin_confirm_sign_up, 54
admin_create_user, 54
admin_delete_user, 54
admin_delete_user_attributes, 54
admin_disable_provider_for_user, 54
admin_disable_user, 54
admin_enable_user, 54
admin_forget_device, 54
admin_get_device, 54
admin_get_user, 54
admin_initiate_auth, 54
admin_link_provider_for_user, 54
admin_list_devices, 54
admin_list_groups_for_user, 54
admin_list_user_auth_events, 54
admin_remove_user_from_group, 54
admin_reset_user_password, 54
admin_respond_to_auth_challenge, 54
admin_set_user_mfa_preference, 54
admin_set_user_password, 54
admin_set_user_settings, 54
admin_update_auth_event_feedback, 54
admin_update_device_status, 54
admin_update_user_attributes, 54
admin_user_global_sign_out, 54
advertise_byoip_cidr, 84
allocate_address, 84
allocate_connection_on_interconnect, 72
allocate_hosted_connection, 72
allocate_private_virtual_interface, 72
allocate_public_virtual_interface, 72
allocate_static_ip, 155
allocate_transit_virtual_interface, 72
analyze_document, 266
apigateway, 7
apigatewaymanagementapi, 11
apigatewayv2, 12
applicationautoscaling, 14
applicationinsights, 16
apply_environment_managed_action, 104
apply_pending_maintenance_action, 78, 168, 198
apply_schema, 32
apply_security_groups_to_client_vpn_target_network, 84
apply_security_groups_to_load_balancer, 108
appmesh, 18
approve_assignment, 166
archive_findings, 128
assign_instance, 171
assign_ipv_6_addresses, 84
assign_private_ip_addresses, 84
assign_tape_pool, 256
assign_volume, 171
associate_address, 84
associate_admin_account, 117
associate_budget_with_resource, 234
associate_client_vpn_target_network, 84
associate_connection_with_lag, 72
associate_dhcp_options, 84
associate_drt_log_bucket, 246
associate_drt_role, 246
associate_elastic_ip, 171
associate_hosted_connection, 72
associate_iam_instance_profile, 84
associate_kms_key, 50
associate_member_account, 159
associate_node, 174
associate_principal_with_portfolio, 234
associate_product_with_portfolio, 235
associate_qualification_with_worker, 166
associate_resolver_endpoint_ip_address, 218
associate_resolver_rule, 218
associate_resource_share, 196
associate_resource_share_permission, 196
associate_route_table, 84
associate_s3_resources, 159
associate_service_action_with_provisioning_artifact, 235
associate_service_quota_template, 239
associate_software_token, 54
associate_subnet_cidr_block, 84
associate_tag_option_with_resource, 235
associate_transit_gateway_multicast_domain, 84
associate_transit_gateway_route_table, 84
associate_trial_component, 224
associate_virtual_interface, 72
associate_vpc_cidr_block, 84
associate_vpc_with_hosted_zone, 214
associate_web_acl, 272
assume_role, 260
assume_role_with_saml, 260
assume_role_with_web_identity, 260
athena, 20
attach_classic_link_vpc, 84
attach_disk, 155
attach_elastic_load_balancer, 171
attach_group_policy, 133
attach_instances, 22
cancel_ingestion, 193
cancel_job, 29
cancel_key_deletion, 146
cancel_maintenance_window_execution, 252
cancel_ml_task_run, 124
cancel_reserved_instances_listing, 84
cancel_resize, 203
cancel_retrieval, 257
cancel_rotate_secret, 229
cancel_schema_extension, 75
cancel_spot_fleet_requests, 84
cancel_spot_instance_requests, 84
cancel_steps, 112
cancel_update_stack, 35
change_message_visibility, 251
change_message_visibility_batch, 251
change_password, 54, 133
change_resource_record_sets, 214
change_tags_for_resource, 214
check_dns_availability, 104
check_domain_availability, 216
check_domain_transferability, 216
check_if_phone_number_is_opted_out, 248
classify_document, 58
clone_receipt_rule_set, 240
clone_stack, 171
close_instance_public_ports, 155
cloudformation, 34
cloudfront, 36
cloudhsm, 38
cloudhsmv2, 39
cloudsearch, 41
cloudsearchdomain, 42
cloudtrail, 44
cloudwatch, 45
cloudwatchevents, 47
cloudwatchlogs, 49
cognitoidentity, 51
cognitoidentityprovider, 53
cognitosync, 56
commit_transaction, 201
compare_faces, 206
complete_layer_upload, 94
complete_lifecycle_action, 22
complete_migration, 102
complete multipart upload, 120, 219
complete_vault_lock, 120
compose_environments, 104
comprehend, 58
comprehendmedical, 60
configservice, 61
configure_health_check, 108
confirm_connection, 72
confirm_device, 54
confirm_forgot_password, 54
confirm_private_virtual_interface, 72
confirm_product_instance, 84
confirm_public_virtual_interface, 72
confirm_sign_up, 54
confirm_subscription, 248
confirm transit virtual interface, 73
costexplorer, 67
costandusagereportservice, 65
costexplorer, 67
count_closed_workflow_executions, 264
count_open_workflow_executions, 264
count_pending_activity_tasks, 264
count_pending_decision_tasks, 264
create accelerator, 123
create_access_key, 133
create_access_point, 222
create_account, 176
create_account_alias, 133
create_action_target, 231
create_activation, 252
create_activity, 243
create_additional_assignments_for_hit,
create_algorithm, 224
create_alias, 75, 146, 148
create_api, 13
create_api_key, 8
create_api_mapping, 13
create_app, 171, 184, 224
create_app_cookie_stickiness_policy, 108
create_application, 17, 104, 142, 144, 233
create_application_snapshot, 104, 233
create_assessment_target, 137
create_assessment_template, 137
create_association, 252
create_association_batch, 252
create_authorizer, 8, 13
create_auto_ml_job, 224
create_auto_scaling_group, 22
create_backup, 80, 118, 174
create_backup_plan, 25
create_backup_selection, 25
create_backup_vault, 25
create_base_path_mapping, 8
create_batch_inference_job, 178
create_batch_prediction, 158
create_bgp_peer, 73
create_bot_version, 150
create_broker, 164
create_bucket, 219
create_budget, 30
create_byte_match_set, 269, 272
create_cache_cluster, 102
create_cache_parameter_group, 102
create_cache_security_group, 102
create_cache_subnet_group, 102
create_cachedi_scsi_volume, 257
create_campaign, 178, 184
create_capacity_provider, 96
create_capacity_reservation, 85
create_case, 262
create_certificate_authority, 6
create_certificate_authority_audit_report, 6
create_change_set, 35
create_classifier, 124
create_client_vpn_endpoint, 85
create_client_vpn_route, 85
create_cloudformation_change_set, 233
create_cloudformation_stack, 155
create_cloudformation_template, 233
create_cloud_front_origin_access_identity, 37
create_cluster, 40, 71, 96, 100, 139, 203
create_cluster_parameter_group, 203
create_cluster_security_group, 203
create_cluster_snapshot, 203
create_cluster_subnet_group, 203
create_code_repository, 224
create_collection, 206
create_compilation_job, 224
create_component, 17
create_compute_environment, 29
create_computer, 75
create_conditional_forwarder, 75
create_configuration, 139, 164
create_configuration_set, 187, 189, 240
create_configuration_set_event_destination, 187, 189, 240
create_configuration_set_tracking_options, 240
create_configuration_template, 104
create_connection, 73, 124
create_constraint, 235
create_cost_category_definition, 68
create_crawler, 124
create_custom_availability_zone, 198
create_custom_key_store, 146
create_custom_verification_email_template, 240
create_customer_gateway, 85
create_dashboard, 193
create_data_repository_task, 118
create_data_set, 193
create_data_source, 193
create_data_source_from_rds, 158
create_data_source_from_redshift, 158
create_data_source_from_s3, 158
create_database, 124
create_dataset, 178
create_dataset_group, 178
create_dataset_import_job, 178
create_db_cluster, 78, 168, 198
create_db_cluster_endpoint, 198
create_db_cluster_parameter_group, 78, 168, 198
create_db_cluster_snapshot, 78, 168, 198
create_db_instance, 78, 168, 198
create_db_instance_read_replica, 198
create_db_parameter_group, 168, 198
create_db_proxy, 198
create_db_security_group, 198
create_db_snapshot, 198
create_db_subnet_group, 78, 168, 198
create_dedicated_ip_pool, 187
create_default_subnet, 85
create_default_vpc, 85
create_deliverability_test_report, 187
create_delivery_stream, 115
create_deployment, 8, 13, 171
create_detector, 128
create_dev_endpoint, 124
create_dhcp_options, 85
create_direct_connect_gateway, 73
create_direct_connect_gateway_association, 73
create_direct_connect_gateway_association_proposal, 73
create_directory, 32, 75
create_disk, 155
create_disk_from_snapshot, 155
create_disk_snapshot, 155
create_distribution, 37
create_distribution_with_tags, 37
create_document, 252
create_document_classifier, 58
create_documentation_part, 8
create_documentation_version, 8
create_domain, 42, 155, 224, 247
create_domain_entry, 155
create_domain_name, 8, 13
create_egress_only_internet_gateway, 85
create_elasticsearch_domain, 106
create_email_identity, 187
create_email_template, 184
create_endpoint, 58, 224
create_endpoint_config, 224
create_endpoint_group, 123
create_entity_recognizer, 58
create_environment, 104
create_evaluation, 158
create_event_bus, 48, 114
create_event_source_mapping, 148
create_event_subscription, 168, 198, 203
create_event_tracker, 178
create_exclusions_preview, 137
create_experiment, 224
create_export_job, 184
create_export_task, 50
create_facet, 32
create_fargate_profile, 100
create_field_level_encryption_config, 37
create_field_level_encryption_profile, 37
create_file_system, 98, 118
create_file_system_from_backup, 118
create_filter, 128
create_fleet, 85
create_flow_definition, 224
create_flow_logs, 85
create_fpga_image, 85
create_function, 148
create_geo_match_set, 269, 272
create_global_cluster, 198
create_global_table, 80
create_gov_cloud_account, 176
create_grant, 146
create_group, 54, 133, 193, 208
create_group_membership, 193
create_hapg, 39
create_health_check, 214
create_hit, 166
create_hit_type, 166
create_hit_with_hit_type, 166
create_hosted_zone, 214
create_hsm, 39, 40
create_hsm_client_certificate, 203
create_hsm_configuration, 203
create_http_namespace, 237
create_human_task_ui, 224
create_hyper_parameter_tuning_job, 224
create_iam_policy_assignment, 193
create_identity_pool, 52
create_identity_provider, 54
create_image, 85
create_import_job, 184
create_index, 32
create_ingestion, 193
create_insight, 231
create_instance, 171
create_instance_export_task, 85  
create_instance_profile, 133  
create_instance_snapshot, 155  
create_instances, 155  
create_instances_from_snapshot, 155  
create_integration, 13  
create_integration_response, 13  
create_intent_version, 150  
create_interconnect, 73  
createインターネット_gateway, 85  
create_invalidation, 37  
create_ip_set, 128, 269, 272  
create_job, 124, 222  
create_job_queue, 29  
create_journey, 184  
create_key, 146  
create_key_pair, 85, 155  
create_labeling_job, 224  
create_lag, 73  
create_launch_configuration, 22  
create_launch_template, 85  
create_launch_template_version, 85  
create_layer, 171  
create_lb_cookie_stickiness_policy, 108  
create_license_configuration, 153  
create_lifecycle_policy, 77  
create_listener, 110, 123  
create_load_balancer, 108, 110, 155  
create_load_balancer_listeners, 108  
create_load_balancer_policy, 108  
create_load_balancer_tls_certificate, 155  
create_local_gateway_route, 85  
create_local_gateway_route_table_vpc_association, 85  
create_log_group, 50  
create_log_pattern, 17  
create_log_stream, 50  
create_log_subscription, 75  
create_login_profile, 133  
create_luna_client, 39  
create_maintenance_window, 252  
create_members, 128, 231  
create_mesh, 19  
create_microsoft_ad, 75  
create_ml_model, 138  
create_ml_transform, 124  
create_model, 8, 13, 224  
create_model_package, 224  
create_monitoring_schedule, 224  
create_mount_target, 98  
create_multipart_upload, 219  
create_named_query, 21  
create_nat_gateway, 85  
create_network_acl, 85  
create_network_acl_entry, 85  
create_network_interface, 85  
create_network_interface_permission, 85  
create_nfs_file_share, 257  
create_nodegroup, 100  
create_notebook_instance, 224  
create_notebook_instance_lifecycle_config, 224  
create_notification, 30  
create_object, 32  
create_open_id_connect_provider, 133  
create_ops_item, 252  
create_option_group, 198  
create_or_update_tags, 22  
create_organization, 176  
create_organizational_unit, 176  
create_parameter_group, 71  
create_partition, 124  
create_partner_event_source, 48, 114  
create_patch_baseline, 252  
create_permission, 6  
create_pipeline, 69  
create_placement_group, 85  
create_platform_application, 248  
create_platform_endpoint, 248  
create_platform_version, 104  
create_platform_version, 104  
create_policy, 133, 176  
create_policy_version, 133  
create_portfolio, 235  
create_portfolio_share, 235  
create_presigned_domain_url, 224  
create_presigned_notebook_instance_url, 224  
create_private_dns_namespace, 237  
create_private_virtual_interface, 73  
create_processing_job, 224  
create_product, 235  
create_project, 206  
create_project_version, 206
create_protection, 246
create_provisioned_product_plan, 235
create_provisioning_artifact, 235
create_public_dns_namespace, 237
create_public_key, 37
create_public_virtual_interface, 73
create_publishing_destination, 128
create_push_template, 184
create_qualification_type, 166
create_query_logging_config, 214
create_queue, 251
create_rate_based_rule, 269, 272
create_realtime_endpoint, 158
create_receipt_filter, 240
create_receipt_rule, 240
create_receipt_rule_set, 240
create_regex_match_set, 269, 272
create_regex_pattern_set, 269, 272
create_relational_database, 155
create_relational_database_from_snapshot, 155
create_relational_database_snapshot, 155
create_replication_group, 102
create_repository, 94
create_request_validator, 8
create_reserved_instances_listing, 85
create_resolver_endpoint, 218
create_resolver_rule, 218
create_resource, 8
create_resource_data_sync, 252
create_resource_group, 137
create_resource_server, 54
create_resource_share, 196
create_rest_api, 8
create_reusable_delegation_set, 214
create_rule, 133
create_route, 13, 19, 85
create_route_response, 13
create_route_table, 85
create_rule, 110, 269, 272
create_rule_group, 269, 272
create_saml_provider, 133
create_sample_findings, 128
create_scaling_plan, 24
create_scheduled_action, 203
create_schema, 32, 178
create_script, 124
create_secret, 229
create_security_configuration, 112, 125
create_security_group, 85
create_segment, 184
create_server, 174
create_service, 96, 237
create_service_action, 235
create_service_linked_role, 133
create_service_specific_credential, 133
create_size_constraint_set, 269, 272
create_slot_type_version, 150
create_smb_file_share, 257
create_sms_template, 184
create_snapshot, 75, 85, 102, 257
create_snapshot_copy_grant, 203
create_snapshot_from_volume_recovery_point, 257
create_snapshot_schedule, 203
create_snapshots, 85
create_solution, 178
create_solution_version, 178
create_spot_datafeed_subscription, 85
create_sql_injection_match_set, 269, 272
create_stack, 35, 171
create_stack_instances, 35
create_stack_set, 35
create_stage, 8, 13
create_state_machine, 243
create_storage_location, 104
create_storedi_scsi_volume, 257
create_stream, 140
create_stream_processor, 206
create_streaming_distribution, 37
create_streaming_distribution_with_tags, 37
create_subnet, 85
create_subnet_group, 71
create_subscriber, 30
create_subscription, 246
create_table, 80, 125
create_tag_option, 235
create_tags, 85, 98, 164, 203
create_tape_with_barcode, 257
create_tapes, 257
create_target_group, 110
create_task_set, 96
create_template, 193, 240
create_template_alias, 193
create_threat_intel_set, 128
create_topic, 249
create_traffic_mirror_filter, 85
create_traffic_mirror_filter_rule, 85
create_traffic_mirror_session, 85
create_traffic_mirror_target, 85
create_traffic_policy, 214
create_traffic_policy_instance, 214
create_traffic_policy_version, 214
create_trail, 45
create_training_job, 224
create_transform_job, 224
create_transit_gateway, 85
create_transit_gateway_multicast_domain, 85
create_transit_gateway_peering_attachment, 85
create_transit_gateway_route, 85
create_transit_gateway_route_table, 85
create_transit_gateway_vpc_attachment, 85
create_transit_virtual_interface, 73
create_trial, 224
create_trial_component, 224
create_trigger, 125
create_trust, 75
createTypedLinkFacet, 32
create_usage_plan, 8
create_usage_plan_key, 8
create_user, 65, 133, 164
create_user_defined_function, 125
create_user_import_job, 54
create_user_pool, 54
create_user_pool_client, 54
create_user_pool_domain, 54
create_user_profile, 171, 224
create_vault, 120
create_virtual_mfa_device, 133
create_virtual_node, 19
create_virtual_router, 19
create_virtual_service, 19
create_vocabulary, 267
create_vocabulary_filter, 267
create_voice_template, 184
create_volume, 85
create_vpc, 85
create_vpc_association_authorization, 214
create_vpc_endpoint, 85
create_vpc_endpoint_connection_notification, 85
create_vpc_endpoint_service_configuration, 86
create_vpc_link, 8
create_vpc_peering_connection, 86
create_vpn_connection, 86
create_vpn_connection_route, 86
create_vpn_gateway, 86
create_web_acl, 269, 272
create_work_group, 21
create_worker_block, 166
create_workflow, 125
create_workteam, 224
create_xss_match_set, 269, 272
datapipeline, 68
dax, 70
deactivate_event_source, 48, 114
deactivate_mfa_device, 133
deactivate_pipeline, 69
decline_handshake, 176
decode_authorization_message, 260
decrease_replica_count, 102
decrease_replication_factor, 71
decrease_stream_retention_period, 140
decrypt, 146
define_analysis_scheme, 42
define_expression, 42
define_index_field, 42
define_suggester, 42
delete_accelerator, 123
delete_access_key, 133
delete_access_point, 222
delete_access_point_policy, 222
delete_account_alias, 133
delete_account_password_policy, 133
delete_account_setting, 96
delete_action_target, 231
delete_activation, 252
delete_activity, 243
delete_adm_channel, 184
delete_aggregation_authorization, 62
delete alarms, 46
delete_algorithm, 224
delete_alias, 146, 148
delete_analysis_scheme, 42
delete_anomaly_detector, 46
delete_api, 13
delete_api_key, 8
delete_api_mapping, 13
delete_apns_channel, 184
delete_apns_sandbox_channel, 184
delete_apns_voip_channel, 184
delete_app, 171, 184, 224
delete_application, 17, 104, 142, 144, 233
delete_application_cloud_watch_logging_option, 142, 144
define_application_input_processing_configuration, 142, 144
define_application_output, 142, 144
define_application_reference_data_source, 142, 144
define_application_snapshot, 144
define_application_version, 104
define_application_vpc_configuration, 144
define_archive, 120
define_assessment_run, 137
define_assessment_target, 137
define_assessment_template, 137
define_association, 252
define_attributes, 96, 247
define_authorizer, 8, 13
define_auto_scaling_group, 22
define_auto_snapshot, 155
define_backup, 40, 80, 118, 174
define_backup_plan, 26
define_backup_selection, 26
define_backup_vault, 26
define_backup_vault_access_policy, 26
define_backup_vault_notifications, 26
define_baidu_channel, 184
define_bandwidth_rate_limit, 257
define_base_path_mapping, 8
define_batch_prediction, 158
define_bgp_peer, 73
define_bot, 150
define_bot_alias, 150
define_bot_channel_association, 150
define_bot_version, 150
define_broker, 164
delete_bucket, 219
delete_bucket_analytics_configuration, 219
delete_bucket_cors, 219
delete_bucket_encryption, 219
delete_bucket_inventory_configuration, 220
delete_bucket_lifecycle, 220
delete_bucket_metrics_configuration, 220
delete_bucket_policy, 220
delete_backup_plan, 26
define_backup_selection, 26
define_backup_vault, 26
define_backup_vault_access_policy, 26
define_backup_vault_notifications, 26
define_baidu_channel, 184
define_bandwidth_rate_limit, 257
define_base_path_mapping, 8
define_batch_prediction, 158
define_bgp_peer, 73
define_bot, 150
define_bot_alias, 150
define_bot_channel_association, 150
define_bot_version, 150
define_broker, 164
delete_collection, 206
define_component, 17
define_compute_environment, 29
define_conditional_forwarder, 75
define_config_rule, 62
define_configuration_aggregator, 62
define_configuration_recorder, 62
define_configuration_set, 187, 189, 241
define_configuration_set_event_destination, 187, 189, 241
delete_configuration_set_tracking_options,
delete_configuration_template, 104
delete_conformance_pack, 62
delete_connection, 11, 73, 125
delete_constraint, 235
delete_cors_configuration, 13
delete_cost_category_definition, 68
delete_crawler, 125
delete_custom_availability_zone, 198
delete_custom_key_store, 146
delete_customer_gateway, 86
delete_dashboard, 193
delete_dashboards, 46
delete_data_set, 193
delete_data_source, 158, 193
delete_database, 125
delete_dataset, 57, 178
delete_dataset_group, 178
delete_db_cluster, 78, 168, 198
delete_db_cluster_endpoint, 198
delete_db_cluster_parameter_group, 78, 168, 198
delete_db_cluster_snapshot, 78, 168, 198
delete_db_instance, 78, 168, 198
delete_db_instance_automated_backup, 198
delete_db_parameter_group, 168, 198
delete_db_proxy, 198
delete_db_security_group, 198
delete_db_snapshot, 198
delete_db_subnet_group, 78, 168, 198
delete_dedicated_ip_pool, 187
delete_delivery_channel, 62
delete_delivery_stream, 115
delete_deployment, 8, 13
delete_destination, 50
delete_detector, 128
delete_dev_endpoint, 125
delete_dhcp_options, 86
delete_direct_connect_gateway, 73
delete_direct_connect_gateway_association, 73
delete_direct_connect_gateway_association_proposal, 73
delete_directory, 32, 75
delete_disk, 155
delete_disk_snapshot, 155
delete_distribution, 37
delete_document, 252
delete_document_classifier, 58
delete_documentation_part, 8
delete_documentation_version, 8
delete_domain, 42, 155, 224, 247
delete_domain_entry, 155
delete_domain_name, 8, 13
delete_authorized_only_internet_gateway, 86
delete_elasticsearch_domain, 106
delete_elasticsearch_service_role, 106
delete_email_channel, 184
delete_email_identity, 187
delete_email_message, 184
delete_endpoint, 58, 184, 224, 249
delete_endpoint_config, 224
delete_endpoint_group, 123
delete_entity_recognizer, 59
delete_environment_configuration, 104
delete_evaluation, 158
delete_evaluation_results, 62
delete_event_bus, 48, 114
delete_event_source_mapping, 148
delete_event_stream, 184
delete_eventSubscription, 168, 198, 203
delete_event tracker, 178
delete_experiment, 224
delete_expression, 42
delete_faces, 206
delete_facet, 32
delete_fargate_profile, 100
delete_field_level_encryption_config, 37
delete_field_level_encryption_profile, 37
delete_file_share, 257
delete_file system, 98, 118
delete_filter, 128
delete_fleets, 86
delete_flow_definition, 224
delete_flow_logs, 86
delete_fpga_image, 86
delete_function, 148
delete_function_concurrency, 148
delete_function_event_invoke_config, 148
delete_gateway, 257
delete_gateway_response, 8
delete_gcm_channel, 184
delete_geo_match_set, 270, 272
delete_global_cluster, 198
delete_group, 54, 133, 193, 208
delete_group_membership, 193
delete_group_policy, 133
delete_hapg, 39
delete_health_check, 214
delete_hit, 166
delete_hosted_zone, 214
delete_hsm, 39, 40
delete_hsm_client_certificate, 203
delete_hsm_configuration, 203
delete_iam_policy_assignment, 193
delete_identities, 52
delete_identity, 241
delete_identity_policy, 241
delete_identity_pool, 52
delete_identity_provider, 54
delete_imported_key_material, 146
delete_index_field, 42
delete_insight, 231
delete_insight_rules, 46
delete_installation_media, 199
delete_instance, 155, 171
delete_instance_profile, 133
delete_instance_snapshot, 155
delete_integration, 8, 13
delete_integration_response, 8, 13
delete_intent, 150
delete_intent_version, 150
delete_interconnect, 73
delete_internet_gateway, 86
delete_inventory, 252
delete_invitations, 128, 231
delete_ip_set, 128, 270, 272
delete_item, 80
delete_job, 125
delete_job_queue, 29
delete_journey, 184
delete_key_pair, 86, 155
delete_known_host_keys, 155
delete_lag, 73
delete_launch_configuration, 22
delete_launch_template, 86
delete_launch_template_versions, 86
delete_layer, 171
delete_layer_version, 148
delete_lexicon, 190
delete_license_configuration, 153
delete_lifecycle_hook, 22
delete_lifecycle_policy, 77, 94
delete_listener, 110, 123
delete_load_balancer, 108, 110, 155
delete_load_balancer_listeners, 108
delete_load_balancer_policy, 108
delete_load_balancer_tls_certificate, 155
delete_local_gateway_route, 86
delete_local_gateway_route_table_vpc_association, 86
delete_log_group, 50
delete_log_pattern, 17
delete_log_stream, 50
delete_log_subscription, 75
delete_logging_configuration, 270, 272
delete_login_profile, 133
delete_luna_client, 39
delete_maintenance_window, 252
delete_members, 128, 231
delete_mesh, 19
delete_message, 251
delete_message_batch, 251
delete_method, 8
delete_method_response, 8
delete_metric_filter, 50
delete_ml_model, 158
delete_ml_transform, 125
delete_model, 8, 13, 224
delete_model_package, 224
delete_monitoring_schedule, 224
delete_mount_target, 98
delete_named_query, 21
delete_namespace, 237
delete_nat_gateway, 86
delete_network_acl, 86
delete_network_acl_entry, 86
delete_network_interface, 86
delete_network_interface_permission, 86
delete_nodegroup, 100
delete_notebook_instance, 224
delete_notebook_instance_lifecycle_config, 224
delete_notification, 31
delete_notification_channel, 117
delete_notification_configuration, 22
delete_object, 32, 220
delete_object_tagging, 220
delete_objects, 220
delete_open_id_connect_provider, 133
delete_option_group, 199
delete_organization, 176
delete_organization_config_rule, 62
delete_organization_conformance_pack, 62
delete_organizational_unit, 176
delete_parameter, 252
delete_parameter_group, 71
delete_parameters, 252
delete_partition, 125
delete_partner_event_source, 48, 114
delete_patch_baseline, 252
delete_pending_aggregation_request, 62
delete_permission, 7
delete_permission_policy, 270, 272
delete_pipeline, 69
delete_placement_group, 86
delete_platform_application, 249
delete_platform_version, 104
delete_policy, 22, 117, 134, 176
delete_policy_version, 134
delete_portfolio, 235
delete_portfolio_share, 235
delete_product, 235
delete_protection, 246
delete_provisioned_concurrency_config, 148
delete_provisioned_product_plan, 235
delete_provisioning_artifact, 235
delete_public_access_block, 220, 222
delete_public_key, 37
delete_publishing_destination, 128
delete_push_template, 184
delete_qualification_type, 166
delete_query_logging_config, 214
delete_queue, 251
deletequeued_reserved_instances, 86
delete_rate_based_rule, 270, 272
delete_realtime_endpoint, 158
delete_receipt_filter, 241
delete_receipt_rule, 241
delete_receipt_rule_set, 241
delete_recovery_point, 26
delete_regex_match_set, 270, 272
delete_regex_pattern_set, 270, 272
delete_relational_database, 155
delete_relational_database_snapshot, 155
delete_remediation_configuration, 62
delete_remediation_exceptions, 62
delete_replication_group, 102
delete_report_definition, 102
delete_repository, 94
delete_repository_policy, 94
delete_request_validator, 8
delete Resolver_endpoint, 218
delete Resolver_rule, 218
delete resource, 9
delete_resource_config, 62
delete_resource_data_sync, 253
delete_resource_policy, 50, 125, 229
delete resource_server, 54
delete_resource_share, 196
delete rest_api, 9
delete_retention_configuration, 62
delete retention_policy, 50
delete_reusable_delegation_set, 214
delete role, 134
delete role_permissions_boundary, 134
delete role_policy, 134
delete route, 13, 19, 86
delete route_response, 13
delete route_settings, 13
delete route_table, 86
delete rule, 48, 110, 114, 270, 272
delete rule_group, 270, 272
delete_saml_provider, 134
delete scaling_plan, 24
delete scaling_policy, 16
delete scheduled_action, 16, 22, 203
delete schema, 32, 178
delete secret, 229
delete security_configuration, 112, 125
delete security_group, 86
delete segment, 184
delete server, 174
delete_server_certificate, 134
delete service, 96, 237
delete service action, 235
delete_service_linked_role, 134
delete_service_quota_increase_request_from_template, 239
delete_service_specific_credential, 134
delete_session, 152
delete_signing_certificate, 134
delete_size_constraint_set, 270, 272
delete_slot_type, 150
delete_slot_type_version, 150
delete_sms_channel, 184
delete_sms_template, 184
delete_snapshot, 75, 86, 102
delete_snapshot_copy_grant, 203
delete_snapshot_schedule, 203, 257
delete_solution, 178
delete_spot_datafeed_subscription, 86
delete_sql_injection_match_set, 270, 272
delete_ssh_public_key, 134
delete_stack, 35, 171
delete_stack_instances, 35
delete_stack_set, 35
delete_stage, 9, 13
delete_state_machine, 243
delete_stream_processor, 206
delete_streaming_distribution, 37
delete_subnet, 86
delete_subnet_group, 71
delete_subscriber, 31
delete_subscription, 246
delete_subscription_filter, 50
delete_suggester, 42
delete_table, 80, 125
delete_table_version, 125
delete_tag_option, 235
delete_tags, 22, 86, 98, 158, 164, 204, 224
delete_tags_for_domain, 216
delete_tape, 257
delete_tape_archive, 257
delete_target_group, 110
delete_task_set, 96
delete_template, 193, 241
delete_template_alias, 193
delete_terminology, 268
delete_threat_intel_set, 128
delete_topic, 249
delete_traffic_mirror_filter, 86
delete_traffic_mirror_filter_rule, 86
delete_traffic_mirror_session, 86
delete_traffic_mirror_target, 86
delete_traffic_policy, 214
delete_traffic_policy_instance, 214
delete_trail, 45
delete_transcription_job, 267
delete_transit_gateway, 86
delete_transit_gateway_multicast_domain, 86
delete_transit_gateway_peering_attachment, 86
delete_transit_gateway_route, 86
delete_transit_gateway_route_table, 86
delete_transit_gateway_vpc_attachment, 86
delete_trial, 224
delete_trial_component, 224
delete_trigger, 125
delete_trust, 75
delete_typedefed_link_facet, 32
delete_usage_plan, 9
delete_usage_plan_key, 9
delete_user, 54, 65, 134, 164, 193
delete_user_attributes, 54
delete_user_by_principal_id, 193
delete_user_defined_function, 125
delete_user_endpoints, 184
delete_user_permissions_boundary, 134
delete_user_policy, 134
delete_user_pool, 54
delete_user_pool_client, 54
delete_user_pool_domain, 55
delete_user_profile, 171, 224
delete_utterances, 150
delete_vault, 120
delete_vault_access_policy, 120
delete_vault_notifications, 120
delete_verified_email_address, 241
delete_virtual_interface, 73
delete_virtual_mfa_device, 134
delete_virtual_node, 19
delete_virtual_router, 19
delete_virtual_service, 19
delete_vocabulary, 267
delete_vocabulary_filter, 267
delete_voice_channel, 184
delete_voice_template, 184
delete_volume, 86, 257
delete_vpc, 86
delete_vpc_association_authorization, 214
delete_vpc_endpoint_connection_notifications, deregister_type, 35
delete_vpc_endpoint_service_configurations, 86
delete_vpc_endpoints, 86
delete_vpc_link, 9
delete_vpc_peering_connection, 87
delete_vpn_connection, 87
delete_vpn_connection_route, 87
delete_vpn_gateway, 87
delete_web_acl, 270, 272
delete_work_group, 21
delete_worker_block, 166
delete_workflow, 125
delete_workteam, 224
delete_xss_match_set, 270, 272
deliver_config_snapshot, 62
deprecate_activity_type, 264
deprecate_domain, 264
deprecate_workflow_type, 264
deprovision_byoip_cidr, 87
deregister_certificate, 75
deregister_container_instance, 96
deregister_db_proxy_targets, 199
deregister_ecs_cluster, 171
deregister_elastic_ip, 171
deregister_event_topic, 75
deregister_image, 87
deregister_instance, 171, 237
deregister_instances_from_load_balancer, 108
deregister_job_definition, 29
deregister_managed_instance, 253
deregister_patch_baseline_for_patch_group, 253
deregister_rds_db_instance, 171
deregister_scalable_target, 16
deregister_stream_consumer, 140
deregister_target_from_maintenance_window, 253
deregister_targets, 110
deregister_task_definition, 96
deregister_task_from_maintenance_window,
describe_configuration_revision, 139, 164
describe_configuration_set, 241
describe_configuration_settings, 104
describe_conformance_pack_compliance, 62
describe_conformance_pack_status, 62
describe_connection_loa, 73
describe_connections, 73
describe_connections_on_interconnect, 73
describe_constraint, 235
describe_container_instances, 96
describe_continuous_backups, 80
describe_contributor_insights, 80
describe_conversion_tasks, 87
describe_copy_product_status, 235
describe_cost_category_definition, 68
describe_create_account_status, 176
describe_cross_account_access_role, 137
describe_custom_availability_zones, 199
describe_custom_key_stores, 146
describe_customer_gateways, 87
describe_dashboard, 193
describe_dashboard_permissions, 193
describe_data_repository_tasks, 118
describe_data_set, 194
describe_data_set_permissions, 194
describe_data_source, 194
describe_data_source_permissions, 194
describe_data_sources, 158
describe_dataset, 57, 178
describe_dataset_group, 178
describe_dataset_import_job, 178
describe_db_cluster_backtracks, 199
describe_db_cluster_endpoints, 199
describe_db_cluster_parameter_groups, 78, 168, 199
describe_db_cluster_parameters, 78, 168, 199
describe_db_cluster_snapshot_attributes, 78, 168, 199
describe_db_cluster_snapshots, 78, 168, 199
describe_db_clusters, 78, 168, 199
describe_db_engine_versions, 78, 168, 199
describe_db_instance_automated_backups, 199
describe_db_instances, 78, 168, 199
describe_db_log_files, 199
describe_db_parameter_groups, 168, 199
describe_db_parameters, 168, 199
describe_db_proxies, 199
describe_db_proxy_target_groups, 199
describe_db_proxy_targets, 199
describe_db_security_groups, 199
describe_db_snapshot_attributes, 199
describe_db_snapshots, 199
describe_db_subnet_groups, 78, 168, 199
describe_default_cluster_parameters, 204
describe_default_parameters, 71
describe_delivery_channel_status, 63
describe_delivery_channels, 63
describe_delivery_stream, 115
describe_deployments, 171
describe_destinations, 50
describe_dhcp_options, 87
describe_dimension_keys, 183
describe_direct_connect_gateway_association_proposals, 73
describe_direct_connect_gateway_associations, 73
describe_direct_connect_gateway_attachments, 73
describe_direct_connect_gateways, 73
describe_directories, 75
describe_document, 253
describe_document_classification_job, 59
describe_document_classifier, 59
describe_document_permission, 253
describe_domain, 225, 264
describe_domain_controllers, 75
describe_domain_endpoint_options, 42
describe_domains, 42
describe_dominant_language_detection_job, 59
describe_drt_access, 246
describe_ecs_clusters, 171
describe_effective_instance_associations, 253
describe_identity_id_format, 87
describe_identity_pool, 52
describe_identity_pool_usage, 57
describe_identity_provider, 55
describe_identity_usage, 57
describe_image_attribute, 87
describe_image_scan_findings, 94
describe_images, 87, 94
describe_import_image_tasks, 87
describe_import_snapshot_tasks, 87
describe_index_fields, 42
describe_ingestion, 194
describe_insight_rules, 46
describe_installation_media, 199
describe_instance_associations_status, 253
describe_instance_attribute, 87
describe_instance_credit_specifications, 87
describe_instance_health, 108
describe_instance_information, 253
describe_instance_patch_states, 253
describe_instance_patch_states_for_patch_group, 253
describe_instance_patches, 253
describe_instance_status, 88
describe_instance_type_offerings, 88
describe_instance_types, 88
describe_instances, 88, 171
describe_instances_health, 104
describe_interconnect_loa, 73
describe_interconnects, 73
describe_internet_gateways, 88
describe_inventory_deletions, 253
describe_job, 120, 222
describe_job_definitions, 29
describe_job_flows, 112
describe_job_queues, 29
describe_jobs, 29
describe_key, 146
describe_key_pairs, 88
describe_key_phrases_detection_job, 59
describe_labeling_job, 225
describe_lags, 73
describe_launch_configurations, 22
describe_launch_template_versions, 88
describe_launch_templates, 88
describe_layers, 171
describe_ldaps_settings, 75
describe_lifecycle_configuration, 98
describe_lifecycle_hook_types, 22
describe_lifecycle_hooks, 22
describe_limits, 80, 140
describe_listener, 123
describe_listener_certificates, 110
describe_listeners, 110
describe_loa, 73
describe_load_balancer_attributes, 108, 110
describe_load_balancer_policies, 108
describe_load_balancer_policy_types, 108
describe_load_balancer_target_groups, 22
describe_load_balancers, 22, 108, 110
describe_load_based_auto_scaling, 171
describe_local_gateway_route_table_virtual_interface_groups, 88
describe_local_gateway_route_table_vpc_associations, 88
describe_local_gateway_route_tables, 88
describe_local_gateway_virtual_interface_groups, 88
describe_local_gateway_virtual_interfaces, 88
describe_local_gateways, 88
describe_locations, 73
describe_log_groups, 50
describe_log_pattern, 17
describe_log_streams, 50
describe_logging_status, 204
describe_luna_client, 39
describe_maintenance_start_time, 257
describe_maintenance_window_execution_task_invocations, 253
describe_maintenance_window_execution_tasks, 253
describe_maintenance_window_executions, 253
describe_maintenance_window_schedule, 253
describe_maintenance_window_targets, 253
describe_maintenance_window_tasks, 253
describe_maintenance_windows, 253
describe_maintenance_windows_for_target, 253
describe_mesh, 19
describe_metric_collection_types, 23
describe_metric_filters, 50
describe_ml_models, 158
describe_model, 225
describe_model_package, 225
describe_monitoring_schedule, 225
describe_mount_target_security_groups, 98
describe_mount_targets, 98
describe_moving_addresses, 88
describe_my_user_profile, 171
describe_nat_gateways, 88
describe_network_acls, 88
describe_network_interface_attribute, 88
describe_network_interface_permissions, 88
describe_network_interfaces, 88
describe_nfs_file_shares, 257
describe_node_association_status, 174
describe_node_configuration_options, 204
describe_nodegroup, 100
describe_notebook_instance, 225
describe_notebook_instance_lifecycle_config, 225
describe_notification_configurations, 23
describe_notifications_for_budget, 31
describe_objects, 69
describe_observation, 17
describe_operating_systems, 171
describe_ops_items, 253
describe_option_group_options, 199
describe_option_groups, 199
describe_orderable_cluster_options, 204
describe_orderable_db_instance_options, 78, 168, 199
describe_organization, 176
describe_organization_config_rule_statuses, 63
describe_organization_config_rules, 63
describe_organization_conformance_pack_statuses, 63
describe_organization_conformance Packs, 63
describe_organizational_unit, 176
describe_parameter_groups, 71
describe_parameters, 71, 253
describe_partner_event_source, 48, 114
describe_patch_bases, 253
describe_patch_group_state, 253
describe_patch_groups, 253
describe_patch_properties, 253
describe_pending_aggregation_requests, 63
describe_pending_maintenance_actions, 78, 168, 199
describe_permissions, 171
describe_phi_detection_job, 60
describe_pipelines, 69
describe_placement_groups, 88
describe_platform_version, 104
describe_policies, 23
describe_policy, 176
describe_portfolio, 225
describe_portfolio_share_status, 235
describe_prefix_lists, 88
describe_principal_id_format, 88
describe_problem, 17
describe_problem_observations, 17
describe_processing_job, 225
describe_product, 235
describe_product_as_admin, 235
describe_product_view, 235
describe_products, 231
describe_project_versions, 206
describe_projects, 206
describe_protected_resource, 26
describe_protection, 246
describe_provisioned_product, 235
describe_provisioned_product_plan, 235
describe_provisioning_artifact, 235
describe_provisioning_parameters, 235
describe_public_ipv4_pools, 88
describe_publishing_destination, 128
describe_queries, 50
describe_random, 171
describe_rds_db_instances, 171
describe_receipt_rule, 241
describe_receipt_rule_set, 241
describe_recipe, 178
describe_stack_resource, 35
describe_stack_resource_drifts, 35
describe_stack_resources, 35
describe_stack_set, 35
describe_stack_set_operation, 35
describe_stack_summary, 172
describe_stacks, 35, 172
describe_stale_security_groups, 88
describe_state_machine, 243
describe_state_machine_for_execution, 243
describe_step, 112
describe_storage, 204
describe_storedi_scsi_volumes, 257
describe_stream, 82, 140
describe_stream_consumer, 140
describe_stream_processor, 206
describe_stream_summary, 140
describe_subnet_groups, 71
describe_subnets, 88
describe_subscribed_workteam, 225
describe_subscribers_for_notification, 31
describe_subscription, 246
describe_subscription_filters, 50
describe_suggesters, 42
describe_table, 80
describe_table_replica_auto_scaling, 80
describe_table_restore_status, 204
describe_tag_option, 235
describe_tags, 23, 73, 88, 98, 108, 110, 158, 204
describe_tape_archives, 257
describe_tape_recovery_points, 257
describe_tapes, 257
describe_target_groupAttributes, 110
describe_target_groups, 110
describe_target_health, 110
describe_task_definition, 96
describe_task_sets, 97
describe_tasks, 96
describe_template, 194
describe_template_alias, 194
describe_template_permissions, 194
describe_termination_policy_types, 23
describe_text_translation_job, 268
describe_time_based_auto_scaling, 172
describe_time_to_live, 80
describe_topics_detection_job, 59
describe_traffic_mirror_filters, 88
describe_traffic_mirror_sessions, 88
describe_traffic_mirror_targets, 88
describe_trails, 45
describe_training_job, 225
describe_transform_job, 225
describe_transit_gateway_attachments, 89
describe_transit_gateway_multicast_domains, 89
describe_transit_gateway_peering_attachments, 89
describe_transit_gateway_route_tables, 89
describe_transit_gateway_vpc_attachments, 89
describe_transit_gateways, 89
describe_trial, 225
describe_trial_component, 225
describe_trusted_advisor_check_refresh_statuses, 263
describe_trusted_advisor_check_result, 263
describe_trusted_advisor_check_summaries, 263
describe_trusted_advisor_checks, 263
describe_trusts, 75
describe_type, 35
describe_type_registration, 35
describe_update, 100
describe_update_actions, 102
describe_upload_buffer, 257
describe_user, 65, 164, 194
describe_user_hierarchy_group, 65
describe_user_hierarchy_structure, 65
describe_user_import_job, 55
describe_user_pool, 55
describe_user_pool_client, 55
describe_user_pool_domain, 55
describe_user_profile, 225
describe_user_profiles, 172
describe_valid_db_instance_modifications, 168, 199
describe_vault, 120
describe_virtual_gateways, 73
describe_virtual_interfaces, 73
describe_virtual_node, 19
describe_virtual_router, 19
describe_virtual_service, 19
describe_voices, 190
describe_volume_attribute, 89
describe_volume_status, 89
describe_volumes, 89, 172
describe_volumes_modifications, 89
describe_vpc_attribute, 89
describe_vpc_classic_link, 89
describe_vpc_classic_link_dns_support, 89
describe_vpc_endpoint_connection_notifications, 89
describe_vpc_endpoint_connections, 89
describe_vpc_endpoint_service_configurations, 89
describe_vpc_endpoint_service_permissions, 89
describe_vpc_endpoint_services, 89
describe_vpcs, 89
describe_vpn_connections, 89
describe_vpn_gateways, 89
describe_vtl_devices, 257
describe_workflow_execution, 264
describe_workflow_type, 264
describe_working_storage, 257
describe_workteam, 225
detach_classic_link_vpc, 89
detach_disk, 155
detach_elastic_load_balancer, 172
detach_from_index, 32
detach_group_policy, 134
detach_instances, 23
detach_instances_from_load_balancer, 155
detach_internet_gateway, 89
detach_load_balancer_from_subnets, 108
detach_load_balancer_target_groups, 23
detach_load_balancers, 23
detach_network_interface, 89
detach_object, 32
detach_policy, 32, 176
detach_rule_policy, 134
detach_static_ip, 155
detachTyped_link, 32
detach_user_policy, 134
detach_volume, 89, 257
detach_vpn_gateway, 89
detect_custom_labels, 206
detect_document_text, 266
detect_dominant_language, 59
detect_entities, 59, 60
detect_entities_v2, 60
detect_faces, 206
detect_key_phrases, 59
detect_labels, 206
detect_moderation_labels, 206
detect_phi, 60
detect_sentiment, 59
detect_stack_drift, 35
detect_stack_resource_drift, 35
detect_stack_set_drift, 35
detect_syntax, 59
detect_text, 206
directconnect, 71
directoryservice, 74
disable_add_on, 155
disable_alarm_actions, 46
disable_availability_zones_for_load_balancer, 108
disable_aws_organizations_access, 235
disable_aws_service_access, 176
disable_directory, 32
disable_domain_auto_renew, 216
disable_domain_transfer_lock, 216
disable_ebs_encryption_by_default, 89
disable_enhanced_monitoring, 141
disable_fast_snapshot_restores, 89
disable_gateway, 257
disable_health_service_access_for_organization, 131
disable_import_findings_for_product, 231
disable_insight_rules, 46
disable_key, 146
disable_key_rotation, 146
disable_ldaps, 75
disable_logging, 204
disable_metrics_collection, 23
disable_policy_type, 176
disable_radius, 75
disable_rule, 48, 114
disable_security_hub, 231
disable_snapshot_copy, 204
disable_sso, 75
disable_transit_gateway_route_table_propagation, 89
disable_vgw_route_propagation, 89
disable_vpc_classic_link, 89
disable_vpc_classic_link_dns_support, 89
disable_vpc_classic_link_dns_support, 89
disassociate_address, 89
disassociate_admin_account, 117
disassociate_budget_from_resource, 235
disassociate_client_vpn_target_network, 89
disassociate_connection_from_lag, 73
disassociate_drt_log_bucket, 246
disassociate_drt_role, 246
disassociate_elastic_ip, 172
disassociate_from_master_account, 128, 231
disassociate_iam_instance_profile, 89
disassociate_kms_key, 50
disassociate_member_account, 159
disassociate_members, 128, 231
disassociate_node, 174
disassociate_principal_from_portfolio, 235
disassociate_product_from_portfolio, 235
disassociate_qualification_from_worker, 166
disassociate_resolver_endpoint_ip_address, 218
disassociate_resolver_rule, 218
disassociate_resource_share, 196
disassociate_resource_share_permission, 196
disassociate_route_table, 89
disassociate_s3_resources, 159
disassociate_service_action_from_provisioning_artifact, 235
disassociate_servicequota_template, 239
disassociate_subnet_cidr_block, 89
disassociate_tag_option_from_resource, 235
disassociate_transit_gateway_multicast_domain, 89
disassociate_transit_gateway_route_table, 89
disassociate_trial_component, 225
disassociate_vpc_cidr_block, 89
disassociate_vpc_from_hosted_zone, 214
disassociate_web_acl, 272
disconnect_custom_key_store, 146
discover_input_schema, 142, 144
discover_instances, 237
discover_poll_endpoint, 97
dlm, 76
docdb, 77
domain_metadata, 247
download_db_log_file_portion, 199
download_default_key_pair, 155
dynamodb, 79
dynamodbstreams, 82
e2, 83
e2instanceconnect, 92
ecr, 93
ecs, 95
efs, 97
eks, 99
elasticache, 101
elasticbeanstalk, 103
elasticsearchervice, 105
elb, 107
e1, 109
emr, 111
enable_add_on, 155
enable_alarm_actions, 46
enable_all_features, 176
enable_availability_zones_for_load_balancer, 108
enable_aws_organizations_access, 235
enable_aws_service_access, 176
enable_directory, 32
enable_domain_auto_renew, 216
enable_domain_transfer_lock, 216
enable_encryption_by_default, 89
enable_enhanced_monitoring, 141
enable_fast_snapshot_restores, 89
enable_health_service_access_for_organization, 131
enable_import_findings_for_product, 231
enable_insight_rules, 46
enable_key, 146
enable_key_rotation, 146
enable ldaps, 75
enable logging, 204
enable metrics collection, 23
enable mfa device, 134
enable policy type, 176
enable radius, 75
enable rule, 48, 114
enable security hub, 231
enable sharing with aws organization, 196
enable snapshot copy, 204
enable sso, 75
enable transit gateway route table propagation, 89
enable vgw route propagation, 89
enable volume io, 89
enable vpc classic link, 89
enable vpc classic link dns support, 90
encrypt, 146
enter standby, 23
estimate template cost, 35
evaluate expression, 69
eventbridge, 113
execute change set, 35
execute policy, 23
execute provisioned product plan, 235
execute provisioned product service action, 235
execute sql, 201
execute statement, 201
exit standby, 23
export backup plan template, 26
export certificate, 5
export client vpn client certificate revocation, 90
export client vpn client configuration, 90
export image, 90
export server engine attribute, 174
export snapshot, 155
export transit gateway routes, 90
failover db cluster, 78, 168, 199
filter log events, 50
firehose, 115
flush stage authorizers cache, 9
flush stage cache, 9
fms, 116
forget device, 55
forgot password, 55
fsx, 117
generate client certificate, 9
generate credential report, 134
generate data key, 146
generate data key pair, 146
generate data key pair without plaintext, 146
generate data key without plaintext, 146
generate data set, 160
generate organizations access report, 134
generate random, 146
generate service last accessed details, 134
generate service last_accessed_details, 134
get access key info, 260
generate secret, 187
generate service last_accessed_details, 134
get access key info, 260
get access key last used, 134
get access point, 222
get access point policy, 222
generate service last_accessed_details, 134
get account, 9, 187
generate service last_accessed_details, 134
get account authorization details, 134
get account balance, 166
get account limit, 214
get account password policy, 134
get account sending enabled, 241
get account settings, 148
get account summary, 134
get active names, 155
get activity task, 243
get adn account, 117
generate service last_accessed_details, 134
get aggregate compliance details by config rule, 63
get aggregate config rule compliance summary, 63
get aggregate discovered resource counts, 63
get aggregate resource config, 63
get alias, 148
get api, 13
get api key, 9
get api keys, 9
get api mapping, 13
get api mappings, 13
get apis, 13
get_apns_channel, 184
get_apns_sandbox_channel, 184
get_apns_voip_channel, 184
get_apns_voip_sandbox_channel, 184
get_app, 184
get_application, 233
get_application_date_range_kpi, 184
get_application_policy, 233
get_application_settings, 184
get_applied_schema_version, 32
get_apps, 184
get_assessment_report, 137
get_assignment, 166
get_association_for_service_quota_template, 239
get_attribute_values, 192
get_attributes, 247
get_authorization_token, 94
get_authorizer, 9, 13
get_authorizers, 9, 13
get_auto_snapshots, 155
get_automation_execution, 253
get_aws_default_service_quota, 239
get_awsorganizations_access_status, 235
get_backup_plan, 26
get_backup_plan_from_json, 26
get_backup_plan_from_template, 26
get_backup_selection, 26
get_backup_vault_access_policy, 26
get_backup_vault_notifications, 26
get_baidu_channel, 184
get_base_path_mapping, 9
get_base_path_mappings, 9
get_automation_execution, 253
get_calendar_state, 253
get_caller_identity, 260
get_campaign, 184
get_campaign_activities, 184
get_campaign_date_range_kpi, 184
get_campaign_version, 184
get_campaign_versions, 184
get_campaigns, 184
get_change, 214
get_change_token, 270, 272
get_change, 214
get_campaigns, 184
get_capacity_reservation_usage, 90
get_catalog_import_status, 125
get_automation_execution, 253
get_automation_execution, 253
get_celebrity_info, 206
get_celebrity_recognition, 206
get_certificate, 5, 7
get_certificate_authority_certificate, 7
get_certificate_authority_csr, 7
get_change, 214
get_change_token, 270, 272
get_change_token_status, 270, 273
get_channels, 184
get_checker_ip_ranges, 214
get_classifier, 125
get_classifiers, 125
get_client_certificate, 9
get_client_certificates, 9
get_cloud_formation_stack_records, 156
get_cloud_formation_template, 233
get_cloud_front_origin_access_identity, 37
get_cloud_front_origin_access_identity_config, 37
get_cluster_credentials, 204
get_cognito_events, 57
get_coip_pool_usage, 90
get_command_invocation, 253
get_compatible_elasticsearch_versions, 106
get_compliance_detail, 117
get_compliance_details_by_config_rule, 63
get_compliance_details_by_resource, 63
get_compliance_summary, 213
get_compliance_summary_by_config_rule, 63
get_compliance_summary_by_resource_type, 63
get_conformance_pack_compliance_details, 63
get_conformance_pack_compliance_summary, 63
get_connection, 11, 125
get_connection_status, 253
get_connections, 125
get_console_output, 90
get_console_screenshot, 90
get_contact_attributes, 65
get_contact_reachability_status, 216
get_content_moderation, 206
get_context_keys_for_custom_policy, 134
get_context_keys_for_principal_policy, 134
get_cost_and_usage, 68
get_cost_and_usage_with_resources, 68
get_cost_forecast, 68
get_crawler, 125
get_crawler_metrics, 125
get_crawlers, 125
get_credential_report, 134
get_credentials_for_identity, 52
get_csv_header, 55
get_current_metric_data, 65
get_custom_verification_email_template, 241
get_dashboard, 46
get_dashboard_embed_url, 194
get_data_catalog_encryption_settings, 125
get_data_retrieval_policy, 120
get_data_source, 158
get_database, 125
get_databases, 125
get_dataflow_graph, 125
get_dedicated_ip, 187
get_dedicated_ips, 188
get_default_credit_specification, 90
get_default_patch_baseline, 253
get_deliverability_dashboard_options, 188
get_deliverability_test_report, 188
get_deployable_patch_snapshot_for_instance, 253
get_deployment, 9, 13
get_deployments, 9, 13
get_detector, 128
get_dev_endpoint, 125
get_dev_endpoints, 125
get_device, 55
get_dimension_values, 68
get_directory, 32
get_directory_limits, 75
get_discovered_resource_counts, 63
get_disk, 156
get_disk_snapshot, 156
get_disk_snapshots, 156
get_disks, 156
get_distribution, 37
get_distribution_config, 37
get_document, 253
get_document_analysis, 266
INDEX

get_instance_snapshot, 156
get_instance_snapshots, 156
get_instance_state, 156
get_instances, 156
get_instances_health_status, 237
get_integration, 9, 13
get_integration_response, 9, 13
get_integrations, 13
get_intent, 151
get_intent_versions, 151
get_intents, 151
get_invalidation, 37
get_inventory, 253
get_inventory_schema, 253
get_invitations_count, 128, 231
get_ip_set, 128, 270, 273
get_item, 80
get_job, 125
get_job_bookmark, 125
get_job_output, 120
get_job_run, 125
get_job_runs, 125
get_jobs, 125
get_journey, 185
get_journey_date_range_kpi, 185
get_journey_execution_activity_metrics, 185
get_journey_execution_metrics, 185
get_key_pair, 156
get_key_pairs, 156
get_key_policy, 146
get_label_rotation_status, 146
get_label_detection, 206
get_launch_template_data, 90
get_layer_version, 148
get_layer_version_by_arn, 148
get_layer_version_policy, 148
get_lexicon, 190
get_license_configuration, 153
getLifecyclepolicies, 77
getLifecycle_policy, 77, 94
getLifecycle_policy_preview, 94
get_link_attributes, 32
get_load_balancer, 156
get_load_balancer_metric_data, 156
get_load_balancer_ttls_certificates, 156
get_log_events, 50
get_log_group_fields, 50
get_log_record, 50
get_logging_configuration, 270, 273
get_login_profile, 134
get_maintenance_window, 253
get_maintenance_window_execution, 253
get_maintenance_window_execution_task, 254
get_maintenance_window_execution_task_invocation, 254
get_maintenance_window_task, 254
get_mapping, 125
get_master_account, 128, 231
get_members, 128, 231
get_method, 9
get_method_response, 9
get_metric_data, 46, 65
get_metric_statistics, 46
get_metric_widget_image, 46
get_ml_model, 158
get_ml_task_run, 125
get_ml_task_runs, 125
get_ml_transform, 125
get_ml_transforms, 125
get_model, 9, 13
get_model_template, 9, 13
get_models, 9, 13
get_named_query, 21
get_namespace, 237
get_notification_channel, 117
get_object, 220
get_object_acl, 220
get_object_attributes, 32
get_object_information, 33
get_object_legal_hold, 220
get_object_license_configuration, 220
get_object_retention, 220
get_object_tagging, 220
get_object_torrent, 220
get_open_id_connect_provider, 134
get_open_id_token, 52
get_open_id_token_for_developer_identity, 52
get_operation, 156, 237
get_operation_detail, 216
get_operations, 156
INDEX

get_operations_for_resource, 156
get_ops_item, 254
get_ops_summary, 254
get_organization_config_rule_detailed_status, 63
get_organization_conformance_pack_detailed_status, 63
get_organizations_access_report, 134
get_parameter, 254
get_parameter_history, 254
get_parameters, 254
get_parameters_by_path, 254
get_parameters_for_import, 146
get_partition, 125
get_partitions, 125
get_password_data, 90
get_patch_baseline, 254
get_patch_baseline_for_patch_group, 254
get_permission, 196
get_permission_policy, 270, 273
get_person_tracking, 206
get_personalized_ranking, 180
get_pipeline_definition, 69
get_plan, 125
get_platform_application_attributes, 249
get_policy, 117, 134, 148
get_policy_version, 134
get_products, 192
get_protection_status, 117
get_provisioned_concurrency_config, 148
get_public_access_block, 220, 222
get_public_key, 37, 147
get_public_key_config, 37
get_push_template, 185
get_qualification_score, 166
get_qualification_type, 166
get_query_execution, 21
get_query_logging_config, 214
get_query_results, 21, 50
get_queue_attributes, 251
get_queue_url, 251
get_random_password, 229
get_rate_based_rule, 270, 273
get_rate_based_rule_managed_keys, 270, 273
get_recommendations, 180
get_records, 82, 141
get_recovery_point_restore_metadata, 26
get_regex_match_set, 270, 273
get_regex_pattern_set, 270, 273
get_regions, 156
get_relational_database, 156
get_relational_database_blueprints, 156
get_relational_database_bundles, 156
get_relational_database_events, 156
get_relational_database_log_events, 156
get_relational_database_log_streams, 156
get_relational_database_master_user_password, 156
get_relational_database_metric_data, 156
get_relational_database_parameters, 156
get_relational_database_snapshot, 156
get_relational_database_snapshots, 156
get_relational_databases, 156
get_repository_policy, 94
get_request_validator, 9
get_request_validators, 9
get_requested_service_quota_change, 239
get_reservation_coverage, 68
get_reservation_purchase_recommendation, 68
get_reservation_utilization, 68
get_reserved_instances_exchange_quote, 90
get_reserved_node_exchange_offersings, 204
get_resolver_endpoint, 218
get_resolver_rule, 218
get_resolver_rule_association, 218
get_resolver_rule_policy, 218
get_resource, 9
get_resource_config_history, 63
get_resource_metrics, 183
get_resource_policies, 196
get_resource_policy, 125, 229
get_resource_share_associations, 196
INDEX

get_resource_share_invitations, 196
get_resource_shares, 196
get_resources, 9, 213
get_rest_api, 9
get_rest_apis, 9
get_reusable_delegation_set, 214
get_reusable_delegation_set_limit, 214
get_rightsizing_recommendation, 68
get_role, 134
get_role_policy, 134
get_route, 13
get_route_response, 13
get_routes, 13
get_rule, 270, 273
get_rule_group, 270, 273
get_saml_provider, 134
get_sampled_requests, 270, 273
get_savings_plans_coverage, 68
get_savings_plans_purchase_recommendation, 68
get_savings_plans_utilization, 68
get_savings_plans_utilization_details, 68
get_scaling_plan_resource_forecast_data, 24
get_schema_as_json, 33
get_sdk, 9
get_sdk_type, 9
get_search_suggestions, 225
get_secret_value, 229
get_security_configuration, 126
get_security_configurations, 126
get_segment, 185
get_segment_export_jobs, 185
get_segment_import_jobs, 185
get_segment_version, 185
get_segment_versions, 185
get_segments, 185
get_send_quota, 241
get_send_statistics, 241
get_server_certificate, 134
get_service, 237
get_service_last_accessed_details, 134
get_service_last_accessed_details_with_entit
134
get_service_linked_role_deletion_status,
134
get_service_quota, 239
get_service_quota_increase_request_from_template, 239
get_service_setting, 254
get_service_settings, 153
get_session, 152
get_session_token, 260
get_shard_iterator, 82, 141
get_signing_certificate, 55
get_size_constraint_set, 270, 273
get_slot_type, 151
get_slot_type_versions, 151
get_slot_types, 151
get_sms_attributes, 249
get_sms_channel, 185
get_sms_template, 185
get_snapshot_limits, 75
get_solution_metrics, 178
get_speech_synthesis_task, 190
get_sql_injection_match_set, 270, 273
get_ssh_public_key, 134
get_stack_policy, 35
get_stage, 9, 13
get_stages, 9, 14
get_static_ip, 156
get_static_ips, 156
get_streaming_distribution, 37
get_streaming_distribution_config, 37
get_subscription_attributes, 249
get_subscription_state, 246
get_supported_resource_types, 26
get_table, 126
get_table_version, 126
get_table_versions, 126
get_tables, 126
get_tag_keys, 213
get_tag_values, 213
get_tags, 9, 14, 68, 126, 208
get_telemetry_metadata, 137
get_template, 35, 241
get_template_summary, 35
get_terminology, 268
get_threat_intel_set, 128
get_topic_attributes, 249
get_traffic_policy, 214
get_traffic_policy_instance, 214
get_traffic_policy_instance_count, 214
INDEX

get_trail, 45
get_trail_status, 45
get_transcription_job, 267
get_transit_gateway_attachment_propagations, 90
get_transit_gateway_multicast_domain_associations, 90
get_transit_gateway_route_table_associations, 90
get_transit_gateway_route_table_propagations, 90
get_trigger, 126
get_triggers, 126
get_typed_link_facet_information, 33
get_ui_customization, 55
get_upgrade_history, 106
get_upgrade_status, 106
get_usage, 10
get_usage_forecast, 68
get_usage_plan, 10
get_usage_plan_key, 10
get_usage_plan_keys, 10
get_usage_plans, 10
get_user, 55, 134
get_user_attribute_verification_code, 55
get_user_defined_function, 126
get_user_defined_functions, 126
get_user_endpoints, 185
get_user_policy, 134
get_user_pool_mfa_config, 55
get_utterances_view, 151
get_vault_access_policy, 120
get_vault_lock, 120
get_vault_notifications, 120
get_vocabulary, 267
get_vocabulary_filter, 267
get_voice_channel, 185
get_voice_template, 185
get_vpc_link, 10
get_vpc_links, 10
get_web_acl, 270, 273
get_web_acl_for_resource, 273
get_work_group, 21
get_workflow, 126
get_workflow_execution_history, 264
get_workflow_run, 126
get_workflow_run_properties, 126
get_workflow_runs, 126
get_xss_match_set, 270, 273
glacier, 119
globalsign_out, 55
globalaccelerator, 121
guide, 123
grant_access, 172
head_bucket, 220
head_object, 220
health, 129
iam, 131
import_api, 14
import_api_keys, 10
import_catalog_to_glue, 126
import_certificate, 5
import_certificate_authority_certificate, 7
import_client_vpn_client_certificate_revocation_list, 90
import_documentation_parts, 10
import_image, 90
import_installation_media, 199
import_instance, 90
import_key_material, 147
import_key_pair, 90, 156
import_rest_api, 10
import_snapshot, 90
import_terminology, 268
import_volume, 90
increase_replica_count, 102
increase_replication_factor, 71
increase_stream_retention_period, 141
index_documents, 42
index_faces, 206
infer_icd10cm, 60
infer_rx_norm, 60
initialize_cluster, 40
initiate_auth, 55
initiate_job, 120
initiate_layer_upload, 95
initiate_multipart_upload, 120
initiate_vault_lock, 120
inspector, 136
invite_account_to_organization, 176
invite_members, 128, 231
invoke, 148
invoke_async, 148
invoke_endpoint, 227
is_vpc_peered, 156
issue_certificate, 7
join_domain, 257
kafka, 138
kinesis, 140
kinesisanalytics, 141
kinesisanalyticsv2, 143
kms, 144
label_parameter_version, 254
lambda, 147
leave_organization, 176
lexmodelbuildingservice, 149
lexruntimeservice, 151
licensemanager, 152
lightsail, 154
list_accelerators, 123
list_accepted_portfolio_shares, 235
list_access_keys, 134
list_access_points, 222
list_account_aliases, 135
list_account_settings, 97
list_accounts, 176
list_accounts_for_parent, 176
list_activated_rules_in_rule_group, 270, 273
list_activities, 243
list_activity_types, 264
list_aggregate_discovered_resources, 63
list_algorithms, 225
list_aliases, 147, 149
list_allowed_node_type_modifications, 102
list_application_dependencies, 233
list_application_snapshots, 144
list_application_versions, 233
list_applications, 17, 142, 144, 233
list_applied_schema_arns, 33
list_apps, 225
list_assessment_run_agents, 137
list_assessment_runs, 137
list_assessment_targets, 137
list_assessment_templates, 137
list_assignments_for_hit, 166
list_association_versions, 254
list_associations, 254
list_associations_for_license_configuration, 153
list_attached_group_policies, 135
list_attached_indices, 33
list_attached_role_policies, 135
list_attached_user_policies, 135
list_attacks, 246
list_attributes, 97
list_auto_ml_jobs, 225
list_available_solution_stacks, 104
list_available_zones, 39
list_aws_default_service_quotas, 239
list_aws_service_access_for_organization, 176
list_backup_jobs, 26
list_backup_plan_templates, 26
list_backup_plan_versions, 26
list_backup_plans, 26
list_backup_selections, 26
list_backup_vaults, 26
list_backups, 80
list_batch_inference_jobs, 178
list_bonus_payments, 166
list_bootstrap_actions, 112
list_brokers, 164
list_bucket_analytics_configurations, 220
list_bucket_inventory_configurations, 220
list_bucket_metrics_configurations, 220
list_buckets, 220
list_budgets_for_resource, 236
list_byte_match_sets, 270, 273
list_campaigns, 178
list_candidates_for_auto_ml_job, 225
list_certificateAuthorities, 7
list_certificates, 5, 75
list_change_sets, 35
list_children, 176
list_closed_workflow_executions, 264
list_cloud_front_origin_access_identities, 37
list_cluster_operations, 139
list_clusters, 97, 100, 112, 139
list_code_repositories, 225
list_collections, 206
list_command_invocations, 254
list_commands, 254
list_compilation_jobs, 225
list_compliance_items, 254
list_compliance_status, 117
list_compliance_summaries, 254
list_components, 17
list_configuration_revisions, 139, 164
list_configuration_sets, 188, 189, 241
list_configurations, 139, 165
list_constraints_for_portfolio, 236
list_contact_flows, 65
list_container_instances, 97
list_contributor_insights, 80
list_cost_category_definitions, 68
list_crawlers, 126
list_create_account_status, 176
list_custom_verification_email_templates, 241
list_dashboard_versions, 194
list_dashboards, 46, 194
list_data_sets, 194
list_data_sources, 194
list_dataset_groups, 178
list_dataset_import_jobs, 178
list_datasets, 57, 178
list_dead_letter_source_queues, 251
list_dedicated_ip_pools, 188
list_deliverability_test_reports, 188
list_delivery_streams, 115
list_detectors, 129
list_dev_endpoints, 126
list_development_schema_arns, 33
list_devices, 55
list_directories, 33
list_discovered_resources, 63
list_distributions, 37
list_distributions_by_web_acl_id, 37
list_document_classification_jobs, 59
list_document_classifiers, 59
list_document_versions, 254
list_documents, 254
list_domain_deliverability_campaigns, 188
list_domain_names, 42, 106
list_domains, 216, 225, 247, 264
list_dominant_language_detection_jobs,
INDEX

list_groups, 55, 135, 194, 208
list_groups_for_user, 135
list_handshakes_for_account, 176
list_handshakes_for_organization, 176
list_hapgs, 39
list_health_checks, 214
list_hi_ts, 166
list_hi_ts_for_qualification_type, 166
list_hosted_zones, 214
list_hosted_zones_by_name, 214
list_hours_of_operations, 65
list_hsms, 39
list_human_task_uis, 225
list_hyper_parameter_tuning_jobs, 225
list_iam_policy_assignments, 194
list_iam_policy_assignments_for_user, 194
list_identities, 52, 241
list_identity_policies, 241
list_identity_pool_usage, 57
list_identity_pools, 52
list_identity_providers, 55
list_images, 95
list_imports, 35
list_incoming.Typed_links, 33
list_index, 33
list_ingestions, 194
list_instance_fleets, 112
list_instance_groups, 112
list_instance_profiles, 135
list_instance_profiles_for_role, 135
list_instances, 112, 238
list_invalidations, 37
list_inventory_entries, 254
list_invitations, 129, 232
list_ip_routes, 75
list_ip_sets, 129, 270, 273
list_jobs, 29, 120, 126, 222
list_journeys, 185
list_key_phrases_detection_jobs, 59
list_key_policies, 147
list_keys, 147
list_labeling_jobs, 225
list_labeling_jobs_for_workteam, 225
list_launch_paths, 236
list_layer_versions, 149
list_layers, 149
list_lexicons, 190
list_license_configurations, 153
list_license_specifications_for_resource, 153
list_listeners, 123
list_local_disks, 257
list_log_pattern_sets, 17
list_log_patterns, 17
list_log_subscriptions, 75
list_logging_configurations, 270, 273
list_luna_clients, 39
list_managed_schema_arns, 33
list_member_accounts, 117, 159
list_members, 129, 232
list_mes, 19
list_metrics, 46
list_mfa_devices, 135
list_model_packages, 225
list_models, 225
list_monitoring_executions, 225
list_monitoring_schedules, 225
list_multipart_uploads, 120, 220
list_named_queries, 21
list_namespaces, 238
list_nodegroups, 100
list_nodes, 139
list_notebook_instance_lifecycle_configs, 225
list_notebook_instances, 225
list_object_attributes, 33
list_object_children, 33
list_object_parent_paths, 33
list_object_policies, 33
list_object_versions, 221
list_objects, 220
list_objects_v2, 220
list_open_id_connect_providers, 135
list_open_workflow_executions, 264
list_operations, 216, 238
list_organization_portfolio_access, 236
list_organizational_units_for_parent, 176
list_outgoingTyped_links, 33
list_parents, 176
list_partner_event_source_accounts, 48, 114
list_partner_event_sources, 48, 114
list_parts, 120, 221
list_pending_invitation_resources, 196
list_permissions, 7, 196
list_phi_detection_jobs, 60
list_phone_numbers, 65
list_phone_numbers_opted_out, 249
list_pipelines, 69
list_platform_applications, 249
list_platform_versions, 104
list_policies, 117, 135, 176
list_policies_for_target, 176
list_policies_granting_service_access, 135
list_policy_attachments, 33
list_policy_versions, 135
list_portfolio_access, 236
list_portfolios, 236
list_portfolios_for_product, 236
list_principals, 196
list_principals_for_portfolio, 236
list_problems, 17
list_processing_jobs, 226
list_protected_resources, 26
list_protections, 246
list_provisioned_capacity, 120
list_provisioned_concurrency_configs, 149
list_provisioned_product_plans, 236
list_provisioning_artifacts, 236
list_provisioning_artifacts_for_service_action, 236
list_public_keys, 37, 45
list_published_schema_arns, 33
list_publishing_destinations, 129
list_qualification_requests, 166
list_qualification_types, 166
list_query_executions, 21
list_query_logging_configs, 215
list_queue_tags, 251
list_queues, 65, 251
list_rate_based_rules, 270, 273
list_receipt_filters, 241
list_receipt_rule_sets, 241
list_recipes, 178
list_record_history, 236
list_records, 57
list_recovery_points_by_resource, 26
list_recovery_points_by_backup_vault, 26
list_regress_match_sets, 270, 273
list_regress_pattern_sets, 270, 273
list_requested_service_quota_change_history, 239
list_requested_service_quota_change_history_by_quota, 239
list_resolver_endpoint_ip_addresses, 218
list_resolver_endpoints, 218
list_resolver_rule_associations, 218
list_resolver_rules, 218
list_resource_compliance_summaries, 254
list_resource_data_sync, 254
list_resource_inventory, 153
list_resource_record_sets, 215
list_resource_servers, 55
list_resource_share_permissions, 196
list_resource_tags, 147
list_resourses, 196
list_resources, 236
list_resources_for_tag_option, 236
list_resources_for_web_acl, 273
list_restore_jobs, 26
list_retirable_grants, 147
list_reusable_delegation_sets, 215
list_review_policy_results_for_hit, 166
list_reviewable_his, 166
list_role_policies, 135
list_role_tags, 135
list_roles, 135
list_roots, 176
list_routes, 19
list_routing_profiles, 65
list_rule_groups, 270, 273
list_rule_names_by_target, 48, 114
list_rules, 48, 114, 270, 273
list_rules_packages, 137
list_s3_resources, 159
list_saml_providers, 135
list_schema_extensions, 75
list_schemas, 178
list_secret_version_ids, 229
list_secrets, 229
list_security_configurations, 112
list_security_profiles, 65
list_sentiment_detection_jobs, 59
list_server_certificates, 135
list_service_actions, 236
list_service_actions_for_provisioning_artifact, 236
list_service_quota_increase_requests_in_template, 239
list_service_quotas, 239
list_service_specific_credentials, 135
list_services, 97, 238, 239
list_shards, 141
list_signing_certificates, 135
list_size_constraint_sets, 270, 273
list_solution_versions, 178
list_solutions, 178
list_speech_synthesis_tasks, 190
list_sql_injection_match_sets, 270, 273
list_ssh_public_keys, 135
list_stack_instances, 35
list_stack_instances_for_provisioned_product, 236
list_stack_resources, 35
list_stack_set_operation_results, 35
list_stack_set_operations, 35
list_stack_sets, 35
list_stacks, 35
list_state_machines, 243
list_steps, 112
list_stream_consumers, 141
list_stream_processors, 206
list_streaming_distributions, 37
list_stores, 82, 141
list_subscribed_rule_groups, 270, 273
list_subscribed_workteams, 226
list_subscriptions, 249
list_subscriptions_by_topic, 249
list_tables, 80
list_tag_options, 236
list_tags, 7, 26, 40, 45, 71, 106, 149, 165, 172, 226
list_tags_for_certificate, 5
list_tags_for_delivery_stream, 115
list_tags_for_domain, 216
list_tags_for_resource, 17, 19, 21, 33, 37, 39, 46, 48, 52, 55, 59, 63, 65, 75, 77, 78, 95, 97, 100, 102, 104, 114, 117, 118, 129, 137, 139, 142, 144, 153, 168, 174, 176, 185, 188, 194, 199, 215, 218, 232, 243, 249, 254, 257,
list_tags_log_group, 51
list_tags_of_resource, 80
list_tags_stream_group, 51
list_tags_tag_group, 51
list_tasks, 97
list_template_aliases, 194
list_template_versions, 185, 194
list_templates, 185, 194, 241
list_termologies, 268
list_text_translation_jobs, 268
listthreat_intel_sets, 129
list_traffic_policy_versions, 215
list_traffic_policy_instances, 215
list_traffic_policy_instances_by_policy, 215
list_traffic_policy_instances_by_hosted_zone, 215
list_traffic_policies, 215
list_traffic_policy, 215
list_trials, 45
list_training_jobs, 226
list_training_jobs_for_hyper_parameter_tuning_jobs, 226
list_transcription_jobs, 267
list_transform_jobs, 226
list_trial_components, 226
list_trials, 226
list_triggers, 126
list_type_registrations, 35
list_type_versions, 35
list_terminologies, 268
list_type_translation_jobs, 268
list_typo_link_facet_attributes, 33
list_typo_link_facet_names, 33
list_types, 35
list_updates, 100
list_usage_for_license_configuration, 153
list_user_groups, 194
list_user_hierarchy_groups, 65
list_user_import_jobs, 55
list_user_policies, 135
list_user_pool_clients, 55
list_user_pools, 55
list_user_profiles, 226
list_user_tags, 135
list_users, 55, 65, 135, 165, 194
list_users_in_group, 55
list_vaults, 120
list_verified_email_addresses, 241
list_versions_by_function, 149
list_virtual_mfa_devices, 135
list_virtual_nodes, 19
list_virtual_routers, 19
list_virtual_services, 19
list_vocabularies, 267
list_vocabulary_filters, 267
list_volume_initiators, 257
list_volume_recovery_points, 257
list_volumes, 257
list_vpc_association_authorizations, 215
list_web_ac_ls, 270, 273
list_work_groups, 21
list_worker_blocks, 166
list_workers_with_qualification_type, 166
list_workflow_types, 264
list_workflows, 126
list_workteams, 226
list_xss_match_sets, 270, 273
lookup_developer_identity, 52
lookup_events, 45
lookup_policy, 33
machinelearning, 157
macie, 159
marketplacecommerceanalytics, 160
marketplaceentitlementservice, 161
marketplacemetering, 162
merge_developer_identities, 52
merge_shards, 141
meter_usage, 163
modify_cache_cluster, 102
modify_cache_parameter_group, 102
modify_cache_subnet_group, 102
modify_capacity_reservation, 90
modify_certificates, 199
modify_client_vpn_endpoint, 90
modify_cluster, 112, 204
modify_cluster_db_revision, 204
modify_cluster_iam_roles, 204
modify_cluster_maintenance, 204
modify_cluster_parameter_group, 204
modify_cluster_snapshot, 204
modify_cluster_snapshot_schedule, 204
modify_cluster_subnet_group, 204
modify_current_db_cluster_capacity, 199
modify_db_cluster, 79, 168, 199
modify_db_cluster_endpoint, 200
modify_db_cluster_parameter_group, 79, 168, 200
modify_db_cluster_snapshot_attribute, 79, 168, 200
modify_db_instance, 79, 168, 200
modify_db_parameter_group, 168, 200
modify_db_proxy, 200
modify_db_proxy_target_group, 200
modify_db_snapshot, 200
modify_db_snapshot_attribute, 200
modify_db_subnet_group, 79, 169, 200
modify_default_credit_specification, 90
modify_document_permission, 254
modify_ebs_default_kms_key_id, 90
modify_event_subscription, 169, 200, 204
modify_fleet, 90
modify_fpga_image_attribute, 90
modify_global_cluster, 200
modify_hapg, 39
modify_hosts, 90
modify_hsm, 39
modify_id_format, 90
modify_identity_id_format, 90
modify_image_attribute, 90
modify_instance_attribute, 90
modify_instance_capacity_reservation_attributes, 90
modify_instance_credit_specification, 90
modify_instance_event_start_time, 90
modify_instance_fleet, 112
modify_instance_groups, 112
modify_instance_metadata_options, 90
modify_instance_placement, 90
modify_launch_template, 90
modify_listener, 110
modify_load_balancer_attributes, 108
modify_luna_client, 39
modify_mount_target_security_groups, 98
modify_network_interface_attribute, 90
modify_option_group, 200
modify_replication_group, 102
modify_replication_group_shard_configuration, 102
modify_report_definition, 66
modify_reserved_instances, 90
modify_rule, 110
modify_scheduled_action, 204
modify_snapshot_attribute, 90
modify_snapshot_copy_retention_period, 204
modify_snapshot_schedule, 204
modify_spot_fleet_request, 90
modify_subnet_attribute, 90
modify_target_group, 110
modify_target_group_attributes, 110
modify_traffic_mirror_filter_network_services, 91
modify_traffic_mirror_filter_rule, 91
modify_traffic_mirror_session, 91
modify_transit_gateway_vpc_attachment, 91
modify_volume, 91
modify_volume_attribute, 91
modify_vpc_attribute, 91
modify_vpc_endpoint, 91
modify_vpc_endpoint_connection_notification, 91
modify_vpc_endpoint_service_configuration, 91
modify_vpc_endpoint_service_permissions, 91
modify_vpc_peering_connection_options, 91
modify_vpc_tenancy, 91
modify_vpn_connection, 91
modify_vpn_tunnel_certificate, 91
modify_vpn_tunnel_options, 91
monitor_instances, 91
move_account, 176
move_address_to_vpc, 91
mq, 163
mturk, 165
neptune, 167
notify_when_uploaded, 257
notify_workers, 166
open_instance_public_ports, 156
opsworks, 169
opsworkscm, 173
opt_in_phone_number, 249
organizations, 175
peer_vpc, 156
personalize, 177
personalizeevents, 179
personalizeruntime, 180
phone_number_validate, 185
pi, 181
pinpoint, 183
pinpointemail, 186
pinpointsmsvoice, 188
poll_for_activity_task, 264
poll_for_decision_task, 264
poll_for_task, 69
polly, 190
post_content, 152
post_text, 152
post_to_connection, 11
predict, 158
preview_agents, 137
pricing, 191
promote_read_replica, 200
promote_read_replica_db_cluster, 169
promote_resource_share_created_from_policy, 196
provision_byoip_cidr, 91
provision_product, 236
publish, 249
publish_layer_version, 149
publish_schema, 33
publish_version, 149
purchase_host_reservation, 91
purchase_provisioned_capacity, 120
purchase_reserved_cache_nodes_offering, 102
purchase_reserved_db_instances_offering, 200
purchase_reserved_elasticsearch_instance_offering, 106
put_key_policy, 147
put_lexicon, 190
put_lifecycle_configuration, 98
put_lifecycle_hook, 23
put_lifecycle_policy, 95
put_log_events, 51
put_logging_configuration, 270, 273
put_method, 10
put_method_response, 10
put_metric_alarm, 47
put_metric_data, 47
put_metric_filter, 51
put_notification_channel, 117
put_notification_configuration, 23
put_object, 221
put_object_acl, 221
put_object_legal_hold, 221
put_object_lock_configuration, 221
put_object_retention, 221
put_object_tagging, 221
put_organization_config_rule, 63
put_organization_conformance_pack, 63
put_parameter, 254
put_partner_events, 48, 114
put_permission, 48, 114
put_permission_policy, 271, 273
put_pipeline_definition, 70
put_policy, 117
put_provisioned_concurrency_config, 149
put_public_access_block, 221, 223
put_record, 115, 141
put_record_batch, 116
put_records, 141
put Remediation_configurations, 63
put Remediation_exceptions, 63
put_report_definition, 66
put_resolver_rule_policy, 218
put_resource_config, 63
put_resource_policy, 51, 126, 229
put_rest_api, 10
put_retention_configuration, 63
put_retention_policy, 51
put_role_permissions_boundary, 135
put_role_policy, 135
put_rule, 48, 114
put_scaling_policy, 16, 23
put_scheduled_action, 16
put_scheduled_update_group_action, 23
put_schema_from_json, 33
put_secret_value, 229
put_service_quota_increase_request_into_template, 239
put_session, 152
put_slot_type, 151
put_subscription_filter, 51
put_targets, 48, 114
put_user_permissions_boundary, 135
put_user_policy, 135
put_workflow_run_properties, 126
query, 80
query_objects, 70
quicksight, 192
ram, 195
rds, 196
rdsdataservice, 201
re_encrypt, 147
reboot_broker, 165
reboot_cache_cluster, 102
reboot_cluster, 205
reboot_db_instance, 79, 169, 200
reboot_instance, 156, 172
reboot_instances, 91
reboot_node, 71
rebuild_environment, 104
receive_message, 251
recognize-celebrities, 206
record_activity_task_heartbeat, 264
record_handler_progress, 35
record_lifecycle_action_heartbeat, 23
redshift, 202
refresh_cache, 258
refresh_trusted_advisor_check, 263
register_activity_type, 264
register_certificate, 75
register_container_instance, 97
register_cross_account_access_role, 137
register_db_proxy_targets, 200
register_default_patch_baseline, 254
register_device, 57
register_domain, 216, 264
register_ecs_cluster, 172
register_elastic_ip, 172
register_event_topic, 75
register_image, 91
register_instance, 172, 238
register_instances_with_load_balancer, 108
register_job_definition, 29
register_patch_baseline_for_patch_group, 254
register_rds_db_instance, 172
register_scalable_target, 16
register_stream_consumer, 141
register_target_with_maintenance_window, 254
register_targets, 110
register_task_definition, 97
register_task_with_maintenance_window, 254
register_transit_gateway_multicast_group_members, 91
register_transit_gateway_multicast_group_sources, 91
register_type, 35
register_usage, 163
register_user, 194
register_volume, 172
register_workflow_type, 264
reimport_api, 14
reject_assignment, 166
reject_portfolio_share, 236
reject_qualification_request, 166
reject_resource_share_invitation, 196
reject_shared_directory, 75
reject_transit_gateway_peering_attachment, 91
reject_transit_gateway_vpc_attachment, 91
reject_vpc_endpoint_connections, 91
reject_vpc_peering_connection, 91
rekognition, 205
release_address, 91
release_hosts, 91
release_static_ip, 156
remove_account_from_organization, 176
remove_attributes, 185
remove_attributes_from_findings, 138
remove_auto_scaling_policy, 112
remove_client_id_from_open_id_connect_provider, 135
remove_facet_from_object, 33
remove_from_global_cluster, 200
remove_ip_routes, 75
remove_layer_version_permission, 149
remove_listener_certificates, 110
remove_permission, 48, 114, 149, 249, 251
remove_role_from_db_cluster, 169, 200
remove_role_from_db_instance, 200
remove_role_from_instance_profile, 135
remove_source_identifier_from_subscription, 169, 200
remove_tags, 45, 70, 106, 110, 112
remove_tags_from_certificate, 5
remove_tags_from_resource, 39, 75, 79, 102, 169, 200, 254, 258
remove_tags_from_stream, 141
remove_tags_from_vault, 120
remove_targets, 48, 114
remove_user_from_group, 135
renew_certificate, 5
renew_domain, 216
reorder_receipt_rule_set, 241
replace_iam_instance_profile_association, 91
replace_network_acl_association, 91
replace_network_acl_entry, 91
replace_route, 91
replace_route_table_association, 91
replace_transit_gateway_route, 91
report_instance_status, 91
report_task_progress, 70
report_task_runner_heartbeat, 70
request_cancel_workflow_execution, 264
request_certificate, 5
request_environment_info, 104
request_service_quota_increase, 239
request_spot_fleet, 91
request_spot_instances, 91
resend_confirmation_code, 55
resend_contact_reachability_email, 216
resend_validation_email, 5
reset_cache, 258
reset_cache_parameter_group, 102
reset_cluster_parameter_group, 205
reset_db_cluster_parameter_group, 79, 169, 200
reset_db_parameter_group, 169, 200
reset_ebs_default_kms_key_id, 91
reset_fpga_image_attribute, 91
reset_image_attribute, 91
reset_instance_attribute, 91
reset_job_bookmark, 126
reset_network_interface_attribute, 91
reset_service_setting, 254
reset_service_specific_credential, 135
reset_snapshot_attribute, 91
reset_user_password, 75
resize_cluster, 205
resolve_case, 263
resolve_customer, 163
resourcegroups, 207
resourcegroupstaggingapi, 209
respond_activity_task_canceled, 264
respond_activity_task_completed, 264
respond_activity_task_failed, 264
respond_decision_task_completed, 264
respond_to_auth_challenge, 55
restart_app_server, 104
restore_address_to_classic, 91
restore_backup, 40
restore_certificate_authority, 7
restore_db_cluster_from_s3, 200
restore_db_cluster_from_snapshot, 79, 169, 200
restore_db_cluster_to_point_in_time, 79, 169, 200
restore_db_instance_from_db_snapshot, 200
restore_db_instance_from_s3, 200
restore_db_instance_to_point_in_time, 200
restore_from_cluster_snapshot, 205
restart_app_server, 104
resume_processes, 23
resume_session, 254
resync_mfa_device, 135
retire_grant, 147
retrieve_domain_auth_code, 216
retrieve_environment_info, 104
retrieve_tape_archive, 258
retrieve_tape_recovery_point, 258
revoke_cache_security_group_ingress, 102
revoke_certificate, 7
revoke_client_vpn_ingress, 92
revoke_cluster_security_group_ingress, 205
revoke_db_security_group_ingress, 200
revoke_grant, 147
revoke_security_group_egress, 92
revoke_security_group_ingress, 92
revoke_snapshot_access, 205
rollback_transaction, 201
rotate_encryption_key, 205
rotate_secret, 229
route53, 213
route53domains, 215
route53resolver, 217
run_instances, 92
run_job_flow, 112
run_scheduled_instances, 92
run_task, 97
s3, 219
s3control, 222
sagemaker, 223
sagemakerruntime, 227
scan, 81
scan_provisioned_products, 236
schedule_key_deletion, 147
search, 43, 226
search_faces, 206
search_faces_by_image, 206
search_provisioned_products, 236
search_resources, 236
search_tables, 126
search_transit_gateway_multicast_groups, 92
search_transit_gateway_routes, 92
secretsmanager, 228
securityhub, 230
select, 247
select_object_content, 221
select_resource_config, 221
send_automation_signal, 254
send_bonus, 166
send_bounce, 241
send_bulk_templated_email, 241
send_command, 254
send_custom_verification_email, 241
send_diagnostic_interrupt, 92
send_email, 188, 241
send_message, 251
send_message_batch, 251
send_messages, 185
send_raw_email, 241
send_ssh_public_key, 93
send_task_failure, 243
send_task_heartbeat, 243
send_task_success, 243
send_templated_email, 241
send_test_event_notification, 166
send_users_messages, 185
send_voice_message, 189
serverlessapplicationrepository, 232
servicem踝cathol, 234
servicediscovery, 236
servicequotas, 238
ses, 240
set_active_receipt_rule_set, 241
set_alter_state, 47
set_cognito_events, 57
set_data_retrieval_policy, 120
set_default_policy_version, 135
set_desired_capacity, 23
set_endpoint_attributes, 249
set_identity_dkim_enabled, 241
set_identity_feedback_forwarding_enabled, 241
set_identity_headers_in_notifications_enabled, 241
set_identity_mail_from_domain, 241
set_identity_notification_topic, 241
set_identity_pool_configuration, 57
set_identity_pool_roles, 52
set_instance_health, 23
set_instance_protection, 23
set_ip_address_type, 111
set_load_balancer_listener_ssl_certificate, 108
set_load_balancer_policies_for_backend_servers
set_load_balancer_policies_of_listener, 108
set_load_based_auto_scaling, 172
set_local_console_password, 258
set_message, 254
set_permission, 172
set_platform_application_attributes, 249
set_queue_attributes, 251
set_receipt_rule_position, 242
set_repository_policy, 95
set_risk_configuration, 55
set_rule_priorities, 111
set_security_groups, 111
set_security_token_service_preferences, 135
set_smb_guest_password, 258
set_sms_attributes, 249
set_stack_policy, 35
set_status, 70
set_subnets, 111
set_subscription_attributes, 249
set_tags_for_resource, 138
set_task_status, 70
set_termination_protection, 112
set_time_based_auto_scaling, 172
set_topic_attributes, 249
set_type_default_version, 35
set_ui_customization, 55
set_user_mfa_preference, 55
set_user_pool_mfa_config, 55
set_user_settings, 55
set_vault_access_policy, 120
set_vault_notifications, 121
set_visible_to_all_users, 112
sfn, 242
share_directory, 76
shield, 244
shutdown_gateway, 258
sign, 147
sign_up, 55
signal_resource, 36
signal_workflow_execution, 264
simpledb, 246
simulate_custom_policy, 135
simulate_principal_policy, 135
sns, 248
sqs, 249
ssm, 251
start_activity_stream, 200
start_application, 142, 144
start_assessment_run, 138
start_associations_once, 254
start_automation_execution, 254
start_availability_monitor_test, 258
start_backup_job, 26
start_celebrity_recognition, 206
start_chat_contact, 65
start_config_rules_evaluation, 63
start_configuration_recorder, 63
start_content_moderation, 206
start_crawler, 126
start_crawler_schedule, 126
start_db_cluster, 79, 200
start_db_instance, 200
start_delivery_stream_encryption, 116
start_document_analysis, 266
start_document_classification_job, 59
start_document_text_detection, 266
start_dominant_language_detection_job, 59
start_elasticsearch_service_software_update, 106
start_entities_detection_job, 59
start_entities_detection_v2_job, 61
start_execution, 243
start_export_labels_task_run, 126
start_face_detection, 206
start_face_search, 206
start_gateway, 258
start_image_scan, 95
start_import, 151
start_import_labels_task_run, 126
start_instance, 157, 172
start_instances, 92
start_job_run, 126
start_key_phrases_detection_job, 59
start_label_detection, 206
start_lifecycle_policy_preview, 95
start_logging, 45
start_maintenance, 174
start_migration, 103
start_ml_evaluation_task_run, 126
start_ml_labeling_set_generation_task_run, 126
start_monitoring_schedule, 226
start_notebook_instance, 226
start_outbound_voice_contact, 65
start_personTracking, 206
start_phi_detection_job, 61
start_project_version, 206
start_query, 51
start_query_execution, 21
start_relational_database, 157
start_remediation_execution, 63
start_report_creation, 213
start_restore_job, 26
start_schema_extension, 76
start_sentiment_detection_job, 59
start_session, 254
start_speech_synthesis_task, 190
start_stack, 172
start_stream_encryption, 141
start_stream_processor, 206
start_support_data_export, 160
start_task, 97
start_text_translation_job, 268
start_topics_detection_job, 59
start_transcription_job, 267
start_trigger, 126
start_user_import_job, 55
start_vpc_endpoint_service_private_dns_verification, 92
start_workflow_execution, 265
start_workflow_run, 126
stop_activity_stream, 200
stop_application, 142, 144
stop_assessment_run, 138
stop_auto_ml_job, 226
stop_automation_execution, 254
stop_backup_job, 26
stop_compilation_job, 226
stop_configuration_recorder, 63
stop_contact, 65
stop_crawler, 126
stop_crawler_schedule, 126
stop_db_cluster, 79, 200
stop_db_instance, 200
stop_delivery_stream_encryption, 116
stop_dominant_language_detection_job, 59
stop_entities_detection_job, 59
stop_entities_detection_v2_job, 61
stop_execution, 243
stop_hyper_parameter_tuning_job, 226
stop_instance, 157, 172
stop_instances, 92
stop_key_phrases_detection_job, 59
stop_labeling_job, 226
stop_logging, 45
stop_monitoring_members, 129
stop_monitoring_schedule, 226
stop_notebook_instance, 226
stop_phi_detection_job, 61
stop_processing_job, 226
stop_project_version, 206
stop_query, 51
stop_query_execution, 21
stop_relational_database, 157
stop_sentiment_detection_job, 59
stop_stack, 172
stop_stack_set_operation, 36
stop_stream_encryption, 141
stop_stream_processor, 206
stop_task, 97
stop_text_translation_job, 268
stop_training_document_classifier, 59
stop_training_entity_recognizer, 59
stop_training_job, 226
stop_transform_job, 226
stop_trigger, 126
stop_user_import_job, 55
storagegateway, 255
sts, 258
submit_attachment_state_changes, 97
submit_container_state_change, 97
submit_job, 29
submit_task_state_change, 97
subscribe, 249
subscribe_to_dataset, 57
subscribe_to_event, 138
suggest, 43
support, 261
suspend_processes, 23
swap_environment_cnam_es, 104
swf, 263
synthesize_speech, 190

tag, 208
tag_certificate_authority, 7
tag_delivery_stream, 116
tag_log_group, 51
tag_queue, 251
tag_resource, 10, 14, 17, 19, 21, 26, 33, 37, 40, 47, 48, 52, 55, 59, 63, 65, 71, 73, 77, 81, 95, 97, 100, 114, 117, 118, 126, 129, 139, 142, 144, 147, 149, 153, 157, 172, 174, 176, 185, 188, 194, 196, 218, 229, 232, 243, 249, 265, 271, 273
tag_resources, 213
tag_role, 135
tag_user, 135
terminate_client_vpn_connections, 92
terminate_environment, 104
terminate_instance_in_auto_scaling_group, 23
terminate_instances, 92
terminate_job, 29
terminate_job_flows, 112
terminate_provisioned_product, 236
terminate_session, 254
terminate_workflow_execution, 265
test_dns_answer, 215
test_event_pattern, 48, 114
test_failover, 103
test_invoke_authorizer, 10
test_invoke_method, 10
test_metric_filter, 51
test_render_template, 242
textract, 265
transact_get_items, 81
transact_write_items, 81
transcribeservice, 266
transfer_domain, 216
translate, 267
translate_text, 268
unarchive_findings, 129
unassign_instance, 172
unassign_ipv6_addresses, 92
unassign_private_ip_addresses, 92
unassign_volume, 172
undeprecate_activity_type, 265
undeprecate_domain, 265
undeprecate_workflow_type, 265
unlink_developer_identity, 52
unlink_identity, 52
unmonitor_instances, 92
unpeer_vpc, 157
unshare_directory, 76
unsubscribe, 249
unsubscribe_from_dataset, 57
unsubscribe_from_event, 138
untag, 208
untag_certificate_authority, 7
untag_delivery_stream, 116
untag_log_group, 51
untag_queue, 251
untag_resource, 10, 14, 17, 19, 21, 26, 33, 37, 40, 47, 49, 52, 55, 59, 63, 65, 71, 73, 77, 81, 95, 97, 100, 114, 117, 118, 126, 129, 139, 142, 144, 147, 149, 153, 157, 172, 174, 176, 185, 188, 194, 196, 218, 229, 232, 243, 249, 265, 271, 273
untag_resources, 213
untag_role, 135
untag_user, 135
update.accelerator, 123
update_accelerator_attributes, 123
update_access_key, 135
update_account, 10
update_account_password_policy, 135
update_account_sending_enabled, 242
update_action_target, 232
update_adm_channel, 185
update_alias, 147, 149
update_api, 14
update_api_key, 10
update_api_mapping, 14
update_apns_channel, 185
update_apns_sandbox_channel, 185
update_apns_voip_channel, 185
update_apns_voip_sandbox_channel, 185
update_app, 172
update_application, 18, 104, 142, 144, 233
update_application_resource_lifecycle, 105
update_application_settings, 185
update_application_version, 105
update_assessment_target, 138
update_association, 254
update_association_status, 254
update_assume_role_policy, 135
update_auth_event_feedback, 55
update_authorizer, 10, 14
update_auto_scaling_group, 23
update_availability_options, 42
update_backup_plan, 26
update_baidu_channel, 185
update_bandwidth_rate_limit, 258
update_base_path_mapping, 10
update_batch_prediction, 158
update_broker, 165
update_broker_count, 139
update_broker_storage, 139
update_budget, 31
update_byte_match_set, 271, 273
update_campaign, 178, 185
update_certificate_authority, 7
update_certificate_options, 5
update_chap_credentials, 258
update_classifier, 126
update_client_certificate, 10
update_cloud_front_origin_access_identity, 37
update_cluster, 71
update_cluster_config, 100
update_cluster_configuration, 139
update_cluster_settings, 97
update_cluster_version, 100
update_continuous_backups, 97
update_conditional_forwarder, 76
update_configuration, 165
update_configuration_set_event_destination, 188, 189, 242
update_configuration_set_reputation_metrics_enabled, 242
update_configuration_set_sending_enabled, 242
update_configuration_set_tracking_options, 242
update_configuration_template, 105
update_connection, 126
update_constraint, 236
update_contact_attributes, 65
update_container_agent, 97
update_container_instances_state, 97
update_continuous_backups, 81
update_contributor_insights, 81
update_cost_category_definition, 68
update_crawler, 126
update_crawler_schedule, 126
update_custom_key_store, 147
update_custom_verification_email_template, 242
update_dashboard, 194
update_dashboard_permissions, 194
update_dashboard_published_version, 194
update_data_set, 194
update_data_set_permissions, 194
update_data_source, 158, 194
update_data_source_permissions, 194
update_database, 126
update_deployment, 10, 14
update_detector, 129
update_dev_endpoint, 126
update_device_status, 55
update_direct_connect_gateway_association, 73
update_distribution, 37
update_document, 254
update_document_default_version, 254
update_documentation_part, 10
update_documentation_version, 10
update_domain, 226
update_domain_contact, 216
update_domain_contact_privacy, 216
update_domain_endpoint_options, 42
update_domain_entry, 157
update_domain_name, 10, 14
update_domain_nameservers, 216
update_elastic_ip, 172
update_elasticsearch_domain_config, 106
update_email_channel, 185
update_email_template, 185
update_emergency_contact_settings, 246
update_endpoint, 59, 185, 226
update_endpoint_group, 123
update_endpoint_weights_and_capacities, 226
update_endpoints_batch, 186
update_environment, 105
update_evaluation, 158
update_event_source_mapping, 149
update_experiment, 226
update_expiration_for_hit, 166
update_facet, 33
update_field_level_encryption_config, 38
update_field_level_encryption_profile, 38
update_file_system, 98, 118
update_filter, 129
update_findings, 232
update_findings_feedback, 129
update_function_code, 149
update_function_configuration, 149
update_function_event_invoke_config, 149
update_gateway_information, 258
update_gateway_response, 10
update_gateway_software_now, 258
update_gcm_channel, 186
update_geo_match_set, 271, 273
update_global_table, 81
update_global_table_settings, 81
update_group, 55, 136, 194, 208
update_group_query, 208
update_health_check, 215
update_hit_review_status, 166
update_hit_type_of_hit, 166
update_hosted_zone_comment, 215
update_iam_policy_assignment, 194
update_identity_pool, 52
update_identity_provider, 55
update_insight, 232
update_instance, 172
update_instance_custom_health_status, 238
update_integration, 10, 14
update_integration_response, 10, 14
update_item, 129, 271, 273
update_key_description, 147
update_lag, 73
update_layer, 172
update_license_configuration, 153
update_license_specifications_for_resource, 153
update_shard_count, 141
update_signing_certificate, 136
update_size_constraint_set, 271, 273
update_smb_file_share, 258
update_smb_security_strategy, 258
update_sms_channel, 186
update_sms_template, 186
update_snapshot_schedule, 258
update_sql_injection_match_set, 271, 273
update_ssh_public_key, 136
update_stack, 36, 172
update_stack_instances, 36
update_stack_set, 36
update_stage, 10, 14
update_state_machine, 243
update_streaming_distribution, 38
update_subnet_group, 71
update_subscriber, 31
update_subscription, 246
update_table, 81, 127
update_table_replica_auto_scaling, 81
update_tag_option, 236
update_tags_for_domain, 216
update_tags_for_resource, 105
update_task_set, 97
update_template, 194, 242
update_template_active_version, 186
update_template_alias, 194
update_template_permissions, 194
update_termination_protection, 36
update_threat_intel_set, 129
update_time_to_live, 81
update_traffic_policy_comment, 215
update_traffic_policy_instance, 215
update_trail, 45
update_trial, 226
update_trial_component, 226
update_trigger, 127
update_trust, 76
update_TYPED_LINK_FACET, 33
update_usage, 10
update_usage_plan, 10
update_user, 136, 165, 194
update_user_attributes, 56
update_user_defined_function, 127
update_user_hierarchy, 65
update_user_identity_info, 65
update_user_phone_config, 65
update_user_pool, 56
update_user_pool_client, 56
update_user_pool_domain, 56
update_user_profile, 172, 226
update_user_routing_profile, 65
update_user_security_profiles, 65
update_virtual_interface_attributes, 73
update_virtual_node, 19
update_virtual_router, 19
update_virtual_service, 19
update_vocabulary, 267
update_vocabulary_filter, 267
update_voice_channel, 186
update_voice_template, 186
update_volume, 172
update_vpc_link, 10
update_vtl_device_type, 258
update_web_acl, 271, 274
update_work_group, 21
update_workflow, 127
update_workteam, 226
update_xss_match_set, 271, 274
upgrade_applied_schema, 33
upgrade_elasticsearch_domain, 106
upgrade_published_schema, 33
upload_archive, 121
upload_documents, 43
upload_layer_part, 95
upload_multipart_part, 121
upload_part, 221
upload_part_copy, 221
upload_server_certificate, 136
upload_signing_certificate, 136
upload_ssh_public_key, 136
validate_configuration_settings, 105
validate_pipeline_definition, 70
validate_template, 36
verify, 147
verify_domain_dkim, 242
verify_domain_identity, 242
verify_email_address, 242
verify_email_identity, 242
verify_software_token, 56
verify_trust, 76
verify_user_attribute, 56
INDEX

view_billing, 216

waf, 268
wafregional, 271
withdraw_byoip_cidr, 92