Package ‘paws’

September 3, 2021

Title Amazon Web Services Software Development Kit
Version 0.1.12
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)
URL https://github.com/paws-r/paws
BugReports https://github.com/paws-r/paws/issues
Imports paws.analytics (>= 0.1.12), paws.application.integration (>= 0.1.12), paws.compute (>= 0.1.12), paws.cost.management (>= 0.1.12), paws.customer.engagement (>= 0.1.12), paws.database (>= 0.1.12), paws.developer.tools (>= 0.1.12), paws.end.user.computing (>= 0.1.12), paws.machine.learning (>= 0.1.12), paws.networking (>= 0.1.12), paws.security.identity (>= 0.1.12), paws.storage (>= 0.1.12)
Suggests testthat
Encoding UTF-8
RoxygenNote 7.1.1
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 2021-09-03 04:50:11 UTC
R topics documented:

acm ...................................................... 5
acmpca ................................................. 6
apigateway ............................................ 8
apigatewaymanagementapi .................... 12
apigatewayv2 .................................... 13
applicationautoscaling ......................... 15
applicationinsights ............................. 18
appmesh ............................................... 20
appstream .......................................... 22
athena ............................................... 24
autoscaling ......................................... 26
autoscalingplans ................................. 29
backup .............................................. 30
batch ................................................ 32
budgets ............................................. 34
cloud9 ............................................... 36
clouddirectory ................................. 38
cloudformation ............................... 41
cloudfront ..................................... 43
cloudhsm ........................................ 46
cloudhsmv2 .................................... 48
cloudsearch ................................... 49
cloudsearchdomain ......................... 51
cloudtrail ...................................... 52
cloudwatch .................................... 54
cloudwatchevents ......................... 56
cloudwatchlogs ............................. 58
codebuild .................................... 61
codecommit .................................. 64
codedeploy .................................... 71
codeml ........................................ 74
codelar ......................................... 78
cognitoidentity .............................. 80
cognitoidentityprovider .................. 82
cognitosync .................................... 86
comprehend .............................. 88
comprehendmedical .................... 90
configservice .......................... 92
connect ...................................... 95
costandusagereportservice ........ 98
costexplorer ................................ 99
datapipeline .................................. 101
dax ............................................. 103
directconnect ............................ 105
directoryservice ...................... 107
dlm ............................................. 110
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>docdb</td>
<td>111</td>
</tr>
<tr>
<td>dynamodb</td>
<td>113</td>
</tr>
<tr>
<td>dynamodbstreams</td>
<td>116</td>
</tr>
<tr>
<td>ec2</td>
<td>118</td>
</tr>
<tr>
<td>ec2instanceconnect</td>
<td>128</td>
</tr>
<tr>
<td>ecr</td>
<td>129</td>
</tr>
<tr>
<td>ecs</td>
<td>132</td>
</tr>
<tr>
<td>efs</td>
<td>134</td>
</tr>
<tr>
<td>eks</td>
<td>136</td>
</tr>
<tr>
<td>elasticache</td>
<td>138</td>
</tr>
<tr>
<td>elasticbeanstalk</td>
<td>141</td>
</tr>
<tr>
<td>elasticsearchservice</td>
<td>143</td>
</tr>
<tr>
<td>elb</td>
<td>145</td>
</tr>
<tr>
<td>elbv2</td>
<td>147</td>
</tr>
<tr>
<td>emr</td>
<td>150</td>
</tr>
<tr>
<td>eventbridge</td>
<td>152</td>
</tr>
<tr>
<td>firehose</td>
<td>154</td>
</tr>
<tr>
<td>fms</td>
<td>156</td>
</tr>
<tr>
<td>fsx</td>
<td>157</td>
</tr>
<tr>
<td>glacier</td>
<td>159</td>
</tr>
<tr>
<td>globalaccelerator</td>
<td>161</td>
</tr>
<tr>
<td>glue</td>
<td>166</td>
</tr>
<tr>
<td>guardduty</td>
<td>170</td>
</tr>
<tr>
<td>health</td>
<td>173</td>
</tr>
<tr>
<td>iam</td>
<td>175</td>
</tr>
<tr>
<td>inspector</td>
<td>179</td>
</tr>
<tr>
<td>kafka</td>
<td>181</td>
</tr>
<tr>
<td>kinesis</td>
<td>183</td>
</tr>
<tr>
<td>kinesisanalytics</td>
<td>184</td>
</tr>
<tr>
<td>kinesisanalyticsv2</td>
<td>186</td>
</tr>
<tr>
<td>kms</td>
<td>188</td>
</tr>
<tr>
<td>lambda</td>
<td>191</td>
</tr>
<tr>
<td>lexmodelbuildingservice</td>
<td>194</td>
</tr>
<tr>
<td>lexruntimeservice</td>
<td>196</td>
</tr>
<tr>
<td>licensemanager</td>
<td>198</td>
</tr>
<tr>
<td>lightsail</td>
<td>200</td>
</tr>
<tr>
<td>machinelearning</td>
<td>204</td>
</tr>
<tr>
<td>macie</td>
<td>206</td>
</tr>
<tr>
<td>marketplacecommerceanalytics</td>
<td>207</td>
</tr>
<tr>
<td>marketplaceentitlementservice</td>
<td>209</td>
</tr>
<tr>
<td>marketplacemetering</td>
<td>210</td>
</tr>
<tr>
<td>mq</td>
<td>212</td>
</tr>
<tr>
<td>mturk</td>
<td>213</td>
</tr>
<tr>
<td>neptune</td>
<td>215</td>
</tr>
<tr>
<td>opsworks</td>
<td>218</td>
</tr>
<tr>
<td>opsworkscm</td>
<td>222</td>
</tr>
<tr>
<td>organizations</td>
<td>224</td>
</tr>
<tr>
<td>personalize</td>
<td>226</td>
</tr>
</tbody>
</table>
topics documented:

personalizeevents .......................................................... 228
personalizeruntime ............................................................ 230
pi ........................................................................... 231
pinpoint ................................................................. 232
pinpointemail ............................................................... 236
pinpointsmsvoice ............................................................. 238
polly .................................................. 239
pricing ................................................................. 241
quicksight ................................................................. 242
ram ................................................................. 246
rds ........................................................................... 247
rdsdataservice .............................................................. 252
redshift ........................................................................... 253
rekognition ................................................................. 257
resourcegroups ............................................................... 259
resourcegroupstaggingapi ................................................... 261
route53 ................................................................. 266
route53domains ............................................................... 269
route53resolver ............................................................... 271
s3 ................................................................. 273
s3control ................................................................. 276
sagemaker ................................................................. 278
sagemakerruntime ........................................................... 284
secretsmanager ............................................................... 285
securityhub ................................................................. 288
serverlessapplicationrepository ........................................... 291
servicecatalog ............................................................... 293
servicediscovery ............................................................ 296
servicequotas ............................................................... 297
ses ................................................................. 299
sfn ................................................................. 302
shield ................................................................. 304
simpledb ................................................................. 306
sns ................................................................. 307
sqs ................................................................. 309
ssm ................................................................. 311
storagegateway ............................................................ 315
sts ................................................................. 319
support ................................................................. 321
swf ................................................................. 323
textract ................................................................. 325
transcribeservice ........................................................... 326
translate ............................................................... 328
waf ................................................................. 330
wafregional ................................................................. 333
workdocs ................................................................. 336
worklink ................................................................. 338
workspaces ............................................................... 340
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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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 an Amazon-issued certificate and its certificate chain
- **import_certificate**: Imports a certificate into AWS Certificate Manager (ACM) to use with services that are integrated with ACM
- **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 quota that determines the number of times the action can be called per second. For more information, see *API Rate Quotas in ACM Private CA* in the ACM Private CA user guide.

Usage

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

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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 Grants one or more permissions on a private CA to the AWS Certificate Manager (ACM) service principal (acm)
delete_certificate_authority Deletes a private certificate authority (CA)
delete_permission Revokes permissions on a private CA granted to the AWS Certificate Manager (ACM) service principal (acm)
delete_policy Deletes the resource-based policy attached to a private CA
describe_certificate_authority Lists information about your private certificate authority (CA) or one that has been shared with you
describe_certificate_authority_audit_report Lists information about a specific audit report created by calling the CreateCertificateAuthorityAuditReport action
get_certificate Retrieves a certificate from your private CA or one that has been shared with you
get_certificate_authority_certificate Retrieves the certificate and certificate chain for your private certificate authority (CA)
get_certificate_authority_csr Retrieves the certificate signing request (CSR) for your private certificate authority (CA)
get_policy Retrieves the resource-based policy attached to a private CA
import_certificate_authority_certificate Imports a signed private CA certificate into ACM Private CA
issue_certificate Uses your private certificate authority (CA), or one that has been shared with you, to issue a client certificate
list_certificate_authorities Lists the private certificate authorities that you created by using the CreateCertificateAuthority action
list_permissions List all permissions on a private CA, if any, granted to the AWS Certificate Manager (ACM) service principal (acm)
list_tags Lists the tags, if any, that are associated with your private CA or one that has been shared with you
put_policy Attaches a resource-based policy to a private CA
restore_certificate_authority Restores a certificate authority (CA) that is in the DELETED state
revoke_certificate Revokes a certificate that was issued inside ACM Private CA
tag_certificate_authority  Adds one or more tags to your private CA
untag_certificate_authority Remove one or more tags from your private CA
update_certificate_authority Updates the status or configuration of a private certificate authority (CA)

Examples

```r
## Not run:
svc <- acmpca()
svc$create_certificate_authority(
    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**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
svc <- apigateway(
    config = list(
        credentials = list(
            creds = list(
                access_key_id = "string",
                secret_key = "string",
                region = "string",
                endpoint = "string"
            )
        )
    )
)
```

```r
svc$create_certificate_authority(
    Foo = 123
)
```
secret_access_key = "string",
    session_token = "string"
),
    profile = "string"
),
    endpoint = "string",
    region = "string"
)
}

Operations

create_api_key
create_authorizer
create_base_path_mapping
create_deployment
create_documentation_part
create_documentation_version
create_domain_name
create_model
create_request_validator
create_resource
create_rest_api
create_stage
create_usage_plan
create_usage_plan_key
create_vpc_link
delete_api_key
delete_authorizer
delete_base_path_mapping
delete_client_certificate
delete_deployment
delete_documentation_part
delete_documentation_version
delete_domain_name
delete_gateway_response
delete_integration
delete_integration_response
delete_method
delete_method_response
delete_model
delete_request_validator
delete_resource
delete_rest_api
delete_stage
delete_usage_plan
delete_usage_plan_key
delete_vpc_link

Create an ApiKey resource
Adds a new Authorizer resource to an existing RestApi resource
Creates a new BasePathMapping resource
Creates a Deployment resource, which makes a specified RestApi callable over the internet
Create documentation part
Create documentation version
Creates a new domain name
Adds a new Model resource to an existing RestApi resource
Creates a RequestValidator of a given RestApi
Creates a Resource resource
Creates a new RestApi resource
Creates a new Stage resource that references a pre-existing Deployment for the API
Creates a usage plan with the throttle and quota limits, as well as the associated API stages,
Creates a usage plan key for adding an existing API key to a usage plan
Creates a VPC link, under the caller’s account in a selected region, in an asynchronous operation
Deletes the ApiKey resource
Deletes an existing Authorizer resource
Deletes the BasePathMapping resource
Deletes the ClientCertificate resource
Deletes a Deployment resource
Delete documentation part
Delete documentation version
Deletes the DomainName resource
Clears any customization of a GatewayResponse of a specified response type on the given RestApi
Represents a delete integration
Represents a delete integration response
Deletes an existing Method resource
Deletes an existing MethodResponse resource
Deletes a model
Deletes a RequestValidator of a given RestApi
Deletes a Resource resource
Deletes the specified API
Deletes a Stage resource
Deletes a usage plan of a given plan Id
Deletes a usage plan key and remove the underlying API key from the associated usage plan
Deletes an existing VpcLink of a specified identifier
flush_stage_authorizers_cache  
flush_stage_cache  
generate_client_certificate  
get_account  
get_api_key  
get_api_keys  
get_authorizer  
generate_client_certificate  
generate_client_certificates  
generate_deployment  
generate_deployments  
generate_documentation_part  
generate_documentation_parts  
generate_documentation_version  
generate_documentation_versions  
generate_domain_name  
generate_domain_names  
generate_export  
generate_gateway_response  
generate_gateway_responses  
generate_integration  
generate_integration_response  
generate_method  
generate_method_response  
generate_model  
generate_models  
generate_model_template  
generate_request_validator  
generate_request_validators  
generate_resource  
generate_rest_api  
generate_rest_apis  
generate_sdk  
generate_sdk_type  
generate_sdk_types  
generate_stage  
generate_stages  
generate_tags  
generate_usage  
generate_usage_plan  
generate_usage_plan_key  
generate_usage_plan_keys  
generate_usage_plans  
generate_vpc_link  
flush_stage_authorizers_cache  
flush_stage_cache  
generate_client_certificate  
generate_client_certificate  
generate_deployment  
generate_deployments  
generate_documentation_part  
generate_documentation_parts  
generate_documentation_version  
generate_documentation_versions  
generate_domain_name  
generate_domain_names  
generate_export  
generate_gateway_response  
generate_gateway_responses  
generate_integration  
generate_integration_response  
generate_method  
generate_method_response  
generate_model  
generate_models  
generate_model_template  
generate_request_validator  
generate_request_validators  
generate_resource  
generate_rest_api  
generate_rest_apis  
generate_sdk  
generate_sdk_type  
generate_sdk_types  
generate_stage  
generate_stages  
generate_tags  
generate_usage  
generate_usage_plan  
generate_usage_plan_key  
generate_usage_plan_keys  
generate_usage_plans  
generate_vpc_link

Flushes all authorizer cache entries on a stage
Flushes a stage’s cache
Generates a ClientCertificate resource
Gets information about the current Account resource
Gets information about the current ApiKey resource
Gets information about the current ApiKeys resource
Describe an existing Authorizer resource
Describe an existing Authorizers resource
Describe a BasePathMapping resource
Represents a collection of BasePathMapping resources
Gets information about the current ClientCertificate resource
Gets a collection of ClientCertificate resources
Gets information about a Deployment resource
Gets information about a Deployments collection
Get documentation part
Get documentation parts
Get documentation version
Get documentation versions
Represents a domain name that is contained in a simpler, more intuitive URL that can be called
Represents a collection of DomainName resources
Exports a deployed version of a RestApi in a specified format
Gets a GatewayResponse of a specified response type on the given RestApi
Gets the GatewayResponses collection on the given RestApi
Get the integration settings
Represents a get integration response
Describe an existing Method resource
Describes a MethodResponse resource
Describes an existing model defined for a RestApi resource
Describes existing Models defined for a RestApi resource
Generates a sample mapping template that can be used to transform a payload into the structure
Gets a RequestValidator of a given RestApi
Gets the RequestValidators collection of a given RestApi
Lists information about a resource
Lists information about a collection of Resource resources
Lists the RestApi resource in the collection
Lists the RestApis resources for your collection
Generates a client SDK for a RestApi and Stage
Get sdk type
Get sdk types
Get information about a Stage resource
Gets information about one or more Stage resources
Gets the Tags collection for a given resource
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
apigateway

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

---

**Examples**

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

# End(Not run)
```
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.

Value
A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- apigatewaymanagementapi(config = list(  
credentials = list(    
creds = list(      
access_key_id = "string",      
secret_access_key = "string",      
session_token = "string"    
),    
profile = "string"    
),    
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(num = 123)
## End(Not run)
```

## Description

Amazon API Gateway V2

## Usage

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

### Arguments

**config**

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_api</td>
<td>Creates an API resource</td>
</tr>
<tr>
<td>create_api_mapping</td>
<td>Creates an API mapping</td>
</tr>
<tr>
<td>create_authorizer</td>
<td>Creates an Authorizer for an API</td>
</tr>
<tr>
<td>create_deployment</td>
<td>Creates a Deployment for an API</td>
</tr>
<tr>
<td>create_domain_name</td>
<td>Creates a domain name</td>
</tr>
<tr>
<td>create_integration</td>
<td>Creates an Integration</td>
</tr>
<tr>
<td>create_integration_response</td>
<td>Creates an IntegrationResponses</td>
</tr>
<tr>
<td>create_model</td>
<td>Creates a Model for an API</td>
</tr>
<tr>
<td>create_route</td>
<td>Creates a Route for an API</td>
</tr>
<tr>
<td>create_route_response</td>
<td>Creates a RouteResponse for a Route</td>
</tr>
<tr>
<td>create_stage</td>
<td>Creates a Stage for an API</td>
</tr>
<tr>
<td>create_vpc_link</td>
<td>Creates a VPC link</td>
</tr>
<tr>
<td>delete_access_log_settings</td>
<td>Deletes the AccessLogSettings for a Stage</td>
</tr>
<tr>
<td>delete_api</td>
<td>Deletes an API resource</td>
</tr>
<tr>
<td>delete_api_mapping</td>
<td>Deletes an API mapping</td>
</tr>
<tr>
<td>delete_authorizer</td>
<td>Deletes an Authorizer</td>
</tr>
<tr>
<td>delete_cors_configuration</td>
<td>Deletes a CORS configuration</td>
</tr>
<tr>
<td>delete_deployment</td>
<td>Deletes a Deployment</td>
</tr>
<tr>
<td>delete_domain_name</td>
<td>Deletes a domain name</td>
</tr>
<tr>
<td>delete_integration</td>
<td>Deletes an Integration</td>
</tr>
<tr>
<td>delete_integration_response</td>
<td>Deletes an IntegrationResponses</td>
</tr>
<tr>
<td>delete_model</td>
<td>Deletes a Model</td>
</tr>
<tr>
<td>delete_route</td>
<td>Deletes a Route</td>
</tr>
<tr>
<td>delete_route_request_parameter</td>
<td>Deletes a route request parameter</td>
</tr>
<tr>
<td>delete_route_response</td>
<td>Deletes a RouteResponse</td>
</tr>
<tr>
<td>delete_route_settings</td>
<td>Deletes the RouteSettings for a stage</td>
</tr>
<tr>
<td>delete_stage</td>
<td>Deletes a Stage</td>
</tr>
<tr>
<td>delete_vpc_link</td>
<td>Deletes a VPC link</td>
</tr>
<tr>
<td>export_api</td>
<td>Export api</td>
</tr>
<tr>
<td>get_api</td>
<td>Gets an API resource</td>
</tr>
<tr>
<td>get_api_mapping</td>
<td>Gets an API mapping</td>
</tr>
<tr>
<td>get_apiMappings</td>
<td>Gets API mappings</td>
</tr>
<tr>
<td>get_apis</td>
<td>Gets a collection of API resources</td>
</tr>
<tr>
<td>get_authorizer</td>
<td>Gets an Authorizer</td>
</tr>
<tr>
<td>get_authorizers</td>
<td>Gets the Authorizers for an API</td>
</tr>
<tr>
<td>get_deployment</td>
<td>Gets a Deployment</td>
</tr>
<tr>
<td>get_deployments</td>
<td>Gets the Deployments for an API</td>
</tr>
<tr>
<td>get_domain_name</td>
<td>Gets a domain name</td>
</tr>
<tr>
<td>get_domain_names</td>
<td>Gets the domain names for an AWS account</td>
</tr>
<tr>
<td>get_integration</td>
<td>Gets an Integration</td>
</tr>
<tr>
<td>get_integration_response</td>
<td>Gets an IntegrationResponses</td>
</tr>
<tr>
<td>get_integration_responses</td>
<td>Gets the IntegrationResponses for an Integration</td>
</tr>
<tr>
<td>get_integrations</td>
<td>Gets the Integrations for an API</td>
</tr>
<tr>
<td>get_model</td>
<td>Gets a Model</td>
</tr>
<tr>
<td>get_models</td>
<td>Gets the Models for an API</td>
</tr>
<tr>
<td>get_modelTemplate</td>
<td>Gets a model template</td>
</tr>
</tbody>
</table>
applicationautoscaling

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
get_vpc_link          Gets a VPC link
get_vpc_links         Gets a collection of VPC links
import_api            Imports an API
reimport_api          Puts an Api resource
reset_authorizers_cache Resets all authorizer cache entries on a stage
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
update_vpc_link       Updates a VPC link

Examples

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

## End(Not run)
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 and entity recognizer endpoints
- AWS Lambda function provisioned concurrency
- Amazon Keyspaces (for Apache Cassandra) tables
- Amazon Managed Streaming for Apache Kafka cluster storage

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 `register_scalable_target` API action for any Application Auto Scaling scalable target. You can suspend and resume (individually or in combination) scale-out activities that are triggered by a scaling policy, scale-in activities that are 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.
Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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

- `delete_scaling_policy` Deletes the specified scaling policy for an Application Auto Scaling scalable target
- `delete_scheduled_action` Deletes the specified scheduled action for an Application Auto Scaling scalable target
- `deregister_scalable_target` Deregisters an Application Auto Scaling scalable target when you have finished using it
- `describe_scalable_targets` Gets information about the scalable targets in the specified namespace
- `describe_scaling_activities` Provides descriptive information about the scaling activities in the specified namespace from the previous six weeks
- `describe_scaling_policies` Describes the Application Auto Scaling scaling policies for the specified service namespace
- `describe_scheduled_actions` Describes the Application Auto Scaling scheduled actions for the specified service namespace
- `put_scaling_policy` Creates or updates a scaling policy for an Application Auto Scaling scalable target
- `put_scheduled_action` Creates or updates a scheduled action for an Application Auto Scaling scalable target
- `register_scalable_target` Registers or updates a scalable target

Examples

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

Description

Amazon CloudWatch Application Insights is a service that helps you detect common problems with your 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 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.

Usage

applicationinsights(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
create_component
create_log_pattern
delete_application
delete_component
delete_log_pattern
describe_application
describe_component
describe_component_configuration
describe_component_configuration_recommendation
describe_log_pattern
describe_observation
describe_problem
describe_problem_observations
list_applications
list_components
list_configuration_history
list_log_patterns
list_log_pattern_sets
list_problems
list_tags_for_resource
tag_resource
untag_resource
update_application
update_component
update_component_configuration
update_log_pattern

Examples

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

## End(Not run)
Description

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

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

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

Usage

appmesh(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

Examples

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

Amazon AppStream

Description

Amazon AppStream 2.0

This is the Amazon AppStream 2.0 API Reference. This documentation provides descriptions and syntax for each of the actions and data types in AppStream 2.0. AppStream 2.0 is a fully managed, secure application streaming service that lets you stream desktop applications to users without rewriting applications. AppStream 2.0 manages the AWS resources that are required to host and run your applications, scales automatically, and provides access to your users on demand.

You can call the AppStream 2.0 API operations by using an interface VPC endpoint (interface endpoint). For more information, see Access AppStream 2.0 API Operations and CLI Commands Through an Interface VPC Endpoint in the Amazon AppStream 2.0 Administration Guide.

To learn more about AppStream 2.0, see the following resources:

- Amazon AppStream 2.0 product page
- Amazon AppStream 2.0 documentation

Usage

```r
appstream(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

Operations

associate_fleet Associates the specified fleet with the specified stack
batch_associate_user_stack Associates the specified users with the specified stacks
batch_disassociate_user_stack Disassociates the specified users from the specified stacks
copy_image Copies the image within the same region or to a new region within the same AWS account
create_directory_config Creates a Directory Config object in AppStream 2
create_fleet Creates a fleet
create_image_builder Creates an image builder
create_image_builder_streaming_url Creates a URL to start an image builder streaming session
create_stack Creates a stack to start streaming applications to users
create_streaming_url Creates a temporary URL to start an AppStream 2
create_usage_report_subscription Creates a usage report subscription
create_user Creates a new user in the user pool
delete_directory_config Deletes the specified Directory Config object from AppStream 2
delete_fleet Deletes the specified fleet
delete_image Deletes the specified image
delete_image_builder Deletes the specified image builder and releases the capacity
delete_image_permissions Deletes permissions for the specified private image
delete_stack Deletes the specified stack
delete_usage_report_subscription Disables usage report generation
delete_user Deletes a user from the user pool
describe_directory_configs Retrieves a list that describes one or more specified Directory Config objects for AppStream 2
describe_fleets Retrieves a list that describes one or more specified fleets, if the fleet names are provided
describe_image_builders Retrieves a list that describes one or more specified image builders, if the image builder names are provided
describe_image_permissions Retrieves a list that describes the permissions for shared AWS account IDs on a private image
describe_images Retrieves a list that describes one or more specified images, if the image names or image ARNs are provided
describe_sessions Retrieves a list that describes the streaming sessions for a specified stack and fleet
describe_stacks Retrieves a list that describes one or more specified stacks, if the stack names are provided
describe_usage_report_subscriptions Retrieves a list that describes one or more usage report subscriptions
describe_user_stack_associations Retrieves a list that describes one or more specified users in the user pool
disable_user Disables the specified user in the user pool
disassociate_fleet Disassociates the specified fleet from the specified stack
disable_user_stack_associations Disables the specified user in the user pool
disable_user
enable_user Enables a user in the user pool
expire_session Immediately stops the specified streaming session
list_associated_fleets Retrieves the name of the fleet that is associated with the specified stack
list Associated_stacks Retrieves the name of the stack with which the specified fleet is associated
list_tags_for_resource Retrieves a list of all tags for the specified AppStream 2
start_fleet Starts the specified fleet
**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**

<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>
Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
svc <- 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_data_catalog**: Creates (registers) a data catalog with the specified name and properties.
- **create_named_query**: Creates a named query in the specified workgroup.
- **create_work_group**: Creates a workgroup with the specified name.
- **delete_data_catalog**: Deletes a data catalog.
- **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_database**: Returns a database object for the specified database and data catalog.
- **get_data_catalog**: Returns the specified data catalog.
- **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_table_metadata**: Returns table metadata for the specified catalog, database, and table.
- **get_work_group**: Returns information about the workgroup with the specified name.
- **list_databases**: Lists the databases in the specified data catalog.
- **list_data_catalogs**: Lists the data catalogs in the current AWS account.
- **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_table_metadata**: Lists the metadata for the tables in the specified data catalog database.
- **list_tags_for_resource**: Lists the tags associated with an Athena workgroup or data catalog resource.
- **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.
autoscaling

tag_resource
untag_resource
update_data_catalog
update_work_group

Add one or more tags to an Athena resource
Removes one or more tags from a data catalog or workgroup resource
Updates the data catalog that has the specified name
Updates the workgroup with the specified name

Examples

## 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

autoscaling(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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` To attach an Application Load Balancer, Network Load Balancer, or Gateway Load Balancer, use the AttachLoadBalancerTargetGroups API operation instead
- `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
- `cancel_instance_refresh` Cancels an instance refresh operation in progress
- `complete_lifecycle_action` Completes the lifecycle action for the specified token or instance with the specified result
- `create_auto_scaling_group` We strongly recommend using a launch template when calling this operation to ensure full functionality for Amazon EC2 Auto Scaling and Amazon EC2
- `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 quotas for your AWS account
- `describe_adjustment_types` Describes the available adjustment types for Amazon EC2 Auto Scaling scaling policies
- `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_instance_refreshes` Describes one or more instance refreshes
- `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 load balancers for the specified Auto Scaling group
- `describe_load_balancer_target_groups` Describes the target groups for the specified Auto Scaling group
- `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
describe_scaling_activities
describe_scaling_process_types
describe_scheduled_actions
describe_tags
describe_termination_policy_types
detach_instances
detach_load_balancers
detach_load_balancer_target_groups
disable_metrics_collection
enable_metrics_collection
enter_standby
execute_policy
exit_standby
put_lifecycle_hook
put_notification_configuration
put_scaling_policy
put_scheduled_update_group_action
record_lifecycle_action_heartbeat
resume_processes
set_desired_capacity
set_instance_health
set_instance_protection
start_instance_refresh
suspend_processes
terminate_instance_in_auto_scaling_group
update_auto_scaling_group

Describes the policies for the specified Auto Scaling group
Describes one or more scaling activities for the specified Auto Scaling group
Describes the scaling process types for use with the ResumeProcesses and SuspendProcesses APIs
Describes the actions scheduled for your Auto Scaling group that haven’t run or that ran and aren’t running anymore
Describes the specified tags
Describes the termination policies supported by Amazon EC2 Auto Scaling
Removes one or more instances from the specified Auto Scaling group
Detaches one or more Classic Load Balancers from the specified Auto Scaling group
Detaches one or more target groups from the specified Auto Scaling group
Disables group metrics for the specified Auto Scaling group
Enables group metrics for the specified Auto Scaling group
Moves the specified instances into the standby state
Executes the specified policy
Moves the specified instances out of the standby state
Creates or updates a lifecycle hook for the specified Auto Scaling group
Configures an Auto Scaling group to send notifications when specified events take place
Creates or updates a scaling policy for an Auto Scaling group
Creates or updates a scheduled scaling action for an Auto Scaling group
Records a heartbeat for the lifecycle action associated with the specified token or instance
Resumes the specified suspended auto scaling processes, or all suspended processes, for the specified Auto Scaling group
Sets the size of the specified Auto Scaling group
Sets the health status of the specified instance
Updates the instance protection settings of the specified instances
Starts a new instance refresh operation, which triggers a rolling replacement of all instances in the Auto Scaling group according to the specified refresh strategy
Suspends the specified auto scaling processes, or all processes, for the specified Auto Scaling group
Terminates the specified instance and optionally adjusts the desired group size
We strongly recommend that all Auto Scaling groups use launch templates to enable

Examples

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

## End(Not run)
```
Description

AWS Auto Scaling

Use AWS Auto Scaling to create scaling plans for your applications to automatically scale your scalable AWS resources.

API Summary

You can use the AWS Auto Scaling service API to accomplish the following tasks:

- Create and manage scaling plans
- Define target tracking scaling policies to dynamically scale your resources based on utilization
- Scale Amazon EC2 Auto Scaling groups using predictive scaling and dynamic scaling to scale your Amazon EC2 capacity faster
- Set minimum and maximum capacity limits
- Retrieve information on existing scaling plans
- Access current forecast data and historical forecast data for up to 56 days previous

To learn more 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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

backup(config = list())

Arguments

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

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```bash
svc <- 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` Creates a backup plan using a backup plan name and backup rules
- `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` Deletes a backup plan
- `delete_backup_selection` Deletes the resource selection associated with a backup plan that is specified by the SelectionId
- `delete_backup_vault` Deletes the backup vault identified by its name
- `delete_backup_vault_access_policy` Deletes the policy document that manages permissions on a backup vault
- `delete_backup_vault_notifications` Deletes event notifications for the specified backup vault
- `delete_recovery_point` Deletes the recovery point specified by a recovery point ID
- `describe_backup_job` Returns backup job details for the specified BackupJobId
- `describe_backup_vault` Returns metadata about a backup vault specified by its name
- `describe_copy_job` Returns metadata associated with creating a copy of a resource
- `describe_global_settings` The current feature settings for the AWS Account
- `describe_protected_resource` Returns information about a saved resource, including the last time it was backed up, and the ARN
- `describe_recovery_point` Returns metadata associated with a recovery point, including ID, status, encryption, and lifecycle
- `describe_region_settings` Returns the current service opt-in settings for the Region
- `describe_restore_job` Returns metadata associated with a restore job that is specified by a job ID
- `export_backup_plan_template` Returns the backup plan that is specified by the plan ID as a backup template
- `get_backup_plan` Returns BackupPlan details for the specified BackupPlanId
- `get_backup_plan_from_json` Returns a valid JSON document specifying a backup plan or an error
- `get_backup_plan_from_template` Returns the template specified by its templateId as a backup plan
- `get_backup_selection` Returns selection metadata and a document in JSON format that specifies a list of resources
- `get_backup_vault_access_policy` Returns the access policy document that is associated with the named backup vault
- `get_backup_vault_notifications` Returns event notifications for the specified backup vault
Examples

```r
## Not run:
svc <- backup()
svc$create_backup_plan(
  Foo = 123
)

## End(Not run)
```

Description

Using AWS Batch, you can run batch computing workloads on the AWS Cloud. Batch computing is a common means for developers, scientists, and engineers to access large amounts of compute
resources. AWS Batch utilizes the advantages of this computing workload to remove the undifferentiated heavy lifting of configuring and managing required infrastructure, while also adopting a familiar batch computing software approach. Given these advantages, AWS Batch can help you to efficiently provision resources in response to jobs submitted, thus effectively helping to eliminate capacity constraints, reduce compute costs, and deliver your results more quickly.

As a fully managed service, AWS Batch can run batch computing workloads of any scale. AWS Batch automatically provisions compute resources and optimizes workload distribution based on the quantity and scale of your specific workloads. With AWS Batch, there’s no need to install or manage batch computing software. This means that you can focus your time and energy on analyzing results and solving your specific problems.

Usage

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

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
csvc <- 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` Cancels a job in an AWS Batch job queue
- `create_compute_environment` Creates an AWS Batch compute environment
- `create_job_queue` Creates an AWS Batch job queue
- `delete_compute_environment` Deletes an AWS Batch compute environment
- `delete_job_queue` Deletes the specified job queue
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deregister_job_definition</td>
<td>Deregisters an AWS Batch job definition</td>
</tr>
<tr>
<td>describe_compute_environments</td>
<td>Describes one or more of your compute environments</td>
</tr>
<tr>
<td>describe_job_definitions</td>
<td>Describes a list of job definitions</td>
</tr>
<tr>
<td>describe_job_queues</td>
<td>Describes one or more of your job queues</td>
</tr>
<tr>
<td>describe_jobs</td>
<td>Describes a list of AWS Batch jobs</td>
</tr>
<tr>
<td>list_jobs</td>
<td>Returns a list of AWS Batch jobs</td>
</tr>
<tr>
<td>list_tags_for_resource</td>
<td>Lists the tags for an AWS Batch resource</td>
</tr>
<tr>
<td>register_job_definition</td>
<td>Registers an AWS Batch job definition</td>
</tr>
<tr>
<td>submit_job</td>
<td>Submits an AWS Batch job from a job definition</td>
</tr>
<tr>
<td>tag_resource</td>
<td>Associates the specified tags to a resource with the specified resourceArn</td>
</tr>
<tr>
<td>terminate_job</td>
<td>Terminates a job in a job queue</td>
</tr>
<tr>
<td>untag_resource</td>
<td>Deletes specified tags from an AWS Batch resource</td>
</tr>
<tr>
<td>update_compute_environment</td>
<td>Updates an AWS Batch compute environment</td>
</tr>
<tr>
<td>update_job_queue</td>
<td>Updates a job queue</td>
</tr>
</tbody>
</table>

**Examples**

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

## End(Not run)
```

---

**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

```r
budgets(config = list())
```

### Arguments

**config**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```r
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_budget_action**: Creates a budget action
- **create_notification**: Creates a notification
- **create_subscriber**: Creates a subscriber
- **delete_budget**: Deletes a budget
- **delete_budget_action**: Deletes a budget action
- **delete_notification**: Deletes a notification
- **delete_subscriber**: Deletes a subscriber
- **describe_budget**: Describes a budget
- **describe_budget_action**: Describes a budget action detail
- **describe_budget_action_histories**: Describes a budget action history detail
- **describe_budget_actions_for_account**: Describes all of the budget actions for an account
- **describe_budget_actions_for_budget**: Describes all of the budget actions for a budget
- **describe_budget_performance_history**: Describes the history for DAILY, MONTHLY, and QUARTERLY budgets
- **describe_budgets**: Lists the budgets that are associated with an account
- **describe_notifications_for_budget**: Lists the notifications that are associated with a budget
- **describe_subscribers_for_notification**: Lists the subscribers that are associated with a notification
- **execute_budget_action**: Executes a budget action
- **update_budget**: Updates a budget
- **update_budget_action**: Updates a budget action
- **update_notification**: Updates a notification
- **update_subscriber**: Updates a subscriber

**Examples**

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

**Description**

AWS Cloud9 is a collection of tools that you can use to code, build, run, test, debug, and release software in the cloud.

For more information about AWS Cloud9, see the [AWS Cloud9 User Guide](https://docs.aws.amazon.com/cloud9/latest/userguide/). AWS Cloud9 supports these operations:
• `create_environment_ec2`: Creates an AWS Cloud9 development environment, launches an Amazon EC2 instance, and then connects from the instance to the environment.

• `create_environment_membership`: Adds an environment member to an environment.

• `delete_environment`: Deletes an environment. If an Amazon EC2 instance is connected to the environment, also terminates the instance.

• `delete_environment_membership`: Deletes an environment member from an environment.

• `describe_environment_memberships`: Gets information about environment members for an environment.

• `describe_environments`: Gets information about environments.

• `describe_environment_status`: Gets status information for an environment.

• `list_environments`: Gets a list of environment identifiers.

• `list_tags_for_resource`: Gets the tags for an environment.

• `tag_resource`: Adds tags to an environment.

• `untag_resource`: Removes tags from an environment.

• `update_environment`: Changes the settings of an existing environment.

• `update_environment_membership`: Changes the settings of an existing environment member for an environment.

Usage

```python
cloud9(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
csvc <- cloud9(
    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_environment_ec2` creates an AWS Cloud9 development environment, launches an Amazon Elastic Compute Cloud (Amazon EC2) instance, and then connects from the instance to the environment.
- `create_environment_membership` adds an environment member to an AWS Cloud9 development environment.
- `delete_environment` deletes an AWS Cloud9 development environment.
- `delete_environment_membership` deletes an environment member from an AWS Cloud9 development environment.
- `describe_environment_memberships` gets information about environment members for an AWS Cloud9 development environment.
- `describe_environments` gets information about AWS Cloud9 development environments.
- `describe_environment_status` gets status information for an AWS Cloud9 development environment.
- `list_environments` gets a list of AWS Cloud9 development environment identifiers.
- `list_tags_for_resource` gets a list of the tags associated with an AWS Cloud9 development environment.
- `tag_resource` adds tags to an AWS Cloud9 development environment.
- `untag_resource` removes tags from an AWS Cloud9 development environment.
- `update_environment` changes the settings of an existing AWS Cloud9 development environment.
- `update_environment_membership` changes the settings of an existing environment member for an AWS Cloud9 development environment.

Examples

```r
## Not run:
svc <- cloud9()
#
svc$create_environment_ec2(
  name = "my-demo-environment",
  automaticStopTimeMinutes = 60L,
  description = "This is my demonstration environment.",
  instanceType = "t2.micro",
  ownerArn = "arn:aws:iam::123456789012:user/MyDemoUser",
  subnetId = "subnet-1fab8aEX"
)
## End(Not run)
```

Description

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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
delete_schema
delete_typed_link_facet
detach_from_index
detach_object
detach_policy
detach_typed_link
disable_directory
enable_directory
get_applied_schema_version
get_directory
get_facet
get_link_attributes
get_object_attributes
get_object_information
get_schema_as_json
get_typed_link_facet_information
list_applied_schema_arns
list_attached_indices
list_development_schema_arns
list_directories
list_facet_attributes
list_facet_names
list_incoming_typed_links
list_index
list_managed_schema_arns
list_object_attributes
list_object_children
list_object_parent_paths
list_object_policies
list_outgoing_typed_links
list_policy_attachments
list_published_schema_arns
list_tags_for_resource
list_typed_link_facet_attributes
list_typed_link_facet_names
lookup_policy
publish_schema
put_schema_from_json
remove_facet_from_object
tag_resource
untag_resource
update_facet
update_link_attributes
update_object_attributes
update_schema
update_typed_link_facet

deletes an object and its associated attributes
Deletes a given schema
Deletes a TypedLinkFacet
Detaches the specified object from the specified index
Detaches a given object from the parent object
Detaches a policy from an object
Detaches a typed link from a specified source and target object
Disables the specified directory
Enables the specified directory
Returns current applied schema version ARN, including the minor version in use
Retrieves metadata about a directory
Gets details of the Facet, such as facet name, attributes, Rules, or ObjectType
Retrieves attributes that are associated with a typed link
Retrieves attributes within a facet that are associated with an object
Retrieves metadata about an object
Retrieves a JSON representation of the schema
Returns the identity attribute order for a specific TypedLinkFacet
Lists schema major versions applied to a directory
Lists indices attached to the specified object
Retrieves each Amazon Resource Name (ARN) of schemas in the development state
Lists directories created within an account
Retrieves attributes attached to the facet
Retrieves all of the ObjectIdentifiers to which a given policy is attached
Lists the major version families of each managed schema
Lists all attributes that are associated with an object
Returns a paginated list of child objects that are associated with a given object
Retrieves all available parent paths for any object type such as node, leaf node, policy node
Lists parent objects that are associated with a given object in pagination fashion
Returns policies attached to an object in pagination fashion
Returns a paginated list of all the outgoing TypedLinkSpecifier information for an object
Returns all of the ObjectIdentifiers to which a given policy is attached
Lists the major version families of each published schema
Returns tags for a resource
Returns a paginated list of all attribute definitions for a particular TypedLinkFacet
Returns a paginated list of TypedLink facet names for a particular schema
Lists all policies from the root of the Directory to the object specified
Publishes a development schema with a major version and a recommended minor version
Allows a schema to be updated using JSON upload
Removes the specified facet from the specified object
An API operation for adding tags to a resource
An API operation for removing tags from a resource
Does the following:
Updates a given typed link's attributes
Updates a given object’s attributes
Updates the schema name with a new name
Updates a TypedLinkFacet
### 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](https://docs.aws.amazon.com). 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](https://docs.aws.amazon.com).

### Usage

```r
cloudformation(config = list())
```

### Arguments

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
cloudformation

Service syntax

```r
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**: Cancels an update on the specified stack
- **continue_update_rollback**: For a specified stack that is in the UPDATE_ROLLBACK_FAILED state, continues rolling it back to the UPDATE_ROLLBACK_COMPLETE state
- **create_change_set**: Creates a list of changes that will be applied to a stack so that you can review the changes before executing them
- **create_stack**: Creates a stack as specified in the template
- **create_stack_instances**: Creates stack instances for the specified accounts, within the specified Regions
- **create_stack_set**: Creates a stack set
- **delete_change_set**: Deletes the specified change set
- **delete_stack**: Deletes a specified stack
- **delete_stack_instances**: Deletes stack instances for the specified accounts, in the specified Regions
- **delete_stack_set**: Deletes a stack set
- **deregister_type**: Removes a type or type version from active use in the CloudFormation registry
- **describe_account_limits**: Retrieves your account's AWS CloudFormation limits, such as the maximum number of stacks that you can create in your account
- **describe_change_set**: Returns the inputs for the change set and a list of changes that AWS CloudFormation will make if you execute the change set
- **describe_stack_drift_detection_status**: Returns information about a stack drift detection operation
- **describe_stack_events**: Returns all stack related events for a specified stack in reverse chronological order
- **describe_stack_instance**: Returns the stack instance that's associated with the specified stack set, AWS account, and Region
- **describe_stack_resource**: Returns a description of the specified resource in the specified stack
- **describe_stack_resource_drifts**: Returns drift information for the resources that have been checked for drift in the specified stack
- **describe_stack_resources**: Returns AWS resource descriptions for running and deleted stacks
- **describe_stacks**: Returns the description for the specified stack; if no stack name was specified, then it returns the description for all the stacks created
- **describe_stack_set**: Returns the description of the specified stack set
- **describe_stack_set_operation**: Returns the description of the specified stack set operation
- **describe_type**: Returns detailed information about a type that has been registered
- **describe_type_registration**: Returns information about a type's registration, including its current status and type and version identifiers
- **detect_stack_drift**: Detects whether a stack's actual configuration differs, or has drifted, from its expected configuration, as defined in the stack template and any values specified as template parameters
- **detect_stack_resource_drift**: Returns information about whether a resource's actual configuration differs, or has drifted, from its expected configuration, as defined in the stack template and any values specified as template parameters
- **detect_stack_set_drift**: Detects drift on a stack set
- **estimate_template_cost**: Returns the estimated monthly cost of a template
- **execute_change_set**: Updates a stack using the input information that was provided when the specified change set was created
- **get_stack_policy**: Returns the stack policy for a specified stack
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
signal_resource
stop_stack_set_operation
update_stack
update_stack_instances
update_stack_set
update_termination_protection
validate_template

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 StackStatusFilter
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(s)
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
Sends a signal to the specified resource with a success or failure status
 Stops an in-progress operation on a stack set and its associated stack instances
Updates a stack as specified in the template
Updates the parameter values for stack instances for the specified accounts, within the specified Regions
Updates the stack set, and associated stack instances in the specified accounts and Regions
Updates termination protection for the specified stack
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

cloudfront(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Operations

create_cache_policy Creates a cache policy
create_cloud_front_origin_access_identity Creates a new origin access identity
create_distribution Creates a new web distribution
create_distribution_with_tags Create a new distribution with tags
create_field_level_encryption_config Create a new field-level encryption configuration
create_field_level_encryption_profile Create a field-level encryption profile
create_invalidation Create a new invalidation
create_key_group Creates a key group that you can use with CloudFront signed URLs and signed cookies
create_monitoring_subscription Enables additional CloudWatch metrics for the specified CloudFront distribution
create_origin_request_policy Creates an origin request policy
create_public_key Uploads a public key to CloudFront that you can use with signed URLs and signed cookies
create_realtime_log_config Creates a real-time log configuration
create_streaming_distribution This API is deprecated
create_streaming_distribution_with_tags This API is deprecated
delete_cache_policy Deletes a cache policy
delete_cloud_front_origin_access_identity Delete an origin access identity
Delete a distribution
Remove a field-level encryption configuration
Remove a field-level encryption profile
Deletes a key group
Disables additional CloudWatch metrics for the specified CloudFront distribution
Deletes an origin request policy
Remove a public key you previously added to CloudFront
Deletes a real-time log configuration
Delete a streaming distribution

Gets a cache policy, including the following metadata:
Gets a cache policy configuration
Get the information about an origin access identity
Get the configuration information about an origin access identity
Get the information about a distribution
Get the configuration information about a distribution
Get the field-level encryption configuration information
Get the field-level encryption configuration information
Get the field-level encryption profile information
Get the field-level encryption profile configuration information
Get the information about an invalidation
Gets a key group, including the date and time when the key group was last modified
Gets a key group configuration
Gets information about whether additional CloudWatch metrics are enabled
Gets an origin request policy, including the following metadata:
Gets an origin request policy configuration
Gets a public key
Gets a public key configuration
Gets a real-time log configuration
Gets information about a specified RTMP distribution, including the distribution configuration
Get the configuration information about a streaming distribution
Gets a list of cache policies
Lists origin access identities
List CloudFront distributions
Gets a list of distribution IDs for distributions that have a cache behavior that
Gets a list of distribution IDs for distributions that have a cache behavior that
Gets a list of distribution IDs for distributions that have a cache behavior that
Gets a list of distributions that have a cache behavior that’s associated with the
List the distributions that are associated with a specified AWS WAF web ACL
List all field-level encryption configurations that have been created in CloudFront
Request a list of field-level encryption profiles that have been created in CloudFront
Lists invalidation batches
Gets a list of key groups
Gets a list of origin request policies
List all public keys that have been added to CloudFront for this account
Gets a list of real-time log configurations
List streaming distributions
List tags for a CloudFront resource
Add tags to a CloudFront resource
untag_resource
update_cache_policy
update_cloud_front_origin_access_identity
update_distribution
update_field_level_encryption_config
update_field_level_encryption_profile
update_key_group
update_origin_request_policy
update_public_key
update_realtime_log_config
update_streaming_distribution

Remove tags from a CloudFront resource
Updates a cache policy configuration
Update an origin access identity
Updates the configuration for a web distribution
Update a field-level encryption configuration
Update a field-level encryption profile
Updates a key group
Updates an origin request policy configuration
Update public key information
Updates a real-time log configuration
Update a streaming distribution

Examples

```r
## Not run:
svc <- cloudfront()
svc$create_cache_policy(
  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

`cloudhsm(config = list())`

Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
svc <- cloudhsm(
  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` 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)
```

---

**cloudhsmv2**  
**AWS CloudHSM V2**

## Description

For more information about AWS CloudHSM, see [AWS CloudHSM](https://aws.amazon.com/cloudhsm/) and the [AWS CloudHSM User Guide](https://docs.aws.amazon.com/cloudhsm/latest/userguide/).

## Usage

```r
cloudhsmv2(config = list())
```

## Arguments

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

## Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```r
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 certificate authority (CA) and the CA's root certificate
- **list_tags**: Gets a list of tags for the specified AWS CloudHSM cluster
- **modify_backup_attributes**: Modifies attributes for AWS CloudHSM backup
- **modify_cluster**: Modifies 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

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

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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Operations

- `build_suggesters`: Indexes the search suggestions
- `create_domain`: Creates a new search domain
- `define_analysis_scheme`: Configures an analysis scheme that can be applied to a text or text-array field to define language-specific text processing options
- `define_expression`: Configures an Expression for the search domain
- `define_index_field`: Configures an IndexField for the search domain
- `define_suggester`: Configures a suggester for a domain
- `delete_analysis_scheme`: Deletes an analysis scheme
- `delete_domain`: Permanently deletes a search domain and all of its data
- `delete_expression`: Removes an Expression from the search domain
- `delete_index_field`: Removes an IndexField from the search domain
- `delete_suggester`: Deletes a suggester
- `describe_analysis_schemes`: Gets the analysis schemes configured for a domain
- `describe_availability_options`: Gets the availability options configured for a domain
- `describe_domain_endpoint_options`: Returns the domain’s endpoint options, specifically whether all requests to the domain must arrive over HTTPS.
- `describe_domains`: Gets information about the search domains owned by this account
- `describe_expressions`: Gets the expressions configured for the search domain
- `describe_index_fields`: Gets information about the index fields configured for the search domain
- `describe_scaling_parameters`: Gets the scaling parameters configured for a domain
- `describe_service_access_policies`: Gets information about the access policies that control access to the domain’s document and search endpoints
- `describe_suggesters`: Gets the suggesters configured for a domain
### Examples

```r
## Not run:
svc <- cloudsearch()
svc$build_suggesters(
  Foo = 123
)

## End(Not run)
```

---

### cloudsearchdomain

#### Amazon CloudSearch Domain

#### Description

You use the AmazonCloudSearch2013 API to upload documents to a search domain and search those documents.

The endpoints for submitting `upload_documents`, `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.

#### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```
svc <- 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.

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

```r
cloudtrail(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
csvc <- 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

- **add_tags**: Adds one or more tags to a trail, up to a limit of 50
- **create_trail**: Creates a trail that specifies the settings for delivery of log data to an Amazon S3 bucket
- **delete_trail**: Deletes a trail
- **describe_trails**: Retrieves settings for one or more trails associated with the current region for your account
- **get_event_selectors**: Describes the settings for the event selectors that you configured for your trail
- **get_insight_selectors**: Describes the settings for the Insights event selectors that you configured for your trail
- **get_trail**: Returns settings information for a specified trail
### cloudwatch

**get_trail_status**  
Returns a JSON-formatted list of information about the specified trail  

**list_public_keys**  
Returns all public keys whose private keys were used to sign the digest files within the specified time range  

**list_tags**  
Lists the tags for the trail in the current region  

**list_trails**  
Lists trails that are in the current account  

**lookup_events**  
Looks up management events or CloudTrail Insights events that are captured by CloudTrail  

**put_trail_status**  
Updates the settings that specify delivery of log files  

**put_event_selectors**  
Configures an event selector or advanced event selectors for your trail  

**put_insight_selectors**  
Lets you enable Insights event logging by specifying the Insights selectors that you want to enable on your trail  

**remove_tags**  
Removes the specified tags from a trail  

**start_logging**  
Starts the recording of AWS API calls and log file delivery for a trail  

**stop_logging**  
Suspends the recording of AWS API calls and log file delivery for the specified trail  

**update_trail**  
Updates the settings that specify delivery of log files

---

### 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

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

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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 500 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
cloudwatchevents

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

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

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

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- cloudwatchevents(
  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
cancel_replay Cancels the specified replay
create_archive Creates an archive of events with the specified settings
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 You can use this operation to temporarily stop receiving events from the specified partner event source
delete_archive Deletes the specified archive
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_archive Retrieves details about an archive
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 the partner has created
describe_replay Retrieves details about a replay
describe_rule Describes the specified rule
disable_rule Disables the specified rule
enable_rule Enables the specified rule
list_archives Lists your archives
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
list_partner_event_source_accounts
list_partner_event_sources
list_replays
list_rule_names_by_target
list_rules
list_tags_for_resource
list_targets_by_rule
put_events
put_partner_events
put_permission
put_rule
put_targets
remove_permission
remove_targets
start_replay
tag_resource
test_event_pattern
untag_resource
update_archive

You can use this to see all the partner event sources that have been shared with your AWS account.
An SaaS partner can use this operation to display the AWS account ID that a particular partner event source is associated with.
An SaaS partner can use this operation to list all the partner event source names that the partner has created.
Lists your replays.
Lists the rules for the specified target.
Lists your Amazon EventBridge rules.
Displays the tags associated with an EventBridge resource.
Lists the targets assigned to the specified rule.
Sends custom events to Amazon EventBridge so that they can be matched to rules.
This is used by SaaS partners to write events to a customer’s partner event bus.
Running PutPermission permits the specified AWS account or AWS organization to put events to the specified event bus.
Creates or updates the specified rule.
Adds the specified targets to the specified rule, or updates the targets if they are already associated with the rule.
Revolves the permission of another AWS account to be able to put events to the specified event bus.
Removes the specified targets from the specified rule.
Starts the specified replay.
Assigns one or more tags (key-value pairs) to the specified EventBridge resource.
Tests whether the specified event pattern matches the provided event.
Removes one or more tags from the specified EventBridge resource.
Updates the specified archive.

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 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. You can 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.`

**Value**

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```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**
## Not run:

```r
svc <- cloudwatchlogs()
```
```python
svc$associate_kms_key(
    Foo = 123
)

## End(Not run)
```

---

codebuild  AWS CodeBuild

description

AWS CodeBuild is a fully managed build service in the cloud. AWS CodeBuild compiles your source code, runs unit tests, and produces artifacts that are ready to deploy. AWS CodeBuild eliminates the need to provision, manage, and scale your own build servers. It provides prepackaged build environments for the most popular programming languages and build tools, such as Apache Maven, Gradle, and more. You can also fully customize build environments in AWS CodeBuild to use your own build tools. AWS CodeBuild scales automatically to meet peak build requests. You pay only for the build time you consume. For more information about AWS CodeBuild, see the AWS CodeBuild User Guide.

AWS CodeBuild supports these operations:

- **batch_delete_builds**: Deletes one or more builds.
- **batch_get_builds**: Gets information about one or more builds.
- **batch_get_projects**: Gets information about one or more build projects. A build project defines how AWS CodeBuild runs a build. This includes information such as where to get the source code to build, the build environment to use, the build commands to run, and where to store the build output. A build environment is a representation of operating system, programming language runtime, and tools that AWS CodeBuild uses to run a build. You can add tags to build projects to help manage your resources and costs.
- **batch_get_report_groups**: Returns an array of report groups.
- **batch_get_reports**: Returns an array of reports.
- **create_project**: Creates a build project.
- **create_report_group**: Creates a report group. A report group contains a collection of reports.
- **create_webhook**: For an existing AWS CodeBuild build project that has its source code stored in a GitHub or Bitbucket repository, enables AWS CodeBuild to start rebuilding the source code every time a code change is pushed to the repository.
- **delete_project**: Deletes a build project.
- **delete_report**: Deletes a report.
- **delete_report_group**: Deletes a report group.
- **delete_resource_policy**: Deletes a resource policy that is identified by its resource ARN.
- **delete_source_credentials**: Deletes a set of GitHub, GitHub Enterprise, or Bitbucket source credentials.
• **delete_webhook**: For an existing AWS CodeBuild build project that has its source code stored in a GitHub or Bitbucket repository, stops AWS CodeBuild from rebuilding the source code every time a code change is pushed to the repository.

• **describe_test_cases**: Returns a list of details about test cases for a report.

• **get_resource_policy**: Gets a resource policy that is identified by its resource ARN.

• **import_source_credentials**: Imports the source repository credentials for an AWS CodeBuild project that has its source code stored in a GitHub, GitHub Enterprise, or Bitbucket repository.

• **invalidate_project_cache**: Resets the cache for a project.

• **list_builds**: Gets a list of build IDs, with each build ID representing a single build.

• **list_builds_for_project**: Gets a list of build IDs for the specified build project, with each build ID representing a single build.

• **list_curated_environment_images**: Gets information about Docker images that are managed by AWS CodeBuild.

• **list_projects**: Gets a list of build project names, with each build project name representing a single build project.

• **list_report_groups**: Gets a list ARNs for the report groups in the current AWS account.

• **list_reports**: Gets a list ARNs for the reports in the current AWS account.

• **list_reports_for_report_group**: Returns a list of ARNs for the reports that belong to a ReportGroup.

• **list_shared_projects**: Gets a list of ARNs associated with projects shared with the current AWS account or user.

• **list_shared_report_groups**: Gets a list of ARNs associated with report groups shared with the current AWS account or user.

• **list_source_credentials**: Returns a list of SourceCredentialsInfo objects. Each SourceCredentialsInfo object includes the authentication type, token ARN, and type of source provider for one set of credentials.

• **put_resource_policy**: Stores a resource policy for the ARN of a Project or ReportGroup object.

• **start_build**: Starts running a build.

• **stop_build**: Attempts to stop running a build.

• **update_project**: Changes the settings of an existing build project.

• **update_report_group**: Changes a report group.

• **update_webhook**: Changes the settings of an existing webhook.

**Usage**

```python
codebuild(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>
Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...),
where svc is the name you’ve assigned to the client. The available operations are listed in the Op-
erations section.

Service syntax

```r
svc <- codebuild(
  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_builds**: Deletes one or more builds
- **batch_get_build_batches**: Retrieves information about one or more batch builds
- **batch_get_builds**: Gets information about one or more builds
- **batch_get_projects**: Gets information about one or more build projects
- **batch_get_report_groups**: Returns an array of report groups
- **batch_get_reports**: Returns an array of reports
- **create_project**: Creates a build project
- **create_report_group**: Creates a build project
- **create_webhook**: For an existing AWS CodeBuild build project that has its source code stored in a GitHub or Bitbucket repository, enables AWS CodeBuild to start rebuilding the source code every time a code change is pushed to the repository
- **delete_build_batch**: Deletes a batch build
- **delete_project**: Deletes a build project
- **delete_report**: Deletes a report
- **delete_report_group**: Deletes a report group
- **delete_resource_policy**: Deletes a resource policy that is identified by its resource ARN
- **delete_source_credentials**: Deletes a set of GitHub, GitHub Enterprise, or Bitbucket source credentials
- **delete_webhook**: For an existing AWS CodeBuild build project that has its source code stored in a GitHub or Bitbucket repository, stops AWS CodeBuild from rebuilding the source code every time a code change is pushed to the repository
- **describe_code_coverage**: Retrieves one or more code coverage reports
- **describe_test_cases**: Returns a list of details about test cases for a report
- **get_report_group_trend**: Gets a resource policy that is identified by its resource ARN
- **get_resource_policy**: Imports the source repository credentials for an AWS CodeBuild project that has its source code stored in a GitHub or Bitbucket repository
- **invalidate_project_cache**: Resets the cache for a project
- **list_build_batches**: Retrieves the identifiers of your build batches in the current region
- **list_build_batches_for_project**: Retrieves the identifiers of the build batches for a specific project
### CodeCommit API Reference

This is the AWS CodeCommit API Reference. This reference provides descriptions of the operations and data types for AWS CodeCommit API along with usage examples.

You can use the AWS CodeCommit API to work with the following objects:

Repositories, by calling the following:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_builds</td>
<td>Gets a list of build IDs, with each build ID representing a single build</td>
</tr>
<tr>
<td>list_builds_for_project</td>
<td>Gets a list of build IDs for the specified build project, with each build ID representing a single build</td>
</tr>
<tr>
<td>list_curated_environment_images</td>
<td>Gets information about Docker images that are managed by AWS CodeBuild</td>
</tr>
<tr>
<td>list_projects</td>
<td>Gets a list of build project names, with each build project name representing a single build project</td>
</tr>
<tr>
<td>list_report_groups</td>
<td>Gets a list of report groups that are shared with other AWS accounts or users</td>
</tr>
<tr>
<td>list_reports</td>
<td>Returns a list of ARNs for the reports that belong to a ReportGroup</td>
</tr>
<tr>
<td>list_reports_for_report_group</td>
<td>Returns a list of ARNs for the reports in the current AWS account</td>
</tr>
<tr>
<td>list_shared_projects</td>
<td>Gets a list of projects that are shared with other AWS accounts or users</td>
</tr>
<tr>
<td>list_shared_report_groups</td>
<td>Returns a list of SourceCredentialsInfo objects</td>
</tr>
<tr>
<td>list_source_credentials</td>
<td>Stores a resource policy for the ARN of a Project or ReportGroup object</td>
</tr>
<tr>
<td>put_resource_policy</td>
<td>Reverts a failed batch build</td>
</tr>
<tr>
<td>retry_build</td>
<td>Starts running a build</td>
</tr>
<tr>
<td>retry_build_batch</td>
<td>Starts a batch build for a project</td>
</tr>
<tr>
<td>start_build</td>
<td>Attempts to stop running a build</td>
</tr>
<tr>
<td>start_build_batch</td>
<td>Stops a running batch build</td>
</tr>
<tr>
<td>stop_build</td>
<td>Changes the settings of a build project</td>
</tr>
<tr>
<td>stop_build_batch</td>
<td>Updates a report group</td>
</tr>
<tr>
<td>update_project</td>
<td>Updates the webhook associated with an AWS CodeBuild build project</td>
</tr>
</tbody>
</table>

#### Examples

```r
## Not run:
svc <- codebuild()
# The following example gets information about builds with the specified # build IDs.
svc$batch_get_builds(
  ids = list(
    "codebuild-demo-project:9b0ac37f-d19e-4254-9879-f47e9a389eEX",
    "codebuild-demo-project:b79a46f7-1473-4636-a23f-da9c45c208EX"
  )
)

## End(Not run)
```
- batch_get_repositories, which returns information about one or more repositories associated with your AWS account.
- create_repository, which creates an AWS CodeCommit repository.
- delete_repository, which deletes an AWS CodeCommit repository.
- get_repository, which returns information about a specified repository.
- list_repositories, which lists all AWS CodeCommit repositories associated with your AWS account.
- update_repository_description, which sets or updates the description of the repository.
- update_repository_name, which changes the name of the repository. If you change the name of a repository, no other users of that repository can access it until you send them the new HTTPS or SSH URL to use.

Branches, by calling the following:

- create_branch, which creates a branch in a specified repository.
- delete_branch, which deletes the specified branch in a repository unless it is the default branch.
- get_branch, which returns information about a specified branch.
- list_branches, which lists all branches for a specified repository.
- update_default_branch, which changes the default branch for a repository.

Files, by calling the following:

- delete_file, which deletes the content of a specified file from a specified branch.
- get_blob, which returns the base-64 encoded content of an individual Git blob object in a repository.
- get_file, which returns the base-64 encoded content of a specified file.
- get_folder, which returns the contents of a specified folder or directory.
- put_file, which adds or modifies a single file in a specified repository and branch.

Commits, by calling the following:

- batch_get_commits, which returns information about one or more commits in a repository.
- create_commit, which creates a commit for changes to a repository.
- get_commit, which returns information about a commit, including commit messages and author and committer information.
- get_differences, which returns information about the differences in a valid commit specifier (such as a branch, tag, HEAD, commit ID, or other fully qualified reference).

Merges, by calling the following:

- batch_describe_merge_conflicts, which returns information about conflicts in a merge between commits in a repository.
- create_unreferenced_merge_commit, which creates an unreferenced commit between two branches or commits for the purpose of comparing them and identifying any potential conflicts.
• **describe_merge_conflicts**, which returns information about merge conflicts between the base, source, and destination versions of a file in a potential merge.

• **get_merge_commit**, which returns information about the merge between a source and destination commit.

• **get_merge_conflicts**, which returns information about merge conflicts between the source and destination branch in a pull request.

• **get_merge_options**, which returns information about the available merge options between two branches or commit specifiers.

• **merge_branches_by_fast_forward**, which merges two branches using the fast-forward merge option.

• **merge_branches_by_squash**, which merges two branches using the squash merge option.

• **merge_branches_by_three_way**, which merges two branches using the three-way merge option.

Pull requests, by calling the following:

• **create_pull_request**, which creates a pull request in a specified repository.

• **create_pull_request_approval_rule**, which creates an approval rule for a specified pull request.

• **delete_pull_request_approval_rule**, which deletes an approval rule for a specified pull request.

• **describe_pull_request_events**, which returns information about one or more pull request events.

• **evaluate_pull_request_approval_rules**, which evaluates whether a pull request has met all the conditions specified in its associated approval rules.

• **get_comments_for_pull_request**, which returns information about comments on a specified pull request.

• **get_pull_request**, which returns information about a specified pull request.

• **get_pull_request_approval_states**, which returns information about the approval states for a specified pull request.

• **get_pull_request_override_state**, which returns information about whether approval rules have been set aside (overridden) for a pull request, and if so, the Amazon Resource Name (ARN) of the user or identity that overrode the rules and their requirements for the pull request.

• **list_pull_requests**, which lists all pull requests for a repository.

• **merge_pull_request_by_fast_forward**, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the fast-forward merge option.

• **merge_pull_request_by_squash**, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the squash merge option.

• **merge_pull_request_by_three_way**, which merges the source destination branch of a pull request into the specified destination branch for that pull request using the three-way merge option.
• `override_pull_request_approval_rules`, which sets aside all approval rule requirements for a pull request.
• `post_comment_for_pull_request`, which posts a comment to a pull request at the specified line, file, or request.
• `update_pull_request_approval_rule_content`, which updates the structure of an approval rule for a pull request.
• `update_pull_request_approval_state`, which updates the state of an approval on a pull request.
• `update_pull_request_description`, which updates the description of a pull request.
• `update_pull_request_status`, which updates the status of a pull request.
• `update_pull_request_title`, which updates the title of a pull request.

Approval rule templates, by calling the following:

• `associate_approval_rule_template_with_repository`, which associates a template with a specified repository. After the template is associated with a repository, AWS CodeCommit creates approval rules that match the template conditions on every pull request created in the specified repository.
• `batch_associate_approval_rule_template_with_repositories`, which associates a template with one or more specified repositories. After the template is associated with a repository, AWS CodeCommit creates approval rules that match the template conditions on every pull request created in the specified repositories.
• `batch_disassociate_approval_rule_template_from_repositories`, which removes the association between a template and specified repositories so that approval rules based on the template are not automatically created when pull requests are created in those repositories.
• `create_approval_rule_template`, which creates a template for approval rules that can then be associated with one or more repositories in your AWS account.
• `delete_approval_rule_template`, which deletes the specified template. It does not remove approval rules on pull requests already created with the template.
• `disassociate_approval_rule_template_from_repository`, which removes the association between a template and a repository so that approval rules based on the template are not automatically created when pull requests are created in the specified repository.
• `get_approval_rule_template`, which returns information about an approval rule template.
• `list_approval_rule_templates`, which lists all approval rule templates in the AWS Region in your AWS account.
• `list_associated_approval_rule_templates_for_repository`, which lists all approval rule templates that are associated with a specified repository.
• `list_repositories_for_approval_rule_template`, which lists all repositories associated with the specified approval rule template.
• `update_approval_rule_template_description`, which updates the description of an approval rule template.
• `update_approval_rule_template_name`, which updates the name of an approval rule template.
• **update_approval_rule_template_content**, which updates the content of an approval rule template.

Comments in a repository, by calling the following:

• **delete_comment_content**, which deletes the content of a comment on a commit in a repository.
• **get_comment**, which returns information about a comment on a commit.
• **get_comment_reactions**, which returns information about emoji reactions to comments.
• **get_comments_for_compared_commit**, which returns information about comments on the comparison between two commit specifiers in a repository.
• **post_comment_for_compared_commit**, which creates a comment on the comparison between two commit specifiers in a repository.
• **post_comment_reply**, which creates a reply to a comment.
• **put_comment_reaction**, which creates or updates an emoji reaction to a comment.
• **update_comment**, which updates the content of a comment on a commit in a repository.

Tags used to tag resources in AWS CodeCommit (not Git tags), by calling the following:

• **list_tags_for_resource**, which gets information about AWS tags for a specified Amazon Resource Name (ARN) in AWS CodeCommit.
• **tag_resource**, which adds or updates tags for a resource in AWS CodeCommit.
• **untag_resource**, which removes tags for a resource in AWS CodeCommit.

Triggers, by calling the following:

• **get_repository_triggers**, which returns information about triggers configured for a repository.
• **put_repository_triggers**, which replaces all triggers for a repository and can be used to create or delete triggers.
• **test_repository_triggers**, which tests the functionality of a repository trigger by sending data to the trigger target.

For information about how to use AWS CodeCommit, see the [AWS CodeCommit User Guide](https://aws.amazon.com/documentation/codecommit/).

**Usage**

```python
codecommit(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
svc <- codecommit(
  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_approval_rule_template_with_repository` creates an association between an approval rule template and a specified repository.
- `batch_associate_approval_rule_template_with_repositories` creates an association between an approval rule template and one or more specified repositories.
- `batch_describe_merge_conflicts` returns information about one or more merge conflicts in the attempted merge of two commit specifiers using the squash or three-way merge strategy.
- `batch_disassociate_approval_rule_template_from_repositories` removes the association between an approval rule template and one or more specified repositories.
- `batch_get_commits` returns information about the contents of one or more commits in a repository.
- `batch_get_repositories` returns information about one or more repositories.
- `create_approval_rule_template` creates a template for approval rules that can then be associated with one or more repositories in your AWS account.
- `create_branch` creates a branch in a repository and points the branch to a commit.
- `create_commit` creates a commit for a repository on the tip of a specified branch.
- `create_pull_request` creates a pull request in the specified repository.
- `create_pull_request_approval_rule` creates an approval rule for a pull request.
- `create_repository` creates a new, empty repository.
- `create_unreferenced_merge_commit` creates an unreferenced commit that represents the result of merging two branches using a specified merge strategy.
- `delete_approval_rule_template` deletes a specified approval rule template.
- `delete_branch` deletes a branch from a repository, unless that branch is the default branch for the repository.
- `delete_comment_content` deletes the content of a comment made on a change, file, or commit in a repository.
- `delete_file` deletes a specified file from a specified branch.
- `delete_pull_request_approval_rule` deletes an approval rule from a specified pull request.
- `delete_repository` deletes a repository.
- `describe_merge_conflicts` returns information about one or more merge conflicts in the attempted merge of two commit specifiers using the squash or three-way merge strategy.
- `describe_pull_request_events` returns information about one or more pull request events.
- `disassociate_approval_rule_template_from_repository` removes the association between a template and a repository.
- `evaluate_pull_request_approval_rules` evaluates whether a pull request has met all the conditions specified in its associated approval rules.
- `get_approval_rule_template` returns information about a specified approval rule template.
- `get_blob` returns the base-64 encoded content of an individual blob in a repository.
- `get_branch` returns information about a repository branch, including its name and the last commit ID.
- `get_comment` returns the content of a comment made on a change, file, or commit in a repository.
- `get_comment_reactions` returns information about reactions to a specified comment ID.
- `get_comments_for_compared_commit` returns information about comments made on the comparison between two commits.
- `get_comments_for_pull_request` returns comments made on a pull request.
get_commit
get_differences
get_file
get_folder
get_merge_commit
get_merge_conflicts
get_merge_options
get_pull_request
get_pull_request_approval_states
get_pull_request_override_state
get_repository
get_repository_triggers
list_approval_rule_templates
list_associated_approval_rule_templates_for_repository
list_branches
list_pull_requests
list_repositories
list_repositories_for Approval_rule_template
list_tags_for_resource
merge_branches_by_fast_forward
merge_branches_by_squash
merge_branches_by_three_way
merge_pull_request_by_fast_forward
merge_pull_request_by_squash
merge_pull_request_by_three_way
override_pull_request_approval_rules
post_comment_for_compared_commit
post_comment_for_pull_request
post_comment_reply
put_comment_reaction
put_file
put_repository_triggers
tag_resource
test_repository_triggers
untag_resource
update_approval_rule_template_content
update_approval_rule_template_description
update_approval_rule_template_name
update_comment
update_default_branch
update_pull_request_approval_rule_content
update_pull_request_approval_state
update_pull_request_description
update_pull_request_status
update_pull_request_title
update_repository_description
update_repository_name

Returns information about a commit, including commit message and committer information.
Returns information about the differences in a valid commit specifier (such as a branch, tag, HEAD, commit ID, or other fully qualified reference).
Returns the base-64 encoded contents of a specified file and its metadata.
Returns the contents of a specified folder in a repository.
Returns information about a specified merge commit.
Returns information about merge conflicts between the before and after commit IDs for a pull request in a repository.
Returns information about the merge options available for merging two specified branches.
Gets information about a pull request in a specified repository.
Gets information about the approval states for a specified pull request.
Returns information about whether approval rules have been overridden for a specified pull request.
Gets information about a repository.
Gets information about triggers configured for a repository.
Lists all approval rule templates in the specified AWS Region.
Lists all approval rule templates that are associated with a specified repository.
Gets information about one or more branches in a repository.
Returns a list of pull requests for a specified repository.
Gets information about one or more repositories.
Lists all repositories associated with the specified approval rule template.
Gets information about AWS tags for a specified Amazon Resource Name (ARN) in AWS CodeCommit.
Merges two branches using the fast-forward merge strategy.
Merges two branches using the squash merge strategy.
Merges two specified branches using the three-way merge strategy.
Attempts to merge the source commit of a pull request into the specified destination branch at the specified commit using the fast-forward merge strategy.
Attempts to merge the source commit of a pull request into the specified destination branch at the specified commit using the squash merge strategy.
Attempts to merge the source commit of a pull request into the specified destination branch at the specified commit using the three-way merge strategy.
Sets aside (overrides) all approval rule requirements for a specified pull request.
Posts a comment on the comparison between two commits.
Posts a comment on a pull request.
Posts a comment in reply to an existing comment on a comparison between commits or a pull request.
Adds or updates a file in a branch in an AWS CodeCommit repository.
Replaces all triggers for a repository.
Adds or updates tags for a resource in AWS CodeCommit.
Tests the functionality of repository triggers by sending information to the trigger target.
Removes tags for a resource in AWS CodeCommit.
Updates the content of an approval rule template.
Updates the description for a specified approval rule template.
Updates the name of a specified approval rule template.
Replaces the contents of a comment.
Sets or changes the default branch name for the specified repository.
Updates the structure of an approval rule created specifically for a pull request.
Updates the state of a user’s approval on a pull request.
Replaces the contents of the description of a pull request.
Updates the status of a pull request.
Replaces the title of a pull request.
Sets or changes the comment or description for a repository.
Renames a repository.
Examples

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

## AWS CodeDeploy

### Description

AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances running in your own facility, serverless AWS Lambda functions, or applications in an Amazon ECS service.

You can deploy a nearly unlimited variety of application content, such as an updated Lambda function, updated applications in an Amazon ECS service, code, web and configuration files, executables, packages, scripts, multimedia files, and so on. AWS CodeDeploy can deploy application content stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. You do not need to make changes to your existing code before you can use AWS CodeDeploy.

AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

### AWS CodeDeploy Components

Use the information in this guide to help you work with the following AWS CodeDeploy components:

- **Application**: A name that uniquely identifies the application you want to deploy. AWS CodeDeploy uses this name, which functions as a container, to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.

- **Deployment group**: A set of individual instances, CodeDeploy Lambda deployment configuration settings, or an Amazon ECS service and network details. A Lambda deployment group specifies how to route traffic to a new version of a Lambda function. An Amazon ECS deployment group specifies the service created in Amazon ECS to deploy, a load balancer, and a listener to reroute production traffic to an updated containerized application. An EC2/On-premises deployment group contains individually tagged instances, Amazon EC2 instances in Amazon EC2 Auto Scaling groups, or both. All deployment groups can specify optional trigger, alarm, and rollback settings.
• **Deployment configuration**: A set of deployment rules and deployment success and failure conditions used by AWS CodeDeploy during a deployment.

• **Deployment**: The process and the components used when updating a Lambda function, a containerized application in an Amazon ECS service, or of installing content on one or more instances.

• **Application revisions**: For an AWS Lambda deployment, this is an AppSpec file that specifies the Lambda function to be updated and one or more functions to validate deployment lifecycle events. For an Amazon ECS deployment, this is an AppSpec file that specifies the Amazon ECS task definition, container, and port where production traffic is rerouted. For an EC2/On-premises deployment, this is an archive file that contains source content—source code, web pages, executable files, and deployment scripts—along with an AppSpec file. Revisions are stored in Amazon S3 buckets or GitHub repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.

This guide also contains information to help you get details about the instances in your deployments, to make on-premises instances available for AWS CodeDeploy deployments, to get details about a Lambda function deployment, and to get details about Amazon ECS service deployments.

**AWS CodeDeploy Information Resources**

• AWS CodeDeploy User Guide
• AWS CodeDeploy API Reference Guide
• AWS CLI Reference for AWS CodeDeploy
• AWS CodeDeploy Developer Forum

**Usage**

```python
codedeploy(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```python
csvc <- codedeploy(
  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_on_premises_instances Adds tags to on-premises instances
batch_get_application_revisions Gets information about one or more application revisions
batch_get_applications Gets information about one or more applications
batch_get_deployment_groups Gets information about one or more deployment groups
This method works, but is deprecated
batch_get_deployment_instances Gets information about one or more deployments
batch_get_deployment_targets Gets information about one or more on-premises instances
continue_deployment For a blue/green deployment, starts the process of rerouting traffic from instances in the original environment to instances in the replacement environment
create_application Create an application
create_deployment Deploys an application revision through the specified deployment group
create_deployment_config Creates a deployment configuration
create_deployment_group Creates a deployment group to which application revisions are deployed
delete_application Deletes an application
delete_deployment_config Deletes a deployment configuration
delete_deployment_group Deletes a deployment group
delete_git_hub_account_token Deletes a GitHub account connection
delete_resources_by_external_id Deletes resources linked to an external ID
deregister_on_premises_instance Deregisters an on-premises instance
get_application Gets information about an application
get_application_revision Gets information about an application revision
get_deployment Gets information about a deployment
get_deployment_config Gets information about a deployment configuration
get_deployment_group Gets information about a deployment group
get_deployment_instance Gets information about an instance as part of a deployment
get_deployment_target Returns information about a deployment target
get_on_premises_instance Gets information about an on-premises instance
list_application_revisions Lists information about revisions for an application
list_applications Lists the applications registered with the IAM user or AWS account
list_deployment_configs Lists the deployment configurations with the IAM user or AWS account
list_deployment_groups Lists the deployment groups for an application registered with the IAM user or AWS account
list_deployment_instances The newer BatchGetDeploymentTargets should be used instead because it works
list_deployments Lists the deployments in a deployment group for an application registered with the IAM user or AWS account
list_deployment_targets Retruns an array of target IDs that are associated a deployment
list_git_hub_account_token_names Lists the names of stored connections to GitHub accounts
list_on_premises_instances Gets a list of names for one or more on-premises instances
list_tags_for_resource Returns a list of tags for the resource identified by a specified Amazon Resource Name (ARN)
put_lifecycle_event_hook_execution_status Sets the result of a Lambda validation function
**Register Application Revision**
Registers with AWS CodeDeploy a revision for the specified application.

**Register On-Premises Instance**
Registers an on-premises instance.

**Remove Tags from On-Premises Instances**
Removes one or more tags from one or more on-premises instances.

**Skip Wait Time for Instance Termination**
In a blue/green deployment, overrides any specified wait time and starts terminating instances immediately after the traffic routing is complete.

**Stop Deployment**
Attempts to stop an ongoing deployment.

**Tag Resource**
Associates the list of tags in the input Tags parameter with the resource identified by the ResourceArn input parameter.

**Untag Resource**
Disassociates a resource from a list of tags.

**Update Application**
Changes the name of an application.

**Update Deployment Group**
Changes information about a deployment group.

### Examples

```r
## Not run:
svc <- codedeploy()
svc$add_tags_to_on_premises_instances(
  Foo = 123
)

## End(Not run)
```

## Description

### Overview

This is the AWS CodePipeline API Reference. This guide provides descriptions of the actions and data types for AWS CodePipeline. Some functionality for your pipeline can only be configured through the API. For more information, see the [AWS CodePipeline User Guide](https://docs.aws.amazon.com/codepipeline/latest/userguide/).

You can use the AWS CodePipeline API to work with pipelines, stages, actions, and transitions.

**Pipelines** are models of automated release processes. Each pipeline is uniquely named, and consists of stages, actions, and transitions.

You can work with pipelines by calling:

- `create_pipeline`, which creates a uniquely named pipeline.
- `delete_pipeline`, which deletes the specified pipeline.
- `get_pipeline`, which returns information about the pipeline structure and pipeline metadata, including the pipeline Amazon Resource Name (ARN).
- `get_pipeline_execution`, which returns information about a specific execution of a pipeline.
- `get_pipeline_state`, which returns information about the current state of the stages and actions of a pipeline.
- list_action_executions, which returns action-level details for past executions. The details include full stage and action-level details, including individual action duration, status, any errors that occurred during the execution, and input and output artifact location details.
- list_pipelines, which gets a summary of all of the pipelines associated with your account.
- list_pipeline_executions, which gets a summary of the most recent executions for a pipeline.
- start_pipeline_execution, which runs the most recent revision of an artifact through the pipeline.
- stop_pipeline_execution, which stops the specified pipeline execution from continuing through the pipeline.
- update_pipeline, which updates a pipeline with edits or changes to the structure of the pipeline.

Pipelines include stages. Each stage contains one or more actions that must complete before the next stage begins. A stage results in success or failure. If a stage fails, the pipeline stops at that stage and remains stopped until either a new version of an artifact appears in the source location, or a user takes action to rerun the most recent artifact through the pipeline. You can call get_pipeline_state, which displays the status of a pipeline, including the status of stages in the pipeline, or get_pipeline, which returns the entire structure of the pipeline, including the stages of that pipeline. For more information about the structure of stages and actions, see AWS CodePipeline Pipeline Structure Reference.

Pipeline stages include actions that are categorized into categories such as source or build actions performed in a stage of a pipeline. For example, you can use a source action to import artifacts into a pipeline from a source such as Amazon S3. Like stages, you do not work with actions directly in most cases, but you do define and interact with actions when working with pipeline operations such as create_pipeline and get_pipeline_state. Valid action categories are:

- Source
- Build
- Test
- Deploy
- Approval
- Invoke

Pipelines also include transitions, which allow the transition of artifacts from one stage to the next in a pipeline after the actions in one stage complete.

You can work with transitions by calling:

- disable_stage_transition, which prevents artifacts from transitioning to the next stage in a pipeline.
- enable_stage_transition, which enables transition of artifacts between stages in a pipeline.

Using the API to integrate with AWS CodePipeline

For third-party integrators or developers who want to create their own integrations with AWS CodePipeline, the expected sequence varies from the standard API user. To integrate with AWS CodePipeline, developers need to work with the following items:
Jobs, which are instances of an action. For example, a job for a source action might import a revision of an artifact from a source.

You can work with jobs by calling:

- **acknowledge_job**, which confirms whether a job worker has received the specified job.
- **get_job_details**, which returns the details of a job.
- **poll_for_jobs**, which determines whether there are any jobs to act on.
- **put_job_failure_result**, which provides details of a job failure.
- **put_job_success_result**, which provides details of a job success.

Third party jobs, which are instances of an action created by a partner action and integrated into AWS CodePipeline. Partner actions are created by members of the AWS Partner Network.

You can work with third party jobs by calling:

- **acknowledge_third_party_job**, which confirms whether a job worker has received the specified job.
- **get_third_party_job_details**, which requests the details of a job for a partner action.
- **poll_for_third_party_jobs**, which determines whether there are any jobs to act on.
- **put_third_party_job_failure_result**, which provides details of a job failure.
- **put_third_party_job_success_result**, which provides details of a job success.

Usage

codepipeline(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

Operations

acknowledge_job
acknowledge_third_party_job
create_custom_action_type
create_pipeline
delete_custom_action_type
delete_pipeline
delete_webhook
deregister_webhook_with_third_party
disable_stage_transition
enable_stage_transition
get_job_details
get_pipeline
get_pipeline_execution
get_pipeline_state
get_third_party_job_details
list_action_executions
list_action_types
list_pipeline_executions
list_pipelines
list_tags_for_resource
list_webhooks
poll_for_jobs
poll_for_third_party_jobs
put_action_revision
put_approval_result
put_job_failure_result
put_job_success_result
put_third_party_job_failure_result
put_third_party_job_success_result
put_webhook
register_webhook_with_third_party
retry_stage_execution
start_pipeline_execution
stop_pipeline_execution
tag_resource
untag_resource
update_pipeline

Returns information about a specified job and whether that job has been received by the pipeline worker
Confirms a job worker has received the specified job
Creates a new custom action that can be used in all pipelines associated with the AWS account
Creates a pipeline
Marks a custom action as deleted
Deletes the specified pipeline
Deletes a previously created webhook by name
Removes the connection between the webhook that was created by CodePipeline and the external tool
Prevents artifacts in a pipeline from transitioning to the next stage in the pipeline
Enables artifacts in a pipeline to transition to a stage in a pipeline
Returns information about a job
Returns the metadata, structure, stages, and actions of a pipeline
Returns information about an execution of a pipeline, including details about artifacts
Requests the details of a job for a third party action
Lists the action executions that have occurred in a pipeline
Gets a summary of all AWS CodePipeline action types associated with your account
Gets a summary of the most recent executions for a pipeline
Gets a summary of all of the pipelines associated with your account
Gets the set of key-value pairs (metadata) that are used to manage the resource
Gets a listing of all the webhooks in this AWS Region for this account
Determines whether there are any third party jobs for a job worker to act on
Provides information to AWS CodePipeline about new revisions to a source
Provides the response to a manual approval request to AWS CodePipeline
Represents the failure of a job as returned to the pipeline by a job worker
Represents the success of a job as returned to the pipeline by a job worker
Represents the failure of a third party job as returned to the pipeline by a job worker
Represents the success of a third party job as returned to the pipeline by a job worker
Defines a webhook and returns a unique webhook URL generated by CodePipeline
Configures a connection between the webhook that was created and the external tool
Resumes the pipeline execution by retrying the last failed actions in a stage
Starts the specified pipeline
Stops the specified pipeline execution
Adds to or modifies the tags of the given resource
Removes tags from an AWS resource
Updates a specified pipeline with edits or changes to its structure
Examples

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

---

**AWS CodeStar**

Description

This is the API reference for AWS CodeStar. This reference provides descriptions of the operations and data types for the AWS CodeStar API along with usage examples.

You can use the AWS CodeStar API to work with:

Projects and their resources, by calling the following:

- `delete_project`, which deletes a project.
- `describe_project`, which lists the attributes of a project.
- `list_projects`, which lists all projects associated with your AWS account.
- `list_resources`, which lists the resources associated with a project.
- `list_tags_for_project`, which lists the tags associated with a project.
- `tag_project`, which adds tags to a project.
- `untag_project`, which removes tags from a project.
- `update_project`, which updates the attributes of a project.

Teams and team members, by calling the following:

- `associate_team_member`, which adds an IAM user to the team for a project.
- `disassociate_team_member`, which removes an IAM user from the team for a project.
- `list_team_members`, which lists all the IAM users in the team for a project, including their roles and attributes.
- `update_team_member`, which updates a team member’s attributes in a project.

Users, by calling the following:

- `create_user_profile`, which creates a user profile that contains data associated with the user across all projects.
- `delete_user_profile`, which deletes all user profile information across all projects.
- `describe_user_profile`, which describes the profile of a user.
- `list_user_profiles`, which lists all user profiles.
- `update_user_profile`, which updates the profile for a user.
Usage

codestar(config = list())

Arguments

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

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...),
where svc is the name you've assigned to the client. The available operations are listed in the Op-
erations section.

Service syntax

svc <- codestar(
  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_team_member Adds an IAM user to the team for an AWS CodeStar project
create_project Creates a project, including project resources
create_user_profile Creates a profile for a user that includes user preferences, such as the display name and email address
delete_project Deletes a project, including project resources
delete_user_profile Deletes a user profile in AWS CodeStar, including all personal preference data associated with that profile
describe_project Describes a project and its resources
describe_user_profile Describes a user in AWS CodeStar and the user attributes across all projects
disassociate_team_member Removes a user from a project
list_projects Lists all projects in AWS CodeStar associated with your AWS account
list_resources Lists resources associated with a project in AWS CodeStar
list_tags_for_project Gets the tags for a project
list_team_members Lists all team members associated with a project
list_user_profiles Lists all the user profiles configured for your AWS account in AWS CodeStar
tag_project Adds tags to a project
untag_project Removes tags from a project
update_project Updates a project in AWS CodeStar
cognitoidentity

update_team_member  Updates a team member’s attributes in an AWS CodeStar project
update_user_profile Updates a user’s profile in AWS CodeStar

Examples

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

cognitoidentity  Amazon Cognito Identity

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

cognitoidentity(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
csvc <- 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 UserPoolDeveloperProviderIds associated with an IdentityId for an existing identity
- `merge_developer_identities` Merges two users having different IdentityIds, existing in the same identity pool, identified by the same developer provider
- `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
)```
## Cognito Identity Provider

**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.

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```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
admin_user_global_sign_out
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_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
delete_user_pool_domain
describe_identity_provider

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 devices, as an administrator
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
Deletes a domain for a user pool
Gets information about a specific identity provider
## Cognito Identity Provider

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>describe_resource_server</code></td>
<td>Describes a resource server</td>
</tr>
<tr>
<td><code>describe_risk_configuration</code></td>
<td>Describes the risk configuration</td>
</tr>
<tr>
<td><code>describe_user_import_job</code></td>
<td>Describes the user import job</td>
</tr>
<tr>
<td><code>describe_user_pool</code></td>
<td>Returns the configuration information and metadata of the specified user pool</td>
</tr>
<tr>
<td><code>describe_user_pool_domain</code></td>
<td>Client method for returning the configuration information and metadata of the specified domain</td>
</tr>
<tr>
<td><code>describe_user_pool_client</code></td>
<td>Gets information about a domain</td>
</tr>
<tr>
<td><code>forget_device</code></td>
<td>Forgets the specified device</td>
</tr>
<tr>
<td><code>forgot_password</code></td>
<td>Calling this API causes a message to be sent to the end user with a confirmation code that is required to change the user's password</td>
</tr>
<tr>
<td><code>get_csv_header</code></td>
<td>Gets the header information for the specified user pool</td>
</tr>
<tr>
<td><code>get_device</code></td>
<td>Gets the device</td>
</tr>
<tr>
<td><code>get_group</code></td>
<td>Gets a group</td>
</tr>
<tr>
<td><code>get_identity_provider_by_identifier</code></td>
<td>Gets the specified identity provider</td>
</tr>
<tr>
<td><code>get_signing_certificate</code></td>
<td>This method takes a user pool ID, and returns the signing certificate</td>
</tr>
<tr>
<td><code>get_ui_customization</code></td>
<td>Gets the UI Customization information for a particular app client’s app UI, if there is something set for the app UI</td>
</tr>
<tr>
<td><code>get_user</code></td>
<td>Gets the user attributes and metadata for a user</td>
</tr>
<tr>
<td><code>get_user_attribute_verification_code</code></td>
<td>Gets the user attribute verification code for the specified attribute name</td>
</tr>
<tr>
<td><code>get_user_pool_mfa_config</code></td>
<td>Gets the user pool multi-factor authentication (MFA) configuration</td>
</tr>
<tr>
<td><code>global_sign_out</code></td>
<td>Signs out users from all devices</td>
</tr>
<tr>
<td><code>initiate_auth</code></td>
<td>Initiates the authentication flow</td>
</tr>
<tr>
<td><code>list_devices</code></td>
<td>Lists the devices</td>
</tr>
<tr>
<td><code>list_groups</code></td>
<td>Lists the groups associated with a user pool</td>
</tr>
<tr>
<td><code>list_identity_providers</code></td>
<td>Lists information about all identity providers for a user pool</td>
</tr>
<tr>
<td><code>list_resource_servers</code></td>
<td>Lists the resource servers for a user pool</td>
</tr>
<tr>
<td><code>list_tags_for_resource</code></td>
<td>Lists the tags that are assigned to an Amazon Cognito user pool</td>
</tr>
<tr>
<td><code>list_user_import_jobs</code></td>
<td>Lists the user import jobs</td>
</tr>
<tr>
<td><code>list_user_pools</code></td>
<td>Lists the clients that have been created for the specified user pool</td>
</tr>
<tr>
<td><code>list_user_pools Clients</code></td>
<td>Lists the user pools associated with an AWS account</td>
</tr>
<tr>
<td><code>list_users</code></td>
<td>Lists the users in the Amazon Cognito user pool</td>
</tr>
<tr>
<td><code>list_users_in_group</code></td>
<td>Lists the users in the specified group</td>
</tr>
<tr>
<td><code>resend_confirmation_code</code></td>
<td>Resends the confirmation (for confirmation of registration) to a specific user in the user pool</td>
</tr>
<tr>
<td><code>respond_to_auth_challenge</code></td>
<td>Responds to the authentication challenge</td>
</tr>
<tr>
<td><code>set_risk_configuration</code></td>
<td>Configures actions on detected risks</td>
</tr>
<tr>
<td><code>set_ui_customization</code></td>
<td>Sets the UI customization information for a user pool's built-in app UI</td>
</tr>
<tr>
<td><code>set_user_mfa_preference</code></td>
<td>Set the user’s multi-factor authentication (MFA) method preference, including which MFA factors to enable and preferred MFA methods and factors for a user pool</td>
</tr>
<tr>
<td><code>set_user_pool_mfa_config</code></td>
<td>Set the user pool multi-factor authentication (MFA) configuration</td>
</tr>
<tr>
<td><code>set_user_settings</code></td>
<td>This action is no longer supported</td>
</tr>
<tr>
<td><code>sign_up</code></td>
<td>Registers the user in the specified user pool and creates a user name, password, and user attributes</td>
</tr>
<tr>
<td><code>sign_out</code></td>
<td>Signs out users from all devices</td>
</tr>
<tr>
<td><code>start_user_import_job</code></td>
<td>Starts the user import</td>
</tr>
<tr>
<td><code>stop_user_import_job</code></td>
<td>Stops the user import</td>
</tr>
<tr>
<td><code>update_auth_event_feedback</code></td>
<td>Updates the device status</td>
</tr>
<tr>
<td><code>update_device_status</code></td>
<td>Updates the specified group with the specified attributes</td>
</tr>
<tr>
<td><code>update_group</code></td>
<td>Updates identity provider information for a user pool</td>
</tr>
<tr>
<td><code>update_resource_server</code></td>
<td>Updates the name and scopes of resource server</td>
</tr>
<tr>
<td><code>update_user_attributes</code></td>
<td>Allows a user to update a specific attribute (one at a time)</td>
</tr>
<tr>
<td><code>update_user_pool</code></td>
<td>Updates the specified user pool with the specified attributes</td>
</tr>
</tbody>
</table>
update_user_pool_client
update_user_pool_domain
verify_software_token
verify_user_attribute

Updates the specified user pool app client with the specified attributes
Updates the Secure Sockets Layer (SSL) certificate for the custom domain for your user pool
Use this API to register a user’s entered TOTP code and mark the user’s software token as verified
Verifies the specified user attributes in the user pool

Examples

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

**cognitosync**  
*Amazon Cognito Sync*

**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**

cognitosync(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>
Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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_usage` Gets usage details (for example, data storage) about a particular identity pool
- `describe_identity_pool_usage` Gets usage information for an identity, including number of datasets and data usage
- `describe_identity_pool_usage` Gets 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_configuration` 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()
```
comprehend

```r
csvc$bulk_publish(
  Foo = 123
)
```

## End(Not run)

### 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

```r
comprehend(config = list())
```

### Arguments

- **config**

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```r
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
batch_detect_entities
batch_detect_key_phrases
batch_detect_sentiment
batch_detect_syntax
classify_document
create_document_classifier
create_endpoint
create_entity_recognizer
delete_document_classifier
delete_endpoint
delete_entity_recognizer
describe_document_classification_job
describe_document_classifier
describe_dominant_language_detection_job
describe_endpoint
describe_entities_detection_job
describe_entity_recognizer
describe_events_detection_job
describe_key_phrases_detection_job
describe_pii_entities_detection_job
describe_sentiment_detection_job
describe_topics_detection_job
detect_dominant_language
detect_entities
detect_key_phrases
detect_sentiment
detect_syntax
list_document_classification_jobs
list_document_classifiers
list_dominant_language_detection_jobs
list_endpoints
list_entities_detection_jobs
list_entity_recognizers
list_events_detection_jobs
list_key_phrases_detection_jobs
list_pii_entities_detection_jobs
list_sentiment_detection_jobs
list_tags_for_resource
list_topics_detection_jobs
start_document_classification_job
start_dominant_language_detection_job
start_entities_detection_job
start_events_detection_job
start_key_phrases_detection_job
start_pii_entities_detection_job
start_sentiment_detection_job

Determines the dominant language of the input text for a batch of documents
Inspects the text of a batch of documents for named entities and returns information about them
Dets the key noun phrases found in a batch of documents
Inspects a batch of documents and returns an inference of the prevailing sentiment
Inspects the text of a batch of documents for the syntax and part of speech of the words
Creates a new document classification request to analyze a single document in real-time
Creates a new document classifier that you can use to categorize documents
Creates a model-specific endpoint for synchronous inference for a previously trained custom model
Creates an entity recognizer using submitted files
Deletes a previously created document classifier
Deletes a model-specific endpoint for a previously-trained custom model
Deletes an entity recognizer
Gets the properties associated with a document classification job
Gets the properties associated with a document classifier
Gets the properties associated with a dominant language detection job
Gets the properties associated with a specific endpoint
Gets the properties associated with an entities detection job
Provides details about an entity recognizer including status, S3 buckets containing training data, and so on
Gets the status and details of an events detection job
Gets the properties associated with a key phrases detection job
Gets the properties associated with a PII entities detection job
Gets the properties associated with a sentiment detection job
Gets the properties associated with a topic detection job
Determines the dominant language of the input text
Inspects text for named entities, and returns information about them
Detects the key noun phrases found in the text
Inspects the input text for entities that contain personally identifiable information
Inspects text and returns an inference of the prevailing sentiment (POSITIVE, NEGATIVE, MIXED)
Inspects text for syntax and the part of speech of words in the document
Gets a list of the documentation classification jobs that you have submitted
Gets a list of the document classifiers that you have created
Gets a list of the dominant language detection jobs that you have submitted
Gets a list of all existing endpoints that you’ve created
Gets a list of the entity detection jobs that you have submitted
Gets a list of the properties of all entity recognizers that you created, including status
Gets a list of the events detection jobs that you have submitted
Get a list of key phrase detection jobs that you have submitted
Gets a list of the PII entity detection jobs that you have submitted
Gets a list of sentiment detection jobs that you have submitted
Lists all tags associated with a given Amazon Comprehend resource
Gets a list of the topic detection jobs that you have submitted
Starts an asynchronous document classification job
Starts an asynchronous dominant language detection job for a collection of documents
Starts an asynchronous entity detection job for a collection of documents
Starts an asynchronous event detection job for a collection of documents
Starts an asynchronous key phrase detection job for a collection of documents
Starts an asynchronous PII entity detection job for a collection of documents
Starts an asynchronous sentiment detection job for a collection of documents
comprehendmedical starts an asynchronous topic detection job. It stops a dominant language detection job in progress. It stops an entities detection job in progress. It stops an events detection job in progress. It stops a key phrases detection job in progress. It stops a PII entities detection job in progress. It stops a sentiment detection job in progress. It stops a document classifier training job while in progress. It stops an entity recognizer training job while in progress. It associates a specific tag with an Amazon Comprehend resource. It removes a specific tag associated with an Amazon Comprehend resource. It updates information about the specified endpoint.

Examples

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

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

```r
comprehendmedical(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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_icd10cm_inference_job` - Gets the properties associated with an InferICD10CM job
- `describe_phi_detection_job` - Gets the properties associated with a protected health information (PHI) detection job
- `describe_rx_norm_inference_job` - Gets the properties associated with an InferRxNorm job
- `detect_entities` - The DetectEntities operation is deprecated
- `detect_entities_v2` - 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_phi` - Inspects the clinical text for protected health information (PHI) entities and returns the entity category, location, and confidence score for each entity
- `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_icd10cm_inference_jobs` - Gets a list of InferICD10CM jobs that you have submitted
- `list_phi_detection_jobs` - Gets a list of protected health information (PHI) detection jobs that you have submitted
- `list_rx_norm_inference_jobs` - Gets a list of InferRxNorm jobs that you have submitted
- `start_entities_detection_v2_job` - Starts an asynchronous medical entity detection job for a collection of documents
- `start_icd10cm_inference_job` - Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM ontology
- `start_phi_detection_job` - Starts an asynchronous job to detect protected health information (PHI)
- `start_rx_norm_inference_job` - Starts an asynchronous job to detect medication entities and link them to the RxNorm ontology
- `stop_entities_detection_v2_job` - Stops a medical entities detection job in progress
- `stop_icd10cm_inference_job` - Stops an InferICD10CM inference job in progress
- `stop_phi_detection_job` - Stops a protected health information (PHI) detection job in progress
- `stop_rx_norm_inference_job` - Stops an InferRxNorm inference job in progress

Examples

```r
## Not run:
svc <- comprehendmedical()
svc$describe_entities_detection_v2_job(Foo = 123)
```
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
cfg <- configservice(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like ` svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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
batch_get_resource_config
delete_aggregation_authorization
delete_config_rule
delete_configuration_aggregator
delete_configuration_recorder
delete_conformance_pack
delete_delivery_channel
delete_evaluation_results
delete_organization_config_rule
delete_organization_conformance_pack
delete_pending_aggregation_request
delete_remediation_configuration
delete_remediation_exceptions
delete_resource_config
delete_retention_configuration
delete_stored_query
deliver_config_snapshot
describe_aggregate_compliance_by_config_rules
describe_aggregation_authorizations
describe_compliance_by_config_rule
describe_compliance_by_resource
describe_config_rule_evaluation_status
describe_config_rules
describe_configuration_aggregator_sources_status
describe_configuration_aggregator_status
describe_configuration_recorder_status
describe_conformance_pack_compliance
describe_conformance_packs
describe_conformance_pack_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

Returns the current configuration items for resources that are present in your AWS Config aggregator.
Returns the current configuration for one or more requested resources.
Deletes the authorization granted to the specified configuration aggregator account.
Deletes the specified AWS Config rule and all of its evaluation results.
Deletes the specified configuration aggregator and the aggregated data associated with it.
Deletes the configuration recorder.
Deletes the specified conformance pack and all the AWS Config rules, remediation actions, and all evaluation results within that conformance pack.
Deletes the delivery channel.
Deletes the evaluation results for the specified AWS Config rule.
Deletes the specified organization config rule and all of its evaluation results.
Deletes the specified organization conformance pack and all the AWS Config rules, remediation actions, and all evaluation results within that conformance pack.
Deletes pending authorization requests for a specified aggregator account.
Deletes the remediation configuration.
Deletes one or more remediation exceptions mentioned in the resource key.
Records the configuration state for a custom resource that has been deleted.
Deletes the retention configuration.
Deletes the stored query for an AWS account in an AWS Region.
Schedules delivery of a configuration snapshot to the Amazon S3 bucket specified in the delivery channel.
Returns a list of compliant and noncompliant rules with the number of resources for compliant and noncompliant rules.
Returns a list of authorizations granted to various aggregator accounts.
Indicates whether the specified AWS Config rules are compliant.
Indicates whether the specified AWS resources are compliant.
Returns status information for each of your AWS managed Config rules.
Returns details about your AWS Config rules.
Returns the details of one or more configuration aggregators.
Returns status information for sources within an aggregator.
Returns the details for the specified configuration recorders.
Returns the current status of the specified configuration recorder.
Returns compliance details for each rule in that conformance pack.
Returns a list of one or more conformance packs.
Provides one or more conformance packs deployment status.
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.
describe_pending_aggregation_requests
describe_remediation_configurations
describe_remediation_exceptions
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
get_stored_query
list_aggregate_discovered_resources
list_discovered_resources
list_stored_queries
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_external_evaluation
put_organization_config_rule
put_organization_conformance_pack
put_remediation_configurations
put_remediation_exceptions
put_resource_config
put_retention_configuration
put_stored_query
select_aggregate_resource_config
select_resource_config
start_config_rules_evaluation
start_configuration_recorder
start_remediation_execution
stop_configuration_recorder
tag_resource
untag_resource

Returns a list of all pending aggregation requests
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 set of resources
Returns the number of compliant and noncompliant rules for one or more accounts
Returns the resource counts across accounts and regions that are present
Returns configuration item that is aggregated for your specific resource
Returns the evaluation results for the specified AWS Config rule
Returns the evaluation results for the specified AWS resource
Returns the number of AWS Config rules that are compliant and noncompliant for one or more accounts
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 status
Returns the resource types, the number of each resource type, and the number of total 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
Returns the details of a specific stored query
Accepts a resource type and returns a list of resource identifiers that are compliant
Accepts a resource type and returns a list of resource identifiers for the latest compliance state
List the stored queries for an AWS account in an AWS Region
List the tags for AWS Config resource
Authorizes the aggregator account and region to collect data from the source account and region
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 sources
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 AWS Config
Used by an AWS Lambda function to deliver evaluation results to AWS Config
Put external evaluation
Adds or updates organization config rule for your entire organization or a specific member account
Deploys conformance packs across member accounts in an AWS Organization
Adds or updates 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
Creates and updates the retention configuration with details about retention period (number of days)
Saves a new query or updates an existing saved query
Accepts a structured query language (SQL) SELECT command and an AWS Config resource
Accepts a structured query language (SQL) SELECT command, performs a query
Runs an on-demand evaluation for the specified AWS Config rules against the last known configuration state
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
Stops 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

```r
## 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 Quotas](https://docs.aws.amazon.com/connect/latest/adminguide/service-quotas) in the *Amazon Connect Administrator Guide*.

To connect programmatically to an AWS service, you use an endpoint. For a list of Amazon Connect endpoints, see [Amazon Connect Endpoints](https://docs.aws.amazon.com/connect/latest/adminguide/endpoints).

Working with contact flows? Check out the [Amazon Connect Flow language](https).

**Usage**

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

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

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

Operations

- associate_approved_origin
- associate_instance_storage_config
- associate_lambda_function
- associate_lex_bot
- associate_routing_profile_queues
- associate_security_key
- create_contact_flow
- create_instance
- create_integration_association
- create_quick_connect
- create_routing_profile
- create_use_case
- create_user
- create_user_hierarchy_group
- delete_instance
- delete_integration_association
- delete_quick_connect
- delete_use_case
- delete_user
- delete_user_hierarchy_group
- describe_contact_flow
- describe_instance
- describe_instance_attribute
- describe_instance_storage_config
- describe_quick_connect
- describe_routing_profile
- describe_user
- describe_user_hierarchy_group
- describe_user_hierarchy_structure
- disassociate_approved_origin

This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
This API is in preview release for Amazon Connect and is subject to change.
disassociate_instance_storage_config
This API is in preview release for Amazon Connect and is subject to change

disassociate_lambda_function
This API is in preview release for Amazon Connect and is subject to change

disassociate_lex_bot
This API is in preview release for Amazon Connect and is subject to change

disassociate_routing_profile_queues
Disassociates a set of queues from a routing profile

disassociate_security_key
This API is in preview release for Amazon Connect and is subject to change

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

get_contact_attributes
This API is in preview release for Amazon Connect and is subject to change

get_current_metric_data
Provides information about the contact flows for the specified Amazon Connect instance

get_federation_token
Provides information about the hours of operation for the specified Amazon Connect instance

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_federation_token
Provides information about the phone numbers for the specified Amazon Connect instance

get_metric_data
Provides information about the prompts for the specified Amazon Connect instance

get_federation_token
Provides information about the queues for the specified Amazon Connect instance

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_federation_token
List the queues associated with a routing profile

get_metric_data
Provides summary information about the routing profiles for the specified Amazon Connect instance

get_federation_token
Provides summary information about the security profiles for the specified Amazon Connect instance

get_metric_data
Lists the tags for the specified resource

get_federation_token
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
Provides summary information about the security profiles for the specified Amazon Connect instance

get_federation_token
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
Provides summary information about the hierarchy groups for the specified Amazon Connect instance

get_federation_token
Lists the tags for the specified resource

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_federation_token
Provides summary information about the hierarchy groups for the specified Amazon Connect instance

get_metric_data
When a contact is being recorded, and the recording has been suspended using SuspendContactRecording, this API resumes recording the call

get_metric_data
Initiates a contact flow to start a new chat for the customer

get_metric_data
This API starts recording the contact when the agent joins the call

get_metric_data
This API places an outbound call to a contact, and then initiates the contact flow

get_metric_data
Ends the specified contact

get_metric_data
When a contact is being recorded, this API stops recording the call

get_metric_data
When a contact is being recorded, this API suspends recording the call

get_metric_data
Removes the specified tags from the specified resource

get_metric_data
Creates or updates the contact attributes associated with the specified contact

get_metric_data
Updates the specified contact flow

tag_resource
The name of the contact flow

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
This API is in preview release for Amazon Connect and is subject to change

get_metric_data
Updates the channels that agents can handle in the Contact Control Panel (CCP)
update_routing_profile_default_outbound_queue Updates the default outbound queue of a routing profile
update_routing_profile_name Updates the name and description of a routing profile
update_routing_profile_queues Updates the properties associated with a set of queues for a routing profile
update_user_hierarchy Assigns the specified hierarchy group to the specified user
update_user_hierarchy_group_name Updates the name of the user hierarchy group
update_user_hierarchy_structure Updates the user hierarchy structure: add, remove, and rename user hierarchy levels
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$associate_approved_origin(
  Foo = 123
)

## End(Not run)
```

---

costandusagereportservice

**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**

```r
costandusagereportservice(config = list())
```

**Arguments**

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

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
csvc <- 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

- `delete_report_definition` Deletes the specified report
- `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

Examples

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

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

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- 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_anomaly_monitor` Creates a new cost anomaly detection monitor with the requested type and monitor specification
- `create_anomaly_subscription` Adds a subscription to a cost anomaly detection monitor
- `create_cost_category_definition` Creates a new Cost Category with the requested name and rules
delete_anomaly_monitor
delete_anomaly_subscription
delete_cost_category_definition
describe_cost_category_definition
get_anomalies
get_anomaly_monitors
get_anomaly_subscriptions
get_cost_and_usage
get_cost_and_usage_with_resources
get_cost_categories
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
provide_anomaly_feedback
update_anomaly_monitor
update_anomaly_subscription
update_cost_category_definition

deletes a cost anomaly monitor
Deletes a cost anomaly subscription
Deletes a Cost Category
Returns the name, ARN, rules, definition, and effective dates of a Cost Category
Retrieves all of the cost anomalies detected on your account, during the time period specified by the DateInterval
Retrieves the cost anomaly monitor definitions for your account
Retrieves the cost anomaly subscription objects for your account
Retrieves cost and usage metrics for your account
Retrieves cost and usage metrics with resources for your account
Retrieves an array of Cost Category names and values incurred cost
Retrieves a forecast for how much Amazon Web Services predicts that you will spend over the forecast time period
Retrieves all available filter values for a specified filter over a period of time
Retrieves the reservation coverage for your account
Gets recommendations for which reservations to purchase
Retrieves the reservation utilization for your account
Creates recommendations that help you save cost by identifying idle and underutilized Amazon EC2 instances
Retrieves the Savings Plans covered for your account
Retrieves your request parameters, Savings Plan Recommendations Summary
Retrieves the Savings Plans utilization for your account across date ranges with daily or monthly granularity
Queries attribute data along with aggregate utilization and savings data for a specified period
Retrieves a forecast for how much Amazon Web Services predicts that you will use over the forecast time period
Returns the name, ARN, NumberOfWeeks and effective dates of all Cost Category
Modifies the feedback property of a given cost anomaly
Updates an existing cost anomaly monitor
Updates an existing cost anomaly monitor subscription
Updates an existing Cost Category

Examples

## Not run:
svc <- costexplorer()
svc$create_anomaly_monitor(
  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, such as performing database queries and running data analysis using Amazon Elastic MapReduce (Amazon EMR). You can use AWS Data Pipeline Task Runner as your task runner, or you can write your own task runner to provide custom data management.

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

datapipeline(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate_pipeline</td>
<td>Validates the specified pipeline and starts processing pipeline tasks</td>
</tr>
<tr>
<td>add_tags</td>
<td>Adds or modifies tags for the specified pipeline</td>
</tr>
<tr>
<td>create_pipeline</td>
<td>Creates a new, empty pipeline</td>
</tr>
<tr>
<td>deactivate_pipeline</td>
<td>Deactivates the specified running pipeline</td>
</tr>
<tr>
<td>delete_pipeline</td>
<td>Deletes a pipeline, its pipeline definition, and its run history</td>
</tr>
<tr>
<td>describe_objects</td>
<td>Gets the object definitions for a set of objects associated with the pipeline</td>
</tr>
<tr>
<td>describe_pipelines</td>
<td>Retrieves metadata about one or more pipelines</td>
</tr>
<tr>
<td>evaluate_expression</td>
<td>Task runners call EvaluateExpression to evaluate a string in the context of the specified object</td>
</tr>
<tr>
<td>get_pipeline_definition</td>
<td>Gets the definition of the specified pipeline</td>
</tr>
<tr>
<td>list_pipelines</td>
<td>Lists the pipeline identifiers for all active pipelines that you have permission to access</td>
</tr>
<tr>
<td>poll_for_task</td>
<td>Task runners call PollForTask to receive a task to perform from AWS Data Pipeline</td>
</tr>
<tr>
<td>put_pipeline_definition</td>
<td>Adds tasks, schedules, and preconditions to the specified pipeline</td>
</tr>
<tr>
<td>query_objects</td>
<td>Queries the specified pipeline for the names of objects that match the specified set of conditions</td>
</tr>
<tr>
<td>remove_tags</td>
<td>Removes existing tags from the specified pipeline</td>
</tr>
<tr>
<td>report_task_progress</td>
<td>Task runners call ReportTaskProgress when assigned a task to acknowledge that it has the task</td>
</tr>
<tr>
<td>report_task_runner_heartbeat</td>
<td>Task runners call ReportTaskRunnerHeartbeat every 15 minutes to indicate that they are operational</td>
</tr>
<tr>
<td>set_status</td>
<td>Requests that the status of the specified physical or logical pipeline objects be updated in the specified pipeline</td>
</tr>
<tr>
<td>set_task_status</td>
<td>Task runners call SetTaskStatus to notify AWS Data Pipeline that a task is completed and provide details about the task status</td>
</tr>
<tr>
<td>validate_pipeline_definition</td>
<td>Validates the specified pipeline definition to ensure that it is well formed and can be run without errors</td>
</tr>
</tbody>
</table>

Examples

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

## End(Not run)
```

---

**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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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

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

## End(Not run)
```

directconnect AWS Direct Connect

Description

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

Usage

directconnect(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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`: Accepts a proposal request to attach a virtual private gateway or transit gateway to a Direct Connect gateway
- `allocate_connection_on_interconnect`: Creates a hosted connection on the specified interconnect or a link aggregation group (LAG) bundle
- `allocate_hosted_connection`: Deprecated
- `allocate_private_virtual_interface`: Provisions a private virtual interface to be owned by the specified AWS account
- `allocate_public_virtual_interface`: Provisions a public virtual interface to be owned by the specified AWS account
- `allocate_transit_virtual_interface`: Provisions a transit virtual interface to be owned by the specified AWS account
- `associate_connection_with_lag`: Associates an existing connection with a link aggregation group (LAG)
- `associate_hosted_connection`: Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG) or interconnect
- `associate_virtual_interface`: Associates a virtual interface with a specified link aggregation group (LAG) or connection
- `confirm_connection`: Confirms the creation of the specified hosted connection on an interconnect
- `confirm_private_virtual_interface`: Accepts ownership of a private virtual interface created by another AWS account
- `confirm_public_virtual_interface`: Accepts ownership of a public virtual interface created by another AWS account
- `confirm_transit_virtual_interface`: Accepts ownership of a transit virtual interface created by another AWS account
- `create_bgp_peer`: Creates a BGP peer on the specified virtual interface
- `create_connection`: Creates a connection between a customer network and a specific AWS Direct Connect location
- `create_direct_connect_gateway`: Creates a Direct Connect gateway, which is an intermediate object that enables you to connect a set of virtual interfaces and virtual private gateways
- `create_direct_connect_gateway_association`: Creates an association between a Direct Connect gateway and a virtual private gateway
- `create_direct_connect_gateway_association_proposal`: Creates a proposal to associate the specified virtual private gateway or transit gateway with the specified Direct Connect gateway
- `create_interconnect`: Creates an interconnect between an AWS Direct Connect Partner's network and a specific AWS Direct Connect location
- `create_lag`: Creates a link aggregation group (LAG) with the specified number of bundled physical dedicated connections between the customer network and a specific AWS Direct Connect location
- `create_private_virtual_interface`: Creates a private virtual interface
- `create_public_virtual_interface`: Creates a public virtual interface
- `create_transit_virtual_interface`: Creates a transit virtual interface
- `delete_bgp_peer`: Deletes the specified BGP peer on the specified virtual interface with the specified customer address and ASN
- `delete_connection`: Deletes the specified connection
- `delete_direct_connect_gateway`: Deletes the specified Direct Connect gateway
- `delete_direct_connect_gateway_association`: Deletes the association between the specified Direct Connect gateway and virtual private gateway
- `delete_direct_connect_gateway_association_proposal`: Deletes the association proposal request between the specified Direct Connect gateway and virtual private gateway or transit gateway
- `delete_interconnect`: Deletes the specified interconnect
- `delete_lag`: Deletes the specified link aggregation group (LAG)
**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

**Examples**

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

## End(Not run)
```

**Delete Virtual Interface**

Deletes a virtual interface

**Describe Connection Loa**

Displays the specified connection or all connections in this Region

**Describe Connections**

Displays the specified connection or all connections in this Region

**Describe Direct Connect Gateway Association Proposals**

Describes one or more association proposals for connection between your Direct Connect gateway and virtual private gateway

**Describe Direct Connect Gateway Attachments**

Lists the attachments between your Direct Connect gateways and virtual private gateways

**Describe Direct Connect Gateways**

Lists all your Direct Connect gateways or only the specified Direct Connect gateway

**Describe Hosted Connections**

Lists the hosted connections that have been provisioned on the specified Direct Connect gateway

**Describe Interconnect Loa**

Lists the interconnects owned by the AWS account or only the specified interconnect

**Describe Interconnects**

Lists all your link aggregation groups (LAG) or the specified LAG

**Describe Locations**

Lists the AWS Direct Connect locations in the current AWS Region

**Describe Tags**

Displays all virtual interfaces for an AWS account

**Describe Virtual Gateways**

Lists the virtual private gateways owned by the AWS account

**Describe Virtual Interfaces**

Disassociates a connection from a link aggregation group (LAG)

**Disassociate Connection From Lag**

Lists the virtual interface failover test history

**List Virtual Interface Test History**

Starts the virtual interface failover test that verifies your configuration meets your resiliency requirements

**Start Bgp Failover Test**

Starts the virtual interface failover test that verifies your configuration meets your resiliency requirements

**Stop Bgp Failover Test**

Stops the virtual interface failover test

**Tag Resource**

Adds one or more tags to the specified AWS Direct Connect resource

**Update Direct Connect Gateway Association**

Removes one or more tags from the specified AWS Direct Connect resource

**Update Gateway Association**

Updates the specified attributes of the Direct Connect gateway association

**Update Lag**

Updates the attributes of the specified link aggregation group (LAG)

**Update Virtual Interface Attributes**

Updates the specified attributes of the specified virtual private interface
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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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_region
- add_tags_to_resource
- cancel_schema_extension
- connect_directory
- create_alias

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
Add two domain controllers in the specified Region for the specified directory
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
create_computer
create_conditional_forwarder
create_directory
create_log_subscription
create_microsoft_ad
create_trust
delete_conditional_forwarder
delete_directory
delete_log_subscription
delete_trust
deregister_certificate
deregister_event_topic
describe_certificate
describe_conditional_forwarders
describe_directories
describe_domain_controllers
describe_event_topics
describe_ldaps_settings
describe_regions
describe_shared_directories
describe_snapshots
describe_trusts
disable_client_authentication
disable_ldaps
disable_radius
disable_sso
enable_client_authentication
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_region
remove_tags_from_resource
reset_user_password
restore_from_snapshot
share_directory

create_computer
create_conditional_forwarder
create_directory
create_log_subscription
create_microsoft_ad
create_trust
delete_conditional_forwarder
delete_directory
delete_log_subscription
delete_trust
deregister_certificate
deregister_event_topic
describe_certificate
describe_conditional_forwarders
describe_directories
describe_domain_controllers
describe_event_topics
describe_ldaps_settings
describe_regions
describe_shared_directories
describe_snapshots
describe_trusts
disable_client_authentication
disable_ldaps
disable_radius
disable_sso
enable_client_authentication
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_region
remove_tags_from_resource
reset_user_password
restore_from_snapshot
share_directory

Creates an Active Directory computer object in the specified 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 domain
Deletes from the system the certificate that was registered for secure LDAP or client certificate authentication
Removes the specified directory as a publisher to the specified SNS topic
Displays information about the certificate registered for secure LDAP or client certificate authentication
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
Provides information about the Regions that are configured for multi-Region replication
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
Disables alternative client authentication methods for the specified directory
Deactivates LDAP secure calls for the specified directory
Disables multi-factor authentication (MFA) with the Remote Authentication Dial In User Service (RADIUS) server for an AD Connector or Microsoft AD directory
Disables single sign-on for a directory
Enables alternative client authentication methods for the specified 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) server 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 secure LDAP or client certificate authentication
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 a secure LDAP or client certificate authentication
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
Stops all replication and removes the domain controllers from the specified Region
Removes tags from a directory
Resets the password for any user in your AWS Managed Microsoft AD or Simple AD directory
Restores a directory using an existing directory snapshot
Shares a specified directory (DirectoryId) in your AWS account (directory owner) with another AWS account.
### start_schema_extension
Applies a schema extension to a Microsoft AD directory

### unshare_directory
Stops the directory sharing between the directory owner and consumer accounts

### update_conditional_forwarder
Updates a conditional forwarder that has been set up for your AWS directory

### update_number_of_domain_controllers
Adds or removes domain controllers to or from the directory

### update_radius
Updates the Remote Authentication Dial In User Service (RADIUS) server information for an AD Connector or Microsoft AD directory

### update_trust
Updates the trust that has been set up between your AWS Managed Microsoft AD directory and an on-premises Active Directory

### verify_trust
AWS Directory Service for Microsoft Active Directory allows you to configure and verify trust relationships.

---

#### Examples

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

## End(Not run)
```

---

### dlm

#### 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.

#### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
svc <- dlm(
  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_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)
```

---

**docdb**

*Amazon DocumentDB with MongoDB compatibility*

**Description**

Amazon DocumentDB API documentation
Usage

docdb(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- docdb(
  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 metadata tags to an Amazon DocumentDB resource
apply_pending_maintenance_action Applies a pending maintenance action to a resource (for example, to an Amazon DocumentDB instance)
copy_db_cluster_parameter_group Copies the specified cluster parameter group
copy_db_cluster_snapshot Copies a snapshot of a cluster
create_db_cluster Creates a new Amazon DocumentDB cluster
create_db_cluster_parameter_group Creates a new cluster parameter group
create_db_cluster_snapshot Creates a snapshot of a cluster
create_db_instance Creates a new instance
create_db_subnet_group Creates a new subnet group
delete_db_cluster Deletes a previously provisioned cluster
delete_db_cluster_parameter_group Deletes a specified cluster parameter group
delete_db_cluster_snapshot Deletes a cluster snapshot
delete_db_instance Deletes a previously provisioned instance
delete_db_subnet_group Deletes a subnet group
describe_certificates Returns a list of certificate authority (CA) certificates provided by Amazon DocumentDB
describe_db_cluster_parameter_groups Returns a list of DBClusterParameterGroup descriptions
describe_db_cluster_parameters
describe_db_clusters
describe_db_cluster_snapshot_attributes
describe_db_cluster_snapshots
describe_db_engine_versions
describe_db_instances
describe_db_subnet_groups
describe_engine_default_cluster_parameters
describe_event_categories
describe_events
describe_orderable_db_instance_options
describe_pending_maintenance_actions
failover_db_cluster
list_tags_for_resource
modify_db_cluster
modify_db_cluster_parameter_group
modify_db_cluster_snapshot_attribute
modify_db_instance
modify_db_subnet_group
reboot_db_instance
remove_tags_from_resource
reset_db_cluster_parameter_group
restore_db_cluster_from_snapshot
restore_db_cluster_to_point_in_time
start_db_cluster
stop_db_cluster

Returns the detailed parameter list for a particular cluster parameter group
Returns information about provisioned Amazon DocumentDB clusters
Returns a list of cluster snapshot attribute names and values for a manual DB cluster snapshot
Returns information about cluster snapshots
Returns a list of the available engines
Returns information about provisioned Amazon DocumentDB instances
Returns a list of DBSubnetGroup descriptions
Returns the default engine and system parameter information for the cluster 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 instances, security groups, snapshots, and DB parameter groups
Returns a list of orderable instance options for the specified engine
Returns a list of resources (for example, instances) that have at least one pending maintenance action
Forces a failover for a cluster
Lists all tags on an Amazon DocumentDB resource
Modifies a setting for an Amazon DocumentDB cluster
Modifies the parameters of a cluster parameter group
Adds an attribute and values to, or removes an attribute and values from, a manual Amazon DocumentDB cluster snapshot
Modifies settings for an instance
Modifies an existing subnet group
You might need to reboot your instance, usually for maintenance reasons
Removes metadata tags from an Amazon DocumentDB resource
Modifies the parameters of a cluster parameter group to the default value
Creates a new cluster from a snapshot or cluster snapshot
Restores a cluster to an arbitrary point in time
Restarts the stopped cluster that is specified by DBClusterIdentifier
Stops the running cluster that is specified by DBClusterIdentifier

Examples

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

## End(Not run)
```

---

dynamodb  Amazon DynamoDB

Description

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 provision, 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

dynamodb(config = list())

Arguments

config

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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_execute_statement This operation allows you to perform batch reads and writes on data stored in DynamoDB.
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
create_global_table
create_table
delete_backup
delete_item
delete_table
describe_backup
describe_continuous_backups
describe_contributor_insights
describe_endpoints
describe_global_table
describe_global_table_settings
describe_kinesis_streaming_destination
describe_limits
describe_table
describe_table_replica_auto_scaling
describe_time_to_live
disable_kinesis_streaming_destination
enable_kinesis_streaming_destination
execute_statement
execute_transaction
export_table_to_point_in_time
get_item
list_backups
list_contributor_insights
list_exports
list_global_tables
list_tables
list_tags_of_resource
put_item
query
restore_table_from_backup
restore_table_to_point_in_time
scan
tag_resource	ranspose_get_items
transpose_write_items
untag_resource
update_continuous_backups
update_contributor_insights
update_global_table
update_global_table_settings
update_item
update_table
update_table_replica_auto_scaling
update_time_to_live

create a backup for an existing table
Creates a backup for an existing table

create global table from an existing table
The CreateTable operation adds a new table to your account

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

describe_backup
Describes an existing backup of a table

describe continuous backups
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 endpoints
Returns the regional endpoint information

describe export
Describes an existing table export

describe global table
Returns information about the specified global table

describe global table settings
Describes Region-specific settings for a global table

describe limits
Returns information about the status of Kinesis streaming

describe table
Returns the current provisioned-capacity quotas for your AWS account in a Region, for all services

describe table_replica_auto_scaling
Describes auto scaling settings across replicas of the global table at once

describe time to live
Gives a description of the Time to Live (TTL) status on the specified table

disable kinesis streaming destination
Stops replication from the DynamoDB table to the Kinesis data stream

enable kinesis streaming destination
Starts table data replication to the specified Kinesis data stream at a timestamp chosen during the enable workflow

execute statement
This operation allows you to perform reads and singleton writes on data stored in DynamoDB, using PartiQL

execute transaction
This operation allows you to perform transactional reads or writes on data stored in DynamoDB

export table to point in time
Exports table data to an S3 bucket

get item
The GetItem operation returns a set of attributes for the item with the given primary key

list backups
List backups associated with an AWS account

list contributor insights
Lists completed exports within the past 90 days

list global tables
Lists all global tables that have a replica in the specified Region

list tables
Returns an array of table names associated with the current account and endpoint

list tags of resource
List all tags on an Amazon DynamoDB resource

put item
Creates a new item, or replaces an old item with a new item

query
This operation allows you to perform reads and singleton writes on data stored in DynamoDB, using PartiQL

restore table from backup
The Query operation finds items based on primary key values

restore table to point in time
Creates a new table from an existing backup

scan
Restores the specified table to the specified point in time within EarliestRestorableDateTime and LatestRestorableDateTime

tag resource
The Scan operation returns one or more items and item attributes by accessing every item

transpose get items
Associate a set of tags with an Amazon DynamoDB resource

transpose write items
TransactGetItems is a synchronous operation that atomically retrieves multiple items

untag resource
TransactWriteItems is a synchronous write operation that groups up to 25 action requests

update continuous backups
Updates the association of tags from an Amazon DynamoDB resource

update contributor insights
UpdateContinuousBackups enables or disables point in time recovery for the specified table

update global table
Updates the status for contributor insights for a specific table or index

update global table settings
Adds or removes replicas in the specified global table

update item
Updates settings for a global table

update table
Edits an existing item’s attributes, or adds a new item to the table if it does not already exist

update table replica auto scaling
Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams settings for a given table

update time to live
Updates auto scaling settings on your global tables at once

The UpdateTimeToLive method enables or disables Time to Live (TTL) for the specified table.
Examples

```r
## Not run:
svc <- dynamodb()
# This example reads multiple items from the Music table using a batch of
# three GetItem requests. Only the AlbumTitle attribute is returned.
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"
          )
        ),
        ProjectionExpression = "AlbumTitle"
      )
    )
  )
)
## End(Not run)
```

---

**dynamodbstreams**  
*Amazon DynamoDB Streams*
**Description**

Amazon DynamoDB Streams provides 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.

**Value**

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```python
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), the composition of its shards, and its corresponding DynamoDB table
- **get_records**
  Retrieves the stream records from a given shard
- **get_shard_iterator**
  Returns a shard iterator
- **listStreams**
  Returns an array of stream ARNs associated with the current account and endpoint
Examples

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

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

```r
ec2(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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_multicast_domain_associations
  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_ipv_6_addresses
  assign_private_ip_addresses
  associate_address
  associate_client_vpn_target_network
  associate_dhcp_options
  associate_enclave_certificate_iam_role
  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

Accepts the Convertible Reserved Instance exchange quote
Accepts a request to associate subnets with a transit gateway
Accepts a transit gateway peering attachment request
Accepts a request to attach a VPC to a transit gateway
Accepts one or more interface VPC endpoint connections
Accepts a VPC peering connection request
Advertises an IPv4 or IPv6 address range that is provisioned for use with your VPC
Allocates an Elastic IP address to your AWS account
Allocates a Dedicated Host to your account
Applies a security group to the association between a VPC and a subnet
Assigns one or more IPv6 addresses to the specified subnet
Assigns one or more secondary private IP addresses to a VPC
Associates an Elastic IP address, or carrier IP address (for instances that are in subnets in Wavelength Zones) with an instance or a network interface
Associates a target network with a Client VPN endpoint
Associates a set of DHCP options (that you’ve previously created) with the specified VPC
Associates an AWS Identity and Access Management (IAM) role with an AWS Certificate Manager (ACM) certificate
Associates an IAM instance profile with a running or stopped instance
Associates a subnet in your VPC or an internet gateway
Associates a CIDR block with your subnet
Associates the specified subnets and transit gateway attachments with the specified transit gateway multicast domain
Assigns the specified attachment with the specified VPC
Associates a CIDR block with your VPC
Links an EC2-Classic instance to a ClassicLink-enabled VPC
Attaches an internet gateway or a virtual private gateway to a VPC
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
[Temporary] Adds the specified egress rules to a security group
Adds the specified ingress rules to a security group
Bundles an Amazon instance store-backed Windows instance
Cancels a bundling operation for an instance store-backed Windows instance
Cancels the specified Capacity Reservation, release associated reserved capacity, and change the Capacity Reservation’s state to cancelled
Cancels an active conversion task
Cancels an active export task
Cancels an in-process import virtual machine operation
cancel_reserved_instances_listing
cancel_spot_fleet_requests
cancel_spot_instance_requests
confirm_product_instance
copy_fpga_image
copy_image
copy_snapshot
create_capacity_reservation
create_carrier_gateway
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_managed_prefix_list
create_nat_gateway
create_network_acl
create_network_acl_entry
create_network_insights_path
create_network_interface
create_network_interface_permission
create_placement_group
create_reserved_instances_listing
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

cancels the specified Reserved Instance listing in the Reserved Instance 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) to the current Region.
Initiates the copy of an AMI from the specified source Region to the current Region.
Copies a point-in-time snapshot of an EBS volume.
Creates a new Capacity Reservation with the specified attributes.
Creates a carrier gateway.
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.
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 for your VPC.
Launches an EC2 Fleet.
Create one or more flow logs to capture information for a network interface.
Creates an Amazon FPGA Image (AFI) from the specified design checkpoint (DCP).
Exports an AMI from the specified source Region.
Creates an internet gateway for use with a VPC.
Creates a 2048-bit RSA key pair with the specified key pair 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 managed prefix list.
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 parameters.
Creates a path to analyze for reachability.
Creates a network interface in the specified subnet.
Grants an AWS-authorized account permission to launch an instance in the specified VPC.
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 a specified VPC.
Adds or overwrites only the specified tags for the specified EC2 instance.
Creates a Traffic Mirror filter.
Creates a Traffic Mirror filter rule.
Creates a Traffic Mirror session.
Creates a target for your Traffic Mirror session.
create_transit_gateway
create_transit_gateway_connect
create_transit_gateway_connect_peer
create_transit_gateway_multicast_domain
create_transit_gateway_peering_attachment
create_transit_gateway_prefix_list_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
create_vpc_endpoint_service_configuration
create_vpn_connection
create_vpn_connection_route
create_vpn_gateway
delete_carrier_gateway
delete_client_vpn_endpoint
delete_client_vpn_route
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_managed_prefix_list
delete_nat_gateway
delete_network_acl
delete_network_acl_entry
delete_network_insights_analysis
delete_network_insights_path
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

Creates a transit gateway
Creates a Connect attachment from a specified transit gateway
Creates a Connect peer for a specified transit gateway
Creates a multicast domain using the specified transit gateway
Requests a transit gateway peering attachment between two transit gateways
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 EC2 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 service configuration
Requests a VPC peering connection between two VPCs
Creates a VPN connection between an existing VPC and a customer gateway
Creates a static route associated with a VPN connection
Creates a virtual private gateway
Deletes a carrier 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 and local gateway route table
Deletes the specified managed prefix list
Deletes the specified NAT gateway
Deletes the specified network ACL
Deletes the specified ingress or egress entry (rule) from the specified network ACL
Deletes the specified network insights analysis
Deletes the specified path
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 Instance
Deletes the specified route from the specified route table
Deletes the specified route table
Deletes a security group
Deletes the specified snapshot
Deletes the data feed for Spot Instances
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_connect
delete_transit_gateway_connect_peer
delete_transit_gateway_multicast_domain
delete_transit_gateway_peering_attachment
delete_transit_gateway_prefix_list_reference
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
delete_vpc_peering_connection
delete_vpn_connection
delete_vpn_connection_route
delete_vpn_gateway
deprovision_byoip_cidr
deregister_instance_event_notification_attributes
deregister_traffic_mirror_filter

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 Connect attachment
Deletes the specified Connect peer
Deletes the specified transit gateway multicast domain
Deletes a transit gateway peering attachment
Deletes a reference (route) to a prefix list in a specified VPC
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
Deletes a VPC peering connection
Deletes the specified VPN connection
Deletes the specified static route associated with an endpoint
Deletes the specified virtual private gateway
Releases the specified address range that you provisioned
Deregisters the specified AMI
Deregisters tag keys to prevent tags that have the specified key from being included in scheduled event notifications
Deregisters the specified members (network interfaces) from the specified transit gateway multicast group
Describes attributes of your AWS account
Describes the specified Elastic IP addresses or all of your Elastic IP addresses
Describes the longer ID format settings for all resources
Describes the Availability Zones, Local Zones, and Wavelength Zones that are available to you
Describes the specified bundle tasks or all of your bundle tasks
Describes the IP address ranges that were specified
Describes one or more of your Capacity Reservations
Describes one or more of your carrier gateways
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 types
Describes the routes for the specified Client VPN endpoint
Describes the target networks associated with the specified Client VPN endpoint
Describes the specified conversion tasks or all your conversion tasks
Describes one or more of your VPN customer gateways
Describes one or more of your DHCP options sets
Describes one or more of your 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_host_reservation_offerings
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
describe_instance_event_notification_attributes
describe_instances
describe_instance_status
describe_instance_type_offerings
describe_instance_types
describe_internet_gateways
describe_ipv_6_pools
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_managed_prefix_lists
describe_moving_addresses
describe_nat_gateways
describe_network_acls
describe_network_insights_analyses
describe_network_insights_paths
describe_network_interface_attribute
describe_network_interface_permissions
describe_network_interfaces
describe_placement_groups

Describes the Elastic Graphics accelerator associated with your instances.
Describes the specified export image tasks or all of your export image tasks.
Describes the specified export instance tasks or all of your export instance tasks.
Describes the state of fast snapshot restores for your snapshots.
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 associated with Dedicated Hosts.
Describes reservations that are associated with Dedicated Hosts.
Describes your IAM instance profile associations.
Describes the ID format settings for resources for your account.
Describes the specified attribute of the specified instance or all instances.
Describes the specified images (AMIs, AKIs, and RIs).
Displays details about an import virtual machine or import snapshot tasks.
Describes your import snapshot tasks.
Describes the specified attribute of the specified instance.
Describes the credit option for CPU usage of the specified instance.
Describes the tag keys that are registered to appear in scheduled event notifications for resources.
Describes the specified instances or all instances.
Describes the status of the specified instances or all instances.
Returns a list of all instance types offered.
Describes the details of the instance types that are available.
Describes one or more of your internet gateways.
Describes your IPv6 address pools.
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 associations between virtual interfaces, virtual interface groups, and VPCs.
Describes one or more local gateways.
Describes the specified local gateway virtual interface.
Describes the specified local gateway virtual interfaces.
Describes your managed prefix lists and any associated resources.
Describes your Elastic IP addresses that are assigned to your account.
Describes one or more of your NAT gateways.
Describes one or more of your network ACLs.
Describes one or more of your network insights.
Describes one or more of your network interfaces.
Describes one or more of your paths.
Describes a network interface attribute.
Describes the permissions for your network interfaces.
Describes one or more of your network interfaces.
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_offerings
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
describe_transit_gateway_attachments
describe_transit_gateway_connect_peers
describe_transit_gateway_connects
describe_transit_gateway_multicast_domains
describe_transit_gateway_peering_attachments
describe_transit_gateway_route_tables
describe_transit_gateways
describe_transit_gateway_vpc_attachments
describe_volume_attribute
describe_volumes
describe_volumes_modifications
describe_volume_status
describe_vpc_attribute
describe_vpc_classic_link
describe_vpc_classic_link_dns_support
describe_vpc_endpoint_connection_notifications
describe_vpc_endpoint_connections
describe_vpc_endpoints
describe_vpc_endpoint_service_configurations
describe_vpc_endpoint_service_permissions
describe_vpc_endpoint_services

Describes available AWS services in a prefix list format.
Describes the ID format settings for the root user and all IAM users.
Describes the specified IPv4 address pools.
Describes the Regions that are enabled for your account.
Describes one or more of the Reserved Instances that you purchased.
Describes your account’s Reserved Instance listings.
Describes the modifications made to your Reserved Instances.
Describes Reserved Instance offerings that are available.
Describes one or more of your route tables.
Finds available schedules that meet the specified criteria.
Describes the specified Scheduled Instances or all Scheduled Instances.
[Only for VPC] 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 snapshot.
Describes the specified EBS snapshots available to you.
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.
[Only for VPC] 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.
Describes one or more Traffic Mirror targets.
Information about one or more Traffic Mirror targets.
Describes one or more attachments between resources.
Describes one or more Connect peers.
Describes one or more attachments to transit gateways.
Describes one or more transit gateway interconnect attachments.
Describes the ClassicLink status of one or more VPCs.
Describes the ClassicLink DNS support status of one or more VPCs.
Describes the connection notifications for VPC endpoints.
Describes the VPC endpoint connections to your VPC.
Describes one or more of your VPC endpoints.
Describes the VPC endpoint service configuration.
Describes the principals (service consumers) that are permitted to find your VPC endpoints.
Describes available services to which you can create VPC endpoints.
describe_vpc_peering_connections
describe_vpcs
describe_vpn_connections
describe_vpn_gateways
detach_classic_link_vpc
detach_internet_gateway
detach_network_interface
detach_volume
detach_vpn_gateway
disable_ebs_encryption_by_default
disable_fast_snapshot_restores
disable_transit_gateway_route_table_propagation
disable_vgw_route_propagation
disable_vpc_classic_link
disable_vpc_classic_link_dns_support
disable_vpc_dhcp_options
disassociate_address
disassociate_client_vpn_target_network
disassociate_enclave_certificate_iam_role
disassociate_subnet_cidr_block
disassociate_transit_gateway_multicast_domain
disassociate_transit_gateway_route_table
disassociate_vpc_cidr_block
enable_ebs_encryption_by_default
enable_fast_snapshot_restores
enable_transit_gateway_route_table_propagation
enable_vgw_route_propagation
enable_volume_io
enable_vpc_classic_link
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_associated_enclave_certificate_iam_roles
get_associated_ipv6_pool_cidrs
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_groups_for_capacity_reservation
get_host_reservation_purchase_preview
get_launch_template_data
get.managed_prefix_listAssociations

Describes one or more of your VPC peering connections
Describes one or more of your VPCs
Describes one or more of your VPN connections
Unlinks (detaches) a linked EC2-Classic instance
Detaches an internet gateway from a VPC, disabling connectivity between the internet and the VPC
Detaches a network interface from an instance
Detaches an EBS volume from an instance
Detaches a virtual private gateway from a VPC
Disables EBS encryption by default for your account
Disables fast snapshot restores for the specified snapshot
Disables the specified resource attachment from propagating routes
Disables a virtual private gateway (VGW) from propagating routes
Disables ClassicLink for a VPC
Disables ClassicLink DNS support for a VPC
Disassociates an Elastic IP address from the instance
Disassociates a target network from the specified subnet
Disassociates an IAM role from an AWS Certificate Manager (ACM) certificate
Disassociates an IAM instance profile from a running or stopped instance
Disassociates a subnet or gateway from a route table
Disassociates a CIDR block from a subnet
Disassociates the specified subnets from the specified transit gateway
Disassociates a resource attachment from a transit gateway
Disassociates a CIDR block from a VPC
Enables EBS encryption by default for your account
Enables fast snapshot restores for the specified snapshot
Enables the specified attachment to propagate routes
Enables a virtual private gateway (VGW) to propagate routes
Enables I/O operations for a volume that had I/O operations disabled
Enables a VPC for ClassicLink
Enables a VPC to support DNS hostname resolution
Downloads the client certificate revocation list for the specified Client VPN endpoint
Downloads the contents of the Client VPN endpoint configuration file
Exports an Amazon Machine Image (AMI) to a VM file
Exports routes from the specified transit gateway
Returns the IAM roles that are associated with the specified certificate
Gets information about the IPv6 CIDR block assignments for a transit gateway
Gets usage information about a Capacity Reservation
Describes the allocations from the specified custom resource
Gets the console output for the specified instance
Retrieves a JPG-format screenshot of a running instance
Describes the default credit option for CPU usage
Describes the default customer master key (CMK)
Lists the resource groups to which a Capacity Reservation is associated
Preview a reservation purchase with configurable parameters
Retrieves the configuration data of the specified instance
Gets information about the resources that are associated with the specified certificate
Get managed prefix list entries
Get password data
Get reserved instances exchange quote
Get transit gateway attachment propagations
Get transit gateway multicast domain associations
Get transit gateway prefix list references
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 availability zone group
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 managed prefix list
Modify network interface attribute
Modify reserved instances
Modify snapshot attribute
Modify spot fleet request
Modify subnet attribute
Modify Traffic Mirror filter network services
Modify Traffic Mirror filter rule
Modify Traffic Mirror session
Modify transit gateway
Modify transit gateway prefix list reference
Modify transit gateway VPC attachment
Modify volume
Modify volume attribute
Modify VPC attribute
Modify VPC endpoint

Gets information about the entries for a specified managed prefix list.
Retrieves the encrypted administrator password for a running Windows instance.
Returns a quote and exchange information for exchanging Convertible Reserved Instances.
Gets information about the associations for the specified resource attachment.
Gets information about the prefix list references in a specified transit gateway route table.
Gets information about the associations for the specified transit gateway route table.
Gets information about the route table propagations for the specified transit gateway route table.
Uploads a client certificate revocation list to the specified Client VPN endpoint.
Imports a single or multi-volume disk images or EBS snapshots into an 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.
Encrypts an EBS snapshot.
Creates an import volume task using metadata from the specified disk image.
Changes the opt-in status of the Local Zone and VPC
Modifies a Capacity Reservation’s capacity and placement attributes.
Modifies the specified Client VPN endpoint.
Modifies the default credit option for CPU usage.
Changes the default customer master key (CMK).
Modifies the specified EC2 fleet.
Modifies the specified attribute of the specified AMI.
Modifies the auto-placement setting of a Dedicated Host.
Modifies the ID format of a resource for a specific IAM user.
Modifies the ID format for the specified resource
Modifies the specified attribute of the specified AMI.
Modifies the specified attribute of the specified instance.
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 event.
Modifies the instance metadata parameters on a running instance.
Modifies the placement attributes for a specified instance.
Modifies a launch template.
Modifies the specified managed prefix list.
Modifies the specified network interface attribute.
Modifies the Availability Zone, instance count, instance type, or VPC.
Adds or removes permission settings for the specified snapshot.
Modifies the specified Spot Fleet request.
Modifies a subnet attribute.
Allows or restricts mirroring network services.
Modifies the specified Traffic Mirror rule.
Modifies a Traffic Mirror session.
Modifies the specified transit gateway.
Modifies a reference (route) to a prefix list in a specified transit gateway.
Modifies the specified VPC attachment.
You can modify several parameters of an existing volume.
Modifies a volume attribute.
Modifies the specified attribute of the specified VPC.
Modifies attributes of a specified 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_connection_options
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_instance_event_notification_attributes
register_transit_gateway_multicast_group_members
register_transit_gateway_multicast_group_sources
reject_transit_gateway_multicast_domain_associations
reject_transit_gateway_peering_attachment
reject_transit_gateway_vpc_attachment
reject_vpc_endpoint_connections
reject_vpc_peering_connection
release_address
release_host reserves
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
restore_managed_prefix_list_version
revoke_client_vpn_ingress
revoke_security_group_egress
revoke_security_group_ingress
run_instances

Modifies a connection notification for VPC endpoint service connection notifications.
Modifies the attributes of your VPC endpoint service configuration.
Modifies the permissions for your VPC endpoint service.
Modifies the VPC peering connection options on one side of a VPC peering connection.
Modifies the instance tenancy attribute of the specified VPC.
Modifies the customer gateway or the target gateway of an AWS Site-to-Site VPN connection.
Modifies the connection options for your Site-to-Site VPN connection.
Modifies the VPN tunnel endpoint certificate.
Modifies the options for a VPN tunnel in an AWS Site-to-Site VPN connection.
Enables detailed monitoring for a running instance.
Moves an Elastic IP address from the EC2-Classic platform to the EC2-VPC platform.
Provisions an IPv4 or IPv6 address range for use with your AWS resources through bring your own IP addresses (BYOIP) and creates a corresponding address pool.
Purchases a reservation with configurations that match those of your Dedicated Host.
Purchases a Reserved Instance for use with your account.
Purchases the Scheduled Instances with the specified schedule.
Requests a reboot of the specified instances.
Registers an AMI.
Registers a set of tag keys to include in scheduled event notifications.
Registers members (network interfaces) with the specified transit gateway.
Registers sources (network interfaces) with the specified transit gateway.
Rejects a request to associate cross-account subnet with a transit gateway.
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 Dedicated Host, it can be released.
Replaces an IAM instance profile association.
Replaces a network ACL association.
Replaces an entry (rule) in a network ACL.
Replaces existing route within a route table in a VPC.
Changes the route table associated with the specified subnet.
Replaces the specified route in the specified transit gateway route table.
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 EBS encryption.
Resets the specified attribute of the specified Amazon FPGA Image (AFI).
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 released.
Restores the entries from a previous version of a network ACL.
Removes an ingress authorization rule from a Classic Link VPC.
Removes the specified egress rules from a security group.
Removes the specified ingress rules from a security group.
Launches the specified number of instances using the specified AMI.
run_scheduled_instances
search_local_gateway_routes
search_transit_gateway_multicast_groups
search_transit_gateway_routes
send_diagnostic_interrupt
start_instances
start_network_insights_analysis
start_vpc_endpoint_service_private_dns_verification
stop_instances
terminate_client_vpn_connections
terminate_instances
unassign_ipv_6_addresses
unassign_private_ip_addresses
unmonitor_instances
update_security_group_rule_descriptions_egress
update_security_group_rule_descriptions_ingress
withdraw_byoip_cidr

Examples

```r
## Not run:
svc <- ec2()
# This example allocates an Elastic IP address to use with an instance in
# a VPC.
svc$allocate_address(
  Domain = "vpc"
)

## End(Not run)
```

---

**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

```r
ec2instanceconnect(config = list())
```

### Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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
## Not run:
svc <- ec2instanceconnect()  
# 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.  
svc$send_ssh_public_key(    
  AvailabilityZone = "us-west-2a",  
  InstanceId = "i-abcd1234",  
  InstanceOSUser = "ec2-user",  
  SSHPublicKey = "ssh-rsa AAAAB3NzaC1yc2EAAAADQABAAABABQF1Hqj2eqCdrGU6d..."  
)
## End(Not run)
```

---

**ecr**  
*Amazon EC2 Container Registry*
Description

Amazon Elastic Container Registry

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

Usage

```python
ecr(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
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

- **batch_check_layer_availability**: Checks the availability of one or more image layers in a repository
- **batch_delete_image**: Deletes a list of specified images within a repository
- **batch_get_image**: Gets detailed information for an image
- **complete_layer_upload**: Informs Amazon ECR that the image layer upload has completed for a specified registry, repository name, and upload ID
- **create_repository**: Creates a repository
- **delete_lifecycle_policy**: Deletes the lifecycle policy associated with the specified repository
delete_registry_policy
delete_repository
delete_repository_policy
describe_images
describe_image_scan_findings
describe_registry
describe_repositories
get_authorization_token
get_download_url_for_layer
get_lifecycle_policy
get_lifecycle_policy_preview
get_registry_policy
get_repository_policy
initiate_layer_upload
list_images
list_tags_for_resource
put_image
put_image_scanning_configuration
put_image_tag_mutability
put_lifecycle_policy
put_registry_policy
put_replication_configuration
set_repository_policy
start_image_scan
start_lifecycle_policy_preview
tag_resource
untag_resource
upload_layer_part

Deletes the registry permissions policy
Deletes a repository
Deletes the repository policy associated with the specified repository
Returns metadata about the images in a repository
Returns the scan findings for the specified image
Describes the settings for a registry
Describes image repositories in a registry
Retrieves an authorization token
Retrieves the pre-signed Amazon S3 download URL corresponding to an image layer
Retrieves the lifecycle policy for the specified repository
Retrieves the results of the lifecycle policy preview request for the specified repository
Retrieves the permissions policy for a registry
Retrieves the repository policy for the specified repository
Notifies Amazon ECR that you intend to upload an image layer
Lists all the image IDs for the specified repository
List the tags for an Amazon ECR resource
Creates or updates the image manifest and tags associated with an image
Updates the image scanning configuration for the specified repository
Updates the image tag mutability settings for the specified repository
Updates the lifecycle policy for the specified repository
Creates or updates the permissions policy for your registry
Creates or updates the replication configuration for a registry
Applies a repository policy to the specified repository to control access permissions
Starts an image vulnerability scan
Starts a preview of a lifecycle policy for the specified repository
Adds specified tags to a resource with the specified ARN
Deletes specified tags from a resource
Uploads an image layer part to Amazon ECR

Examples

```r
## Not run:
svc <- ecr()
# This example deletes images with the tags precise and trusty in a
# repository called ubuntu in the default registry for an account.
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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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_capacity_provider**: Deletes the specified capacity provider
- **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_task_sets**: Describes a specified task or tasks
- **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**: List 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 task 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**: This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent
- **register_task_definition**: Registers a new task definition from the supplied family and containerDefinitions
- **run_task**: Starts a new task using the specified task definition
- **start_task**: Starts a new task from the specified task definition on the specified container instance or instances
- **stop_task**: Stops a running task
- **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_capacity_provider Modifies the parameters for a capacity provider
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 Updating the task placement strategies and constraints on an Amazon ECS service remains in preview and is a Beta Service as defined by and subject to the Beta Service Participation Service Terms located at https://aws
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
## Not run:
svc <- ecs()
# This example creates a cluster in your default region.
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**

```r
esf(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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_access_point` Creates an EFS access point
- `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_access_point` Deletes the specified access point
- `delete_file_system` Deletes a file system, permanently severing access to its contents
- `delete_file_system_policy` Deletes the FileSystemPolicy for the specified file system
- `delete_mount_target` Deletes the specified mount target
- `delete_tags` Deletes the specified tags from a file system
- `describe_access_points` Returns the description of a specific Amazon EFS access point if the AccessPointId is provided
- `describe_backup_policy` Returns the backup policy for the specified EFS file system
- `describe_file_system_policy` Returns the FileSystemPolicy for the specified EFS 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
- `list_tags_for_resource` Lists all tags for a top-level EFS resource
- `modify_mount_target_security_groups` Modifies the set of security groups in effect for a mount target
- `put_backup_policy` Updates the file system’s backup policy
- `put_file_system_policy` Applies an Amazon EFS FileSystemPolicy to an Amazon EFS file system
- `put_lifecycle_configuration` Enables lifecycle management by creating a new LifecycleConfiguration object
- `tag_resource` Creates a tag for an EFS resource
- `untag_resource` Removes tags from an EFS resource
- `update_file_system` Updates the throughput mode or the amount of provisioned throughput of an existing file system
Examples

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

### 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.

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```python
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_addon` - Creates an Amazon EKS add-on
- `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_addon` - Deletes an Amazon EKS add-on
- `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_addon` - Describes an Amazon EKS add-on
- `describe_addon_versions` - Describes the Kubernetes versions that the add-on can be used with
- `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_addons` - Lists the available add-ons
- `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 managed 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_addon` - Updates an Amazon EKS add-on
- `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

```r
## Not run:
svc <- eks()
# The following example creates an Amazon EKS cluster called prod.
svc$create_cluster(
  version = "1.10",
  name = "prod",
  clientRequestToken = "1d2129a1-3d38-460a-9756-e5b91fddb951",
  resourcesVpcConfig = list(
    securityGroupIds = list(
      "sg-6979fe18"
    ),
    subnetIds = list(
      "subnet-6782e71e",
      "subnet-e7e761ac"
    )
  ),
  roleArn = "arn:aws:iam::012345678910:role/eks-service-role-AWSServiceRole..."
)
## End(Not run)
```

# elasticache

## Amazon 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

```r
elasticache(config = list())
```

### Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
Value
A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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_global_replication_group` Global Datastore for Redis offers fully managed, fast, reliable and secure cross-region replication
- `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
- `create_user` For Redis engine version 6
- `create_user_group` For Redis engine version 6
- `decrease_node_groups_in_global_replication_group` Decreases the number of node groups in a Global Datastore
- `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 a cache subnet group
- `delete_global_replication_group` Deleting a Global Datastore is a two-step process: Deletes an existing replication group
- `delete_replication_group` Deletes an existing snapshot
- `delete_snapshot`
delete_user
delete_user_group
describe_cache_clusters
describe_cache_engine_versions
describe_cache_parameter_groups
describe_cache_parameters
describe_cache_security_groups
describe_cache_subnet_groups
describe_engine_default_parameters
describe_events
describe_global_replication_groups
describe_replication_groups
describe_reserved_cache_nodes
describe_reserved_cache_nodes_offerings
describe_service_updates
describe_snapshots
describe_update_actions
describe_user_groups
describe_users
disassociate_global_replication_group
failover_global_replication_group
increase_node_groups_in_global_replication_group
increase_replica_count
list_allowed_node_type_modifications
list_tags_for_resource
modify_cache_cluster
modify_cache_parameter_group
modify_cache_subnet_group
modify_global_replication_group
modify_replication_group
modify_replication_group_shard_configuration
modify_user
modify_user_group
purchase_reserved_cache_nodes_offering
rebalance_slots_in_global_replication_group
reboot_cache_cluster
remove_tags_from_resource
reset_cache_parameter_group
revoke_cache_security_group_ingress
start_migration
test_failover

For Redis engine version 6
For Redis engine version 6
Returns information about all provisioned clusters if no cluster identifier is specified
Returns a list of the available cache engines and their versions
Returns a list of cache parameter group descriptions
Returns the detailed parameter list for a particular cache parameter group
Returns a list of cache security group descriptions
Returns a list of cache subnet group descriptions
Returns the default engine and system parameter information for the specified engine
Returns events related to clusters, cache security groups, and cache parameter groups
Returns information about a particular global replication group
Returns information about a particular replication group
Returns information about reserved cache nodes for this account, or about a specified reserved node
Lists available reserved cache node offerings
Returns details of the service updates
Returns information about cluster or replication group snapshots
Returns details of the update actions
Returns a list of user groups
Returns a list of users
Remove a secondary cluster from the Global Datastore using the Global Datastore name
Used to failover the primary region to a selected secondary region
Increase the number of node groups in the Global Datastore
Dynamically increases the number of replicas in a Redis (cluster mode disabled) replication group
Lists all available node types that you can scale your Redis cluster’s or replication group’s current node type
Lists all cost allocation tags currently on the named resource
Modifies the settings for a cluster
Modifies the parameters of a cache parameter group
Modifies an existing cache subnet group
Modifies the settings for a Global Datastore
Modifies the settings for a replication group
Modifies a replication group’s shards (node groups) by allowing you to
Changes user password(s) and/or access string
Changes the list of users that belong to the user group
Allows you to purchase a reserved cache node offering
Redistribute slots to ensure uniform distribution across existing shards
Reboots some, or all, of the cache nodes within a provisioned cluster
Removes the tags identified by the TagKeys list from the named resource
Modifies the parameters of a cache parameter group to the engine or system default
Revokes ingress from a cache security group
Start the migration of data
Represents the input of a TestFailover operation which test automatic failover for

Examples

## Not run:
svc <- elasticache()
svc$add_tags_to_resource(}

```r
```
elasticbeanstalk

Foo = 123
)

## End(Not run)

---

**elasticbeanstalk**  
**AWS Elastic Beanstalk**

**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 [https://elasticbeanstalk.s3.amazonaws.com/doc/2010-12-01/AWSElasticBeanstalk.wsdl](https://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](https://aws.amazon.com/tools/).

**Endpoints**

For a list of region-specific endpoints that AWS Elastic Beanstalk supports, go to [Regions and Endpoints](https://docs.aws.amazon.com/general/latest/gr/rande.html) in the *Amazon Web Services Glossary*.

**Usage**

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

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

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

Operations

abort_environment_update
apply_environment_managed_action
associate_environment_operations_role
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
disassociate_environment_operations_role
list_available_solution_stacks
list_platform_branches
list_platform_versions
list_tags_for_resource
rebuild_environment
request_environment_info
restart_app_server
retrieve_environment_info
swap_environment_cnam_es
terminate_environment

Cancels in-progress environment configuration update or application version deployment.
Applies a scheduled managed action immediately.
Add or change the operations role used by an environment.
Checks if the specified CNAME is available.
Create or update a group of environments that each run a separate component of an application.
Creates an application that has one configuration template named default and no application versions.
Creates an application version for the specified application.
Creates an AWS Elastic Beanstalk configuration template, associated with an application.
Launches an AWS Elastic Beanstalk environment for the specified application.
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.
Delete 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 set.
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.
Returns information about the overall health of the specified 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 a platform version.
Disassociate the operations role from an environment.
Returns a list of the available solution stack names, with the public version first and then in reverse chronological order.
Lists the platform branches available for your account in an AWS Region.
Lists the platform versions available for your account in an AWS Region.
Return 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.
update_application                  Updates the specified application to have the specified properties
update_application_resource_lifecycle Modifies lifecycle settings for an application
update_application_version          Updates the specified application version to have the specified properties
update_configuration_template       Updates the specified configuration template to have the specified properties
update_environment                 Updates 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_tags_for_resource           Update the list of tags applied to an AWS Elastic Beanstalk resource
validate_configuration_settings    Takes a set of configuration settings and either a configuration template or environment.

Examples

```
## Not run:
svc <- elasticbeanstalk()
# The following code aborts a running application version deployment for
# an environment named my-env:
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.
Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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

- `accept_inbound_cross_cluster_search_connection` Allows the destination domain owner to accept an inbound cross-cluster search connection request
- `add_tags` Attaches tags to an existing Elasticsearch domain
- `associate_package` Associates a package with an Amazon ES domain
- `cancel_elasticsearch_service_software_update` Cancels a scheduled service software update for an Amazon ES domain
- `create_elasticsearch_domain` Creates a new Elasticsearch domain
- `create_outbound_cross_cluster_search_connection` Creates a new cross-cluster search connection from a source domain to a destination domain
- `create_package` Create a package for use with Amazon ES domains
- `delete_elasticsearch_domain` Permanently deletes the specified Elasticsearch domain and all of its data
- `delete_elasticsearch_service_role` Deletes the service-linked role that Elasticsearch Service uses to manage and maintain VPC domains
- `delete_inbound_cross_cluster_search_connection` Allows the destination domain owner to delete an existing inbound cross-cluster search connection request
- `delete_outbound_cross_cluster_search_connection` Allows the source domain owner to delete an existing outbound cross-cluster search connection request
- `delete_package` Delete the package
- `describe_elasticsearch_domain` Returns domain configuration information about the specified Elasticsearch domain
- `describe_elasticsearch_domain_config` Provides cluster configuration information about the specified Elasticsearch domain
- `describe_elasticsearch_domains` Returns domain configuration information about the specified Elasticsearch domain
- `describe_elasticsearch_instance_type_limits` Describe Elasticsearch Limits for a given InstanceType and ElasticsearchVersion
- `describe_outbound_cross_cluster_search_connections` Lists all the inbound cross-cluster search connections for a destination domain
- `describe_packages` Describes all packages available to Amazon ES
- `describe_reserved_elasticsearch_instance_offerings` Lists available reserved Elasticsearch instance offerings
- `describe_reserved_elasticsearch_instances` Returns information about reserved Elasticsearch instances for this account
- `dissociate_package` Dissociates a package from the Amazon ES domain
- `get_compatible_elasticsearch_versions` Returns a list of upgrade compatible Elasticsearch versions
- `get_package_version_history` Returns a list of versions of the package, along with their creation time
get_upgrade_history
get_upgrade_status
list_domain_names
list_domains_for_package
list_elasticsearch_instance_types
list_elasticsearch_versions
list_packages_for_domain
list_tags
purchase_reserved_elasticsearch_instance_offering
reject_inbound_cross_cluster_search_connection
remove_tags
start_elasticsearch_service_software_update
update_elasticsearch_domain_config
update_package
upgrade_elasticsearch_domain

Examples

```r
## Not run:
svc <- elasticsearchservice()
svc$accept_inbound_cross_cluster_search_connection(
  Foo = 123
)

## End(Not run)
```

---

**Description**

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

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

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

To get started, create a load balancer with one or more listeners using `create_load_balancer`. Register your instances with the load balancer using `register_instances_with_load_balancer`. 

---

**elb**

**Elastic Load Balancing**
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

```python
elb(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>

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Operations

- **add_tags**
  Adds the specified tags to the specified load balancer
- **apply_security_groups_to_load_balancer**
  Associates one or more security groups with your load balancer in a virtual private cloud (VPC)
- **attach_load_balancer_to_subnets**
  Adds one or more subnets to the set of configured subnets for the specified load balancer
- **configure_health_check**
  Specifies the health check settings to use when evaluating the health state of your EC2 instances
- **create_app_cookie_stickiness_policy**
  Generates a stickiness policy with sticky session lifetimes that follow that of an application-generated cookie
- **create_lb_cookie_stickiness_policy**
  Generates a stickiness policy with sticky session lifetimes controlled by the lifetime of the browser (user-agent) or a specified expiration period
- **create_load_balancer**
  Creates a Classic Load Balancer
- **create_load_balancer_listeners**
  Creates one or more listeners for the specified load balancer
- **create_load_balancer_policy**
  Creates a policy with the specified attributes for the specified load balancer
- **delete_load_balancer**
  Deletes the specified load balancer
- **delete_load_balancer_listeners**
  Deletes the specified listeners from the specified load balancer
- **delete_load_balancer_policy**
  Deletes the specified policy from the specified load balancer
- **deregister_instances_from_load_balancer**
  Deregisters the specified instances from the specified load balancer
describe_account_limits
Describes the current Elastic Load Balancing resource limits for your AWS account

describe_instance_health
Describes the state of the specified instances with respect to the specified load balancer

describe_load_balancer_attributes
Describes the attributes for the specified load balancer

describe_load_balancer_policies
Describes the specified policies

describe_load_balancer_policy_types
Describes the specified load balancer policy types or all load balancer policy types

describe_load_balancers
Describes the specified the load balancers

describe_tags
Describes the tags associated with the specified load balancers

detach_load_balancer_from_subnets
Removes the specified subnets from the set of configured subnets for the load balancer

disable_availability_zones_for_load_balancer
Removes the specified Availability Zones from the set of Availability Zones for the specified load balancer

enable_availability_zones_for_load_balancer
Adds the specified Availability Zones to the set of Availability Zones for the load balancer

modify_load_balancer_attributes
Modifies the attributes of the specified load balancer

register_instances_with_load_balancer
Adds the specified instances to the specified load balancer

remove_tags
Removes one or more tags from the specified load balancer

set_load_balancer_listener_ssl_certificate
Sets the certificate that terminates the specified listener’s SSL connections

set_load_balancer_policies_for_backend_server
Replaces the set of policies associated with the specified port on which the EC2 instance is listening with a new set of policies

set_load_balancer_policies_of_listener
Replaces the current set of policies for the specified load balancer port with

Examples

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

---

elbv2

**Elastic Load Balancing**
Description

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

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

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

For more information, see the Elastic Load Balancing User Guide.

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

Usage

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

Arguments

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

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...),` where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- elbv2(
    config = list(
        credentials = list(
            creds = list(
                access_key_id = "string",
                secret_access_key = "string",
                session_token = "string"
            ),
            profile = "string"
        ),
```
elmv2

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

Operations

add_listener_certificates
add_tags
create_listener
create_load_balancer
create_rule
create_target_group
delete_listener
delete_load_balancer
delete_rule
delete_target_group
deregister_targets
describe_account_limits
describe_listener_certificates
describe_listeners
describe_load_balancer_attributes
describe_load_balancers
describe_rules
describe_ssl_policies
describe_tags
describe_target_group_attributes
describe_target_groups
describe_target_health
modify_listener
modify_load_balancer_attributes
modify_rule
modify_target_group
modify_target_group_attributes
register_targets
remove_listener_certificates
remove_tags
set_ip_address_type
set_rule_priorities
set_security_groups
set_subnets

Adds the specified SSL server certificate to the certificate list for the specified HTTPS or TLS listener
Adds the specified tags to the specified Elastic Load Balancing resource
Creates a listener for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
Creates an Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
Creates a rule for the specified listener
Creates a target group
Deletes the specified listener
Deletes the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
Deletes the specified rule
Deletes the specified target group
Deregisters the specified targets from the specified target group
Describes the default certificate and the certificate list for the specified HTTPS or TLS listener
Describes the specified listeners or the listeners for the specified Application Load Balancer Load Balancer
Describes the attributes for the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
Describes the specified load balancers or all of your load balancers
Describes the specified policies or all policies used for SSL negotiation
Describes the specified Elastic Load Balancing resources
Describes the attributes for the specified target group
Describes the specified target groups or all of your target groups
Describes the health of the specified targets or all of your targets
Replaces the specified properties of the specified listener
Modifies the specified attributes of the specified Application Load Balancer, Network Load Balancer, or Gateway Load Balancer
Replaces the specified properties of the specified rule
Modifies the health checks used when evaluating the health state of the targets in the specified target group
Modifies the specified attributes of the specified target group
Registers the specified targets with the specified target group
Removes the specified certificate from the certificate list for the specified HTTPS or TLS listener
Removes the specified tags from the specified Elastic Load Balancing resources
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 Application Load Balancer

Examples

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

---

**emr**

**Amazon Elastic MapReduce**

## Description

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

## Usage

```
emr(config = list())
```

## Arguments

**config**

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

## Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

## Service syntax

```
svc <- emr(
    config = list(
        credentials = list(
```
```python
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
- `create_studio`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `create_studio_session_mapping`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `delete_security_configuration`: Deletes a security configuration
- `delete_studio`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `delete_studio_session_mapping`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `describe_cluster`: Provides cluster-level details including status, hardware and software configuration, VPC settings, and so on
- `describe_job_flows`: This API is no longer supported and will eventually be removed
- `describe_notebook_execution`: Provides details of a notebook execution
- `describe_security_configuration`: Provides the details of a security configuration by returning the configuration JSON
- `describe_step`: Provides more detail about the cluster step
- `describe_studio`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `get_block_public_access_configuration`: Returns the Amazon EMR block public access configuration for your AWS account
- `get_managed_scaling_policy`: Fetches the attached managed scaling policy for an Amazon EMR cluster
- `get_studio_session_mapping`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `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 information for all active EC2 instances and EC2 instances terminated in the last 30 days
- `list_notebook_executions`: Provides summaries of all notebook executions
- `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
- `list_studios`: The Amazon EMR Studio APIs are in preview release for Amazon EMR and are subject to change
- `modify_cluster`: Modifies the number of steps that can be executed concurrently for the cluster specified using ClusterID
- `modify_instance_fleet`: Modifies the target On-Demand and target Spot capacities for the instance fleet with the specified InstanceFleetID within the cluster specified using ClusterID
- `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
**Examples**

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

## End(Not run)
```

---

**eventbridge**

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 an 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**

```r
eventbridge(config = list())
```
**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
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
- `cancel_replay`  
  Cancels the specified replay
- `create_archive`  
  Creates an archive of events with the specified settings
- `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`  
  You can use this operation to temporarily stop receiving events from the specified partner event source
- `delete_archive`  
  Deletes the specified archive
- `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_archive`  
  Retrieves details about an archive
- `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_replay`  
  Retrieves details about a replay
- `describe_rule`  
  Describes the specified rule
- `disable_rule`  
  Disables the specified rule
- `enable_rule`  
  Enables the specified rule
- `list_archives`  
  Lists your archives
- `list_event_buses`  
  Lists all the event buses in your account, including the default event bus, custom event buses.
list_event_sources
list_partner_event_source_accounts
list_partner_event_sources
list_replays
list_rule_names_by_target
list_rules
list_tags_for_resource
list_targets_by_rule
put_events
put_partner_events
put_permission
put_rule
put_targets
remove_permission
remove_targets
start_replay
tag_resource
test_event_pattern
untag_resource
update_archive

You can use this to see all the partner event sources that have been shared with your AWS account.
An SaaS partner can use this operation to display the AWS account ID that a particular partner event source is associated with.
An SaaS partner can use this operation to list all the partner event source names that they have created.
Lists your replays.
Lists the rules for the specified target.
Lists your Amazon EventBridge rules.
Displays the tags associated with an EventBridge resource.
Lists the targets associated with the specified rule.
Sends custom events to Amazon EventBridge so that they can be matched to rules.
This is used by SaaS partners to write events to a customer’s partner event bus.
Running PutPermission permits the specified AWS account or AWS organization to put events to the specified event bus.
Creates or updates the specified rule.
Adds the specified targets to the specified rule, or updates the targets if they are already associated with the rule.
Revoke the permission of another AWS account to be able to put events to the specified event bus.
Removes the specified targets from the specified rule.
Starts the specified replay.
Assigns one or more tags (key-value pairs) to the specified EventBridge resource.
Tests whether the specified event pattern matches the provided event.
Removes one or more tags from the specified EventBridge resource.
Updates the specified archive.

Examples

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

---

firehose  Amazon Kinesis 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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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` Writes multiple data records into a delivery stream in a single call, which can achieve high throughput
- `start_delivery_stream_encryption` Enables server-side encryption (SSE) for the delivery stream
- `stop_delivery_stream_encryption` Disables server-side encryption (SSE) for the delivery stream
- `tag_delivery_stream` Adds or updates tags for the specified delivery stream
- `untag_delivery_stream` Removes tags from the specified delivery stream
- `update_destination` Updates the specified destination of the specified delivery stream

Examples

```r
## Not run:
svc <- firehose()
svc$create_delivery_stream()
```
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. Some API actions require explicit resource permissions. For information, see the developer guide topic Firewall Manager required permissions for API actions.

Usage

fms(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Foo = 123
)

## End(Not run)
Operations

associate_admin_account  Sets the AWS Firewall Manager administrator account
delete_apps_list        Permanently deletes an AWS Firewall Manager applications list
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 AWS Firewall Manager SNS logs
delete_policy           Permanently deletes an AWS Firewall Manager policy
delete_protocols_list   Permanently deletes an AWS Firewall Manager protocols list
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 as the AWS Firewall Manager administrator
get_apps_list           Returns information about the specified AWS Firewall Manager applications list
get_compliance_detail   Returns detailed compliance information about the specified member account
get_notification_channel Returns 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
get_protocols_list      Returns information about the specified AWS Firewall Manager protocols list
get_violation_details   Retrieves violations for a resource based on the specified AWS Firewall Manager policy and AWS account
list_apps_lists         Returns an array of AppsListDataSummary objects
list_compliance_status  Returns an array of PolicyComplianceStatus objects
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
list_protocols_lists    Returns an array of ProtocolsListDataSummary objects
list_tags_for_resource  Retrieves the list of tags for the specified AWS resource
put_apps_list           Creates an AWS Firewall Manager applications list
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
put_protocols_list      Creates an AWS Firewall Manager protocols list
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)
```

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

```python
fsx(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
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

- `associate_file_system_aliases`: Use this action to associate one or more Domain Name Server (DNS) aliases with an existing Amazon FSx for Windows File Server file system.
- `cancel_data_repository_task`: Cancels an 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 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 backup.
- `delete_backup`: Deletes an Amazon FSx backup, deleting its contents.
- `delete_file_system`: Deletes a file system, deleting its contents.
- `describe_backups`: Returns the description of specific Amazon FSx backups, if a BackupIds value is provided.
- `describe_data_repository_tasks`: Returns the description of specific Amazon FSx for Lustre data repository tasks, if one or more TaskIds values are provided.
- `describe_file_system_aliases`: Returns the DNS aliases that are associated with the specified Amazon FSx for Windows File Server file system.
- `describe_file_systems`: Returns the description of specific Amazon FSx file systems, if a FileSystemIds value is provided.
- `disassociate_file_system_aliases`: Use this action to disassociate, or remove, one or more Domain Name Service (DNS) aliases from an Amazon FSx for Windows File Server file system.
- `list_tags_for_resource`: Lists tags for an Amazon FSx file systems and backups in the case of Amazon FSx for Windows File Server.
- `tag_resource`: Tags an Amazon FSx file system.
- `untag_resource`: This action removes a tag from an Amazon FSx file system.
- `update_file_system`: Use this operation to update the configuration of an existing Amazon FSx file system.
Examples

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

---

**glacier**  
*Amazon Glacier*

**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

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

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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` This operation aborts a multipart upload identified by the upload ID
- `abort_vault_lock` This operation aborts the vault locking process if the vault lock is not in the Locked state
- `add_tags_to_vault` This operation adds the specified tags to a vault
- `complete_multpart_upload` You call this operation to inform Amazon S3 Glacier (Glacier) that all the archive parts have been uploaded and that Glacier can then assemble the archive from the uploaded parts
- `complete_vault_lock` This operation completes the vault locking process by transitioning the vault lock from the InProgress state to the Locked state, which causes the vault lock policy to become unchangeable
- `create_vault` This operation creates a new vault with the specified name
- `delete_archive` This operation deletes an archive from a vault
- `delete_vault` This operation deletes a vault
- `delete_vault_access_policy` This operation deletes the access policy associated with the specified vault
- `delete_vault_notifications` This operation deletes the notification configuration set for a vault
- `describe_job` This operation returns information about a job you previously initiated, including the job initiation date, the job status code/message and the Amazon SNS topic to notify after Amazon S3 Glacier (Glacier) completes the job
- `describe_vault` This operation returns information about a vault, including the vault’s Amazon Resource Name (ARN), the date the vault was created, the number of archives it contains, and the total size of all the archives in the vault
- `get_data_retrieval_policy` This operation returns the current data retrieval policy for the account and region specified in the GET request
- `get_job_output` This operation downloads the output of the job you initiated using `InitiateJob`
- `get_vault_access_policy` This operation retrieves the access-policy subresource set on the vault; for more information, see Set Vault Access Policy (PUT access-policy)
- `get_vault_lock` This operation retrieves the following attributes from the lock-policy subresource set on the
get_vault_notifications
initiate_job
initiate_multipart_upload
initiate_vault_lock
list_jobs
list_multpart_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
set_vault_notifications
upload_archive
upload_multipart_part

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,
This operation initiates a multipart upload
This operation initiates the vault locking process by doing the following:
This operation lists jobs for a vault, including jobs that are in-progress and jobs that have recently finished.
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 lists all vaults owned by the calling user’s account
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
This operation configures notifications that will be sent when specific events happen to a vault
This operation adds an archive to a vault
This operation uploads a part of an archive

Examples

```r
## Not run:
svc <- glacier()
# The example deletes an in-progress multipart upload to a vault named
# my-vault:
svc$abort_multipart_upload(
  accountId = "-",
  uploadId = "19gaRezEXAMPLES6RYyqthHOC_kGRCT03L9ytr220UmPtBYKk-OssZtLq...",
  vaultName = "my-vault"
)
## End(Not run)
```

---

**Description**

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

AWS Global Accelerator is a service in which you create *accelerators* to improve the performance of your applications for local and global users. Depending on the type of accelerator you choose, you can gain additional benefits.
• By using a standard accelerator, you can improve availability of your internet applications that are used by a global audience. With a standard accelerator, Global Accelerator directs traffic to optimal endpoints over the AWS global network.

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

Global Accelerator is a global service that supports endpoints in multiple AWS Regions but you must specify the US West (Oregon) Region to create or update accelerators.

By default, Global Accelerator provides you with two static IP addresses that you associate with your accelerator. With a standard accelerator, instead of using the IP addresses that Global Accelerator provides, you can configure these entry points to be IPv4 addresses from your own IP address ranges that you bring to Global Accelerator. The static IP addresses are anycast from the AWS edge network. For a standard accelerator, they distribute incoming application traffic across multiple endpoint resources in multiple AWS Regions, which increases the availability of your applications. Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that are located in one AWS Region or multiple Regions. For custom routing accelerators, you map traffic that arrives to the static IP addresses to specific Amazon EC2 servers in endpoints that are virtual private cloud (VPC) subnets.

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

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

For a list of the AWS Regions where Global Accelerator and other services are currently supported, see the AWS Region Table.

AWS Global Accelerator includes the following components:

**Static IP addresses:**

Global Accelerator provides you with a set of two static IP addresses that are anycast from the AWS edge network. If you bring your own IP address range to AWS (BYOIP) to use with a standard accelerator, you can instead assign IP addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator.

The IP addresses serve as single fixed entry points for your clients. If you already have Elastic Load Balancing load balancers, Amazon EC2 instances, or Elastic IP address resources set up for your applications, you can easily add those to a standard accelerator in Global Accelerator. This allows Global Accelerator to use static IP addresses to access the resources.

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies like tag-based permissions with Global Accelerator to delete an accelerator. For more information, see Tag-based policies.
Accelerator:
An accelerator directs traffic to endpoints over the AWS global network to improve the performance of your internet applications. Each accelerator includes one or more listeners. There are two types of accelerators:

- A *standard* accelerator directs traffic to the optimal AWS endpoint based on several factors, including the user’s location, the health of the endpoint, and the endpoint weights that you configure. This improves the availability and performance of your applications. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses.

- A *custom routing* accelerator directs traffic to one of possibly thousands of Amazon EC2 instances running in a single or multiple virtual private clouds (VPCs). With custom routing, listener ports are mapped to statically associate port ranges with VPC subnets, which allows Global Accelerator to determine an EC2 instance IP address at the time of connection. By default, all port mapping destinations in a VPC subnet can’t receive traffic. You can choose to configure all destinations in the subnet to receive traffic, or to specify individual port mappings that can receive traffic.

For more information, see [Types of accelerators](#).

DNS name:
Global Accelerator assigns each accelerator a default Domain Name System (DNS) name, similar to `a1234567890abcdef.awsglobalaccelerator.com`, that points to the static IP addresses that Global Accelerator assigns to you or that you choose from your own IP address range. Depending on the use case, you can use your accelerator’s static IP addresses or DNS name to route traffic to your accelerator, or set up DNS records to route traffic using your own custom domain name.

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, by default, 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 port (or port range) and protocol (or protocols) that you configure. A listener can be configured for TCP, UDP, or both TCP and UDP protocols. 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. With a standard accelerator, 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. With a standard accelerator, 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 example, for new releases across different AWS Regions.
Endpoint:
An endpoint is a resource that Global Accelerator directs traffic to.
Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. An Application Load Balancer endpoint can be internet-facing or internal. Traffic for standard accelerators is routed to endpoints based on the health of the endpoint along with 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.
Endpoints for custom routing accelerators are virtual private cloud (VPC) subnets with one or many EC2 instances.

Usage
```
globalaccelerator(config = list())
```

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

Value
A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

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

Operations
- `add_custom_routing_endpoints` Associate a virtual private cloud (VPC) subnet endpoint with your custom routing accelerator
- `advertise_byoip_cidr` Advertises an IPv4 address range that is provisioned for use with your AWS resources through bring your own IP addresses (BYOIP)
- `allow_custom_routing_traffic` Specify the Amazon EC2 instance (destination) IP addresses and ports for a VPC subnet endpoint that can receive traffic for a custom routing accelerator
- `create_accelerator` Create an accelerator
create_custom_routing_accelerator
create_custom_routing_endpoint_group
create_custom_routing_listener
create_endpoint_group
create_listener
delete_accelerator
delete_custom_routing_accelerator
delete_custom_routing_endpoint_group
delete_custom_routing_listener
delete_endpoint_group
delete_listener
deny_custom_routing_traffic
deprovision_byoip_cidr
describe_accelerator
describe_accelerator_attributes
describe_custom_routing_accelerator
describe_custom_routing_accelerator_attributes
describe_custom_routing_endpoint_group
describe_custom_routing_listener
describe_endpoint_group
describe_listener
describe_accelerator_attributes
describe_custom_routing_accelerator
describe_custom_routing_accelerator_attributes
describe_custom_routing_endpoint_group
describe_custom_routing_listener
describe_endpoint_group
describe_listener
describe_accelerator_attributes
describe_custom_routing_accelerator
describe_custom_routing_accelerator_attributes
describe_custom_routing_endpoint_group
describe_custom_routing_listener
describe_endpoint_group
describe_listener
list_accelerators
list_byoip_cidrs
list_custom_routing_accelerators
list_custom_routing_endpoint_groups
list_custom_routing_listeners
list_custom_routing_port_mappings
list_custom_routing_port_mappings_by_destination
list_endpoint_groups
list_listeners
list_tags_for_resource
provision_byoip_cidr
remove_custom_routing_endpoints
tag_resource
untag_resource
update_accelerator
update_accelerator_attributes
update_custom_routing_accelerator
update_custom_routing_accelerator_attributes
update_custom_routing_listener
update_endpoint_group
update_listener
withdraw_byoip_cidr

Create a custom routing accelerator
Create an endpoint group for the specified listener for a custom routing accelerator
Create a listener to process inbound connections from clients to a custom routing accelerator
Create an endpoint group for the specified listener
Create a listener to process inbound connections from clients to an accelerator
Delete an accelerator
Delete a custom routing accelerator
Delete an endpoint group from a listener for a custom routing accelerator
Delete a listener for a custom routing accelerator
Delete an endpoint group from a listener
Delete a listener from an accelerator
Specify the Amazon EC2 instance (destination) IP addresses and ports
Release the specified address range that you provisioned to use with your AWS resources through bring your own IP addresses (BYOIP)
Describe an accelerator
Describe the attributes of an accelerator
Describe a custom routing accelerator
Describe the attributes of a custom routing accelerator
Describe an endpoint group for a custom routing accelerator
Describe an endpoint group
Describe a listener
List the accelerators for an AWS account
List the IP address ranges that were specified in calls to ProvisionByoipCidr
List the custom routing accelerators for an AWS account
List the endpoint groups that are associated with a listener for a custom routing accelerator
List the listeners for a custom routing accelerator
List the port mappings for a specific EC2 instance (destination) in a VPC subnet endpoint
List the port mappings for a specific EC2 instance (destination) in a VPC subnet endpoint
List the port mappings
List the listeners for a custom routing accelerator
List all tags for an accelerator
Provision an IP address range to use with your AWS resources through bring your own IP addresses (BYOIP)
Remove endpoints from a custom routing accelerator
Add tags to an accelerator resource
Remove tags from a Global Accelerator resource
Update an accelerator
Update the attributes for an accelerator
Update a custom routing accelerator
Update the attributes for a custom routing accelerator
Update a listener for a custom routing accelerator
Update an endpoint group
Update a listener
Stop advertising an address range that is provisioned as an address pool

Examples

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

## End(Not run)

---

**glue**

*AWS Glue*

**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.

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
svc <- glue(
  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
batch_delete_connection
batch_delete_partition
batch_delete_table
batch_delete_partition_version
batch_get_crawlers
batch_get_dev_endpoints
batch_get_jobs
batch_get_partition
batch_get_triggers
batch_get_workflows
batch_stop_job_run
batch_update_partition
cancel_ml_task_run
check_schema_version_validity
create_classifier
create_connection
create_crawler
create_database
create_dev_endpoint
create_job
create_ml_transform
create_partition
create_partition_index
create_registry
create_schema
create_script
create_security_configuration
create_table
create_trigger
create_user_defined_function
create_workflow
delete_classifier
delete_column_statistics_for_partition
delete_column_statistics_for_table
delete_connection
delete_crawler
delete_database
delete_dev_endpoint
delete_job
delete_ml_transform
delete_partition
delete_partition_index
delete_registry
delete_resource_policy
delete_schema
delete_schema_versions
delete_security_configuration

Creates one or more partitions in a batch operation
Deletes a list of connection definitions from the Data Catalog
Deletes one or more partitions in a batch operation
Deletes multiple tables at once
Deletes a specified batch of versions of a table
Returns a list of resource metadata for a given list of crawler names
Returns a list of resource metadata for a given list of development endpoint names
Returns a list of resource metadata for a given list of job names
Retrieves partitions in a batch request
Returns a list of resource metadata for a given list of trigger names
Returns a list of resource metadata for a given list of workflow names
Stops one or more job runs for a specified job definition
Updates one or more partitions in a batch operation
Cancels (stops) a task run
Validates the supplied schema
Creates a classifier in the user’s account
Creates a connection definition in the Data Catalog
Creates a new crawler with specified targets, role, configuration, and optional schedule
Creates a new database in a Data Catalog
Creates a new development endpoint
Creates a new job definition
Creates an AWS Glue machine learning transform
Creates a new partition
Creates a specified partition index in an existing table
Creates a new registry which may be used to hold a collection of schemas
Creates a new schema set and registers the schema definition
Transforms a directed acyclic graph (DAG) into code
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
Delete the partition column statistics of a column
Retrieves table statistics of columns
Deletes a connection from the Data Catalog
Removes a specified crawler from the AWS Glue Data Catalog, unless the crawler state is STOPPED
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 partition index from an existing table
Delete the entire registry including schema and all of its versions
Deletes a specified policy
Deletes the entire schema set, including the schema set and all of its versions
Remove versions from the specified schema
Deletes a specified security configuration
delete_table
delete_table_version
delete_trigger
delete_user_defined_function
delete_workflow
get_catalog_import_status
get_classifier
get_classifiers
get_column_statistics_for_partition
get_column_statistics_for_table
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_partition_indexes
get_partitions
get_plan
get_registry
get_resource_policies
get_resource_policy
get_schema
get_schema_by_definition
get_schema_version
get_schema_versions_diff
get_security_configuration
get_security_configurations
get_table
get_tables
get_table_version
get_table_versions

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 partition statistics of columns
Retrieves table statistics of columns
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 the partition indexes associated with a table
Retrieves information about the partitions in a table
Gets code to perform a specified mapping
Describes the specified registry in detail
Retrieves the security configurations for the resource policies set on individual resources
Retrieves a specified resource policy
Describes the specified schema in detail
Retrieves a schema by the SchemaDefinition
Get the specified schema by its unique ID assigned when a version of the schema is created or registered
Fetched the schema version difference in the specified difference type between two versions
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
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_ml_transforms
list_registries
list_schemas
list_schema_versions
list_triggers
list_workflows
put_data_catalog_encryption_settings
put_resource_policy
put_schema_version_metadata
put_workflow_run_properties
query_schema_version_metadata
register_schema_version
remove_schema_version_metadata
reset_job_bookmark
resume_workflow_run
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
stop_workflow_run
tag_resource
untag_resource
update_classifier
update_column_statistics_for_partition
update_column_statistics_for_table
update_connection

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 a sortable, filterable list of existing AWS Glue machine learning transforms
Returns a list of registries that you have created, with minimal registry information
Returns a list of schemas with minimal details
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 metadata key value pair for a specified schema version ID
Puts the specified workflow run properties for the given workflow run
Queries for the schema version metadata information
Adds a new version to the existing schema
Removes a key value pair from the schema version metadata for the specified schema version ID
Resets a bookmark entry
Retorts selected nodes of a previous partially completed workflow run and resubmits them
Searches a set of tables based on properties in the table metadata as well as on the database name
Starts a crawl using the specified crawler, regardless of what is scheduled
Changes the schedule state of the specified crawler to SCHEDULED, unless the state is currently NOT_SCHEDULED, but does not start the crawler if it is already running
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 transform
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 the transform
Starts an existing trigger
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 it if it is already running
Stops a specified trigger
Stops the execution of the specified workflow run
Adds tags to a resource
Removes tags from a resource
Modifies an existing classifier (a GrokClassifier, an XMLClassifier, a JsonClassifier, or a CsvClassifier)
Creates or updates partition statistics of columns
Creates or updates table statistics of columns
Updates a connection definition in the Data Catalog
**update_crawler** Updates a crawler
**update_crawler_schedule** Updates the schedule of a crawler using a cron expression
**update_database** Updates an existing database definition in a Data Catalog
**update_dev_endpoint** Updates a specified development endpoint
**update_job** Updates an existing job definition
**update_ml_transform** Updates an existing machine learning transform
**update_partition** Updates a partition
**update_registry** Updates an existing registry which is used to hold a collection of schemas
**update_schema** Updates the description, compatibility setting, or version checkpoint for a schema set
**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, 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 that serve malware or mine bitcoin.

GuardDuty also monitors AWS account access behavior for signs of compromise. Some examples of this are unauthorized infrastructure deployments such as EC2 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 the *Amazon GuardDuty User Guide*. 

---
Usage

guardduty(config = list())

Arguments

cfg 

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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 GuardDuty administrator account
- **archive_findings**: Archives GuardDuty findings that are 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, which is called a 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 export findings to
- **create_sample_findings**: Generates example findings of types specified by the list of finding types
- **create_threat_intel_set**: Creates a new ThreatIntelSet
- **decline_invitations**: Declines invitations sent to the current member account by AWS accounts specified by their account IDs
- **delete_detector**: Deletes an Amazon GuardDuty detector that is 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 administrator account)
- **delete_publishing_destination**: Deletes the publishing definition with the specified destinationId
delete_threat_intel_set
describe_organization_configuration
describe_publishing_destination
disable_organization_admin_account
disable_members
disable_member_detectors
disassociate_from_master_account
disassociate_threat_intel_set
get_detector
get_filter
guardduty
guardduty
get_findings
get_findings_statistics
get_findings_statistics
get_invitations_count
get_ip_set
get_master_account
guardduty
guardduty
guardduty
get_threat_intel_set
get_usage_statistics
get_ip_set
get_members
get_member_detectors
get_threat_intel_set
invitations
invite_members
driver
list_detectors
list_findings
list_invitations
list_ip_sets
list_members
list_threat_intel_sets
list_publishing_destinations
list_tags_for_resource
list_threat_intel_sets
list_threat_intel_sets
list_threat_intel_sets
start_monitoring_members
stop_monitoring_members
update_detector
update_filter
update_findings_feedback
update_filter
update_ip_set
update_member_detectors
update_organization_configuration
update_publishing_destination
update_threat_intel_set

Deliberate

**Examples**

```r
## Not run:
svc <- guardduty()
```
## AWS Health APIs and Notifications

### Description

**AWS Health**

The AWS Health API provides programmatic access to the AWS Health information that appears in the AWS Personal Health Dashboard. You can use the API operations to get information about AWS Health events that affect your AWS services and resources.

You must have a Business or Enterprise support plan from AWS Support to use the AWS Health API. If you call the AWS Health API from an AWS account that doesn’t have a Business or Enterprise support plan, you receive a `SubscriptionRequiredException` error.

AWS Health has a single endpoint: health.us-east-1.amazonaws.com (HTTPS). Use this endpoint to call the AWS Health API operations.

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

If your AWS account is part of AWS Organizations, you can use the AWS Health organizational view feature. This feature provides a centralized view of AWS Health events across all accounts in your organization. You can aggregate AWS Health events in real time to identify accounts in your organization that are affected by an operational event or get notified of security vulnerabilities. Use the organizational view API operations to enable this feature and return event information. For more information, see Aggregating AWS Health events in the AWS Health User Guide.

When you use the AWS Health API operations to return AWS Health events, see the following recommendations:

- Use the `eventScopeCode` parameter to specify whether to return AWS Health events that are public or account-specific.
- Use pagination to view all events from the response. For example, if you call the `describe_events_for_organization` operation to get all events in your organization, you might receive several page results. Specify the `nextToken` in the next request to return more results.

### Usage

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

### Arguments

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

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
cvc <- 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 specified 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
- `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 AWS Organizations
- `disable_health_service_access_for_organization` Disables AWS 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
## Not run:
svc <- health()
svc$describe_affected_accounts_for_organization(
  Foo = 123
)
```
**Description**

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS services. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users and applications can access. For more information about IAM, see AWS Identity and Access Management (IAM) and the AWS Identity and Access Management User Guide.

**Usage**

iam(config = list())

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

svc <- 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
add_role_to_instance_profile
add_user_to_group
attach_group_policy
attach_role_policy
attach_user_policy
change_password
create_access_key
create_account_alias
create_group
create_instance_profile
create_login_profile
create_open_id_connect_provider
create_policy
create_policy_version
create_role
create_saml_provider
create_service_linked_role
create_service_specific_credential
create_user
create_virtual_mfa_device
deactivate_mfa_device
delete_access_key
delete_account_alias
delete_account_password_policy
delete_group
delete_group_policy
delete_instance_profile
delete_login_profile
delete_open_id_connect_provider
delete_policy
delete_policy_version
delete_role
delete_role_permissions_boundary
delete_role_policy
delete_saml_provider
delete_server_certificate
delete_service_linked_role
delete_service_specific_credential
delete_signing_certificate
delete_ssh_public_key
delete_user
delete_user_permissions_boundary
delete_user_policy
delete_virtual_mfa_device
detach_group_policy

detaches the specified managed policy from the specified IAM group

deploys a new virtual MFA device for the AWS account

deletes the specified AWS MFA device and removes it from association with the user name for which it was originally enabled

deletes the specified server certificate

deletes a SAML provider resource in IAM

deletes the specified server certificate
 submits a service-linked role deletion request and returns a DeletionTaskId, which you can use to check the status of the deletion

deletes the specified IAM user

deletes the specified inline policy that is embedded in the specified IAM user

deletes the specified virtual MFA device

detaches the specified managed policy from the specified IAM group

detaches the specified IAM role from the specified instance profile

adds a new client ID (also known as audience) to the list of client IDs associated with the specified IAM OpenID Connect (OIDC) provider resource

adds the specified IAM role to the specified instance profile

creates a new instance profile

creates a new user for your AWS account

creates a new version of the specified managed policy

creates a new IAM resource that describes an identity provider (IdP) that supports SAML 2

creates a new IAM entity to describe an identity provider (IdP) that supports OpenID Connect (OIDC)

creates a new IAM user for your AWS account

creates a new virtual MFA device for the AWS account

creates an alias for your AWS account

creates a new managed policy for your AWS account

creates a new AWS secret access key and corresponding AWS access key ID for the specified user

creates an IAM entity to describe an identity provider (IdP) that supports OpenID Connect (OIDC)

creates an IAM resource that describes an identity provider (IdP) that supports SAML 2

creates an AWS account alias

creates an IAM user for your AWS account

creates an IAM entity to describe an identity provider (IdP) that supports OpenID Connect (OIDC)

creates an IAM entity to describe an identity provider (IdP) that supports SAML 2
detach_role_policy
detach_user_policy
enable_mfa_device
generate_credential_report
generate_organizations_access_report
generate_service_last_accessed_details
get_access_key_last_used
get_account_authorization_details
get_account_password_policy
get_account_summary
get_context_keys_for_custom_policy
get_context_keys_for_principal_policy
get_credential_report
group
get_group_policy
group
get_instance_profile
group
get_login_profile
group
get_open_id_connect_provider
group
get_organizations_access_report
get_policy
group
get_policy_version
get_role
group
get_role_policy
get_saml_provider
group
get_server_certificate
group
get_service_last_accessed_details
group
get_service_last_accessed_details_with_entities
get_service_linked_role_deletion_status
group
get_ssh_public_key
get_user
get_user_policy
list_access_keys
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

Removes the specified managed policy from the specified role
Removes the specified managed policy from the specified user
Enables the specified MFA device and associates it with the specified IAM user
Generates a credential report for the AWS account
Generates a report for service last accessed data for AWS Organizations
Generates a report that includes details about when an IAM resource (user, group, role, or policy) was last used
Retrieves information about the specified access key
Retrieves information about all IAM users, groups, roles, and policies in the AWS account
Retrieves the password policy for the AWS account
Retrieves information about IAM entity usage and IAM quotas in the AWS account
Retrieves a list of all of the context keys referenced in the input policies
Gets a list of all of the context keys referenced in all the IAM policies that are attached to the AWS account
Retrieves a list of IAM users that are in the specified IAM group
Retrieves the specified inline policy document that is embedded in the specified IAM user
Retrieves the specified inline policy document that is embedded in the specified IAM user
Retrieves information about the specified identity provider (OIDC) provider in IAM
Retrieves the service last accessed data report for AWS Organizations
Retrieves information about the specified version of the specified managed policy
Retrieves information about the specified version of the specified managed policy
Retrieves information about the specified version of the specified managed policy
Retrieves the specified inline policy document that is embedded in the specified IAM group
Retrieves the specified inline policy document that is embedded in the specified IAM group
Retrieves the specified inline policy document that is embedded in the specified IAM group
Retrieves information about when the specified access key was last used
Retrieves information about the specified OpenID Connect (OIDC) provider resource object in IAM
Retrieves the specified SSH public key, including metadata about the key
Retrieves information about the specified IAM user, including the user’s username, groups, and policies
Retrieves the specified inline policy document that is embedded in the specified IAM user
Retrieves information about the access key IDs associated with the specified IAM user
Lists the account alias associated with the AWS account (Note: you can use this to list all accounts)
Lists all managed policies that are attached to the specified IAM group
Lists 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 can be used to access
Lists the names of the inline policies that are embedded in the specified IAM 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 information about the IAM OpenID Connect (OIDC) provider resource object in IAM
Lists all managed policies that are available in your AWS account, including your own customer-defined managed policies and all AWS managed policies
Retrieves a list of policies that the IAM identity (user, group, or role) can access
Lists information about the versions of the specified managed policy
Lists the names of the inline policies that are embedded in the specified IAM group

After you generate a group or policy report using the GenerateServiceLastAccessedDetails operation, you can use the JobId parameter in GetServiceLastAccessedDetailsWithEntities to retrieve the details about when an IAM resource (user, group, role, or policy) was last used. For example, you can generate a report that includes details about when an IAM user was last used in an attempt to access AWS services and then use the JobId parameter in GetServiceLastAccessedDetailsWithEntities to retrieve the details about when the user was last used. This can be useful for identifying potential security risks or for monitoring the usage of IAM resources in your account.
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
resync_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
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

Lists the IAM roles that have the specified path prefix
Lists the tags 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 user
Returns information about the signing certificates associated with the specified 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 tags 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 IAM role
Adds or updates the policy that is specified as the IAM role’s permissions boundary
Adds or updates an inline policy document that is embedded in the specified IAM user
Adds or updates the policy that is specified as the IAM user’s permissions boundary
Removes the specified client ID (also known as audience) from the list of registered client IDs
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 (operative) 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
Updates the name and/or the path of the specified IAM group
Changes the password for the specified IAM user
Replaces the existing list of server certificate thumbprints associated with the specified role
Updates the description or maximum session duration setting of a role
Use UpdateRole instead
Updates the metadata document for an existing SAML provider resource object
Updates the name and/or the path of the specified server certificate store
Sets the status of a service-specific credential to Active or Inactive
Changes the status of the specified user signing certificate from active to inactive
Sets the status of an IAM user’s SSH public key to active or inactive
Updates the name and/or the path of the specified IAM user
Uploads a server certificate entity for the AWS account
Uploads an X
Uploads an SSH public key and associates it with the specified IAM user
Examples

## Not run:
svc <- iam()

# 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:
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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- inspector(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    ...))
```python
    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 template
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
generate_cross_account_access_role  Produces an assessment report that includes detailed and comprehensive results of a specified assessment run
get_assessment_report  Retrieves the exclusions preview (a list of ExclusionPreview objects) specified by the preview token
get_assessment_run  Information about the data that is collected for the specified assessment run
get_event_subscriptions  Lists the agents of the assessment runs that are specified by the ARNs of the assessment runs
get_event_subscriptions  Lists the assessment runs that correspond to the assessment templates that are specified by the ARNs of the assessment templates
get_exclusions  Lists the ARNs of the assessment targets within this AWS account
generate_cross_account_access_role  Lists the event subscriptions that correspond to the assessment targets that are specified by the ARNs of the assessment targets
generate_cross_account_access_role  Lists all the event subscriptions for the assessment template that is specified by the ARN of the assessment template
generate_cross_account_access_role  Lists findings that are generated by the assessment runs that are specified by the ARNs of the assessment runs
generate_cross_account_access_role  Lists all available Amazon Inspector rules packages
generate_cross_account_access_role  Lists all tags associated with an assessment template
get_assessment_run_agents  Previews the agents installed on the EC2 instances that are part of the specified assessment run
get_assessment_run_agents  Registers the IAM role that grants Amazon Inspector access to AWS Services needed to perform security assessments
get_assessment_run_agents  Removes entire attributes (key and value pairs) from the findings that are specified by the ARNs of the assessment runs
get_assessment_run_agents  Sets tags (key and value pairs) to the assessment template that is specified by the ARN of the assessment template
get_assessment_run_agents  Starts the assessment run specified by the ARN of the assessment template
generate_cross_account_access_role  Stops the assessment run that is specified by the ARN of the assessment run
generate_cross_account_access_role  Enables the process of sending Amazon Simple Notification Service (SNS) notifications
generate_cross_account_access_role  Disables the process of sending Amazon Simple Notification Service (SNS) notifications
generate_cross_account_access_role  Updates the assessment target that is specified by the ARN of the assessment target
```
## Examples

```r
## Not run:
svc <- inspector()
# Assigns attributes (key and value pairs) to the findings that are
# specified by the ARNs of the findings.
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-..."
  )
)
## End(Not run)
```

---

**kafka**

*Managed Streaming for Kafka*

### Description

The operations for managing an Amazon MSK cluster.

### Usage

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

### Arguments

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```r
svc <- kafka(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        ...),
        region = "us-west-2"
      ),
      endpoint = "https://kafka.example.com"
    )
  )
```

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

Operations

batch_associate_scram_secret Associates one or more Scram Secrets with an Amazon MSK cluster
batch_disassociate_scram_secret Disassociates one or more Scram Secrets from an Amazon MSK cluster
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
delete_configuration Deletes an MSK Configuration
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
get_compatible_kafka_versions Gets the Apache Kafka versions to which you can update the MSK cluster
group_cluster_operations Returns a list of all the operations that have been performed on the specified MSK cluster
group_clusters Returns a list of all the MSK clusters in the current Region
group_configuration_revisions Returns a list of all the MSK configurations in this Region
group_configurations Returns a list of all the MSK configurations in this Region
group_kafka_versions Returns a list of Kafka versions
group_nodes Returns a list of the broker nodes in the cluster
group_scram_secrets Returns a list of the Scram Secrets associated with an Amazon MSK cluster
group_tags_for_resource Returns a list of the tags associated with the specified resource
reboot_broker Reboots brokers
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_cluster_kafka_version Updates the Apache Kafka version for the cluster
update_configuration Updates an MSK configuration
update_monitoring Updates the monitoring settings for the cluster

Examples

## Not run:
svc <- kafka()
**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**

```r
kinesis(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
cvc <- 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 accessible after they are added to the stream
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 list
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 provides information about each consumer
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 a 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)
```

---

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.

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
csvc <- 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`
**Examples**

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

---

**Amazon Kinesis Analytics**

**Description**

Amazon Kinesis Data Analytics is a fully managed service that you can use to process and analyze streaming data using Java, SQL, or Scala. The service enables you to quickly author and run Java, SQL, or Scala code against streaming sources to perform time series analytics, feed real-time dashboards, and create real-time metrics.

**Usage**

```r
kinesisanalyticsv2(config = list())
```

**Arguments**

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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 Kinesis Data Analytics application
- `add_application_input_processing_configuration`: Adds an InputProcessingConfiguration to a SQL-based Kinesis Data Analytics application
- `add_application_output`: Adds an external destination to your SQL-based Kinesis Data Analytics application
- `add_application_reference_data_source`: Adds a reference data source to an existing SQL-based Kinesis Data Analytics application
- `add_application_vpc_configuration`: Adds a Virtual Private Cloud (VPC) configuration to the application
- `create_application`: Creates a Kinesis Data Analytics application
- `create_application_presigned_url`: Creates and returns a URL that you can use to connect to an application’s extension
- `create_application_snapshot`: Creates a snapshot of the application’s state data
- `delete_application`: Deletes the specified application
- `delete_application_cloud_watch_logging_option`: Deletes an Amazon CloudWatch log stream from a Kinesis Data Analytics application
- `delete_application_input_processing_configuration`: Deletes an InputProcessingConfiguration from an input
- `delete_application_output`: Deletes the output destination configuration from your SQL-based Kinesis Data Analytics application
- `delete_application_reference_data_source`: Deletes a reference data source configuration from the specified SQL-based Kinesis Data Analytics application
- `delete_application_vpc_configuration`: Deletes a VPC configuration from a Kinesis Data Analytics application
- `describe_application`: Returns information about a specific Kinesis Data Analytics application
- `describe_application_snapshot`: Returns information about a snapshot of application state data
- `discover_input_schema`: Infers a schema for a SQL-based Kinesis Data Analytics application by evaluating sample records on the specified streaming source (Kinesis data stream or Kinesis Data Firehose delivery stream) or Amazon S3 object
- `list_applications`: Lists information about the current application snapshots
- `list_application_snapshots`: Lists information about the current application snapshots
- `list_tags_for_resource`: Retrieves the list of key-value tags assigned to the application
- `start_application`: Starts the specified Kinesis Data Analytics application
- `stop_application`: Stops the application from processing data
tag_resource  
untag_resource  
update_application

Adds one or more key-value tags to a Kinesis Data Analytics application
Removes one or more tags from a Kinesis Data Analytics application
Updates an existing Kinesis Data Analytics application

Examples

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

---

**kms**

**AWS Key Management Service**

---

**Description**

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](https://docs.aws.amazon.com/kms/latest/developerguide/).

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 the [Tools for Amazon Web Services](https://aws.amazon.com/tools/).

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`
- `generate_data_key`
- `generate_data_key_without_plaintext`

Usage

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

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

cancel_key_deletion
connect_custom_key_store
create_alias
create_custom_key_store
create_grant
create_key
decrypt
delete_alias
delete_custom_key_store
delete_imported_key_material
describe_custom_key_stores
describe_key
disable_key
disable_key_rotation
disconnect_custom_key_store
disable
encrypt
generate_data_key
generate_data_key_pair
generate_data_key_pair_without_plaintext
generate_data_key_without_plaintext
generate_random
get_key_policy
get_key_rotation_status
get_parameters_for_import
get_public_key
import_key_material
list_aliases
list_grants
list_key_policies
list_keys
list_resource_tags
list_retirable_grants
put_key_policy
re_encrypt
revoke_grant

Cancel the deletion of a customer master key (CMK)
Connects or reconnects a custom key store to its associated AWS CloudHSM cluster
Creates a friendly name for a customer master key (CMK)
Creates a custom key store that is associated with an AWS CloudHSM cluster that you own and manage
Adds a grant to a customer master key (CMK)
Creates a unique customer managed customer master key (CMK) in your AWS account
Decrypts ciphertext that was encrypted by a AWS KMS customer master key (CMK)
Deletes the specified alias
Deletes a custom key store
Deletes key material that you previously imported
Gets information about custom key stores in the account and region
Provides detailed information about a customer master key (CMK)
Sets the state of a customer master key (CMK) to disabled
Disables automatic rotation of the key material for the specified symmetric customer key
Disconnects the custom key store from its associated AWS CloudHSM cluster
Sets the key state of a customer master key (CMK) to enabled
Enables automatic rotation of the key material for the specified symmetric customer key
Encrypts plaintext into ciphertext by using a customer master key (CMK)
Generates a unique symmetric data key for client-side encryption
Generates a unique asymmetric data key pair
Generates a unique asymmetric data key pair
Returns a random byte string that is cryptographically secure
Gets a key policy attached to the specified customer master key (CMK)
Gets a Boolean value that indicates whether automatic rotation of the key material is enabled
Returns the items you need to import key material into a symmetric, customer managed AWS KMS customer master key
Imports key material into an existing symmetric AWS KMS customer master key
Gets a list of aliases in the caller’s AWS account and region
Gets a list of all grants for the specified customer master key (CMK)
Gets the names of the key policies that are attached to a customer master key (CMK)
Gets a list of all customer master keys (CMKs) in the caller’s AWS account and Region
Returns all tags on the specified customer master key (CMK)
Returns all grants in which the specified principal is the RetiringPrincipal in the grant
Attaches a key policy to the specified customer master key (CMK)
Decrypts ciphertext and then reencrypts it entirely within AWS KMS
Retires a grant
Re revoked the specified grant for the specified customer master key (CMK)
Example

## Not run:

```r
svc <- kms()

# The following example cancels deletion of the specified CMK.
svc$cancel_key_deletion(
  KeyId = "1234abcd-12ab-34cd-56ef-1234567890ab"
)

## End(Not run)
```

---

lambda  

AWS Lambda

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.

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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_code_signing_config**: Creates a code signing configuration.
- **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_code_signing_config**: Deletes the code signing configuration.
- **delete_event_source_mapping**: Deletes an event source mapping.
- **delete_function**: Deletes a Lambda function.
- **delete_function_code_signing_config**: Removes the code signing configuration from the function.
- **delete_function_concurrency**: Deletes 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_code_signing_config**: Returns information about the specified code signing configuration.
- **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_code_signing_config**: Returns the code signing configuration for the specified function.
- **get_function_concurrency**: Returns details about the reserved concurrency configuration for a function.
- **get_function_event_invoke_config**: Returns 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 layer archive.
- **get_layer_version_by_arm**: Returns information about a version of an AWS Lambda layer, with a link to download the layer archive.
- **get_layer_version_policy**: Returns the permission policy for a version of an AWS Lambda layer.
- **get_policy**: Returns the resource-based IAM policy for a function, version, or alias.
- **get_provisioned_concurrency_config**: Retrieves the provisioned concurrency configuration for a function’s alias or version.
- **invoke**: Invokes a Lambda function.
invoke_async
list_aliases
list_code_signing_configs
list_event_source_mappings
list_function_event_invoke_configs
list_functions
list_functions_by_code_signing_config
list_layers
list_layer_versions
list_provisioned_concurrency_configs
list_tags
list_versions_by_function
publish_layer_version
publish_version
put_function_code_signing_config
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_code_signing_config
update_event_source_mapping
update_function_code
update_function_configuration
update_function_event_invoke_config

For asynchronous function invocation, use Invoke
Returns a list of aliases for a Lambda function
Returns a list of code signing configurations
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
List the functions that use the specified code signing configuration
Lists AWS Lambda layers and shows information about the latest version of each
Lists the versions of an AWS Lambda layer
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
Update the code signing configuration for the function
Sets the maximum number of simultaneous executions for a function, and reserves capacity for that concurrency level
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 layer
Revoke 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
Update the code signing configuration
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
## Not run:
svc <- lambda()

# The following example grants permission for the account 223456789012 to
# use version 1 of a layer named my-layer.
svc$sadd_layer_version_permission(
  Action = "lambda:GetLayerVersion",
  LayerName = "my-layer",
  Principal = "223456789012",
  StatementId = "xaccount",
  VersionNumber = 1L
)

## End(Not run)
```
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
config Optional configuration of credentials, endpoint, and/or region.

Value
A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax
svc <- 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
Delete 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_bot_aliases
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

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

get_bots
Returns bot information as follows:

get_bot_versions
Gets information about all of the versions of a bot

get_builtin_intent
Returns 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 a 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:

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:

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

put_bot
Creates an alias for the specified version of the bot or replaces an alias for the specified bot

prompt
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

tag_resource
Adds the specified tags to the specified resource

untag_resource
Removes tags from a bot, bot alias or bot channel

Examples

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

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

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
licensemanager

## 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)
```

licensemanager

AWS License Manager

Description

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.

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- 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

accept_grant
check_in_license
checkout_borrow_license
checkout_license
create_grant
create_grant_version
create_license
create_license_configuration
create_license_version
create_token
delete_grant
delete_license
delete_license_configuration
delete_token
extend_license_consumption
get_access_token
get_grant
get_license
get_license_configuration
get_license_usage
get_service_settings
list_associations_for_license_configuration
list_distributed_grants
list_failures_for_license_configuration_operations
list_license_configurations
list_licenses
list_license_specifications_for_resource
list_license_versions
list_received_grants
list_received_licenses
list_resource_inventory
list_tags_for_resource
list_tokens
list_usage_for_license_configuration
reject_grant
tag_resource

Accepts the specified grant
Checks in the specified license
Checks out the specified license for offline use
Checks out the specified license
Creates a grant for the specified license
Creates a new version of the specified grant
Creates a license
Creates a license configuration
Creates a new version of the specified license
Creates a long-lived token
Deletes the specified grant
Deletes the specified license
Deletes the specified license configuration
Deletes the specified token
Extends the expiration date for license consumption
Gets a temporary access token to use with AssumeRoleWithWebIdentity
Gets detailed information about the specified grant
Gets detailed information about the specified license
Gets detailed information about the specified license configuration
Gets detailed information about the usage of the specified license
Gets the License Manager settings for the current Region
Lists the resource associations for the specified license configuration
Lists the grants distributed for the specified license
Lists the license configuration operations that failed
Lists the license configurations for your account
Lists the licenses for your account
Describes the license configurations for the specified resource
Lists all versions of the specified license
Lists grants that are received but not accepted
Lists received licenses
Lists resources managed using Systems Manager inventory
Lists the tags for the specified license configuration
Lists your tokens
Lists all license usage records for a license configuration, displaying license consumption details by resource at a selected point in time
Rejects the specified grant
Adds the specified tags to the specified license configuration
untag_resource
update_license_configuration
update_license_specifications_for_resource
update_service_settings

Removes the specified tags from the specified license configuration
Modifies the attributes of an existing license configuration
Adds or removes the specified license configurations for the specified AWS resource
Updates License Manager settings for the current Region

Examples

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

## End(Not run)
```

---

**lightsail**

*Amazon Lightsail*

**Description**

Amazon Lightsail is the easiest way to get started with Amazon Web Services (AWS) for developers who need to build websites or web applications. It includes everything you need to launch your project quickly - instances (virtual private servers), container services, managed databases, SSD-based block storage, static IP addresses, load balancers, content delivery network (CDN) distributions, DNS management of registered domains, and resource snapshots (backups) - for a low, predictable monthly price.

You can manage your Lightsail resources using the Lightsail console, Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information about Lightsail concepts and tasks, see the Lightsail Dev Guide.

This API Reference provides detailed information about the actions, data types, parameters, and errors of the Lightsail service. For more information about the supported AWS Regions, endpoints, and service quotas of the Lightsail service, see Amazon Lightsail Endpoints and Quotas in the AWS General Reference.

**Usage**

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

**Arguments**

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

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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_certificate_to_distribution
attach_disk
attach_instances_to_load_balancer
attach_load_balancer_tls_certificate
attach_static_ip
close_instance_public_ports
copy_snapshot
create_certificate
create_cloud_formation_stack
create_contact_method
create_container_service
create_container_service_deployment
create_container_service_registry_login
create_disk
create_disk_from_snapshot
create_disk_snapshot
create_distribution
create_domain
create_domain_entry
create_instances
create_instances_from_snapshot
create_instance_snapshot
create_key_pair

Allocates a static IP address
Attaches an SSL/TLS certificate to your Amazon Lightsail content delivery network
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 ports for a specific Amazon Lightsail instance
Copies a manual snapshot of an instance or disk as another manual snapshot, or copies an automatic snapshot of an instance or disk
Creates an AWS CloudFormation stack, which creates a new Amazon EC2 instance
Creates an email or SMS text message contact method
Creates an Amazon Lightsail container service
Creates a deployment for your Amazon Lightsail container service
Creates a temporary set of log in credentials that you can use to log in to the Lightsail instance
Creates a block storage disk that can be attached to an Amazon Lightsail instance
Creates a block storage disk from a manual or automatic snapshot of a disk
Creates a snapshot of a block storage disk
Creates an Amazon Lightsail content delivery network (CDN) distribution
Creates a domain resource for the specified domain (e.g., example.com)
Creates one of the following domain name system (DNS) records in a domain (e.g., A, CNAME, MX, NS, SOA, SRV, TXT)
Creates one or more Amazon Lightsail instances
Creates one or more new instances from a manual or automatic snapshot of an instance
Creates a snapshot of a specific virtual private server, or instance
Creates an SSH key pair
create_load_balancer
create_load_balancer_tls_certificate
create_relational_database
create_relational_database_from_snapshot
create_relational_database_snapshot
delete_alarm
delete_auto_snapshot
delete_certificate
delete_contact_method
delete_container_image
delete_container_service
delete_disk
delete_disk_snapshot
delete_distribution
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_certificate_from_distribution
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_alarms
get_auto_snapshots
get_blueprints
get_bundles
get_certificates
get_cloud_formation_stack_records
get_contact_methods
get_container_api_metadata
get_container_images
get_container_log
get_container_service_deployments
get_container_service_metric_data
get_container_service_powers
get_container_services
get_disk

create_load_balancer
Create a Lightsail load balancer
create_load_balancer_tls_certificate
Create a Lightsail load balancer TLS certificate
create_relational_database
Create a new database in Amazon Lightsail
create_relational_database_from_snapshot
Create a new database from an existing database snapshot in Amazon Lightsail
create_relational_database_snapshot
Create a snapshot of your database in Amazon Lightsail
delete_alarm
Delete an alarm
delete_auto_snapshot
Deletes an automatic snapshot of an instance or disk
delete_certificate
Deletes an SSL/TLS certificate for your Amazon Lightsail content delivery
delete_contact_method
Deletes a contact method
delete_container_image
Deletes a container image that is registered to your Amazon Lightsail container service
delete_container_service
Deletes your Amazon Lightsail container service
delete_disk
Deletes the specified block storage disk
delete_disk_snapshot
Deletes the specified disk snapshot
delete_distribution
Deletes your Amazon Lightsail content delivery network (CDN) distribution
delete_domain
Deletes the specified domain recordset and all of its domain records
delete_domain_entry
Deletes a specific domain entry
delete_instance
Deletes an Amazon Lightsail instance
delete_instance_snapshot
Deletes a specific snapshot of a virtual private server (or instance)
delete_key_pair
Deletes a specific SSH key pair
delete_known_host_keys
Deletes the known host key or certificate used by the Amazon Lightsail browser-based SSH or RDP clients to authenticate an instance
delete_load_balancer
Deletes a Lightsail load balancer and all its associated SSL/TLS certificates
delete_load_balancer_tls_certificate
Deletes an SSL/TLS certificate associated with a Lightsail load balancer
delete_relational_database
Deletes a database in Amazon Lightsail
delete_relational_database_snapshot
Deletes a database snapshot in Amazon Lightsail
detach_certificate_from_distribution
Detaches an SSL/TLS certificate from your Amazon Lightsail content delivery
detach_disk
Detaches a stopped block storage disk from a Lightsail instance
detach Instances_from_load_balancer
Detaches the specified instances from a Lightsail load balancer
detach_static_ip
Detaches a static IP from the Amazon Lightsail instance to which it is attached
disable_add_on
Disables an add-on for an Amazon Lightsail resource
download_default_key_pair
Downloads the default SSH key pair from the user’s account
enable_add_on
Enables or modifies an add-on for an Amazon Lightsail resource
export_snapshot
Exports an Amazon Lightsail instance or block storage disk snapshot to Amazon Elastic Block Store
get_active_names
Returns the names of all active (not deleted) resources
get_alarms
Returns information about the configured alarms
get_auto_snapshots
Returns the available automatic snapshots for an instance or disk
get_blueprints
Returns the list of available instance images, or blueprints
get_bundles
Returns the list of bundles that are available for purchase
get_certificates
Returns information about one or more Amazon Lightsail SSL/TLS certificates
get_cloud_formation_stack_records
Returns the CloudFormation stack record created as a result of the create cloud formation stack operation
get_contact_methods
Returns information about the configured contact methods
get_container_api_metadata
Returns information about Amazon Lightsail containers, such as the current version of the Lightsail Control (lightsailctl) plugin
get_container_images
Returns the container images that are registered to your Amazon Lightsail container service
get_container_log
Returns the log events of a container of your Amazon Lightsail container service
get_container_service_deployments
Returns the deployments for your Amazon Lightsail container service
get_container_service_metric_data
Returns the data points of a specific metric of your Amazon Lightsail container service
get_container_service_powers
Returns the list of powers that can be specified for your Amazon Lightsail container service
get_container_services
Returns information about one or more of your Amazon Lightsail container services
get_disk
Returns information about a specific block storage disk
get_disks
get_disk_snapshots
get_disk_snapshots
get_distribution_bundles
get_distribution_latest_cache_reset
get_distribution_metric_data
get_distributions
get_domain
get_domains
get_export_snapshot_records
get_instance
get_instance_access_details
get_instance_port_states
get_instances
get_instance_snapshot
get_instance_snapshots
get_instance_state
get_key_pair
get_key_pairs
get_load_balancer
get_load_balancer_metric_data
get_load_balancers
get_load_balancer_tls_certificates
get_operation
get_operations
get_operations_for_resource
get_regions
get_reational_database
get_reational_database_blueprints
get_reational_database_bundles
get_reational_database_events
get_reational_database_log_events
get_reational_database_log_streams
get_reational_database_master_user_password
get_relationbal_database_metric_data
get_reational_databases
get_reational_database_snapshot
get_reational_database_snapshots
get_static_ip
get_static_ips
import_key_pair
is_vpc_peered
open_instance_public_ports
peer_vpc
put_alarm
put_instance_public_ports

Returns information about all block storage disks in your AWS account and region.
Returns information about a specific block storage disk snapshot.
Returns information about all block storage disk snapshots in your AWS account and region.
Returns the list bundles that can be applied to your Amazon Lightsail content delivery network (CDN) distributions.
Returns the timestamp and status of the last cache reset of a specific Amazon Lightsail content delivery network (CDN) distribution.
Returns the data points of a specific metric for an Amazon Lightsail content delivery network (CDN) distribution.
Returns information about one or more of your Amazon Lightsail content delivery network (CDN) distributions.
Returns a list of all domains in the user’s account.
Returns the export snapshot record created as a result of the export snapshot operation.
Returns information about a specific Amazon Lightsail instance, which is a virtual private server.
Returns temporary SSH keys you can use to connect to a specific virtual private server.
Returns the data points for the specified Amazon Lightsail instance metric, given an instance ID.
Returns the firewall port states for a specific Amazon Lightsail instance, the IP addresses allowed to connect to the instance through the ports, and the protocol.
Returns information about all Amazon Lightsail virtual private servers, or instances.
Returns information about a specific instance snapshot.
Returns all instance snapshots for the user’s account.
Returns the state of a specific instance.
Returns information about a specific key pair.
Returns information about all key pairs in the user’s account.
Returns information about the specified Lightsail load balancer.
Returns information about health metrics for your Lightsail load balancer.
Returns information about all load balancers in an account.
Returns information about the TLS certificates that are associated with the specified Lightsail load balancer.
Returns information about a specific operation.
Returns information about all operations.
Gets operations for a specific resource (e.g., an Amazon Lightsail instance or load balancer).
Returns a list of all valid regions for Amazon Lightsail.
Returns information about a specific database in Amazon Lightsail.
Returns a list of available database blueprints in Amazon Lightsail.
Returns the list of bundles that are available in Amazon Lightsail.
Returns a list of events for a specific database in Amazon Lightsail.
Returns a list of log events for a database in Amazon Lightsail.
Returns a list of available log streams for a specific database in Amazon Lightsail.
Returns the current, previous, or pending versions of the master user password.
Returns the data points of the specified metric for a database in Amazon Lightsail.
Returns all of the runtime parameters offered by the underlying database software.
Returns information about all of your databases in Amazon Lightsail.
Returns information about a specific database snapshot in Amazon Lightsail.
Returns information about all of your database snapshots in Amazon Lightsail.
Returns information about a specific static IP.
Returns information about all static IPs in the user’s account.
Imports a public SSH key from a specific key pair.
Returns a Boolean value indicating whether your Lightsail VPC is peered.
Opens ports for a specific Amazon Lightsail instance, and specifies the IP addresses allowed to connect to the instance through the ports.
Tries to peer the Lightsail VPC with the user’s default VPC.
Creates or updates an alarm, and associates it with the specified metric.
Opens ports for a specific Amazon Lightsail instance, and specifies the IP addresses allowed to connect to the instance through the ports.
Examples

```r
## Not run:
svc <- lightsail()
svc$allocate_static_ip(
  Foo = 123
)

## End(Not run)
```

### Description

Definition of the public APIs exposed by Amazon Machine Learning

### Usage

```r
machinelearning(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>
Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

---

## Description

Amazon Macie Classic

Amazon Macie Classic is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Macie Classic 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 Amazon Macie Classic User Guide.

A new Amazon Macie is now available with significant design improvements and additional features, at a lower price and in most AWS Regions. We encourage you to explore and use the new and improved features, and benefit from the reduced cost. To learn about features and pricing for the new Amazon Macie, see Amazon Macie.

## Usage

```r
macie(config = list())
```

## Arguments

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

## Value

A client for the service. You can call the service’s operations using syntax like `svc\$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
csvc <- 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 Classic as a member account
- **associate_s3_resources** Associates specified S3 resources with Amazon Macie Classic for monitoring and data classification
- **disassociate_member_account** Removes the specified member account from Amazon Macie Classic
- **disassociate_s3_resources** Removes specified S3 resources from being monitored by Amazon Macie Classic
- **list_member_accounts** Lists all Amazon Macie Classic member accounts for the current Amazon Macie Classic master account
- **list_s3_resources** Lists all the S3 resources associated with Amazon Macie Classic
- **update_s3_resources** Updates the classification types for the specified S3 resources

Examples

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

Description

Provides AWS Marketplace business intelligence data on-demand.
Usage

marketplacecommerceanalytics(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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

Examples

## 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

marketplaceentitlementservice(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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

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

## End(Not run)
```

**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. For information on the permissions you need to use this API, see AWS Marketing metering and entitlement API permissions in the AWS Marketplace Seller Guide.

**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

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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

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

## End(Not run)
AmazonMQ

Description

Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ 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

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

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
list_configurations: Returns a list of all configurations
list_tags: Lists tags for a resource
list_users: Returns a list of all ActiveMQ users
reboot_broker: Reboots a broker
update_broker: Adds a pending configuration change to a broker
update_configuration: Updates the specified configuration
update_user: Updates the information for an ActiveMQ user

Examples

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

mturk

Amazon Mechanical Turk

Description

Amazon Mechanical Turk API Reference

Usage

mturk(config = list())

Arguments

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

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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`
Examples

```r
## Not run:
svc <- mturk()
svc$accept_qualification_request(  
  Foo = 123  
)

## End(Not run)
```

Description

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

```python
neptune(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add_role_to_db_cluster</code></td>
<td>Associates an Identity and Access Management (IAM) role from an Neptune DB</td>
</tr>
<tr>
<td><code>add_source_identifier_to_subscription</code></td>
<td>Adds a source identifier to an existing event notification subscription</td>
</tr>
<tr>
<td><code>apply_pending_maintenance_action</code></td>
<td>Applies a pending maintenance action to a resource (for example, to a DB instance)</td>
</tr>
<tr>
<td><code>copy_db_cluster_parameter_group</code></td>
<td>Copies the specified DB cluster parameter group</td>
</tr>
<tr>
<td><code>copy_db_cluster_snapshot</code></td>
<td>Copies a snapshot of a DB cluster</td>
</tr>
<tr>
<td><code>copy_db_cluster</code></td>
<td>Creates a new Amazon Neptune DB cluster</td>
</tr>
<tr>
<td><code>create_db_cluster</code></td>
<td>Creates a custom endpoint and associates it with an Amazon Neptune DB cluster</td>
</tr>
<tr>
<td><code>create_db_cluster_parameter_group</code></td>
<td>Creates a new DB cluster parameter group</td>
</tr>
<tr>
<td><code>create_db_cluster_snapshot</code></td>
<td>Creates a snapshot of a DB cluster</td>
</tr>
<tr>
<td><code>create_db_instance</code></td>
<td>Creates a new DB instance</td>
</tr>
<tr>
<td><code>create_db_parameter_group</code></td>
<td>Creates a new DB parameter group</td>
</tr>
<tr>
<td><code>create_db_subnet_group</code></td>
<td>Creates a new DB subnet group</td>
</tr>
<tr>
<td><code>create_event_subscription</code></td>
<td>Creates an event notification subscription</td>
</tr>
<tr>
<td><code>delete_db_cluster</code></td>
<td>The DeleteDBCluster action deletes a previously provisioned DB cluster</td>
</tr>
</tbody>
</table>
delete_db_cluster_endpoint
delete_db_cluster_parameter_group
delete_db_cluster_snapshot
delete_db_instance
delete_db_parameter_group
delete_db_subnet_group
delete_event_subscription
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_instances
describe_db_parameter_groups
describe_db_parameters
describe_db_subnet_groups
describe_engine_default_cluster_parameters
describe_engine_default_parameters
describe_event_categories
describe_events
describe_event_subscriptions
describe_orderable_db_instance_options
describe_pending_maintenance_actions
describe_valid_db_instance_modifications
failover_db_cluster
list_tags_for_resource
modify_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_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
start_db_cluster
stop_db_cluster

Deletes a custom endpoint and removes it from an Amazon Neptune DB cluster
Deletes a specified DB cluster parameter group
Deletes a DB cluster snapshot
The DeleteDBInstance action deletes a previously provisioned DB instance
Deletes a specified DBParameterGroup
Deletes a DB subnet group
Deletes an event notification subscription
Returns information about endpoints for an Amazon Neptune DB cluster
Returns a list of DBClusterParameterGroup descriptions
Returns the detailed parameter list for a particular DB cluster parameter group
Returns information about provisioned DB clusters, and supports pagination
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
Returns information about provisioned instances, and supports pagination
Returns a list of DBParameterGroup descriptions
Returns the detailed parameter list for a particular DB parameter group
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 for the past 14 days
Lists all the subscription descriptions for a customer account
Returns a list of orderable DB instance options for the specified database engine
Returns a list of resources (for example, DB instances) that have at least one pending maintenance action
You can call DescribeValidDBInstanceModifications to learn what modifications are possible
Forces a failover for a DB cluster
Lists all tags on an Amazon Neptune resource
Modify a setting for a DB cluster
Modifies the properties of an endpoint in an Amazon Neptune 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
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 value
Creates a new DB cluster from a DB snapshot or DB cluster snapshot
Restores a DB cluster to an arbitrary point in time
Starts an Amazon Neptune DB cluster that was stopped using the AWS console
Stops an Amazon Neptune DB cluster
Examples

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

## End(Not run)
```

---

**opsworks**  
*AWS OpsWorks*

### 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/opsws/) 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 `create_stack`, `clone_stack`, or `update_stack` 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.

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Operations

    assign_instance
    assign_volume
    associate_elastic_ip
    attach_elastic_load_balancer
    clone_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
    describe_service_errors
    describe_stack_provisioning_parameters
    describe_stacks
    describe_stack_summary
    describe_time_based_auto_scaling
```

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
Deregisters 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
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
Describes time-based auto scaling configurations for specified instances
describe_user_profiles  Describe specified users
describe_volumes    Describes an instance’s Amazon EBS volumes
detach_elastic_load_balancer  Detaches a specified Elastic Load Balancing instance from its layer
disassociate_elastic_ip  Disassociates an Elastic IP address from its instance
get_hostname_suggestion  Gets a generated host name for the specified layer, based on the current host name
grant_access        This action can be used only with Windows stacks
list_tags           Returns a list of tags that are applied to the specified stack or layer
reboot_instance     Reboots a specified instance
register_ecs_cluster  Registers a specified Amazon ECS cluster with a stack
register_elastic_ip     Registers an Elastic IP address with a specified stack
register_instance       Registers instances that were created outside of AWS OpsWorks Stacks with a specified stack
register_rds_db_instance  Registers an Amazon RDS instance with a stack
register_volume         Registers an Amazon EBS volume with a specified stack
set_load_based_auto_scaling  Specify the load-based auto scaling configuration for a specified layer
set_permission         Specifies a user’s permissions
set_time_based_auto_scaling  Specify the time-based auto scaling configuration for a specified instance
start_instance       Starts a specified instance
start_stack            Starts a stack’s instances
stop_instance          Stops a specified instance
stop_stack             Stops a specified stack
tag_resource           Apply cost-allocation tags to a specified stack or layer in AWS OpsWorks Stacks
unassign_instance      Unassigns a registered instance from all layers that are using the instance
unassign_volume        Unassigns an assigned Amazon EBS volume
untag_resource         Removes tags from a specified stack or layer
update_app               Updates a specified app
update_elastic_ip        Updates a registered Elastic IP address’s name
update_instance          Updates a specified instance
update_layer             Updates a specified layer
update_my_user_profile    Updates a user’s SSH public key
update_rds_db_instance    Updates an Amazon RDS instance
update_stack              Updates a specified stack
update_user_profile       Updates a specified user profile
update_volume             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
opsworkscm

- opsworks-cm.eu-west-1.amazonaws.com

For more information, see AWS OpsWorks endpoints and quotas in the AWS General Reference.

**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.
```

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```
svc <- 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
organizations

Disassociates a node from an AWS OpsWorks CM server, and removes the node from the server's managed nodes.

Exports a specified server engine attribute as a base64-encoded string.

Returns a list of tags that are applied to the specified AWS OpsWorks for Chef Automate server.

Restores a backup to a server that is in a CONNECTION_LOST, HEALTHY, RUNNING, UNHEALTHY, or TERMINATED state.

Manually starts server maintenance.

Applies tags to an AWS OpsWorks for Chef Automate or AWS OpsWorks for Puppet Enterprise server.

Removes specified tags from an AWS OpsWorks-CM server or backup.

Updates settings for a server.

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

```r
config

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

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.
organizations

Service syntax

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

Operations

- `accept_handshake`: Sends a response to the originator of a handshake agreeing to the action proposed in the handshake request.
- `attach_policy`: Attaches a policy to a root, an organizational unit (OU), or an individual AWS account.
- `cancel_handshake`: Cancels a handshake.
- `create_account`: Creates an AWS account that is automatically a member of the organization whose credentials made the request.
- `create_gov_cloud_account`: This action is available if all of the following are true:
- `create_organization`: Creates an AWS organization.
- `create_organizational_unit`: Creates an organizational unit (OU) within a root or parent OU.
- `create_policy`: Creates a policy of a specified type that you can attach to a root, an organizational unit, or an individual AWS account.
- `decline_handshake`: Declines a handshake request.
- `delete_organization`: Deletes the organization.
- `delete_organizational_unit`: Deletes an organizational unit (OU) from a root or another OU.
- `delete_policy`: Deletes the specified policy from your organization.
- `deregister_delegated_administrator`: Removes the specified member AWS account as a delegated administrator for the specified AWS service.
- `describe_account`: Retrieves AWS Organizations-related information about the specified account.
- `describe_create_account_status`: Retrieves the current status of an asynchronous request to create an account.
- `describe_effective_policy`: Returns the contents of the effective policy for specified policy type and account.
- `describe_handshake`: Retrieves information about a previously requested handshake.
- `describe_organizational_unit`: Retrieves information about an organizational unit (OU).
- `describe_organization`: Retrieves information about the organization that the user’s account belongs to.
- `describe_policy`: Retrieves information about a policy.
- `detach_policy`: Detaches a policy from a target root, organizational unit (OU), or account.
- `disable_aws_service_access`: Disables the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.
- `disable_all_features`: Enables all features in an organization.
- `enable_aws_service_access`: Enables the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.
- `enable_policy_type`: Enables a policy type in a root.
- `invite_account_to_organization`: Sends an invitation to another account to join your organization as a member account.
- `leave_organization`: Removes a member account from its parent organization.
- `list_accounts`: Lists all the accounts in the organization.
- `list_accounts_for_parent`: Lists the accounts in an organization that are contained by the specified target root.
list_aws_service_access_for_organization
list_children
list_create_account_status
list_delegated_administrators
list_delegated_services_for_account
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
register_delegated_administrator
remove_account_from_organization
tag_resource
untag_resource
update_organizational_unit
update_policy

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 parent.
Lists the account creation requests that match the specified status that is currently being tracked for the organization.
Lists the AWS accounts that are designated as delegated administrators in this organization.
List the AWS services for which the specified account is a delegated administrator.
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 child.
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, organizational unit (OU), or account.
Lists the roots that are defined in the current organization.
Lists tags that are attached to 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) to the specified destination.
Enables the specified member account to administer the Organizations features of the specified AWS service.
Removes the specified account from the organization.
Adds one or more tags to the specified resource.
Removes any tags with the specified keys from the specified resource.
Renames the specified organizational unit (OU).
Updates an existing policy with a new name, description, or content.

Examples

```r
## Not run:
svc <- organizations()
# 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.
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

```r
personalize(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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**: Creates a batch inference job
- **create_campaign**: Creates a campaign by deploying a solution version
- **create_dataset**: Creates an empty dataset and adds it to the specified dataset group
- **create_dataset_group**: Creates an empty dataset group
- **create_dataset_import_job**: Creates a job that imports training data from your data source (an Amazon S3 bucket) to an Amazon Personalize dataset
- **create_event_tracker**: Creates an event tracker that you use when sending event data to the specified dataset group using the PutEvents API
- **create_filter**: Creates a recommendation filter
- **create_schema**: Creates an Amazon Personalize schema from the specified schema string
- **create_solution**: Creates the configuration for training a model
- **create_solution_version**: Trains or retracts an active solution
- **delete_campaign**: Removes a campaign by deleting the solution deployment
- **delete_dataset**: Deletes a dataset
- **delete_dataset_group**: Deletes a dataset group
- **delete_event_tracker**: Deletes the event tracker
- **delete_filter**: Deletes a filter
- **delete_schema**: Deletes a 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_filter
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_filters
list_recipes
list_schemas
list_solutions
list_solution_versions
update_campaign

Deletes all versions of a solution and the Solution object itself
Describes the given algorithm
Gets the properties of a batch inference job including name, Amazon Resource Name (ARN), status, and the ARN of the solution version used to generate the recommendations
Describes the given campaign, including its status
Describes the given dataset
describes the given dataset group
Describes the dataset import job created by CreateDatasetImportJob, including the import job status
Describes an event tracker
Describes the given feature transformation
Describes a filter’s properties
Describes a recipe
Describes a schema
Describes a solution
Describes a specific version of a solution
Gets the metrics for the specified solution version
Gets a list of the batch inference jobs that have been performed off of a solution version
Returns a list of campaigns that use the given solution
Returns a list of dataset groups
Returns a list of dataset import jobs that use the given dataset
Returns the list of datasets contained in the given dataset group
Returns the list of event trackers associated with the account
Lists all filters that belong to a given dataset group
Returns a list of available recipes
Returns the list of schemas associated with the account
Returns a list of solutions that use the given dataset group
Returns a list of solution versions for the given solution
Updates a campaign by either deploying a new solution or changing the value of the campaign’s minProvisionedTPS parameter

Examples

## Not run:
svc <- personalize()
svc$create_batch_inference_job(
  Foo = 123
)

## End(Not run)

---

**personalizeevents**

Amazon Personalize Events

---

**Description**

Amazon Personalize can consume real-time user event data, such as *stream* or *click* data, and use it for model training either alone or combined with historical data. For more information see recording-events.
Usage

personalizeevents(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
- put_items Adds one or more items to an Items dataset
- put_users Adds one or more users to a Users dataset

Examples

## Not run:
svc <- personalizeevents()
svc$put_events(
  Foo = 123
)

## End(Not run)
Amazon Personalize Runtime

Description

Amazon Personalize Runtime

Usage

personalizeruntime(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
AWS Performance Insights

Description

Amazon RDS Performance Insights enables you to monitor and explore different dimensions of database load based on data captured from a running DB instance. The guide provides detailed information about Performance Insights data types, parameters and errors.

When Performance Insights is enabled, the Amazon RDS Performance Insights API provides visibility into the performance of your DB instance. Amazon CloudWatch provides the authoritative source for AWS service-vended monitoring metrics. Performance Insights offers a domain-specific view of DB load.

DB load is measured as Average Active Sessions. Performance Insights provides the data to API consumers as a two-dimensional time-series dataset. The time dimension provides DB load data for each time point in the queried time range. Each time point decomposes overall load in relation to the requested dimensions, measured at that time point. Examples include SQL, Wait event, User, and Host.

- To learn more about Performance Insights and Amazon Aurora DB instances, go to the Amazon Aurora User Guide.
- To learn more about Performance Insights and Amazon RDS DB instances, go to the Amazon RDS User Guide.

Usage

pi(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```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)
```

Description

Doc Engage API - Amazon Pinpoint API

Usage

`pinpoint(config = list())`

Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
svc <- 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

- `create_app` Creates an application
- `create_campaign` Creates a new campaign for an application or updates the settings of an existing campaign for an application
- `create_email_template` Creates a message template for messages that are sent through the email channel
- `create_export_job` Creates an export job for an application
- `create_import_job` Creates an import job for an application
- `create_journey` Creates a journey for an application
- `create_push_template` Creates a message template for messages that are sent through a push notification channel
- `create_recommender_configuration` Creates an Amazon Pinpoint configuration for a recommender model
- `create_segment` 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
- `create_sms_template` Creates a message template for messages that are sent through the SMS channel
- `create_voice_template` Creates a message template for messages that are sent through the voice channel
- `delete_adm_channel` Disables the ADM channel for an application and deletes any existing settings for the channel
- `delete_apns_channel` Disables the APNs channel for an application and deletes any existing settings for the channel
- `delete_apns_sandbox_channel` Disables the APNs sandbox channel for an application and deletes any existing settings for the channel
- `delete_apns_voip_channel` Disables the APNs VoIP channel for an application and deletes any existing settings for the channel
- `delete_apns_voip_sandbox_channel` Disables the APNs VoIP sandbox channel for an application and deletes any existing settings for the channel
- `delete_app` Deletes an application
- `delete_baidu_channel` Disables the Baidu channel for an application and deletes any existing settings for the channel
- `delete_campaign` Deletes a campaign from an application
- `delete_email_channel` Disables the email channel for an application and deletes any existing settings for the channel
- `delete_email_template` Deletes a message template for messages that were sent through the email channel
- `delete_endpoint` Deletes an endpoint from an application
- `delete_event_stream` Deletes the event stream for an application
- `delete_gcm_channel` Disables the GCM channel for an application and deletes any existing settings for the channel
delete_journey
delete_push_template
delete_recommender_configuration
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
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_recommender_configuration
get_recommender_configurations
get_segment
get_segment_export_jobs
get_segment_import_jobs
get_segments
get_segment_version
get_segment_versions

deletes a journey from an application
Deletes a message template for messages that were sent through a push notification channel
Deletes an Amazon Pinpoint configuration for a recommender model
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 information about 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 a specific application
Retrieves information about the status, configuration, and other settings for a specific version of a campaign
Retrieves information about the status, configuration, and other settings for all versions of a specific campaign
Retrieves information about the history and status of each channel for an application
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 the content and settings of a message template for messages that are sent through the push template
Retrieves information about an Amazon Pinpoint configuration for a recommender model
Retrieves information about all the recommender model configurations that are associated with the application
Retrieves information about the configuration, dimension, and other settings for a specific segment
Retrieves information about the status and settings of the export jobs for a specific segment
Retrieves information about the status and settings of the import jobs for a specific segment
Retrieves information about the configuration, dimension, and other settings for all segments
Retrieves information about the configuration, dimension, and other settings for all specific segments
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_events
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
update_endpoints_batch
update_gcm_channel
update_journey
update_journey_state
update_push_template
update_recommender_configuration
update_segment
update_sms_channel
update_sms_template
update_template_active_version
update_voice_channel
update_voice_template

Retrieves information about the status and settings of the SMS channel for an application
Retrieves the content and settings of a message template for messages that are sent through the SMS channel
Retrieves information about all the endpoints that are associated with a specific user ID
Retrieves information about the status and settings of the voice channel for an application
Retrieves the content and settings of a message template for messages that are sent through the voice channel
Retrieves information about the status, configuration, and other settings for all the journeys that are associated with an application
Retrieves all the tags (keys and values) that are associated with an application, campaign, message template, or segment
Retrieves information about all the message templates 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 exists already
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 to a list of users
Adds one or more tags (keys and values) to an application, campaign, message template, segment, or resource
Removes one or more tags (keys and values) from an application, campaign, message template, segment, or resource
Enables the ADM channel for an application or updates the status and settings of the ADM channel for an application
Enables the APNs channel for an application or updates the status and settings of the APNs channel for an application
Enables the APNs sandbox channel for an application or updates the status and settings of the APNs sandbox channel for an application
Enables the APNs VoIP channel for an application or updates the status and settings of the APNs VoIP channel for an application
Enables the APNs VoIP sandbox channel for an application or updates the status and settings of the APNs VoIP sandbox channel for an application
Enables the Baidu channel for an application or updates the status and settings of the Baidu channel for an application
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 for an application
Updates an existing email 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
Creates a new batch of endpoints for an application or updates the settings of an existing batch of endpoints
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
Updates an Amazon Pinpoint configuration for a recommender model
Creates a new segment for an application or updates the configuration, dimension, and other settings of an existing segment
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

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

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
csvc <- pinpointemail(
    config = 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 pool that it's associated with, whether or not it's enabled for sending email, and more
- `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 that it's associated with, as well information about the automatic warm-up process for the address
- `get_dedicated_ips`: List the dedicated IP addresses that are associated with your Amazon Pinpoint account
- `get_deliverability_dashboard_options`: Retrieve information about the status of the Deliverability dashboard for your Amazon Pinpoint account
- `get_deliverability_test_report`: Retrieve the results of a predictive inbox placement test
- `get_domain_deliverability_campaign`: Retrieve all the deliverability data for a specific campaign
- `get_domain_statistics_report`: Retrieve inbox placement and engagement rates for the domains that you use to send email
- `get_email_identity`: Provides information about a specific identity associated with your Amazon Pinpoint account
- `list_configuration_sets`: List all of the configuration sets associated with your Amazon Pinpoint account
- `list_dedicated_ip_pools`: List all of the dedicated IP pools that exist in your Amazon Pinpoint account
- `list_deliverability_test_reports`: Show a list of the predictive inbox placement tests that you've performed, regardless of their statuses
- `list_domain_deliveryability_campaigns`: Retrieve deliverability data for all the campaigns that used a specific domain to send email
- `list_email_identities`: Returns a list of all of the email identities that are associated with your Amazon Pinpoint account
- `list_tags_for_resource`: Retrieve a list of the tags (keys and values) that are associated with a specified resource
- `put_account_dedicated_ip_warmup_attributes`: Enable or disable the automatic warm-up feature for dedicated IP addresses
- `put_account_sending_attributes`: Enable or disable the ability of your account to send email
- `put_configuration_set_delivery_options`: Associate a configuration set with a dedicated IP pool
- `put_configuration_set_reputation_options`: Enable or disable collection of reputation metrics for emails that you send using this configuration set
Examples

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

---

**Description**

Pinpoint SMS and Voice Messaging public facing APIs

**Usage**

`pinpointsmsvoice(config = list())`

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

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

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

## End(Not run)
```
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.

Value
A client for the service. You can call the service’s operations using syntax like svc$operation(...),
where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax
svc <- 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

```r
## Not run:
svc <- polly()
# Deletes a specified pronunciation lexicon stored in an AWS Region.
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 `get_attribute_values` to see what values are available for an attribute. With the service code and an attribute name and value, you can use `get_products` 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

```r
pricing(config = list())
```

## Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
csvc <- 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
Description

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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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_account_customization Creates Amazon QuickSight customizations the current AWS Region
create_analysis Creates an analysis in Amazon QuickSight
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
create_ingestion
create_namespace
create_template
create_template_alias
create_theme
create_theme_alias
delete_account_customization
delete_analysis
delete_dashboard
delete_data_set
delete_data_source
delete_group
delete_group_membership
delete_iam_policy_assignment
delete_namespace
delete_template
delete_template_alias
delete_theme
delete_theme_alias
delete_user
delete_user_by_principal_id
describe_account_customization
describe_account_settings
describe_analysis
describe_analysis_permissions
describe_dashboard
describe_dashboard_permissions
describe_data_set
describe_data_set_permissions
describe_data_source
describe_data_source_permissions
describe_group
describe_iam_policy_assignment
describe_ingestion
describe_namespace
describe_template
describe_template_alias
describe_template_permissions
describe_theme
describe_theme_alias
describe_theme_permissions
describe_user
get_dashboard_embed_url
get_session_embed_url
list_analyses
list_dashboards
list_dashboard_versions

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.
create_namespace
(Enterprise edition only) Creates a new namespace for you to use with Amazon QuickSight.
create_template
Creates a template from an existing QuickSight analysis or template.
create_template_alias
Creates a template alias for a template.
create_theme
Creates a theme.
create_theme_alias
Creates a theme alias for a theme.
delete_account_customization
Deletes all Amazon QuickSight customizations in this AWS Region for the specified AWS account.
delete_analysis
Deletes an analysis from Amazon QuickSight.
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_namespace
Deletes a namespace and the users and groups that are associated with the namespace.
delete_template
Deletes a template.
delete_template_alias
Deletes the item that the specified template alias points to.
delete_theme
Deletes a theme.
delete_theme_alias
Deletes the version of the theme that the specified theme 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_account_customization
Describes the customizations associated with the provided AWS account and Amazon QuickSight namespace in an AWS Region.
describe_account_settings
Describes the settings that were used when your QuickSight subscription was first created in this AWS account.
describe_analysis
Provides a summary of the metadata for an analysis.
describe_analysis_permissions
Describes the permissions on an analysis.
describe_dashboard
Describes 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 resource permissions for a data source.
describe_group
Returns an Amazon QuickSight group’s description and Amazon Resource Name (ARN).
describe_iam_policy_assignment
Describes a SPICE ingestion.
describe_ingestion
Describes the current namespace.
describe_namespace
Describes a template’s metadata.
describe_template
Describes the template alias for a template.
describe_template_permissions
Describes read and write permissions on a template.
describe_theme
Describes a theme.
describe_theme_alias
Describes the alias for a theme.
describe_theme_permissions
Describes the read and write permissions for a theme.
describe_user
Returns information about a user, given the user name.
get_dashboard_embed_url
Generates a session URL and authorization code that you can use to embed an Amazon QuickSight read-only dashboard in your web server code.
get_session_embed_url
Generates a session URL and authorization code that you can use to embed the Amazon QuickSight console in your web server code.
list_analyses
Lists Amazon QuickSight analyses that exist in the specified AWS account.
list_dashboards
Lists dashboards in an AWS account.
list_dashboard_versions
Lists all the versions of the dashboards in the QuickSight subscription.
### Examples

```
## Not run:
```
ram

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

`ram(config = list())`

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
cvc <- quicksight()
cvc$cancel_ingestion(
  Foo = 123
)

## End(Not run)
```
Operations

accept_resource_share_invitation
accept_resource_share
associate_resource_share_permission
create_resource_share
delete_resource_share
disassociate_resource_share
disassociate_resource_share_permission
enable_sharing_with_aws_organization
get_permission
get_resource_policies
get_resource_share_associations
get_resource_share_invitations
get_resource_shares
list_pending_invitation_resources
list_permissions
list_principals
list_resources
list_resource_share_permissions
list_resource_types
promote_resource_share_created_from_policy
reject_resource_share_invitation
tag_resource
untag_resource
update_resource_share

Accepts an invitation to a resource share from another AWS account
Associates the specified resource share with the specified principals and resources
Associates a permission with a resource share
Creates a resource share
Deletes the specified resource share
Disassociates the specified principals or resources from the specified resource share
Disassociates an AWS RAM permission from a resource share
Enables resource sharing within your AWS Organization
Gets the contents of an AWS RAM permission in JSON format
Gets the policies for the specified resources that you own and have shared
Gets the resources or principals for the resource shares that you own
Gets the invitations for resource sharing that you’ve received
Gets the resource shares that you own or the resource shares that are shared with you
Lists the resources in a resource share that is shared with you but that the invitation is still pending for
Lists the AWS RAM permissions
Lists the principals that you have shared resources with or that have shared resources with you
Lists the resources that you added to a resource shares or the resources that are shared with you
Lists the AWS RAM permissions that are associated with a resource share
Lists the shareable resource types supported by AWS RAM
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
Rejects an invitation to a resource share from another AWS account
Adds the specified tags to the specified resource share that you own
Removes the specified tags from the specified resource share that you own
Updates the specified resource share that you own

Examples

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

## End(Not run)
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. 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

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

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```rds
svc <- rds(
  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 Amazon Aurora DB cluster.
- `add_role_to_db_instance`: Associates an AWS Identity and Access Management (IAM) role with a DB instance.
- `add_source_identifier_to_subscription`: Adds a source identifier to an existing RDS event notification subscription.
- `add_tags_to_resource`: Adds metadata tags to an Amazon RDS resource.
- `apply_pending_maintenance_action`: Applies a pending maintenance action to a resource (for example, to a DB cluster).
- `authorize_db_security_group_ingress`: Enables ingress to a DBSecurityGroup using one of two forms of authorization.
- `backtrack_db_cluster`: Backtracks a DB cluster to a specific time, without creating a new DB cluster.
- `build_auth_token`: Return an authentication token for a database connection.
- `cancel_export_task`: Cancels an export task in progress that is exporting a snapshot to Amazon S3.
- `copy_db_cluster`: Copies the specified DB cluster parameter group.
- `copy_db_cluster_parameter_group`: Copies a snapshot of a DB cluster.
- `copy_db_cluster_snapshot`: Copies a DB cluster parameter group.
- `copy(db)`: Copies the specified DB parameter group.
- `copy(db)`: Copies the specified option group.
- `create_custom_availability_zone`: Creates a custom Availability Zone (AZ).
- `create_db_cluster`: Creates a new Amazon Aurora DB cluster.
- `create_db_cluster_endpoint`: Creates a new custom endpoint and associates it with an Amazon Aurora 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_instance_read_replica`: Creates a new DB instance that acts as a read replica for an existing source DB instance.
- `create_db_proxy`: Creates a new DB proxy.
- `create_db_security_group`: Creates a new DB security group.
- `create_db_snapshot`: Creates a snapshot of a DB instance.
- `create_db_subnet_group`: Creates a new DB subnet group.
- `create_event_subscription`: Creates an RDS event notification subscription.
- `create_global_cluster`: Creates an Aurora global database spread across multiple AWS Regions.
- `create_option_group`: Creates a new option group.
- `delete_custom_availability_zone`: 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 using the DbiResourceId value of the source.
Deletes a specified DB parameter group.
Deletes an existing proxy.
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.
Deletes the installation medium for a DB engine that requires an on-prem license.
Deletes an existing option group.
Remove the association between one or more DBProxyTarget data structures and a DBProxyTargetGroup.
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 specified 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.
Returns information about DB proxies.
Returns information about DB proxy target groups, represented by DBProxyTarget objects.
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.
Returns the default engine and system parameter information for the specified engine.
Displays a list of categories for all event source types, or, if specified, for a specified source.
Returns events related to DB instances, DB clusters, DB parameter groups, and so on.
Lists all the subscription descriptions for a customer account.
Returns information about a snapshot export to Amazon S3.
Returns information about Aurora global database clusters.
Describes the available installation media for a DB engine that requires an on-prem license.
Describes all available 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
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_offerings
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

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, copy a DB snapshot from, or replicate automated backups from
You can call DescribeValidDBInstanceModifications to learn what modifications you can make to your DB instance
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, such as SQL Server
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 temporarily, 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
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 DB cluster parameter group
Modifies settings for a DB instance
Modifies the parameters of a DB parameter group
Changes the settings for an existing DB proxy
Modifies the properties of a DBProxyTargetGroup
Updates a manual DB snapshot with a new engine version
Adds an attribute and values to, or removes an attribute and values from, a DB snapshot
Modifies an existing DB subnet group
Modifies an existing RDS event notification subscription
Modify 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
Associate one or more DBProxyTarget data structures with a DBProxy Target Group
Detaches an Aurora secondary cluster from an Aurora global database cluster
Disassociates an AWS Identity and Access Management (IAM) role from an Aurora global database cluster
Disassociates an AWS Identity and Access Management (IAM) role from a DB instance
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 values
Modifies the parameters of a DB parameter group to the engine/system default values
Creates an Amazon Aurora DB cluster from MySQL 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 a DB instance to an arbitrary point in time
Reverts to a previous version of a database and all associated read-only replicas
Starts a database activity stream to monitor activity on the database
Starts an Amazon Aurora DB cluster that was stopped using the AWS console
**rdsdataservice**

**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**

```r
rdsdataservice(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

---

**Examples**

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

## End(Not run)
```
Service syntax

```r
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.

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
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

authorize_cluster_security_group_ingress
authorize_snapshot_access
batch_delete_cluster_snapshots
batch_modify_cluster_snapshots
cancel_resize
copy_cluster_snapshot
create_cluster
create_cluster_parameter_group
create_cluster_security_group
create_cluster_snapshot
create_cluster_subnet_group
create_event_subscription
create_hsm_client_certificate
create_hsm_configuration
create_scheduled_action
create_snapshot_copy_grant
create_snapshot_schedule
create_tags
create_usage_limit
delete_cluster
delete_cluster_parameter_group
delete_cluster_security_group
delete_cluster_snapshot
delete_cluster_subnet_group
delete_event_subscription
delete_hsm_client_certificate
delete_hsm_configuration
delete_scheduled_action
delete_snapshot_copy_grant
delete_snapshot_schedule
delete_tags

delete_usage_limit
describe_account_attributes
describe_cluster_db_revisions
describe_cluster_parameter_groups
describe_cluster_parameters
describe_clusters
describe_cluster_security_groups
describe_cluster_snapshots
describe_cluster_subnet_groups
describe_cluster_tracks
describe_cluster_versions
describe_default_cluster_parameters
describe_event_categories
describe_events
describe_event_subscriptions
describe_hsm_client_certificates
describe_hsm_configurations

Add an inbound (ingress) rule to an Amazon Redshift security group
Authorizes the specified AWS customer account to restore the specified snapshot
Deletes a set of cluster snapshots
Modifies the settings for a set of cluster snapshots
Cancels a resize operation for a cluster
Copies the specified automated cluster snapshot to a new manual cluster snapshot
Creates a new cluster with the specified parameters
Creates an Amazon Redshift parameter group
Creates a new Amazon Redshift security group
Creates a manual snapshot of the specified cluster
Creates a new Amazon Redshift subnet group
Creates an Amazon Redshift event notification subscription
Creates an HSM client certificate that an Amazon Redshift cluster will use to connect
Creates an HSM configuration that contains the information required by an Amazon Redshift cluster
Creates a scheduled action
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 a snapshot schedule that can be associated to a cluster and which overrides the default system backup schedule
Adds tags to a cluster
Creates a usage limit for a specified Amazon Redshift feature on a cluster
Deletes a previously provisioned cluster without its final snapshot being created
Deletes a specified Amazon Redshift parameter group
Deletes an Amazon Redshift security group
Deletes the specified manual snapshot
Deletes the specified cluster subnet group
Deletes an Amazon Redshift event notification subscription
Deletes the specified HSM client certificate
Deletes the specified Amazon Redshift HSM configuration
Deletes a scheduled action
Deletes the specified snapshot copy grant
Deletes a snapshot schedule
Deletes tags from a resource
Deletes a usage limit from a cluster
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
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, and security and access properties
Returns information about Amazon Redshift security groups
Returns one or more snapshot objects, which contain metadata about your cluster
Returns one or more cluster subnet group objects, which contain metadata about your cluster
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 customer account
Returns information about the specified HSM client certificate
Returns information about the specified Amazon Redshift HSM configuration
describe_logging_status

describe_node_configuration_options

describe_orderable_cluster_options

describe_reserved_node_offerings

describe_reserved_nodes

describe_resize

describe_scheduled_actions

describe_snapshot_copy_grants

describe_snapshot_schedules

describe_storage

describe_table_restore_status

describe_tags

describe_usage_limits

disable_logging

disable_snapshot_copy

enable_logging

enable_snapshot_copy

get_cluster_credentials

get_reserved_node_exchange_offering

modify_cluster

modify_cluster_db_revision

modify_cluster_iam_roles

modify_cluster_maintenance

modify_cluster_parameter_group

modify_cluster_snapshot

modify_cluster_snapshot_schedule

modify_cluster_subnet_group

modify_event_subscription

modify_scheduled_action

modify_snapshot_copy_retention_period

modify_snapshot_schedule

modify_usage_limit

pause_cluster

purchase_reserved_node_offering

reboot_cluster

reset_cluster_parameter_group

resize_cluster

restore_from_cluster_snapshot

restore_table_from_cluster_snapshot

resume_cluster

revoke_cluster_security_group_ingress

revoke_snapshot_access

rotate_encryption_key

---

**Examples**

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

## End(Not run)

**rekognition**

---

**Amazon Rekognition**

---

**Description**

This is the Amazon Rekognition API reference.

**Usage**

rekognition(config = list())

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

svc <- 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
create_collection
create_project
create_project_version
create_stream_processor
delete_collection
delete_faces
delete_project
delete_project_version
delete_stream_processor
describe_collection
describe_projects
describe_project_versions
describe_stream_processor
detect_custom_labels
detect_faces
detect_labels
detect_moderation_labels
detect_protective_equipment
detect_text
get_celebrity_info
get_celebrity_recognition
get_content_moderation
get_face_detection
get_face_search
get_label_detection
get_person_tracking
get_segment_detection
get_text_detection
index_faces
list_collections
list_faces
list_stream_processors
recognize_celebrities
search_faces
search_faces_by_image
start_celebrity_recognition
start_content_moderation
start_face_detection
start_face_search
start_label_detection
start_person_tracking
start_project_version
start_segment_detection
start_stream_processor
start_text_detection
stop_project_version
stop_stream_processor

rekognition

compare_faces
Creates 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
Deletes 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_project
Deletes an Amazon Rekognition Custom Labels project.
delete_project_version
Deletes an Amazon Rekognition Custom Labels model.
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_protective_equipment
Detects Personal Protective Equipment (PPE) worn by people detected in an 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 content moderation 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 the face search results forAmazon 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 person tracking results of a Amazon Rekognition Video analysis started by StartPersonTracking.
get_segment_detection
Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSegmentDetection.
get_text_detection
Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDetection.
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_project_versions
Gets a list of stream processors that you have created with CreateStreamProcessor.
recognize_celebrities
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 specific.
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.
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 the running of the version of a model.
start_segment_detection
Starts asynchronous detection of segment detection in a stored video.
start_stream_processor
Starts processing a stream processor.
start_text_detection
Starts asynchronous detection of text in a stored video.
stop_project_version
Stops a running model.
stop_stream_processor
Stops a running stream processor that was created by CreateStreamProcessor.
## Examples

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

---

**resourcegroups**

### AWS Resource Groups

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

resourcegroups(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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 resource group with the specified name and description
delete_group Deletes the specified resource group
group Resources group Returns information about a specified resource group
get_group_configuration Returns the service configuration associated with the specified resource group
get_group_query Retrieves 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
group_resources Adds the specified resources to the specified group
list_group_resources Returns a list of ARNs of the resources that are members of a specified resource group
list_groups Returns a list of existing resource groups in your account
put_group_configuration Attaches a service configuration to the specified group
resourcegroupstaggingapi

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>search_resources</td>
<td>Returns a list of AWS resource identifiers that matches the specified query</td>
</tr>
<tr>
<td>tag</td>
<td>Adds tags to a resource group with the specified ARN</td>
</tr>
<tr>
<td>ungroup_resources</td>
<td>Removes the specified resources from the specified group</td>
</tr>
<tr>
<td>untag</td>
<td>Deletes tags from a specified resource group</td>
</tr>
<tr>
<td>update_group</td>
<td>Updates the description for an existing group</td>
</tr>
<tr>
<td>update_group_query</td>
<td>Updates the resource query of a group</td>
</tr>
</tbody>
</table>

Examples

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

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.

Do not store personally identifiable information (PII) or other confidential or sensitive information in tags. We use tags to provide you with billing and administration services. Tags are not intended to be used for private or sensitive data.

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.

Services that support the Resource Groups Tagging API

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 Cloud Map
- AWS CloudFormation
- Amazon CloudFront
- AWS CloudHSM
- AWS CloudTrail
- Amazon CloudWatch (alarms only)
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon Cloudwatch Synthetics
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeGuru Profiler
- AWS CodePipeline
- AWS CodeStar
• AWS CodeStar Connections
• Amazon Cognito Identity
• Amazon Cognito User Pools
• Amazon Comprehend
• AWS Config
• Amazon Connect
• AWS Data Exchange
• AWS Data Pipeline
• AWS Database Migration Service
• AWS DataSync
• AWS Device Farm
• AWS Direct Connect
• AWS Directory Service
• Amazon DynamoDB
• Amazon EBS
• Amazon EC2
• EC2 Image Builder
• Amazon ECR
• Amazon ECS
• Amazon EKS
• AWS Elastic Beanstalk
• Amazon Elastic File System
• Elastic Load Balancing
• Amazon Elastic Inference
• Amazon ElastiCache
• Amazon Elasticsearch Service
• AWS Elemental MediaLive
• AWS Elemental MediaPackage
• AWS Elemental MediaPackage VoD
• AWS Elemental MediaTailor
• Amazon EMR
• Amazon EventBridge Schema
• AWS Firewall Manager
• Amazon Forecast
• Amazon Fraud Detector
• Amazon FSx
• Amazon S3 Glacier
• AWS Global Accelerator
• AWS Ground Station
• AWS Glue
• Amazon GuardDuty
• Amazon Inspector
• Amazon Interactive Video Service
• AWS IoT Analytics
• AWS IoT Core
• AWS IoT Device Defender
• AWS IoT Device Management
• AWS IoT Events
• AWS IoT Greengrass
• AWS IoT 1-Click
• AWS IoT Sitewise
• AWS IoT Things Graph
• Amazon Kendra
• AWS Key Management Service
• Amazon Kinesis
• Amazon Kinesis Data Analytics
• Amazon Kinesis Data Firehose
• AWS Lambda
• Amazon Lex
• AWS License Manager
• Amazon Lightsail
• Amazon Macie
• Amazon Machine Learning
• Amazon MQ
• Amazon MSK
• Amazon MSK
• Amazon Neptune
• AWS Network Manager
• AWS OpsWorks
• AWS OpsWorks CM
• AWS Organizations
• Amazon Pinpoint
• Amazon Quantum Ledger Database (QLDB)
• Amazon RDS
resourcegroupstaggingapi

- Amazon Redshift
- AWS Resource Access Manager
- AWS Resource Groups
- AWS RoboMaker
- Amazon Route 53
- Amazon Route 53 Resolver
- Amazon S3 (buckets only)
- Amazon SageMaker
- Savings Plans
- AWS Secrets Manager
- AWS Security Hub
- AWS Service Catalog
- Amazon Simple Email Service (SES)
- Amazon Simple Notification Service (SNS)
- Amazon Simple Queue Service (SQS)
- Amazon Simple Workflow Service
- AWS Step Functions
- AWS Storage Gateway
- AWS Systems Manager
- AWS Transfer for SFTP
- Amazon VPC
- AWS WAF
- AWS WAF Regional
- Amazon WorkLink
- Amazon WorkSpaces

Usage

resourcegroupstaggingapi(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
svc <- resourcegroupstagingapi(
  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 <- resourcegroupstagingapi()
svc$describe_report_creation(
  Foo = 123
)
```

## End(Not run)

---

**route53**

**Amazon Route 53**

Description

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

route53(config = list())

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

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

**Operations**

- `activate_key_signing_key`  
  Activates a key signing key (KSK) so that it can be used for signing by DNSSEC
- `associate_vpc_with_hosted_zone`  
  Associates an Amazon VPC with a private hosted zone
- `change_resource_record_sets`  
  Creates, changes, or deletes a resource record set, which contains authoritative DNS information for a specified domain name or subdomain name
- `change_tags_for_resource`  
  Adds, edits, or deletes tags for a health check or a hosted zone
- `create_health_check`  
  Creates a new health check
- `create_hosted_zone`  
  Creates a new public or private hosted zone
- `create_key_signing_key`  
  Creates a new key signing key (KSK) associated with a hosted zone
- `create_query_logging_config`  
  Creates a configuration for DNS query logging
- `create_reusable_delegation_set`  
  Creates a delegation set (a group of four name servers) that can be reused by multiple hosted zones that were created by the same AWS account
- `create_traffic_policy`  
  Creates a traffic policy, which you use to create multiple DNS resource record sets in a specified hosted zone based on the settings in the traffic policy
- `create_traffic_policy_instance`  
  Creates a new version of an existing traffic policy
- `create_traffic_policy_version`  
  Creates resource record sets in a specified hosted zone based on the settings in a specified traffic policy version
- `create_vpc_association_authorization`  
  Authorizes the AWS account that created a specified VPC to submit an AssociateVPCWithHostedZone request to associate the VPC with a specified hosted zone that was created by a different account
- `deactivate_key_signing_key`  
  Deactivates a key signing key (KSK) so that it will not be used for signing by DNSSEC
- `delete_health_check`  
  Deletes a health check
- `delete_hosted_zone`  
  Deletes a hosted zone
delete_key_signing_key
delete_query_logging_config
deleteReusableDelegationSet
deleteTrafficPolicy
deleteTrafficPolicyInstance
delete_vpc_association_authorization
disable_hosted_zone_dnssec
disassociate_vpc_from_hosted_zone
enable_hosted_zone_dnssec
get_account_limit
get_change
get_checker_ip_ranges
get_dnssec
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
list_hosted_zones_by_vpc
list_query_logging_configs
list_resource_record_sets
list_reusable_delegation_sets
list_tags_for_resource
list_tags_for_resources
list_traffic_policies
list_traffic_policy_instances
list_traffic_policy_instances_by_hosted_zone
list_traffic_policy_instances_by_policy
list_traffic_policy_versions
list_vpc_association_authorizations
test_dns_answer
update_health_check
update_hosted_zone_comment
update_traffic_policy_comment
update_traffic_policy_instance

Deletes a key signing key (KSK)
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 uses to publish DNS records
Removes authorization to submit an AssociateVPCWithHostedZone request to associate a specified VPC with a hosted zone
Disables DNSSEC signing in a specific hosted zone
Disassociates an Amazon Virtual Private Cloud (Amazon VPC) from an Amazon Route 53 hosted zone
Enables DNSSEC signing in a specific hosted zone
Gets the specified limit for the current account, for example, the maximum number of health checks that you can create
Returns the current status of a change batch request
GetCheckerIpRanges still works, but we recommend that you download ip-ranges from the S3 bucket
Returns information about DNSSEC for a specific hosted zone, including the key signing keys (KSKs) and zone signing keys (ZSKs) in the hosted zone
Gets information about whether a specified geographic location is supported for Amazon Route 53 geolocation resource record sets
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 hosted 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 records that you can create in the hosted zone
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 reusable delegation set
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
Lists all the private hosted zones that a specified VPC is associated with, regardless of which AWS account or AWS service owns the hosted zones
Lists the resource record sets in a specified hosted zone
Retrieves a list of the reusable delegation sets that are associated with the current AWS account
Lists tags for one health check or hosted zone
Lists tags for up to 10 health checks or hosted zones
Gets information about the latest version for every traffic policy that is associated with the current AWS account
Gets information about the traffic policy instances that you created by using the command line
Gets information about the traffic policy instances that you created in a specific AWS region
Gets information about the traffic policy instances that you created by using a specific API
Gets information about all of the versions for a specified traffic policy
Gets a list of the VPCs that were created by other accounts and that can be associated with your hosted zones
Gets the value that Amazon Route 53 returns in response to a DNS request for a specified zone
Updates an existing health check
Updates the comment for a specified hosted zone
Updates the comment for a specified traffic policy version
Updates the resource record sets in a specified hosted zone that were created by using the command line.
Examples

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

## End(Not run)
```

### Description

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

### Usage

```r
route53domains(config = list())
```

### Arguments

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```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

accept_domain_transfer_from_another_aws_account
cancel_domain_transfer_to_another_aws_account
check_domain_availability
check_domain_transferability
delete_tags_for_domain
disable_domain_auto_renew
disable_domain_transfer_lock
enable_domain_auto_renew
enable_domain_transfer_lock
get_contact_reachability_status
get_domain_detail
get_domain_suggestions
get_operation_detail
list_domains
list_operations
list_tags_for_domain
register_domain
reject_domain_transfer_from_another_aws_account
renew_domain
resend_contact_reachability_email
retrieve_domain_auth_code
transfer_domain
transfer_domain_to_another_aws_account
update_domain_contact
update_domain_contact_privacy
update_domain_nameservers
update_tags_for_domain
view_billing

Accepts the transfer of a domain from another AWS account to the current AWS account.
Cancels the transfer of a domain from the current AWS account to another AWS account.
This operation checks the availability of one domain name.
Checks whether a domain name can be transferred to Amazon Route 53.
This operation deletes the specified tags for a domain.
This operation disables automatic renewal of domain registration for the specified domain.
This operation removes the transfer lock on the domain (specifically the clientTransferProhibited status).
This operation configures Amazon Route 53 to automatically renew the nameserver updates for the specified domain.
This operation sets the transfer lock on the domain (specifically the clientTransferProhibited status).
For operations that require confirmation that the email address for the registrant contact is valid.
This operation returns detailed information about a specified domain.
The GetDomainSuggestions operation returns a list of suggested domain names for the current domain name.
This operation returns the current state and the associated tags for the specified domain.
This operation returns all the domain names registered with Amazon Route 53.
This operation returns information about all of the operations that return an operation ID.
This operation returns all of the tags that are associated with the specified domain.
This operation registers a domain.
Rejects the transfer of a domain from another AWS account to the current AWS account.
This operation renews a domain for the specified number of years.
For operations that require confirmation that the email address for the registrant contact is valid.
This operation returns the AuthCode for the domain.
Transfers a domain from another registrar to Amazon Route 53.
Transfers a domain from the current AWS account to another AWS account.
This operation updates the contact information for a particular domain.
This operation updates the specified domain contact’s privacy setting.
This operation replaces the current set of name servers for the domain.
This operation adds or updates tags for a specified domain.
Returns all the domain-related billing records for the current AWS account.

Examples

## Not run:
svc <- route53domains()
svc$accept_domain_transfer_from_another_aws_account(
  Foo = 123
)
Amazon Route 53 Resolver

Description

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

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

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

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

Conditionally forward queries from a VPC to resolvers on your network

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

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

Usage

```
route53resolver(config = list())
```

Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

Operations

- `associate_resolver_endpoint_ip_address`: Adds IP addresses to an inbound or an outbound Resolver endpoint
- `associate_resolver_query_log_config`: Associates an Amazon VPC with a specified query logging configuration
- `associate_resolver_rule`: Associates a Resolver rule with a VPC
- `create_resolver_endpoint`: Creates a Resolver endpoint
- `create_resolver_query_log_config`: Creates a Resolver query logging configuration, which defines where you want Resolver to save DNS query logs that originate in your VPCs
- `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_query_log_config`: Deletes a query logging configuration
- `delete_resolver_rule`: Deletes a Resolver rule
- `disassociate_resolver_endpoint_ip_address`: Removes IP addresses from an inbound or an outbound Resolver endpoint
- `disassociate_resolver_query_log_config`: Disassociates a VPC from a query logging configuration
- `disassociate_resolver_rule`: Removes the association between a specified Resolver rule and a specified VPC
- `get_resolver_dnssec_config`: Gets DNSSEC validation information for a specified resource
- `get_resolver_endpoint`: Gets information about a specified Resolver endpoint, such as whether it’s an inbound or an outbound endpoint
- `get_resolver_query_log_config`: Gets information about a specified Resolver query logging configuration, such as the location that logs are sent to
- `get_resolver_query_log_config_association`: Gets information about a specified association between a Resolver query logging configuration and an Amazon VPC
- `get_resolver_query_log_config_policy`: Gets information about the Resolver query logging policy for a specified configuration
- `get_resolver_rule`: Gets information about a specified Resolver rule, such as the domain name that VPCs use to communicate with the rule
- `get_resolver_rule_association`: Gets information about an association between a specified Resolver rule and a VPC
- `get_resolver_rule_policy`: Gets information about the Resolver rule policy for a specified rule
- `list_resolver_dnssec_configs`: Lists the configurations for DNSSEC validation that are associated with the current AWS account
- `list_resolver_endpoint_ip_addresses`: Gets the IP addresses for a specified Resolver endpoint
- `list_resolver_endpoints`: Lists all the Resolver endpoints that were created using the current AWS account
- `list_resolver_query_log_config_associations`: Lists information about associations between Amazon VPCs and query logging configurations
### Examples

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

---

## S3

### Description

Amazon Simple Storage Service

### Usage

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

### Arguments

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```
svc <- 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 S3 bucket
- `create_multipart_upload` This operation initiates a multipart upload and returns an upload ID
- `delete_bucket` Deletes the S3 bucket
- `delete_bucket_analytics_configuration` Deletes an analytics configuration for the bucket (specified by the analytics configuration ID)
- `delete_bucket_cors` This implementation of the DELETE operation removes default encryption
- `delete_bucket_encryption` Deletes the S3 Intelligent-Tiering configuration from the specified bucket
- `delete_bucket_intelligent_tiering_configuration` Deletes an inventory configuration (identified by the inventory ID) from the bucket
- `delete_bucket_inventory_configuration` Deletes the lifecycle configuration from the specified bucket
- `delete_bucket_metrics_configuration` Deletes a metrics configuration for the Amazon CloudWatch request metrics (specified by the metrics configuration ID) from the bucket
- `delete_bucket_ownership_controls` Removes OwnershipControls for an Amazon S3 bucket
- `delete_bucket_policy` This implementation of the DELETE operation uses the policy subresource to delete the policy of a specified bucket
- `delete_bucket_replication` Deletes the replication configuration from the bucket
- `delete_bucket_tagging` Deletes the tags from the bucket
- `delete_bucket_website` This operation removes the website configuration for a bucket
- `delete_file` Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object
- `delete_objects` This operation enables you to delete multiple objects from a bucket using a single HTTP request
- `delete_object` Removes the PublicAccessBlock configuration for an Amazon S3 bucket
- `download_file` Download a file from S3 and store it at a specified file location
- `get_bucket_accelerate_configuration` This implementation of the GET operation uses the accelerate subresource to return the Transfer Acceleration state of a bucket, which is either Enabled or Suspended
- `get_bucket_acl` This implementation of the GET operation uses the acl subresource to return the access control list (ACL) of a bucket
- `get_bucket_analytics_configuration` Returns an analytics configuration (identified by the analytics configuration ID) from the bucket
- `get_bucket_cors` Returns the cors configuration information set for the bucket
- `get_bucket_encryption` Returns the default encryption configuration for an Amazon S3 bucket
- `get_bucket_intelligent_tiering_configuration` Gets the S3 Intelligent-Tiering configuration from the specified bucket
- `get_bucket_inventory_configuration` Returns an inventory configuration (identified by the inventory configuration ID)
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_ownership_controls
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_intelligent_tiering_configurations
list_bucket_inventory_configurations
list_bucket_metrics_configurations
list_buckets
list_multipart_uploads
list_objects
list_objects_v2
list_object_versions
list_parts
put_bucket_accelerate_configuration
put_bucket_acl
put_bucket_analytics_configuration
put_bucket_cors
put_bucket_encryption
put_bucket_intelligent_tiering_configuration
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

For an updated version of this API, see GetBucketLifecycleConfiguration
Bucket lifecycle configuration now supports specifying a lifecycle rule using:
Returns the Region the bucket resides in
Returns the logging status of a bucket and the permissions users have to view and modify the logging configuration
Gets a metrics configuration (specified by the metrics configuration ID) from a bucket
No longer used, see GetBucketNotificationConfiguration
Returns the notification configuration of a bucket
Retrieves OwnershipControls for an Amazon S3 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
Returns 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 content
Lists the analytics configurations for the bucket
Lists the S3 Intelligent-Tiering configuration from the specified bucket
Retains 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
Returns metadata about all versions of the 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 operation uses the encryption subresource to configure default encryption
Puts a S3 Intelligent-Tiering configuration to the specified bucket
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 configuration
Set the logging parameters for a bucket and to specify permissions for who can view and modify the logging configuration
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
PutBucketOwnershipControls
PutBucketPolicy
PutBucketReplication
PutBucketRequestPayment
PutBucketTagging
PutBucketVersioning
PutBucketWebsite
PutObject
PutObjectACL
PutObjectLegalHold
PutObjectLockConfiguration
PutObjectRetention
PutObjectTagging
PutPublicAccessBlock
RestoreObject
SelectObjectContent
UploadPart
UploadPartCopy

Examples

## Not run:
svc <- s3()
# The following example aborts a multipart upload.
svc$abort_multipart_upload(
  Bucket = "examplebucket",
  Key = "bigobject",
  UploadId = "xadcOB_7YPB0Ju0FiQ9cz4P3Pe6F1Zw04f7wN93uHsNBEw97pl5eNwzExg0LA..."
)

## End(Not run)

s3control  AWS S3 Control

Description

AWS S3 Control provides access to Amazon S3 control plane operations.

Usage

s3control(config = list())

Arguments

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

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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_bucket` This API operation creates an Amazon S3 on Outposts bucket
- `create_job` S3 Batch Operations performs large-scale Batch Operations on Amazon S3 objects
- `delete_access_point` Deletes the specified access point
- `delete_access_point_policy` Deletes the access point policy for the specified access point
- `delete_bucket` This API operation deletes an Amazon S3 on Outposts bucket
- `delete_bucket_lifecycle_configuration` This API action deletes an Amazon S3 on Outposts bucket’s lifecycle configuration
- `delete_bucket_policy` This API operation deletes an Amazon S3 on Outposts bucket policy
- `delete_bucket_tagging` This operation deletes an Amazon S3 on Outposts bucket’s tags
- `delete_job_tagging` Removes the entire tag set from the specified S3 Batch Operations job
- `delete_public_access_block` Removes the PublicAccessBlock configuration for an AWS account
- `delete_storage_lens_configuration` Deletes the Amazon S3 Storage Lens configuration
- `delete_storage_lens_configuration_tagging` Deletes the Amazon S3 Storage Lens configuration tags
- `describe_job` Retrieves the configuration parameters and status for a Batch Operations job
- `get_access_point` Returns configuration information about the specified access point
- `get_access_point_policy` Returns the access point policy associated with the specified access point
- `get_access_point_policy_status` Indicates whether the specified access point currently has a policy that allows public access
- `get_bucket` Gets an Amazon S3 on Outposts bucket
- `get_bucket_lifecycle_configuration` This operation gets an Amazon S3 on Outposts bucket’s lifecycle configuration
- `get_bucket_policy` This action gets a bucket policy for an Amazon S3 on Outposts bucket
- `get_bucket_tagging` This operation gets an Amazon S3 on Outposts bucket’s tags
- `get_job_tagging` Returns the tags on an S3 Batch Operations job
- `get_public_access_block` Retrieves the PublicAccessBlock configuration for an AWS account
- `get_storage_lens_configuration` Gets the Amazon S3 Storage Lens configuration
### Amazon SageMaker Service

**Description**

Provides APIs for creating and managing Amazon SageMaker resources.

Other Resources:

- Amazon SageMaker Developer Guide
- Amazon Augmented AI Runtime API Reference

**Usage**

```r
sagemaker(config = list())
```

**Arguments**

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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_association`: Creates an association between the source and the destination
- `add_tags`: Adds or overwrites one or more tags for the specified Amazon SageMaker resource
- `associate_trial_component`: Associates a trial component with a trial
- `create_action`: Creates an action
- `create_algorithm`: Create a machine learning algorithm that you can use in Amazon SageMaker and list in the AWS Marketplace
- `create_app`: Creates a running App for the specified UserProfile
- `create_app_image_config`: Creates a configuration for running a SageMaker image as a KernelGateway app
- `create_artifact`: Creates an artifact
- `create_auto_ml_job`: Creates an Autopilot job
- `create_code_repository`: Creates a Git repository as a resource in your Amazon SageMaker account
- `create_compilation_job`: Starts a model compilation job
- `create_context`: Creates a context
- `create_data_quality_job_definition`: Creates a definition for a job that monitors data quality and drift
- `create_device_fleet`: Creates a device fleet
- `create_domain`: Creates a Domain used by Amazon SageMaker Studio
- `create_edge_packaging_job`: Starts a SageMaker Edge Manager model packaging job
- `create_endpoint`: Creates an endpoint using the endpoint configuration specified in the request
- `create_endpoint_config`: Creates an endpoint configuration that Amazon SageMaker hosting services uses to deploy models
- `create_experiment`: Creates a new FeatureGroup
- `create_flow_definition`: Creates a flow definition
- `create_human_task_ui`: Defines the settings you will use for the human review workflow user interface
- `create_hyper_parameter_tuning_job`: Starts a hyperparameter tuning job
- `create_image`: Creates a custom SageMaker image
create_image_version
create_labeling_job
create_model
create_model_bias_job_definition
create_model_explainability_job_definition
create_model_package
create_model_package_group
create_model_quality_job_definition
create_monitoring_schedule
create_notebook_instance
create_notebook_instance_lifecycle_config
create_pipeline
create_presigned_domain_url
create_presigned_notebook_instance_url
create_processing_job
create_project
create_training_job
create_transform_job
create_trial
create_trial_component
create_user_profile
create_workforce
create_workteam
delete_action
delete_algorithm
delete_app
delete_app_image_config
delete_artifact
delete_association
delete_code_repository
delete_context
delete_data_quality_job_definition
delete_device_fleet
delete_domain
delete_endpoint
delete_endpoint_config
delete_experiment
delete_feature_group
delete_flow_definition
delete_human_task_ui
delete_image
delete_image_version
delete_model
delete_model_bias_job_definition
delete_model_explainability_job_definition
delete_model_package
delete_model_package_group
delete_model_package_group_policy

create_image_version
Creates a version of the SageMaker image specified by ImageName
create_labeling_job
Creates a job that uses workers to label the data objects in your input data
create_model
Creates a model in Amazon SageMaker
create_model_bias_job_definition
Creates the definition for a model bias job
create_model_explainability_job_definition
Creates the definition for a model explainability job
create_model_package
Creates a model package that you can use to create Amazon SageMaker models
create_model_package_group
Creates a model group
create_model_quality_job_definition
Creates a definition for a job that monitors model quality and drift
create_monitoring_schedule
Creates a schedule that regularly starts Amazon SageMaker Processing Jobs
create_notebook_instance
Creates an Amazon SageMaker notebook instance
create_notebook_instance_lifecycle_config
Creates a lifecycle configuration that you can associate with a notebook instance
create_pipeline
Creates a pipeline using a JSON pipeline definition
create_presigned_domain_url
Creates a URL for a specified UserProfile in a Domain
create_presigned_notebook_instance_url
Returns a URL that you can use to connect to the Jupyter server from a notebook instance
create_processing_job
Creates a processing job
create_project
Creates a machine learning (ML) project that can contain one or more templates that set up an ML pipeline from training to deploying an approved model
create_training_job
Starts a model training job
create_transform_job
Starts a transform job
create_trial
Creates an Amazon SageMaker trial
create_trial_component
Creates a trial component, which is a stage of a machine learning trial
create_user_profile
Creates a user profile
create_workforce
Use this operation to create a workforce
create_workteam
Creates a new work team for labeling your data
delete_action
Deletes an action
delete_algorithm
Removes the specified algorithm from your account
delete_app
Used to stop and delete an app
delete_app_image_config
Deletes an AppImageConfig
delete_artifact
Deletes an artifact
delete_association
Deletes an association
delete_code_repository
Deletes the specified Git repository from your account
delete_context
Deletes a context
delete_data_quality_job_definition
Deletes a data quality monitoring job definition
delete_device_fleet
Deletes a fleet
delete_domain
Used to delete a domain
delete_endpoint
Deletes an endpoint
delete_endpoint_config
Deletes an endpoint configuration
delete_experiment
Deletes an Amazon SageMaker experiment
delete_feature_group
Delete the FeatureGroup and any data that was written to the OnlineStore
delete_flow_definition
Deletes the specified flow definition
delete_human_task_ui
Use this operation to delete a human task user interface (worker task template)
delete_image
Deletes a SageMaker image and all versions of the image
delete_image_version
Deletes a version of a SageMaker image
delete_model
Deletes a model
delete_model_bias_job_definition
Deletes an Amazon SageMaker model bias job definition
delete_model_explainability_job_definition
Deletes an Amazon SageMaker model explainability job definition
delete_model_package
Deletes a model package
delete_model_package_group
Deletes the specified model group
delete_model_package_group_policy
Deletes a model group resource policy
delete_model_quality_job_definition
delete_monitoring_schedule
delete_notebook_instance
delete_notebook_instance_lifecycle_config
delete_pipeline
delete_project
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workforce
delete_workteam
deregister_devices
describe_action
describe_algorithm
describe_app
describe_app_image_config
describe_artifact
describe_auto_ml_job
describe_code_repository
describe_compilation_job
describe_context
describe_data_quality_job_definition
describe_device
describe_device_fleet
describe_domain
describe_edge_packaging_job
describe_endpoint
describe_endpoint_config
describe_experiment
describe_feature_group
describe_flow_definition
describe_human_task_ui
describe_hyper_parameter_tuning_job
describe_image
describe_image_version
describe_labeling_job
describe_model
describe_model_bias_job_definition
describe_model_explainability_job_definition
describe_model_package
describe_model_package_group
describe_model_quality_job_definition
describe_monitoring_schedule
describe_notebook_instance
describe_notebook_instance_lifecycle_config
describe_pipeline
describe_pipeline_definition_for_execution

Deletes the specified model quality monitoring job definition
Deletes a monitoring schedule
Deletes an Amazon SageMaker notebook instance
Deletes a notebook instance lifecycle configuration
Deletes a pipeline if there are no in-progress executions
Delete the specified project
Deletes the specified tags from an Amazon SageMaker resource
Deletes the specified trial
Deletes the specified trial component
Deletes a user profile
Use this operation to delete a workforce
Deletes an existing work team
Deregisters the specified devices
Describes an action
Returns a description of the specified algorithm that is in your account
Describes the app
Describes an AppImageConfig
Describes an artifact
Returns information about an Amazon SageMaker job
Gets details about the specified Git repository
Returns information about a model compilation job
Describes a context
Gets the details of a data quality monitoring job definition
Describes the device
A description of the fleet the device belongs to
The description of the domain
A description of edge packaging jobs
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
Use this operation to describe a FeatureGroup
Returns information about the specified flow definition
Returns information about the requested human task user interface (worker)
Ggets a description of a hyperparameter tuning job
Describes a SageMaker image
Describes a version of a SageMaker image
Gets information about a labeling job
Describes a model that you created using the CreateModel API
Returns a description of a model bias job definition
Returns a description of a model explainability job definition
Returns a description of the specified model package, which is used to create models
Gets a description for the specified model group
Returns a description of a model quality job definition
Describes the schedule for a monitoring job
Returns information about a notebook instance
Returns a description of a notebook instance lifecycle configuration
Describes the details of a pipeline
Describes the details of an execution’s pipeline definition
describe_pipeline_execution
describe_processing_job
describe_project
describe_subscribed_workteam
describe_training_job
describe_transform_job
describe_trial
describe_trial_component
describe_user_profile
describe_workforce
disable_sagemaker_servicecatalog_portfolio
disassociate_trial_component
enable_sagemaker_servicecatalog_portfolio
get_device_fleet_report
get_model_package_group_policy
get_sagemaker_servicecatalog_portfolio_status
get_search_suggestions
list_actions
list_algorithms
list_app_image_configs
list_apps
list_artifacts
list_associations
list_auto_ml_jobs
list_candidates_for_auto_ml_job
list_code_repositories
list_compilation_jobs
list_contexts
list_data_quality_job_definitions
list_device_fleets
list_devices
list_domains
list_edge_packaging_jobs
list_endpoint_configs
list_endpoints
list_experiments
list_feature_groups
list_flow_definitions
list_human_task_uis
list_hyper_parameter_tuning_jobs
list_images
list_image_versions
list_labeling_jobs
list_labeling_jobs_for_workteam
list_model_bias_job_definitions
list_model_explainability_job_definitions
list_model_package_groups

Describes the details of a pipeline execution
Returns a description of a processing job
Describes the details of a project
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 a user profile
Lists private workforce information, including workforce name, Amazon Resource Name (ARN), and, if applicable, allowed IP address ranges (CIDRs)
Gets information about a specific work team
Disassociates a trial component from a trial
Enables using Service Catalog in SageMaker
Describes a fleet
Gets a resource policy that manages access for a model group
Gets the status of Service Catalog in SageMaker
An auto-complete API for the search functionality in the Amazon SageMaker console
Lists the actions in your account and their properties
Lists the machine learning algorithms that have been created
Lists the AppImageConfigs in your account and their properties
Lists apps
Lists the artifacts in your account and their properties
Lists the associations in your account and their properties
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 contexts in your account and their properties
Lists the data quality job definitions in your account
Returns a list of devices in the fleet
A list of devices
Lists the domains
Returns a list of edge packaging jobs
Lists endpoint configurations
Lists endpoints
Lists all the experiments in your account
List FeatureGroups based on given filter and order
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
Lists the images in your account and their properties
Lists the versions of a specified image and their properties
Gets a list of labeling jobs
Gets a list of labeling jobs assigned to a specified work team
Lists model bias jobs definitions that satisfy various filters
Lists model explainability job definitions that satisfy various filters
Gets a list of the model groups in your AWS account
list_model_packages
list_model_quality_job_definitions
list_models
list_monitoring_executions
list_monitoring_schedules
list_notebook_instance_lifecycle_configs
list_notebook_instances
list_pipeline_executions
list_pipeline_execution_steps
list_pipeline_parameters_for_execution
list_pipelines
list_processing_jobs
list_projects
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_workforces
list_workteams
put_model_package_group_policy
register_devices
render_ui_template
search
start_monitoring_schedule
start_notebook_instance
start_pipeline_execution
stop_auto_ml_job
stop_compilation_job
stop_edge_packaging_job
stop_hyper_parameter_tuning_job
stop_labeling_job
stop_monitoring_schedule
stop_notebook_instance
stop_pipeline_execution
stop_processing_job
stop_training_job
stop_transform_job
update_action
update_app_image_config
update_artifact
update_code_repository
update_context
update_device_fleet
update_devices

Lists the model packages that have been created
Gets a list of model quality monitoring job definitions in your account
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 requester's account
Gets a list of the pipeline executions
Gets a list of PipelineExecutionStep objects
Gets a list of parameters for a pipeline execution
Gets a list of pipelines
Lists processing jobs that satisfy various filters
Gets a list of the projects in an AWS account
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 launched
Lists transform jobs
Lists the trial components in your account
Lists the trials in your account
Lists user profiles
Use this operation to list all private and vendor workforces in an AWS Region
Gets a list of private work teams that you have defined in a region
Adds a resource policy to control access to a model group
Register devices
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 and attaches your ML storage volume
Starts a pipeline execution
A method for forcing the termination of a running job
Stops a model compilation job
Request to stop an edge packaging 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 pipeline execution
Stops a processing job
Stops a training job
Stops a transform job
Updates an action
Updates the properties of an AppImageConfig
Updates an artifact
Updates the specified Git repository with the specified values
Updates a context
Updates a fleet of devices
Updates one or more devices in a fleet
update_domain
update_endpoint
update_endpoint_weights_and_capacities
update_experiment
update_image
update_model_package
update_monitoring_schedule
update_notebook_instance
update_notebook_instance_lifecycle_config
update_pipeline
update_pipeline_execution
update_training_job
update_trial
update_trial_component
update_user_profile
update_workforce
update_workteam

Updates the default settings for new user profiles in the domain
Deploys the new EndpointConfig specified in the request, switches to using newly created endpoint, and then deletes resources provisioned for the endpoint using the previous EndpointConfig (there is no availability loss)
Updates variant weight of one or more variants associated with an existing endpoint
Adds, updates, or removes the description of an experiment
Updates the properties of a SageMaker image
Updates a versioned model
Updates a previously created schedule
Updates a notebook instance
Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
Updates a pipeline
Updates a pipeline execution
Update a model training job to request a new Debugger profiling configuration
Updates the display name of a trial
Updates one or more properties of a trial component
Updates a user profile
Use this operation to update your workforce
Updates an existing work team with new member definitions or descriptions

Examples

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

---

**sagemakerruntime**

*Amazon SageMaker Runtime*

**Description**

The Amazon SageMaker runtime API.

**Usage**

`sagemakerruntime(config = list())`

**Arguments**

- **config**
  
  Optional configuration of credentials, endpoint, and/or region.
Value
A client for the service. You can call the service’s operations using syntax like `svc(operation(...))`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

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

Description

AWS Secrets Manager API Reference

AWS Secrets Manager provides a service to enable 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, 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 provide cryptographically signing requests, managing errors, and retrying requests automatically. For more information about the AWS SDKs, including downloading and installing them, see Tools for Amazon Web Services.

We recommend 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 API supports GET and POST requests for all actions, and 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 the service returns as a response to HTTP query requests contain single, long strings without line breaks or white space formatting. The JSON shown in the examples displays the code formatted with both line breaks and white space to improve readability. When example input parameters can also cause long strings extending beyond the screen, you can 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 the requests successfully made to Secrets Manager, who made the request, when it was made, and so on. For more about AWS Secrets Manager and 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 enabling it 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.
Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cancel_rotate_secret</code></td>
<td>Disables automatic scheduled rotation and cancels the rotation of a secret if currently in progress</td>
</tr>
<tr>
<td><code>create_secret</code></td>
<td>Creates a new secret</td>
</tr>
<tr>
<td><code>delete_resource_policy</code></td>
<td>Deletes the resource-based permission policy attached to the secret</td>
</tr>
<tr>
<td><code>delete_secret</code></td>
<td>Deletes an entire secret and all of its versions</td>
</tr>
<tr>
<td><code>describe_secret</code></td>
<td>Retrieves the details of a secret</td>
</tr>
<tr>
<td><code>get_random_password</code></td>
<td>Generates a random password of the specified complexity</td>
</tr>
<tr>
<td><code>get_resource_policy</code></td>
<td>Retrieves the JSON text of the resource-based policy document attached to the specified secret</td>
</tr>
<tr>
<td><code>get_secret_value</code></td>
<td>Retrieves the contents of the encrypted fields SecretString or SecretBinary from the specified version</td>
</tr>
<tr>
<td><code>list_secrets</code></td>
<td>Lists all of the secrets that are stored by Secrets Manager in the AWS account</td>
</tr>
<tr>
<td><code>list_secret_version_ids</code></td>
<td>Lists all of the versions attached to the specified secret</td>
</tr>
<tr>
<td><code>put_resource_policy</code></td>
<td>Attaches the contents of the specified resource-based permission policy to a secret</td>
</tr>
<tr>
<td><code>put_secret_value</code></td>
<td>Stores a new encrypted secret value in the specified secret</td>
</tr>
<tr>
<td><code>restore_secret</code></td>
<td>Cancels the scheduled deletion of a secret by removing the DeletedDate time stamp</td>
</tr>
<tr>
<td><code>rotate_secret</code></td>
<td>Configures and starts the asynchronous process of rotating this secret</td>
</tr>
<tr>
<td><code>tag_resource</code></td>
<td>Attaches one or more tags, each consisting of a key name and a value, to the specified secret</td>
</tr>
<tr>
<td><code>untag_resource</code></td>
<td>Removes one or more tags from the specified secret</td>
</tr>
<tr>
<td><code>update_secret</code></td>
<td>Modifies many of the details of the specified secret</td>
</tr>
<tr>
<td><code>update_secret_version_stage</code></td>
<td>Modifies the staging labels attached to a version of a secret</td>
</tr>
<tr>
<td><code>validate_resource_policy</code></td>
<td>Validates the JSON text of the resource-based policy document attached to the specified secret</td>
</tr>
</tbody>
</table>
Examples

```r
## Not run:
svc <- secretsmanager()
# 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.
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 readiness status of your environment based on controls from supported security standards. 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](https://docs.aws.amazon.com/securityhub/latest/APIReference/).

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 `create_members` 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 invitation was sent from.

The following throttling limits apply to using Security Hub API operations.

- **batch_enable_standards** - RateLimit of 1 request per second, BurstLimit of 1 request per second.
- **get_findings** - RateLimit of 3 requests per second, BurstLimit of 6 requests per second.
- **update_findings** - RateLimit of 1 request per second, BurstLimit of 5 requests per second.
- **update_standards_control** - RateLimit of 1 request per second, BurstLimit of 5 requests per second.
- All other operations - RateLimit of 10 requests per second, BurstLimit of 30 requests per second.
Usage

securityhub(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- 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 that the invitation was sent from
- **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
- **batch_update_findings**: Used by Security Hub customers to update information about their investigation into a finding
- **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_organization_configuration**: Returns information about the Organizations configuration for Security Hub
describe_products
describe_standards
describe_standards_controls
disable_import_findings_for_product
disable_organization_admin_account
disable_security_hub
disassociate_from_master_account
disassociate_members
enable_import_findings_for_product
enable_organization_admin_account
enable_security_hub
get_enabled_standards
get_findings
get_insight_results
get_insights
get_invitations_count
get_master_account
get_members
invite_members
list_enabled_products_for_import
list_invitations
list_members
list_organization_admin_accounts
list_tags_for_resource
tag_resource
untag_resource
update_action_target
update_findings
update_insight
update_organization_configuration
update_security_hub_configuration
update_standards_control

Returns information about the available products that you can subscribe to and integrate with Security Hub
Returns a list of the available standards in Security Hub
Returns a list of security standards controls
Disables the integration of the specified product with Security Hub
Disables Security Hub in your account only in the current Region
Disassociates the current Security Hub member account from the associated master account
Disassociates the specified member accounts from the associated master account
Enables the integration of a partner product with Security Hub
Designates the Security Hub administrator account for an organization
Enables Security Hub for your account in the current Region or the Region you specify
Returns a list of the standards that are currently enabled
Returns a list of findings that match the specified criteria
Lists the results of the Security Hub insight specified by the insight ARN
Lists and describes insights for the specified insight ARNs
Returns the count of all Security Hub membership invitations that were sent to the current member account
Provides the details for the Security Hub master account for the current member account
Returns the details for the Security Hub member accounts for the specified account ID
Invites other AWS accounts to become member accounts for the Security Hub master account
Lists all findings-generating solutions (products) that you are subscribed to receive findings from
Lists all Security Hub membership invitations that were sent to the current AWS account
Lists details about all member accounts for the current Security Hub master account
Lists the Security Hub administrator accounts
Returns a list of tags associated with a resource
Adds one or more tags to a resource
Removes one or more tags from a resource
Updates the name and description of a custom action target in Security Hub
UpdateFindings is deprecated
Updates the Security Hub insight identified by the specified insight ARN
Used to update the configuration related to Organizations
Updates configuration options for Security Hub
Used to control whether an individual security standard control is enabled or disabled

Examples

```r
## Not run:
svc <- securityhub()
svc$accept_invitation(
  Foo = 123
)

## End(Not run)
```
serverlessapplicationrepository

AWS Serverless Application Repository

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

serverlessapplicationrepository(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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
- `unshare_application` Unshares an application from an AWS Organization
- `update_application` Updates the specified application

Examples

```r
## Not run:
svc <- serverlessapplicationrepository()
svc$create_application(
  Foo = 123
)
## End(Not run)
```
AWS Service Catalog enables organizations to create and manage catalogs of IT services that are approved for AWS. To get the most out of this documentation, you should be familiar with the terminology discussed in AWS Service Catalog Concepts.

Usage

```python
servicecatalog(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
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`: Associates the specified product with the specified portfolio
- `associate_service_action_with_provisioning_artifact`: Associates a self-service action with a provisioning artifact
Associate the specified TagOption with the specified portfolio or product.
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 or a new product.
Creates a constraint.
Creates a portfolio.
Shares the specified portfolio with the specified account or organization node.
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.
Stops sharing the specified portfolio with the specified account or organization node.
Deletes the specified product.
Deletes the specified plan.
Deletes the specified provisioning artifact (also known as a version) for the specified product.
Deletes a self-service action.
Deletes the specified TagOption.
Gets information about the specified constraint.
Gets the status of the specified copy product operation.
Gets information about the specified portfolio.
Returns a summary of each of the portfolio shares that were created for the specified portfolio.
Gets the status of the specified portfolio share operation.
Gets information about the specified product.
Gets information about the specified product configuration.
Gets information about the specified product provisioning artifacts.
Gets information about the specified portfolio.
Gets information about the specified portfolio share.
Gets information about the specified resource.
Gets the status of the specified resource.
Gets information about the specified request operation.
Describes a self-service action.
Finds the default parameters for a specific self-service action.
Gets information about the specified TagOption.
Disables portfolio sharing through AWS Organizations feature.
Disassociates the specified budget from the specified resource.
Disassociates a previously associated principal ARN from a specified portfolio.
Disassociates the specified product from the specified portfolio.
Disassociates the specified self-service action association from a specified provisioning artifact.
Disassociates the specified TagOption from the specified resource.
Enables portfolio sharing feature through AWS Organizations.
Provisions or modifies a product based on the resource changes.
Executes a self-service action against a provisioned product.
Get the Access Status for AWS Organization portfolio share configuration.
This API takes either a ProvisionedProductId or a ProvisionedProductName.
Requests the import of a ProvisionedProductId or a ProvisionedProductName.
servicecatalog

list_accepted_portfolio_shares
list_budgets_for_resource
list_constraints_for_portfolio
list_launch_paths
list_organization_portfolio_access
list_portfolio_access
list_portfolios
list_portfolios_for_product
list_principals_for_portfolio
list_provisioned_product_plans
list_provisioning_artifacts
list_provisioning_artifacts_for_service_action
list_record_history
list_resources_for_tag_option
list_service_actions
list_service_actions_for_provisioning_artifact
list_stack_instances_for_provisioned_product
list_tag_options
provision_product
reject_portfolio_share
scan_provisioned_products
search_products
search_products_as_admin
search_provisioned_products
terminate_provisioned_product
update_constraint
update_portfolio
update_portfolio_share
update_product
update_provisioned_product
update_provisioned_product_properties
update_provisioning_artifact
update_service_action
update_tag_option

Lists all portfolios for which sharing was accepted by this account
Lists all the budgets associated to the specified resource
Lists the constraints for the specified portfolio and product
Lists the paths to the specified product
Lists the organization nodes that have access to the specified portfolio
Lists the account IDs that have access to the specified portfolio
Lists all portfolios in the catalog
Lists all portfolios that the specified product is associated with
Lists all principal ARNs associated with the specified portfolio
Lists the plans for the specified provisioned product or all plans
Lists all provisioning artifacts (also known as versions) for the specified product
Lists the specified requests or all performed requests
Lists the resources associated with the specified TagOption
Lists all self-service actions
Returns a paginated list of self-service actions associated with the specified product
Returns summary information about stack instances that are associated with the specified provisioned products
Lists the specified TagOptions or all TagOptions
Provisions the specified product
Rejects an offer to share the specified portfolio
Lists the provisioned products that are available (not terminated)
Gets information about the products to which the caller has access
Gets information about the products for the specified portfolio
Gets information about the provisioned products that meet the specified criteria
Terminates the specified provisioned product
Updates the specified constraint
Updates the specified portfolio
Updates the specified portfolio share
Updates the specified product
Requests updates to the configuration of the specified provisioned product
Requests updates to the properties of the specified provisioned product
Updates the specified provisioning artifact (also known as a version)
Updates a self-service action
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.

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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 specific Amazon VPC.
create_public_dns_namespace
create_service
delete_namespace
delete_service
deregister_instance
discover_instances
get_instance
get_instances_health_status
get_namespace
get_operation
get_service
list_instances
list_namespaces
list_operations
list_services
list_tags_for_resource
register_instance
tag_resource
untag_resource
update_instance_custom_health_status
update_service

creates a public namespace based on DNS, which will be visible on the internet

create_service

creates a service, which defines the configuration for the following entities:

delete_namespace

deletes a namespace from the current account

delete_service

deletes the Amazon Route 53 DNS records and health check, if any, that AWS Cloud Map created for the specified namespace and service

deregister_instance

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 that are associated with a specified service

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

list_instances

lists summary information about the instances that you registered by using a specified service

list_namespaces

lists summary information about the namespaces that were created by the current AWS account

list_operations

lists operations that match the criteria that you specify

list_services

lists summary information for all the services that are associated with one or more specified namespaces

list_tags_for_resource

lists tags for the specified resource

register_instance

creates or updates one or more records and, optionally, creates a health check based on the settings in a specified service

tag_resource

adds one or more tags to the specified resource

untag_resource

removes one or more tags from the specified resource

update_instance_custom_health_status

submits a request to change the health status of a custom health check to healthy or unhealthy

update_service

submits a request to perform the following operations:

Examples

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

---

**Service Quotas**

**Description**

With Service Quotas, you can view and manage your quotas easily as your AWS workloads grow. Quotas, also referred to as limits, are the maximum number of resources that you can create in your AWS account. For more information, see the [Service Quotas User Guide](#).
Usage

```r
servicequotas(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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 your quota request template with your organization
- `delete_service_quota_increase_request_from_template`: Deletes the quota increase request for the specified quota from your organization
- `disassociate_service_quota_template`: Disables your quota request template
- `get_association_for_service_quota_template`: Retrieves the status of the association for the quota request template
- `get_aws_default_service_quota`: Retrieves the default value for the specified quota
- `get_requested_service_quota_change`: Retrieves information about the specified quota increase request
- `get_service_quota`: Retrieves the applied quota value for the specified quota
- `get_service_quota_increase_request_from_template`: Retrieves information about the specified quota increase request in your quota request template
- `list_aws_default_service_quotas`: Lists the default values for the quotas for the specified AWS service
- `list_requested_service_quota_change_history`: Retrieves the quota increase requests for the specified service
- `list_requested_service_quota_change_history_by_quota`: Retrieves the quota increase requests for the specified quota
- `list_service_quota_increase_requests_in_template`: Lists the quota increase requests in the specified quota request template
- `list_service_quotas`: Lists the applied quota values for the specified AWS service
- `list_services`: Lists the names and codes for the services integrated with Service Quotas
- `list_tags_for_resource`: Returns a list of the tags assigned to the specified applied quota
- `put_service_quota_increase_request_into_template`: Adds a quota increase request to your quota request template
**request_service_quota_increase**  
Submits a quota increase request for the specified quota

**tag_resource**  
Adds tags to the specified applied quota

**untag_resource**  
Removes tags from the specified applied quota

### Examples

```r
## Not run:
svc <- servicequotas()
svc$associate_service_quota_template(
  Foo = 123
)

## End(Not run)
```

---

**ses**  
*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

```r
ses(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>

## Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```
svc <- 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 for open and click event tracking
- `create_custom_verification_email_template`: Creates a 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_custom_verification_email_template`: Deletes an existing custom verification email template
- `delete_identity`: Deletes the specified identity (an email address or a domain) from the list of verified identities
- `delete_identity_policy`: Deletes the specified sending authorization policy for the given identity (an email address or a domain)
- `delete_receipt_filter`: Deletes the specified IP address filter
- `delete_receipt_rule`: Deletes the specified receipt rule
- `delete_receipt_rule_set`: Deletes the specified receipt rule set and all of the receipt rules it contains
- `delete_template`: Deletes an email template
- `delete_verified_email_address`: Deprecated
- `describe_active_receipt_rule_set`: Returns the metadata and receipt rules for the receipt rule set that is currently active
- `describe_configuration_set`: Returns the details of the specified configuration set
- `describe_receipt_rule`: Returns the details of the specified receipt rule
- `describe_receipt_rule_set`: Returns the details of the specified receipt rule set
- `get_account_sending_enabled`: Returns the email sending status of the Amazon SES account for the current region
- `get_custom_verification_email_template`: Returns the custom email verification template for the template name you specify
- `get_identity_dkim_attributes`: Returns the current status of Easy DKIM signing for an entity
- `get_identity_mail_from_domain_attributes`: Returns the custom MAIL FROM attributes for a list of identities (email addresses and/or domains), respectively
- `get_identity_notification_attributes`: Returns the requested sending authorization policies for the given identities
- `get_identity_policies`: Returns the requested sending authorization policies for the given identities
**Examples**

```r
## Not run:
svc <- ses()

# The following example creates a receipt rule set by cloning an existing
```
# one:
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](https://docs.aws.amazon.com/stepfunctions/latest/dg/).

**Usage**

```r
sfn(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```r
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
create_state_machine
delete_activity
delete_state_machine
describe_activity
describe_execution
describe_state_machine
describe_state_machine_for_execution
describe_activity_task
describe_execution_history
list_activities
list_executions
list_state_machines
list_tags_for_resource
send_task_failure
send_task_heartbeat
send_task_success
start_execution
start_sync_execution
stop_execution
tag_resource
untag_resource
update_state_machine

Examples

## 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](https://docs.aws.amazon.com/waf/latest/developerguide/).

**Usage**

```python
shield(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```python
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"
    )
)
```
Operations

- **associate_drt_log_bucket**: Authorizes the DDoS Response Team (DRT) to access the specified Amazon S3 bucket containing your AWS WAF logs.
- **associate_drt_role**: Authorizes the DDoS Response Team (DRT), using the specified role, to access your AWS account to assist with DDoS attack mitigation during potential attacks.
- **associate_health_check**: Adds health-based detection to the Shield Advanced protection for a resource.
- **associate_proactive_engagement_details**: Initializes proactive engagement and sets the list of contacts for the DDoS Response Team (DRT).
- **create_protection**: Enables AWS Shield Advanced for a specific AWS resource.
- **create_protection_group**: Creates a grouping of protected resources so they can be handled as a collective.
- **create_subscription**: Activates AWS Shield Advanced for an account.
- **delete_protection**: Deletes an AWS Shield Advanced Protection.
- **delete_protection_group**: Removes the specified protection group.
- **delete_subscription**: Removes AWS Shield Advanced from an account.
- **describe_attack**: Describes the details of a DDoS attack.
- **describe_attack_statistics**: Provides information about the number and type of attacks AWS Shield has detected in the last year for all resources that belong to your account.
- **describe_drt_access**: Returns the current role and list of Amazon S3 log buckets used by the DDoS Response Team (DRT) to access your AWS account while assisting with attack mitigation.
- **describe_emergency_contact_settings**: A list of email addresses and phone numbers that the DDoS Response Team (DRT) can use to contact you if you have proactive engagement enabled, for escalations to the DRT and to initiate proactive customer support.
- **describe_protection**: Lists the details of a Protection object.
- **describe_protection_group**: Returns the specification for the specified protection group.
- **describe_subscription**: Provides details about the AWS Shield Advanced subscription for an account.
- **disable_proactive_engagement**: Removes authorization from the DDoS Response Team (DRT) to notify contacts about escalations to the DRT and to initiate proactive customer support.
- **disassociate_drt_log_bucket**: Removes the DDoS Response Team’s (DRT) access to the specified Amazon S3 bucket.
- **disassociate_drt_role**: Removes the DDoS Response Team’s (DRT) access to your AWS account.
- **disassociate_health_check**: Removes health-based detection from the Shield Advanced protection for a resource.
- **enable_proactive_engagement**: Authorizes the DDoS Response Team (DRT) to use email and phone to notify contacts about escalations to the DRT and to initiate proactive customer support.
- **get_subscription_state**: Returns the SubscriptionState, either Active or Inactive.
- **list_attacks**: Returns all ongoing DDoS attacks or all DDoS attacks during a specified time period.
- **list_protection_groups**: Retrieves the ProtectionGroup objects for the account.
- **list_protections**: Lists all Protection objects for the account.
- **list_resources_in_protection_group**: Retrieves the resources that are included in the protection group.
- **update_emergency_contact_settings**: Updates the details of the list of email addresses and phone numbers that the DDoS Response Team (DRT) can use to contact you if you have proactive engagement enabled, for escalations to the DRT and to initiate proactive customer support.
- **update_protection_group**: Updates an existing protection group.
- **update_subscription**: Updates the details of an existing subscription.

Examples

```r
data <- shield()
data$associate_drt_log_bucket(
  Foo = 123
)
```

## End(Not run)
Amazon SimpleDB

Description

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.


Usage

```r
simpledb(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
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.
- **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

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

---

### 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](https://aws.amazon.com/sns). For detailed information about Amazon SNS features and their associated API calls, see the Amazon SNS Developer Guide.

For information on the permissions you need to use this API, see [Identity and access management in Amazon SNS](https://aws.amazon.com/sns) in 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: cryptographic signing your service requests, retrying requests, and handling error responses. For a list of available SDKs, go to [Tools for Amazon Web Services](https://aws.amazon.com/tools/).

### Usage

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

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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 to the endpoint by an earlier Subscribe action.
- `create_platform_application`: Creates a platform application object for one of the supported push notification services, such as APNS and GCM (Firebase Cloud Messaging), to which devices and mobile apps may register.
- `create_platform_endpoint`: Creates an endpoint for a device and mobile app on one of the supported push notification services, such as GCM (Firebase Cloud Messaging) and APNS.
- `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 GCM (Firebase Cloud Messaging).
- `list_subscriptions`: Returns a list of the requester’s subscriptions.
- `list_subscriptions_by_topic`: 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
opt_in_phone_number
publish
remove_permission
set_endpoint_attributes
set_platform_application_attributes
set_sms_attributes
set_subscription_attributes
set_topic_attributes
subscribe
tag_resource
unsubscribe
untag_resource

Returns a list of the requester’s topics
Use this request to opt in a phone number that is opted out, which enables you to resume sending SMS messages to the number
Sends a message to an Amazon SNS topic, a text message (SMS message) directly to a phone number, or a message to a mobile platform endpoint
Removes a statement from a topic’s access control policy
Sets the attributes for an endpoint for a device on one of the supported push notification services
Sets the attributes of the platform application object for the supported push notification services
Use this request to set the default settings for sending SMS messages and receiving daily SMS usage reports
Allows a subscription owner to set an attribute of the subscription to a new value
Allows a topic owner to set an attribute of the topic to a new value
Subscribes an endpoint to an Amazon SNS topic
Add tags to the specified Amazon SNS topic
Deletes a subscription
Remove tags from the specified Amazon SNS topic

Examples

```r
## Not run:
svc <- sns()
svc$add_permission(
  Foo = 123
)

## End(Not run)
```

---

Amazon Simple Queue Service

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.

For information on the permissions you need to use this API, see Identity and access management in the Amazon Simple Queue Service Developer Guide.

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
Usage

sqs(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- 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
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 Elastic Compute Cloud instance (EC2 instance), or any on-premises server or virtual machine (VM) 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 EC2 instances, see the Amazon EC2 API Reference. For information about how to use a Query API, see Making API requests.

#### Examples

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>delete_message_batch</code></td>
<td>Deletes up to ten messages from the specified queue</td>
</tr>
<tr>
<td><code>delete_queue</code></td>
<td>Deletes the queue specified by the QueueUrl, regardless of the queue’s contents</td>
</tr>
<tr>
<td><code>get_queue_attributes</code></td>
<td>Gets attributes for the specified queue</td>
</tr>
<tr>
<td><code>get_queue_url</code></td>
<td>Returns the URL of an existing Amazon SQS queue</td>
</tr>
<tr>
<td><code>list_dead_letter_source_queues</code></td>
<td>Returns a list of your queues that have the RedrivePolicy queue attribute configured with a DeadLetterQueue URL attribute</td>
</tr>
<tr>
<td><code>list_queues</code></td>
<td>Returns a list of your queues in the current region</td>
</tr>
<tr>
<td><code>list_queue_tags</code></td>
<td>List all cost allocation tags added to the specified Amazon SQS queue</td>
</tr>
<tr>
<td><code>purge_queue</code></td>
<td>Deletes the messages in a queue specified by the QueueURL parameter</td>
</tr>
<tr>
<td><code>receive_message</code></td>
<td>Retrieves one or more messages (up to 10), from the specified queue</td>
</tr>
<tr>
<td><code>remove_permission</code></td>
<td>Revokes any permissions in the queue policy that matches the specified Label parameter</td>
</tr>
<tr>
<td><code>send_message</code></td>
<td>Delivers a message to the specified queue</td>
</tr>
<tr>
<td><code>send_message_batch</code></td>
<td>Delivers up to ten messages to the specified queue</td>
</tr>
<tr>
<td><code>set_queue_attributes</code></td>
<td>Sets the value of one or more queue attributes</td>
</tr>
<tr>
<td><code>tag_queue</code></td>
<td>Add cost allocation tags to the specified Amazon SQS queue</td>
</tr>
<tr>
<td><code>untag_queue</code></td>
<td>Remove cost allocation tags from the specified Amazon SQS queue</td>
</tr>
</tbody>
</table>
Usage

```python
ssm(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
svc <- 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`: Generates an activation code and activation ID you can use to register your on-premises server or virtual machine (VM) with Systems Manager.
- `create_association`: Associates the specified Systems Manager document with the state that you want to maintain on your instances.
- `create_association_batch`: Associates the specified Systems Manager document with the specified instances or targets.
- `create_document`: Creates a Systems Manager (SSM) document.
- `create_maintenance_window`: Creates a new maintenance window.
- `create_ops_item`: Creates a new OpsItem.
- `create_ops_metadata`: If you create a new application in Application Manager, Systems Manager calls this API action to specify information about the new application, including the application type.
- `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 instance.
- `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.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete_maintenance_window</td>
<td>Deletes a maintenance window</td>
</tr>
<tr>
<td>delete_ops_metadata</td>
<td>Delete OpsMetadata related to an application</td>
</tr>
<tr>
<td>delete_parameter</td>
<td>Delete a parameter from the system</td>
</tr>
<tr>
<td>delete_parameters</td>
<td>Delete a list of parameters</td>
</tr>
<tr>
<td>delete_patch_baseline</td>
<td>Deletes a patch baseline</td>
</tr>
<tr>
<td>deregister_managed_instance</td>
<td>Deletes a Resource Data Sync configuration</td>
</tr>
<tr>
<td>deregister_patch_baseline_for_patch_group</td>
<td>Removes the server or virtual machine from the list of registered servers</td>
</tr>
<tr>
<td>deregister_target_from_maintenance_window</td>
<td>Removes a patch group from a patch baseline</td>
</tr>
<tr>
<td>deregister_task_from_maintenance_window</td>
<td>Removes a target from a maintenance window</td>
</tr>
<tr>
<td>describe_activations</td>
<td>Describes details about the activation, such as the date and time</td>
</tr>
<tr>
<td>describe_association</td>
<td>Describes the association for the specified target or instance</td>
</tr>
<tr>
<td>describe_association_executions</td>
<td>Use this API action to view all executions for a specific association</td>
</tr>
<tr>
<td>describe_association_execution_targets</td>
<td>Use this API action to view information about a specific execution</td>
</tr>
<tr>
<td>describe_automation_executions</td>
<td>Provides details about all active and terminated Automation execution</td>
</tr>
<tr>
<td>describe_automation_step_executions</td>
<td>Information about all active and terminated step executions in a Automation</td>
</tr>
<tr>
<td>describe_available_patches</td>
<td>Lists all patches eligible to be included in a patch baseline</td>
</tr>
<tr>
<td>describe_document</td>
<td>Describes the specified Systems Manager document</td>
</tr>
<tr>
<td>describe_effective_instance_associations</td>
<td>Describes the permissions for a Systems Manager document</td>
</tr>
<tr>
<td>describe_effective_patches_for_patch_baseline</td>
<td>All associations for the instance(s)</td>
</tr>
<tr>
<td>describe_instance_associations_status</td>
<td>Retrieves the current effective patches (the patch and the approval state)</td>
</tr>
<tr>
<td>describe_instance_information</td>
<td>The status of the associations for the instance(s)</td>
</tr>
<tr>
<td>describe_instance_information</td>
<td>Describes one or more of your instances, including information</td>
</tr>
<tr>
<td>describe_instance_patches</td>
<td>Retrieves information about the patches on the specified instance</td>
</tr>
<tr>
<td>describe_instance_patch_states</td>
<td>Retrieves the high-level patch state of one or more instances</td>
</tr>
<tr>
<td>describe_instance_patch_states_for_patch_group</td>
<td>Retrieves the high-level patch state for the instances in the specified patch group</td>
</tr>
<tr>
<td>describe_inventory_deletions</td>
<td>Describes a specific delete inventory operation</td>
</tr>
<tr>
<td>describe_maintenance_window_executions</td>
<td>Lists the executions of a maintenance window</td>
</tr>
<tr>
<td>describe_maintenance_window_execution_tasks</td>
<td>Retrieves the individual task executions (one per target) for a patch</td>
</tr>
<tr>
<td>describe_maintenance_window_scheduled</td>
<td>For a given maintenance window execution, lists the tasks that must be run</td>
</tr>
<tr>
<td>describe_maintenance_window_windows</td>
<td>Retrieves the maintenance windows in an AWS account</td>
</tr>
<tr>
<td>describe_maintenance_window_schedule</td>
<td>Retrieves information about upcoming executions of a maintenance window</td>
</tr>
<tr>
<td>describe_maintenance_window_tasks</td>
<td>Retrieves information about the maintenance window targets or tasks</td>
</tr>
<tr>
<td>describe_maintenance_window_targets</td>
<td>Lists the targets registered with the maintenance window</td>
</tr>
<tr>
<td>describe_maintenance_window_tasks</td>
<td>Lists the tasks in a maintenance window</td>
</tr>
<tr>
<td>describe_ops_items</td>
<td>Query a set of OpsItems</td>
</tr>
<tr>
<td>describe_parameters</td>
<td>Get information about a parameter</td>
</tr>
<tr>
<td>describe_patch_baselines</td>
<td>Lists the patch baselines in your AWS account</td>
</tr>
<tr>
<td>describe_patch_groups</td>
<td>Lists all patch groups that have been registered with patch baselines</td>
</tr>
<tr>
<td>describe_patch_group_state</td>
<td>Returns high-level aggregated patch compliance state for a patch</td>
</tr>
<tr>
<td>describe_patch_properties</td>
<td>Lists the properties of available patches organized by product, platform, and so on</td>
</tr>
<tr>
<td>describe_sessions</td>
<td>Retrieves a list of all active sessions (both connected and disconnected)</td>
</tr>
<tr>
<td>get_default_patch_baseline</td>
<td>Get detailed information about a particular Automation execution</td>
</tr>
<tr>
<td>get_deployable_patch_snapshot_for_instance</td>
<td>Gets the state of the AWS Systems Manager Change Calendar</td>
</tr>
<tr>
<td>get_automation_execution</td>
<td>Returns detailed information about command execution for an instance</td>
</tr>
<tr>
<td>get_connection_status</td>
<td>Retrieves the Session Manager connection status for an instance</td>
</tr>
<tr>
<td>get_default_patch_baseline</td>
<td>Retrieves the default patch baseline</td>
</tr>
<tr>
<td>get_deployable_patch_snapshot_for_instance</td>
<td>Retrieves the current snapshot for the patch baseline the instance</td>
</tr>
</tbody>
</table>
get_document
get_inventory
get_inventory_schema
get_maintenance_window
get_maintenance_window_execution
get_maintenance_window_execution_task
get_maintenance_window_execution_task_invocation
get_maintenance_window_task
get_ops_item
get_ops_metadata
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_document_metadata_history
list_documents
list_document_versions
list_inventory_entries
list_ops_item_events
list_ops_metadata
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
startAssociationsOnce

Gets the contents of the specified Systems Manager document
Query inventory information
Return a list of inventory type names for the account, or return a list of attribute names for a specific inventory item type
Retrieves a maintenance window
Retrieves details about a specific maintenance window execution
Retrieves details about a specific task run as part of a maintenance window execution
Retrieves information about a specific task running on a specific instance
Lists the tasks in a maintenance window
Get information about an OpsItem by using the ID
View operational metadata related to an application in Application Manager
View a summary of OpsItems based on specified filters and aggregators
Get information about a parameter by using the parameter name
For a specified resource ID, this API action returns a list of compliance types and other compliance details
Returns a summary count of compliant and non-compliant resources for a compliance type
Returns a list of all Systems Manager (SSM) documents in the current AWS account and Region
List all versions for a document
A list of inventory items returned by the request
Returns a list of all OpsItem events in the current AWS account
List all associations for the current AWS account and region
List all versions of an association for a specific association ID
Lists the commands requested by users of the AWS account
For a specified resource ID, this API action returns a list of compliant and non-compliant resources
Information about approval reviews for a version of an SSM document
Returns a list of the tags assigned to the specified resource
A parameter label is a user-defined alias to help you manage different versions of a parameter
A parameter label is a user-defined alias to help you manage different versions of a parameter
A parameter label is a user-defined alias to help you manage different versions of a parameter
A parameter label is a user-defined alias to help you manage different versions of a parameter
Initiates execution of an Automation document
Starts an 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 the status of the Systems Manager document associated with the specified instance
Updates one or more values for an SSM document
Set the default version of a document
Updates information related to approval reviews for a specific version of a document
Updates an existing maintenance window
Modifies the target of an existing maintenance window
Modifies a task assigned to a maintenance window
Changes the Amazon Identity and Access Management (IAM) role of the on-premises instance or virtual machines (VM)
Edit or change an OpsItem
Systems Manager calls this API action when you edit OpsMetadata
Modify an existing patch baseline
Update a resource data sync
ServiceSetting is an account-level setting for an AWS service

## 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 endpoints and quotas:** 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-78e226633445566ee`.

For more information, see [Announcement: Heads-up – Longer AWS Storage Gateway volume and snapshot IDs coming in 2016](#).

### Usage

```python
storagegateway(config = list())
```

### Arguments

**config**

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

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

### Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

### Service syntax

```python
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
add_cache
add_tags_to_resource
add_upload_buffer
add_working_storage
assign_tape_pool
attach_volume
cancel_archival
cancel_retrieval
createCachediSCSIVolume
createNFSFileShare
createSMBFileShare
create_snapshot
create_snapshot_from_volume_recovery_point
create_storedi_scsi_volume
create_tape_pool
create_tapes
create_tape_with_barcode
delete_automatic_tape_creation_policy
delete_bandwidth_rate_limit
delete_chap_credentials
delete_file_share
delete_gateway
delete_snapshot_schedule
delete_tape
delete_tape_archive
delete_tape_pool
delete_volume
describe_availability_monitor_test
describe_bandwidth_rate_limit
describe_bandwidth_rate_limit_schedule
describe_cache
describeCachediSCSIVolumes
describe_chap_credentials
describe_gateway_information
describe_maintenance_start_time
describe_nfs_file_shares

activates the gateway you previously deployed on your host
Configures one or more gateway local disks as cache for a gateway
Adds one or more tags to the specified resource
Configures one or more gateway local disks as upload buffer for a specified gateway
Configures one or more gateway local disks as working storage for a gateway
Assigns a tape to a tape pool for archiving
Connects a volume to an iSCSI connection and then attaches the volume to the gateway
Cancels archiving of a virtual tape to the virtual tape shelf (VTS) after the archiving process is initiated
Cancels retrieval of a virtual tape from the virtual tape shelf (VTS) to a gateway
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 a new custom tape pool
Creates one or more virtual tapes
Creates a virtual tape by using your own barcode
Deletes the automatic tape creation policy of a gateway
Deletes the bandwidth rate limits of a gateway
Deletes Challenge-Handshake Authentication Protocol (CHAP) credentials from a gateway
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)
Delete a custom tape pool
Deletes the specified storage volume that you previously created using the CreateCachediSCSIVolume or CreateStorEdiSCSIVolume API
Returns information about the most recent High Availability monitoring test
Returns the bandwidth rate limits of a gateway
Returns information about the bandwidth rate limit schedule 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
Returns metadata about a gateway such as its name, network interfaces, configuration
Returns your gateway’s weekly maintenance start time including the day and time
Gets a description for one or more Network File System (NFS) file shares from a gateway.
describe_smb_file_shares
describe_smb_settings
describe_snapshot_schedule
describe_storedi_scsci_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_automatic_tape_creation_policies
list_file_shares
list_gateways
list_local_disks
list_tags_for_resource
list_tape_pools
list_tapes
list_volume_initiators
list_volume_recovery_points
list_volumes
notify_when_uploaded
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_automatic_tape_creation_policy
update_bandwidth_rate_limit
update_bandwidth_rate_limit_schedule
update_chap_credentials
update_gateway_information
update_gateway_software_now
update_maintenance_start_time
update_nfs_file_share
update_smb_file_share
update_smb_file_share_visibility
update_smb_security_strategy
update_snapshot_schedule
update_vtl_device_type

get a description for one or more server message block (SMB) file shares
get 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 tape
returns a description of the specified Amazon Resource Name (ARN) of virtual tapes
returns information about the upload buffer of a gateway
returns a description of virtual tape library (VTL) devices for the specified tape
returns information about the working storage of a gateway
disconnects a volume from an iSCSI connection and then detaches the volume
Disables a tape gateway when the gateway is no longer functioning
Adds a file gateway to an Active Directory domain
Lists the automatic tape creation policies for a gateway
Gets a list of the file shares for a specific file gateway, or the list of file share
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 custom tape pools
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
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
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 automatic tape creation policy of a gateway
Updates the bandwidth rate limits of a gateway
Updates the bandwidth rate limit schedule for a specified gateway
Updates the Challenge-Handshake Authentication Protocol (CHAP) credentials
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
Controls whether the shares on a gateway are visible in a net view or browse
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

```r
## Not run:
svc <- storagegateway()
# Activates the gateway you previously deployed on your host.
svc$activate_gateway(
  ActivationKey = "29AV1-30FV9-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**

AWS Security Token Service (STS) 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 information about using this service, see Temporary Security Credentials.

**Usage**

```r
sts(config = list())
```

**Arguments**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```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 do not normally have permission to access.
- `assume_role_with_saml` Returns a set of temporary security credentials for users who have been authenticated via an 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\":\"Allow\",\"Action\":\"sts:AssumeRole\",\"Principal\":\"amazonaws.com\"],
  RoleArn = "arn:aws:iam::123456789012:role/demo",
  RoleSessionName = "testAssumeRoleSession",
  Tags = list(
    list(
      Key = "Project",
      Value = "Unicorn"
    ),
    list(
      Key = "Team",
      Value = "Automation"
    ),
  ))
```
Key = "Cost-Center",
    Value = "12345"
    }
    ),
  TransitiveTagKeys = list(
    "Project",
    "Cost-Center"
  )
  )
## End(Not run)

support

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.

- You must have a Business or Enterprise support plan to use the AWS Support API.
- If you call the AWS Support API from an account that does not have a Business or Enterprise support plan, the SubscriptionRequiredException error message appears. For information about changing your support plan, see AWS Support.

The AWS Support service also exposes a set of AWS 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 describe_services and describe_severity_levels operations return AWS service names, service codes, service categories, and problem severity levels. You use these values when you call the create_case operation.

- **Case creation, case details, and case resolution.** The create_case, describe_cases, describe_attachment, and resolve_case 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:

- **describe_trusted_advisor_checks** returns the list of checks that run against your AWS resources.
Using the checkId for a specific check returned by `describe_trusted_advisor_checks`, you can call `describe_trusted_advisor_check_result` to obtain the results for the check that you specified.

- `describe_trusted_advisor_check_summaries` returns summarized results for one or more Trusted Advisor checks.
- `refresh_trusted_advisor_check` requests that Trusted Advisor rerun a specified check.
- `describe_trusted_advisor_check_refresh_statuses` 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**

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

**Value**

A client for the service. You can call the service’s operations using syntax like `svc$operation(...),` where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

**Service syntax**

```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 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 for each service

- **describe_severity_levels**
  - Returns the list of severity levels that you can assign to an AWS Support case

- **describe_trusted_advisor_check_refresh_statuses**
  - Returns the refresh status of the AWS Trusted Advisor checks that have the specified check IDs

- **describe_trusted_advisor_check_result**
  - Returns the results of the AWS Trusted Advisor check that has the specified check ID

- **describe_trusted_advisor_checks**
  - Returns information about all available AWS Trusted Advisor checks, including the name, ID, category, description, and metadata

- **describe_trusted_advisor_check_summaries**
  - Returns the results for the AWS Trusted Advisor check summaries for the specified check IDs

- **refresh_trusted_advisor_check**
  - Refreshes the AWS Trusted Advisor check that you specify using the check ID

- **resolve_case**
  - Resolves a support case

Examples

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

---

**swf**  
*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.

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```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 filtering criteria.
- `count_open_workflow_executions` Returns the number of open workflow executions within the given domain that meet the 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 filtering criteria.
- `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 task list.
- `poll_for_decision_task` Used by deciders to get a DecisionTask from the specified decision task list.
Example

```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.
Value
A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

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

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

Operations and objects for transcribing speech to text.

Usage

transcribeservice(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

svc <- transcribeservice(
  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>create_language_model</td>
<td>Creates a new custom language model</td>
</tr>
<tr>
<td>create_medical_vocabulary</td>
<td>Creates a new custom vocabulary that you can use to change how Amazon Transcribe Medical transcribes your audio file</td>
</tr>
<tr>
<td>create_vocabulary</td>
<td>Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription of an audio file</td>
</tr>
<tr>
<td>create_vocabulary_filter</td>
<td>Creates a new vocabulary filter that you can use to filter words, such as profane words, from the transcription output</td>
</tr>
<tr>
<td>delete_language_model</td>
<td>Deletes a custom language model using its name</td>
</tr>
<tr>
<td>delete_medical_transcription_job</td>
<td>Deletes a transcription job generated by Amazon Transcribe Medical and any related information</td>
</tr>
<tr>
<td>delete_medical_vocabulary</td>
<td>Deletes a vocabulary from Amazon Transcribe Medical</td>
</tr>
<tr>
<td>delete_transcription_job</td>
<td>Deletes a previously submitted transcription job along with any other generated results such as models and transcriptions</td>
</tr>
<tr>
<td>delete_vocabulary</td>
<td>Deletes a vocabulary from Amazon Transcribe</td>
</tr>
<tr>
<td>delete_vocabulary_filter</td>
<td>Removes a vocabulary filter</td>
</tr>
<tr>
<td>describe_language_model</td>
<td>Gets information about a single custom language model</td>
</tr>
<tr>
<td>get_medical_transcription_job</td>
<td>Returns information about a transcription job from Amazon Transcribe Medical</td>
</tr>
</tbody>
</table>
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.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
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

- `create_parallel_data`: Creates a parallel data resource in Amazon Translate by importing an input file from Amazon S3.
- `delete_parallel_data`: Deletes a parallel data resource in Amazon Translate.
- `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, status, source and target languages, input/output S3 buckets, and so on.
- `get_parallel_data`: Provides information about a parallel data resource.
- `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_parallel_data`: Provides a list of your parallel data resources in Amazon Translate.
- `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.
- `update_parallel_data`: Updates a previously created parallel data resource by importing a new input file from Amazon S3.

Examples

```r
## Not run:
svc <- translate()
svc$create_parallel_data(
  Foo = 123
)

## End(Not run)
```
Description

This is **AWS WAF Classic** documentation. For more information, see **AWS WAF Classic** in the developer guide.

**For the latest version of AWS WAF**, use the AWS WAFV2 API and see the **AWS WAF Developer Guide**. With the latest version, AWS WAF has a single set of endpoints for regional and global use.

This is the **AWS WAF Classic API Reference** for using AWS WAF Classic with Amazon CloudFront. The AWS WAF Classic 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 Classic API actions, data types, and errors. For detailed information about AWS WAF Classic features and an overview of how to use the AWS WAF Classic API, see the **AWS WAF Classic** in the developer guide.

Usage

```python
waf(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
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
create_geo_match_set
create_ip_set
create_rate_based_rule
create_regex_match_set
create_regex_pattern_set
create_rule
create_rule_group
create_size_constraint_set
create_sql_injection_match_set
create_web_acl
create_web_acl_migration_stack
create_xss_match_set
delete_byte_match_set
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  This is AWS WAF Classic documentation
list_geo_match_sets  This is AWS WAF Classic documentation
list_ip_sets  This is AWS WAF Classic documentation
list_logging_configurations  This is AWS WAF Classic documentation
list_rate_based_rules  This is AWS WAF Classic documentation
list_regex_match_sets  This is AWS WAF Classic documentation
list_regex_pattern_sets  This is AWS WAF Classic documentation
list_rule_groups  This is AWS WAF Classic documentation
list_rules  This is AWS WAF Classic documentation
list_size_constraint_sets  This is AWS WAF Classic documentation
list_sql_injection_match_sets  This is AWS WAF Classic documentation
list_subscribed_rule_groups  This is AWS WAF Classic documentation
list_tags_for_resource  This is AWS WAF Classic documentation
list_web_acl  This is AWS WAF Classic documentation
list_xss_match_sets  This is AWS WAF Classic documentation
put_logging_configuration  This is AWS WAF Classic documentation
put_permission_policy  This is AWS WAF Classic documentation
tag_resource  This is AWS WAF Classic documentation
untag_resource  This is AWS WAF Classic documentation
update_byte_match_set  This is AWS WAF Classic documentation
update_geo_match_set  This is AWS WAF Classic documentation
update_ip_set  This is AWS WAF Classic documentation
update_rate_based_rule  This is AWS WAF Classic documentation
update_regex_match_set  This is AWS WAF Classic documentation
update_regex_pattern_set  This is AWS WAF Classic documentation
update_rule  This is AWS WAF Classic documentation
update_rule_group  This is AWS WAF Classic documentation
update_size_constraint_set  This is AWS WAF Classic documentation
update_sql_injection_match_set  This is AWS WAF Classic documentation
update_web_acl  This is AWS WAF Classic documentation
update_xss_match_set  This is AWS WAF Classic documentation

Examples

## Not run:
svc <- waf()
# The following example creates an IP match set named MyIPSetFriendlyName.
svc$create_ip_set(
  ChangeToken = "abcd12f2-46da-4fdb-b8d5-fbd4c466928f",
  Name = "MyIPSetFriendlyName"
)
## End(Not run)
Description

This is **AWS WAF Classic Regional** documentation. For more information, see **AWS WAF Classic** in the developer guide.

**For the latest version of AWS WAF**, use the AWS WAFV2 API and see the **AWS WAF Developer Guide**. With the latest version, AWS WAF has a single set of endpoints for regional and global use.

This is the **AWS WAF Regional Classic API Reference** for using AWS WAF Classic with the AWS resources, Elastic Load Balancing (ELB) Application Load Balancers and API Gateway APIs. The AWS WAF Classic actions and data types listed in the reference are available for protecting Elastic Load Balancing (ELB) Application Load Balancers and API Gateway APIs. 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 Classic API actions, data types, and errors. For detailed information about AWS WAF Classic features and an overview of how to use the AWS WAF Classic API, see the **AWS WAF Classic** in the developer guide.

Usage

```python
wafregional(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```python
svc <- 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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>associate_web_acl</td>
<td>This is AWS WAF Classic Regional documentation</td>
</tr>
<tr>
<td>create_byte_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_geo_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_ip_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_rate_based_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_regex_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_regex_pattern_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_rule_group</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_size_constraint_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_sql_injection_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_web_acl</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>create_web_acl_migration_stack</td>
<td>Creates an AWS CloudFormation WAFV2 template for the specified web ACL in the specified Amazon S3 bucket</td>
</tr>
<tr>
<td>create_xss_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_byte_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_geo_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_ip_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_logging_configuration</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_permission_policy</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_rate_based_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_regex_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_regex_pattern_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_rule_group</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_size_constraint_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_sql_injection_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>delete_web_acl</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>disassociate_web_acl</td>
<td>This is AWS WAF Classic Regional documentation</td>
</tr>
<tr>
<td>get_byte_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_change_token</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_change_token_status</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_geo_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_ip_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_logging_configuration</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_permission_policy</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_rate_based_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_rate_based_rule_managed_keys</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_regex_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_regex_pattern_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_rule</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_rule_group</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_sampled_requests</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_size_constraint_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_sql_injection_match_set</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
<tr>
<td>get_web_acl</td>
<td>This is AWS WAF Classic documentation</td>
</tr>
</tbody>
</table>
### Examples

```r
## Not run:
svc <- wafregional()
# The following example creates an IP match set named MyIPSetFriendlyName.
svc$create_ip_set(
  ChangeToken = "abcd12f2-46da-4fdb-b8d5-fbd4c466928f",
  Name = "MyIPSetFriendlyName"
)
```

## End(Not run)
Description

The WorkDocs API is designed for the following use cases:

- **File Migration**: File migration applications are supported for users who want to migrate their files from an on-premises or off-premises file system or service. Users can insert files into a user directory structure, as well as allow for basic metadata changes, such as modifications to the permissions of files.

- **Security**: Security applications are supported for users who have additional security needs, such as antivirus or data loss prevention. The API actions, along with AWS CloudTrail, allow these applications to detect when changes occur in Amazon WorkDocs. Then, the application can take the necessary actions and replace the target file. If the target file violates the policy, the application can also choose to email the user.

- **eDiscovery/Analytics**: General administrative applications are supported, such as eDiscovery and analytics. These applications can choose to mimic or record the actions in an Amazon WorkDocs site, along with AWS CloudTrail, to replicate data for eDiscovery, backup, or analytical applications.

All Amazon WorkDocs API actions are Amazon authenticated and certificate-signed. They not only require the use of the AWS SDK, but also allow for the exclusive use of IAM users and roles to help facilitate access, trust, and permission policies. By creating a role and allowing an IAM user to access the Amazon WorkDocs site, the IAM user gains full administrative visibility into the entire Amazon WorkDocs site (or as set in the IAM policy). This includes, but is not limited to, the ability to modify file permissions and upload any file to any user. This allows developers to perform the three use cases above, as well as give users the ability to grant access on a selective basis using the IAM model.

Usage

```python
workdocs(config = list())
```

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```r
svc <- workdocs(
  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_document_version_upload: Aborts the upload of the specified document version that was previously initiated by InitiateDocumentVersionUpload
- activate_user: Activates the specified user
- add_resource_permissions: Creates a set of permissions for the specified folder or document
- create_comment: Adds a new comment to the specified document version
- create_custom_metadata: Adds one or more custom properties to the specified resource (a folder, document, or version)
- create_folder: Creates a folder with the specified name and parent folder
- create_labels: Adds the specified list of labels to the given resource (a document or folder)
- create_notification_subscription: Configure Amazon WorkDocs to use Amazon SNS notifications
- create_user: Creates a user in a Simple AD or Microsoft AD directory
- deactivate_user: Deactivates the specified user, which revokes the user's access to Amazon WorkDocs
- delete_comment: Deletes the specified comment from the document version
- delete_custom_metadata: Deletes custom metadata from the specified resource
- delete_document: Permanently deletes the specified document and its associated metadata
- delete_folder: Permanently deletes the specified folder and its contents
- delete_folder_contents: Deletes the contents of the specified folder
- delete_labels: Deletes the specified list of labels from a resource
- delete_notification_subscription: Deletes the specified subscription from the specified organization
- delete_user: Deletes the specified user from a Simple AD or Microsoft AD directory
- describe_activities: Describes the user activities in a specified time period
- describe_comments: List all the comments for the specified document version
- describe_document_versions: Retrieves the document versions for the specified document
- describe_folder_contents: Describes the contents of the specified folder, including its documents and subfolders
- describe_groups: Describes the groups specified by the query
- describe_notification_subscriptions: Lists the specified notification subscriptions
- describe_resource_permissions: Describes the permissions of a specified resource
- describe_root_folders: Describes the current user's special folders; the RootFolder and the RecycleBin
- describe_users: Describes the specified users
- get_current_user: Retrieves details of the current user for whom the authentication token was generated
- get_document: Retrieves details of a document
- get_document_path: Retrieves the path information (the hierarchy from the root folder) for the requested document
### Examples

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

---

**worklink**

### Amazon WorkLink

#### Description

Amazon WorkLink is a cloud-based service that provides secure access to internal websites and web apps from iOS and Android phones. In a single step, your users, such as employees, can access internal websites as efficiently as they access any other public website. They enter a URL in their web browser, or choose a link to an internal website in an email. Amazon WorkLink authenticates the user’s access and securely renders authorized internal web content in a secure rendering service in the AWS cloud. Amazon WorkLink doesn’t download or store any internal web content on mobile devices.

#### Usage

```
worklink(config = list())
```

#### Arguments

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

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- worklink(
  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_domain` Specifies a domain to be associated to Amazon WorkLink
- `associate_website_authorization_provider` Associates a website authorization provider with a specified fleet
- `associate_website_certificate_authority` Imports the root certificate of a certificate authority (CA) used to obtain TLS certificates used by associated websites within the company network
- `create_fleet` Creates a fleet
- `delete_fleet` Deletes a fleet
- `describe_audit_stream_configuration` Describes the configuration for delivering audit streams to the customer account
- `describe_company_network_configuration` Describes the networking configuration to access the internal websites associated with the specified fleet
- `describe_device` Provides information about a user’s device
- `describe_device_policy_configuration` Describes the device policy configuration for the specified fleet
- `describe_domain` Provides information about the domain
- `describe_fleet_metadata` Provides basic information for the specified fleet, excluding identity provider, networking, and device configuration details
- `describe_identity_provider_configuration` Provides information about the certificate authority
- `disassociate_domain` Disassociates a domain from Amazon WorkLink
- `disassociate_website_authorization_provider` Disassociates a website authorization provider from a specified fleet
- `disassociate_website_certificate_authority` Removes a certificate authority (CA)
- `list_devices` Retrieves a list of devices registered with the specified fleet
- `list_domains` Retrieves a list of domains associated to a specified fleet
- `list_fleets` Retrieves a list of fleets for the current account and Region
- `list_tags_for_resource` Retrieves a list of tags for the specified resource
- `list_website_authorization_providers` Retrieves a list of website authorization providers associated with a specified fleet
- `list_website_certificateAuthorities` Retrieves a list of certificate authorities added for the current account and Region
- `restore_domain_access` Moves a domain to ACTIVE status if it was in the INACTIVE status
- `revoke_domain_access` Moves a domain to INACTIVE status if it was in the ACTIVE status
workspaces

Amazon WorkSpaces

Description

Amazon WorkSpaces Service

Amazon WorkSpaces enables you to provision virtual, cloud-based Microsoft Windows and Amazon Linux desktops for your users.

Usage

workspaces(config = list())

Arguments

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

Value

A client for the service. You can call the service’s operations using syntax like svc$operation(...), where svc is the name you’ve assigned to the client. The available operations are listed in the Operations section.
Service syntax

```python
svc <- workspaces(
    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_connection_alias`: Associates the specified connection alias with the specified directory to enable cross-Region redirection
- `associate_ip_groups`: Associates the specified IP access control group with the specified directory
- `authorize_ip_rules`: Adds one or more rules to the specified IP access control group
- `copy_workspace_image`: Copies the specified image from the specified Region to the current Region
- `create_connection_alias`: Creates the specified connection alias for use with cross-Region redirection
- `create_ip_group`: Creates an IP access control group
- `create_tags`: Creates the specified tags for the specified WorkSpaces resource
- `create_workspaces`: Creates one or more WorkSpaces
- `delete_connection_alias`: Deletes the specified connection alias
- `delete_ip_group`: Deletes the specified IP access control group
- `delete_tags`: Deletes the specified tags from the specified WorkSpaces resource
- `delete_workspace_image`: Deletes the specified image from your account
- `deregister_workspace_directory`: Deregisters the specified directory
- `describe_account`: Retrieves a list that describes the configuration of Bring Your Own License (BYOL) for the specified account
- `describe_account_modifications`: Retrieves a list that describes modifications to the configuration of Bring Your Own License (BYOL) for the specified account
- `describe_client_properties`: Retrieves a list that describes one or more specified Amazon WorkSpaces clients
- `describe_connection_aliases`: Describes the connection aliases used for cross-Region redirection
- `describe_connection_alias_permissions`: Describes the permissions that the owner of a connection alias has granted to another AWS account
- `describe_ip_groups`: Describes one or more of your IP access control groups
- `describe_tags`: Describes the specified tags for the specified WorkSpaces resource
- `describe_workspace_bundles`: Retrieves a list that describes the available WorkSpace bundles
- `describe_workspace_directories`: Describes the available directories that are registered with Amazon WorkSpaces
- `describe_workspace_image_permissions`: Describes the permissions that the owner of an image has granted to other AWS accounts
- `describe_workspace_images`: Retrieves a list that describes one or more specified images, if the image identifiers are provided
- `describe_workspaces`: Describes the specified WorkSpaces
- `describe_workspaces_connection_status`: Describes the connection status of the specified WorkSpaces
- `describe_workspace_snapshots`: Describes the snapshots for the specified WorkSpace
- `disassociate_connection_alias`: Disassociates a connection alias from a directory
- `disassociate_ip_groups`: Disassociates the specified IP access control group from the specified directory
- `import_workspace_image`: Imports the specified Windows 10 Bring Your Own License (BYOL) image into A
list_available_management_cidr_ranges
migrate_workspace
modify_account
modify_client_properties
modify_selfservice_permissions
modify_workspace_access_properties
modify_workspace_creation_properties
modify_workspace_properties
modify_workspace_state
reboot_workspaces
rebuild_workspaces
register_workspace_directory
restore_workspace
revoke_ip_rules
start_workspaces
stop_workspaces
terminate_workspaces
update_connection_alias_permission
update_rules_of_ip_group
update_workspace_image_permission

Retrieves a list of IP address ranges, specified as IPv4 CIDR blocks, that you can use
Migrates a WorkSpace from one operating system or bundle type to another, while
Modifies the configuration of Bring Your Own License (BYOL) for the specified account
Modifies the properties of the specified Amazon WorkSpaces clients
Modifies the self-service WorkSpace management capabilities for your users
Specifies which devices and operating systems users can use to access their WorkSpaces
Modifies the specified WorkSpace properties
Sets the state of the specified WorkSpace
Reboots the specified WorkSpaces
Rebuilds the specified WorkSpace
Registers the specified directory
Restores the specified WorkSpace to its last known healthy state
Removes one or more rules from the specified IP access control group
Starts the specified WorkSpaces
Stops the specified WorkSpaces
Terminates the specified WorkSpaces
Shares or unshares a connection alias with one account by specifying whether that account
Shares or unshares an image with one account in the same AWS Region by specifying

Examples

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

xray

AWS X-Ray

Description

AWS X-Ray provides APIs for managing debug traces and retrieving service maps and other data
created by processing those traces.

Usage

`xray(config = list())`

Arguments

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

A client for the service. You can call the service’s operations using syntax like `svc$operation(...)`, where `svc` is the name you’ve assigned to the client. The available operations are listed in the Operations section.

Service syntax

```r
svc <- xray(
  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_traces`: Retrieves a list of traces specified by ID
- `create_group`: Creates a group resource with a name and a filter expression
- `create_sampling_rule`: Creates a rule to control sampling behavior for instrumented applications
- `delete_group`: Deletes a group resource
- `delete_sampling_rule`: Deletes a sampling rule
- `get_encryption_config`: Retrieves the current encryption configuration for X-Ray data
- `get_group`: Retrieves group resource details
- `get_groups`: Retrieves all active group details
- `get_insight`: Retrieves the summary information of an insight
- `get_insight_events`: X-Ray reevaluates insights periodically until they’re resolved, and records each intermediate state as an event
- `get_insight_impact_graph`: Retrieves a service graph structure filtered by the specified insight
- `get_insight_summaries`: Retrieves the summaries of all insights in the specified group matching the provided filter values
- `get_sampling_rules`: Retrieves all sampling rules
- `get_sampling_rule_summaries`: Retrieves information about recent sampling results for all sampling rules
- `get_sampling_targets`: Requests a sampling quota for rules that the service is using to sample requests
- `get_service_graph`: Retrieves a document that describes services that process incoming requests, and downstream services that they call as a result
- `get_time_series_service_statistics`: Get an aggregation of service statistics defined by a specific time range
- `get_trace_graph`: Retrieves a service graph for one or more specific trace IDs
- `get_trace_summaries`: Retrieves IDs and annotations for traces available for a specified time frame using an optional filter
- `list_tags_for_resource`: Returns a list of tags that are applied to the specified AWS X-Ray group or sampling rule
- `put_encryption_config`: Updates the encryption configuration for X-Ray data
- `put_telemetry_records`: Used by the AWS X-Ray daemon to upload telemetry
- `put_trace_segments`: Uploads segment documents to AWS X-Ray
- `tag_resource`: Applies tags to an existing AWS X-Ray group or sampling rule
untag_resource  Removes tags from an AWS X-Ray group or sampling rule
update_group   Updates a group resource
update_sampling_rule  Modifies a sampling rule’s configuration

Examples

## Not run:
svc <- xray()
svc$batch_get_traces(
  Foo = 123
)

## End(Not run)
Index

abort_document_version_upload, 337
abort_environment_update, 142
abort_multipart_upload, 160, 274
abort_vault_lock, 160
accept_direct_connect_gateway_association_proposal, 106
accept_domain_transfer_from_another_aws_account, 270
accept_grant, 199
accept_handshake, 225
accept_inbound_cross_cluster_search_connection, 144
accept_invitation, 171, 289
accept_portfolio_share, 293
accept_qualification_request, 214
accept_reserved_instances_exchange_quote, 119
accept_reserved_node_exchange, 254
accept_resource_share_invitation, 247
accept_shared_directory, 108
accept_transit_gateway_multicast_domain_associations, 119
accept_transit_gateway_peering_attachment, 119
accept_transit_gateway_vpc_attachment, 119
accept_vpc_endpoint_connections, 119
accept_vpc_peering_connection, 119
acknowledge_job, 76, 77
acknowledge_third_party_job, 76, 77
acm, 5
acmpca, 6
activate_event_source, 57, 153
activate_gateway, 317
activate_key_signing_key, 267
activate_pipeline, 103
activate_user, 337
add_application_cloud_watch_logging_option, 185, 187
add_application_input, 185, 187
add_application_input_processing_configuration, 185, 187
add_application_output, 185, 187
add_application_reference_data_source, 185, 187
add_application_vpc_configuration, 187
add_association, 279
add_attachments_to_set, 321, 323
add_attributes_to_findings, 180
add_cache, 317
add_client_id_to_open_id_connect_provider, 176
add_communication_to_case, 321, 323
add_custom_attributes, 84
add_custom_routing_endpoints, 164
add_facet_to_object, 39
add_instance_fleet, 151
add_instance_groups, 151
add_ip_routes, 108
add_job_flow_steps, 151
add_layer_version_permission, 192
add_listener_certificates, 149
add_permission, 192, 308, 310
add_region, 108
add_resource_permissions, 337
add_role_to_db_cluster, 216, 249
add_role_to_db_instance, 249
add_role_to_instance_profile, 176
add_source_identifier_to_subscription, 216, 249
add_tags, 53, 103, 144, 146, 149, 151, 205, 279
add_tags_to_certificate, 6
add_tags_to_on_premises_instances, 73
add_tags_to_resource, 47, 108, 112, 139, 216, 249, 312, 317
add_tags_to_stream, 184
add_tags_to_vault, 160
<table>
<thead>
<tr>
<th>Function</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_upload_buffer</td>
<td>317</td>
</tr>
<tr>
<td>add_user_to_group</td>
<td>176</td>
</tr>
<tr>
<td>add_working_storage</td>
<td>317</td>
</tr>
<tr>
<td>admin_add_user_to_group</td>
<td>84</td>
</tr>
<tr>
<td>admin_confirm_sign_up</td>
<td>84</td>
</tr>
<tr>
<td>admin_create_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_delete_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_delete_user_attributes</td>
<td>84</td>
</tr>
<tr>
<td>admin_disable_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_disable_provider_for_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_disable_user_attributes</td>
<td>84</td>
</tr>
<tr>
<td>admin_enable_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_forget_device</td>
<td>84</td>
</tr>
<tr>
<td>admin_get_device</td>
<td>84</td>
</tr>
<tr>
<td>admin_get_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_initiate_auth</td>
<td>84</td>
</tr>
<tr>
<td>admin_link_provider_for_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_list_devices</td>
<td>84</td>
</tr>
<tr>
<td>admin_list_groups_for_user</td>
<td>84</td>
</tr>
<tr>
<td>admin_list_user_auth_events</td>
<td>84</td>
</tr>
<tr>
<td>admin_remove_user_from_group</td>
<td>84</td>
</tr>
<tr>
<td>admin_reset_user_password</td>
<td>84</td>
</tr>
<tr>
<td>admin_response_to_auth_challenge</td>
<td>84</td>
</tr>
<tr>
<td>admin_set_user_mfa_preference</td>
<td>84</td>
</tr>
<tr>
<td>admin_set_user_password</td>
<td>84</td>
</tr>
<tr>
<td>admin_set_user_settings</td>
<td>84</td>
</tr>
<tr>
<td>admin_update_auth_event_feedback</td>
<td>84</td>
</tr>
<tr>
<td>admin_update_device_status</td>
<td>84</td>
</tr>
<tr>
<td>advertise_byoip_cidr</td>
<td>119, 164</td>
</tr>
<tr>
<td>allocate_address</td>
<td>119</td>
</tr>
<tr>
<td>allocate_connection_on_interconnect</td>
<td>106</td>
</tr>
<tr>
<td>allocate_hosted_connection</td>
<td>106</td>
</tr>
<tr>
<td>allocate_hosts</td>
<td>119</td>
</tr>
<tr>
<td>allocate_private_virtual_interface</td>
<td>106</td>
</tr>
<tr>
<td>allocate_public_virtual_interface</td>
<td>106</td>
</tr>
<tr>
<td>allocate_static_ip</td>
<td>201</td>
</tr>
<tr>
<td>allocate_transit_virtual_interface</td>
<td>106</td>
</tr>
<tr>
<td>allow_custom_routing_traffic</td>
<td>164</td>
</tr>
<tr>
<td>analyze_document</td>
<td>326</td>
</tr>
<tr>
<td>apigateway</td>
<td>8</td>
</tr>
<tr>
<td>apigatewaymanagementapi</td>
<td>12</td>
</tr>
<tr>
<td>apigatewayv2</td>
<td>13</td>
</tr>
<tr>
<td>applicationautoscaling</td>
<td>15</td>
</tr>
<tr>
<td>applicationinsights</td>
<td>18</td>
</tr>
<tr>
<td>apply_environment_managed_action</td>
<td>142</td>
</tr>
<tr>
<td>apply_pending_maintenance_action</td>
<td>112, 216, 249</td>
</tr>
<tr>
<td>apply_schema</td>
<td>39</td>
</tr>
<tr>
<td>apply_security_groups_to_client_vpn_target_network</td>
<td>119</td>
</tr>
<tr>
<td>apply_security_groups_to_load_balancer</td>
<td>146</td>
</tr>
<tr>
<td>appmesh</td>
<td>20</td>
</tr>
<tr>
<td>approve_assignment</td>
<td>214</td>
</tr>
<tr>
<td>appstream</td>
<td>22</td>
</tr>
<tr>
<td>archive_findings</td>
<td>171</td>
</tr>
<tr>
<td>assign_instance</td>
<td>220</td>
</tr>
<tr>
<td>assign_ipv6_addresses</td>
<td>119</td>
</tr>
<tr>
<td>assign_private_ip_addresses</td>
<td>119</td>
</tr>
<tr>
<td>assign_tape_pool</td>
<td>317</td>
</tr>
<tr>
<td>assign_volume</td>
<td>220</td>
</tr>
<tr>
<td>associate_address</td>
<td>119</td>
</tr>
<tr>
<td>associate_admin_account</td>
<td>157</td>
</tr>
<tr>
<td>associate_approval_rule_template_with_repository</td>
<td>67, 69</td>
</tr>
<tr>
<td>associate_approved_origin</td>
<td>96</td>
</tr>
<tr>
<td>associate_budget_with_resource</td>
<td>293</td>
</tr>
<tr>
<td>associate_client_vpn_target_network</td>
<td>119</td>
</tr>
<tr>
<td>associate_connection_alias</td>
<td>341</td>
</tr>
<tr>
<td>associate_connection_with_lag</td>
<td>106</td>
</tr>
<tr>
<td>associate_dhcp_options</td>
<td>119</td>
</tr>
<tr>
<td>associate_domain</td>
<td>339</td>
</tr>
<tr>
<td>associate_drt_log_bucket</td>
<td>305</td>
</tr>
<tr>
<td>associate_drt_role</td>
<td>305</td>
</tr>
<tr>
<td>associate_elastic_ip</td>
<td>220</td>
</tr>
<tr>
<td>associate_enclave_certificate_iam_role</td>
<td>119</td>
</tr>
<tr>
<td>associate_environment_operations_role</td>
<td>142</td>
</tr>
<tr>
<td>associate_file_system_aliases</td>
<td>158</td>
</tr>
<tr>
<td>associate_fleet</td>
<td>23</td>
</tr>
<tr>
<td>associate_health_check</td>
<td>305</td>
</tr>
<tr>
<td>associate_hosted_connection</td>
<td>106</td>
</tr>
<tr>
<td>associate_iam_instance_profile</td>
<td>119</td>
</tr>
<tr>
<td>associate_instance_storage_config</td>
<td>96</td>
</tr>
<tr>
<td>associate_ip_groups</td>
<td>341</td>
</tr>
<tr>
<td>associate_kms_key</td>
<td>60</td>
</tr>
<tr>
<td>associate_lambda_function</td>
<td>96</td>
</tr>
<tr>
<td>associate_lex_bot</td>
<td>96</td>
</tr>
<tr>
<td>associate_member_account</td>
<td>207</td>
</tr>
<tr>
<td>associate_node</td>
<td>223</td>
</tr>
</tbody>
</table>
batch_delete_cluster_snapshots, 255
batch_delete_connection, 167
batch_delete_image, 130
batch_delete_partition, 167
batch_delete_scheduled_action, 27
batch_delete_table, 167
batch_delete_table_version, 167
batch_describe_merge_conflicts, 65, 69
batch_detect_dominant_language, 89
batch_detect_entities, 89
batch_detect_key_phrases, 89
batch_detect_sentiment, 89
batch_detect_syntax, 89
batch_disable_standards, 289
batch_disassociate_approval_rule_template_from_repositories, 67, 69
batch_disassociate_scram_secret, 182
batch_disassociate_service_action_from_provisioning_artifact, 294
batch_disassociate_user_stack, 23
batch_enable_standards, 288, 289
batch_execute_statement, 114, 253
batch_get_aggregate_resource_config, 93
batch_get_application_revisions, 73
batch_get_applications, 73
batch_get_build_batches, 63
batch_get_builds, 61, 63
batch_get_commits, 65, 69
batch_get_crawlers, 167
batch_get_deployment_groups, 73
batch_get_deployment_instances, 73
batch_get_deployment_targets, 73
batch_get_deployments, 73
batch_get_dev_endpoints, 167
batch_get_image, 130
batch_get_item, 114
batch_get_jobs, 167
batch_get_named_query, 25
batch_get_on_premises_instances, 73
batch_get_partition, 167
batch_get_projects, 61, 63
batch_get_query_execution, 25
batch_get_report_groups, 61, 63
batch_get_reports, 61, 63
batch_get_repositories, 65, 69
batch_get_resource_config, 93
batch_get_traces, 343
batch_get_triggers, 167
batch_get_workflows, 167
batch_import_findings, 289
batch_meter_usage, 211
batch_modify_cluster_snapshots, 255
batch_put_attributes, 307
batch_put_scheduled_update_group_action, 27
batch_read, 39
batch_stop_job_run, 167
batch_stop_update_action, 139
batch_update_findings, 289
batch_update_partition, 167
batch_write, 39
begin_transaction, 253
budgets, 34
build_auth_token, 249
build_suggesters, 50
bulk_publish, 87
bundle_instance, 119
cancel_archival, 317
cancel_bundle_task, 119
cancel_capacity_reservation, 119
cancel_command, 312
cancel_conversion_task, 119
cancel_data_repository_task, 158
cancel_domain_transfer_to_another_aws_account, 270
cancel_elasticsearch_service_software_update, 144
cancel_export_task, 60, 119, 249
cancel_handshake, 225
cancel_import_task, 119
cancel_ingestion, 243
cancel_instance_refresh, 27
cancel_job, 33
cancel_key_deletion, 190
cancel_maintenance_window_execution, 312
cancel_ml_task_run, 167
cancel_replay, 57, 153
cancel_reserved_instances_listing, 120
cancel_resize, 255
cancel_retrieval, 317
cancel_rotate_secret, 287
cancel_schema_extension, 108
cancel_spot_fleet_requests, 120
cancel_spot_instance_requests, 120
cancel_steps, 151
cancel_update_stack, 42
change_message_visibility, 310
change_message_visibility_batch, 310
change_password, 84, 176
change_resource_record_sets, 267
change_tags_for_resource, 267
check_dns_availability, 142
check_domain_availability, 270
check_domain_transferability, 270
check_if_phone_number_is_opted_out, 308
check_in_license, 199
check_schema_version_validity, 167
checkout_borrow_license, 199
checkout_license, 199
classify_document, 89
clone_receipt_rule_set, 300
clone_stack, 219, 220
close_instance_public_ports, 201
cloud9, 36
clouddirectory, 38
cloudformation, 41
cloudfront, 43
cloudhsm, 46
cloudhsmv2, 48
cloudsearch, 49
cloudsearchdomain, 51
cloudtrail, 52
cloudwatch, 54
cloudwatchevents, 56
cloudwatchlogs, 58
codebuild, 61
codecommit, 64
codedeploy, 71
codestar, 78
cognitoidentity, 80
cognitoidentityprovider, 82
cognitosync, 86
commit_transaction, 253
compare_faces, 258
complete_layer_upload, 130
complete_lifecycle_action, 27
complete_migration, 139
complete_multipart_upload, 160, 274
complete_vault_lock, 160
compose_environments, 142
comprehend, 88
comprehendmedical, 90
configservice, 92
configure_health_check, 146
confirm_connection, 106
confirm_device, 84
confirm_forgot_password, 84
confirm_private_virtual_interface, 106
confirm_product_instance, 120
confirm_public_virtual_interface, 106
confirm_sign_up, 84
confirm_subscription, 308
confirm_transit_virtual_interface, 106
connect, 95
connect_custom_key_store, 190
connect_directory, 108
continue_deployment, 73
continue_update_rollback, 42
copy_backup_to_region, 49
copy_cluster_snapshot, 255
copy_db_cluster_parameter_group, 112, 216, 249
copy_db_cluster_snapshot, 112, 216, 249
copy_db_parameter_group, 216, 249
copy_db_snapshot, 249
copy_fpga_image, 120
copy_image, 23, 120
copy_object, 274
copy_option_group, 249
copy_product, 294
copy_snapshot, 120, 139, 201
copy_workspace_image, 341
costandusagereportservice, 98
costexplorer, 99
count_closed_workflow_executions, 324
count_open_workflow_executions, 324
count_pending_activity_tasks, 324
count_pending_decision_tasks, 324
create_accelerator, 164
create_access_key, 176
create_access_point, 135, 277
create_account, 225
create_account_alias, 176
create_account_customization, 243
create_action, 279
create_action_target, 289
create_activation, 312
create_activity, 303
create_additional_assignments_for_hit, 214
create_addon, 137
create_algorithm, 279
create_alias, 108, 190, 192
create_analysis, 243
create_anomaly_monitor, 100
create_anomaly_subscription, 100
create_api, 14
create_api_key, 9
create_api_mapping, 14
create_app, 220, 233, 279
create_app_cookie_stickiness_policy, 146
create_app_image_config, 279
create_application, 19, 73, 142, 185, 187, 292
create_application_presigned_url, 187
create_application_snapshot, 187
create_application_version, 142, 292
create_approval_rule_template, 67, 69
create_archive, 57, 153
create_artifact, 279
create_assessment_target, 180
create_assessment_template, 180
create_association, 312
create_association_batch, 312
create_authorizer, 9, 14
create_auto_ml_job, 279
create_auto_scaling_group, 27
create_backup, 115, 158, 223
create_backup_plan, 31
create_backup_selection, 31
create_backup_vault, 31
create_base_path_mapping, 9
create_batch_inference_job, 227
create_batch_prediction, 205
create_bgp_peer, 106
create_bot_version, 194
create_branch, 65, 69
create_broker, 212
create_bucket, 274, 277
create_budget, 36
create_budget_action, 36
create_byte_match_set, 331, 334
create_cache_cluster, 139
create_cache_parameter_group, 139
create_cache_policy, 44
create_cache_security_group, 139
create_cache_subnet_group, 139
create_cachedi_scsi_volume, 317
create_campaign, 227, 233
create_capacity_provider, 133
create_capacity_reservation, 120
create_carrier_gateway, 120
create_case, 321, 323
create_certificate, 201
create_certificate_authority, 7
create_certificate_authority_audit_report, 7
create_change_set, 42
create_classifier, 167
create_client_vpn_endpoint, 120
create_client_vpn_route, 120
create_cloud_formation_change_set, 292
create_cloud_formation_stack, 201
create_cloud_formation_template, 292
create_cloud_front_origin_access_identity, 44
create_cluster, 49, 104, 133, 137, 182, 255
create_cluster_parameter_group, 255
create_cluster_security_group, 255
create_cluster_snapshot, 255
create_cluster_subnet_group, 255
create_code_repository, 279
create_code_signing_config, 192
create_collection, 258
create_comment, 337
create_commit, 65, 69
create_compilation_job, 279
create_component, 19
create_compute_environment, 33
create_computer, 109
create_conditional_forwarder, 109
create_configuration, 182, 212
create_configuration_set, 237, 239, 300
create_configuration_set_event_destination, 237, 239, 300
create_configuration_set_tracking_options, 300
create_configuration_template, 142
create_connection, 106, 167
create_connection_alias, 341
create_constraint, 294
create_contact_flow, 96
create_contact_method, 201
create_container_service, 201
create_container_service_deployment, 201
create_container_service_registry_login, 201
create_context, 279
create_cost_category_definition, 100
create_crawler, 167
create_custom_action_type, 77
create_custom_availability_zone, 249
create_custom_key_store, 190
create_custom_metadata, 337
create_custom_routing_accelerator, 165
create_custom_routing_endpoint_group, 165
create_custom_routing_listener, 165
create_custom_verification_email_template, 300
create_customer_gateway, 120
create_dashboard, 243
create_data_catalog, 25
create_data_quality_job_definition, 279
create_data_repository_task, 158
create_data_set, 243
create_data_source, 243
create_data_source_from_rds, 205
create_data_source_from_redshift, 205
create_data_source_from_s3, 205
create_database, 167
create_dataset, 227
create_dataset_group, 227
create_dataset_import_job, 227
create_db_cluster, 112, 216, 249
create_db_cluster_endpoint, 216, 249
create_db_cluster_parameter_group, 112, 216, 249
create_db_cluster_snapshot, 112, 216, 249
create_db_instance, 112, 216, 249
create_db_instance_read_replica, 249
create_db_parameter_group, 216, 249
create_db_proxy, 249
create_db_security_group, 249
create_db_snapshot, 249
create_db_subnet_group, 112, 216, 249
create_dedicated_ip_pool, 237
create_default_subnet, 120
create_default_vpc, 120
create_deliverability_test_report, 237
create_delivery_stream, 155
create_deployment, 9, 14, 73, 220
create_deployment_config, 73
create_deployment_group, 73
create_detector, 171
create_dev_endpoint, 167
create_device_fleet, 279
create_dhcp_options, 120
create_direct_connect_gateway, 106
create_direct_connect_gateway_association, 106
create_direct_connect_gateway_association_proposal, 106
create_directory, 39, 109
create_directory_config, 23
create_disk, 201
create_disk_from_snapshot, 201
create_disk_snapshot, 201
create_distribution, 44, 201
create_distribution_with_tags, 44
create_document, 312
create_document_classifier, 89
create_documentation_part, 9
create_documentation_version, 9
create_domain, 50, 201, 279, 307
create_domain_entry, 201
create_domain_name, 9, 14
create_edge_packaging_job, 279
create_egress_only_internet_gateway, 120
create_elasticsearch_domain, 144
create_email_identity, 237
create_email_template, 233
create_endpoint, 89, 279
create_endpoint_config, 279
create_endpoint_group, 165
create_entity_recognizer, 89
create_environment, 142
create_environment_ec2, 37, 38
create_environment_membership, 37, 38
create_evaluation, 205
create_event_bus, 57, 153
create_event_source_mapping, 192
create_event_subscription, 216, 249, 255
create_event_tracker, 227
create_exclusions_preview, 180
create_experiment, 279
create_export_job, 233
create_export_task, 60
create_facet, 39
create_fargate_profile, 137
create_feature_group, 279
create_field_level_encryption_config, 44
create_field_level_encryption_profile, 44
create_file_system, 135, 158
create_file_system_from_backup, 158
create_filter, 171, 227
create_fleet, 23, 120, 339
create_flow_definition, 279
create_flow_logs, 120
create_folder, 337
create_fpga_image, 120
create_function, 175
create_gateway_route, 21
create_geo_match_set, 331, 334
create_global_cluster, 249
create_global_replication_group, 139
create_global_table, 115
create_gov_cloud_account, 225
create_grant, 190, 199
create_grant_version, 199
create_group, 84, 176, 243, 260, 343
create_group_membership, 243
create_hapg, 47
create_health_check, 267
create_hit, 214
create_hit_type, 214
create_hit_with_hit_type, 214
create_hosted_zone, 267
create_hsm, 47, 49
create_hsm_client_certificate, 255
create_hsm_configuration, 255
create_http_namespace, 296
create_human_task_ui, 279
create_hyper_parameter_tuning_job, 279
create_iam_policy_assignment, 244
create_identity_pool, 81
create_identity_provider, 84
create_image, 120, 279
create_image_builder, 23
create_image_builder_streaming_url, 23
create_image_version, 280
create_import_job, 233
create_index, 39
create_ingestion, 244
create_insight, 289
create_instance, 96, 220
create_instance_export_task, 120
create_instance_profile, 176
create_instance_snapshot, 201
create_instances, 201
create_instances_from_snapshot, 201
create_integration, 14
create_integration_association, 96
create_integration_response, 14
create_intent_version, 194
create_interconnect, 106
create_internet_gateway, 120
create_invalidation, 44
create_ip_group, 341
create_ip_set, 171, 331, 334
create_job, 167, 277
create_job_queue, 33
create_journey, 233
create_key, 190
create_key_group, 44
create_key_pair, 120, 201
create_key_signing_key, 267
create_labeling_job, 280
create_labels, 337
create_lag, 106
create_language_model, 327
create_launch_configuration, 27
create_launch_template, 120
create_launch_template_version, 120
create_layer, 220
create_lb_cookie_stickiness_policy, 146
create_license, 199
create_license_configuration, 199
create_license_version, 199
create_lifecycle_policy, 111
create_listener, 149, 165
create_load_balancer, 145, 146, 149, 202
create_load_balancer_listeners, 146
create_load_balancer_policy, 146
create_load_balancer_tls_certificate, 202
create_local_gateway_route, 120
create_local_gateway_route_table_vpc_association  120
create_log_group, 60
create_log_pattern, 19
create_log_stream, 60
create_log_subscription, 109
create_login_profile, 176
create_luna_client, 47
create_maintenance_window, 312
create_managed_prefix_list, 120
create_medical_vocabulary, 327
create_members, 171, 288, 289
create_mesh, 21
create_microsoft_ad, 109
create_ml_model, 205
create ml_transform, 167
create_model, 9, 14, 280
create_model_bias_job_definition, 280
create_model_explainability_job_definition, 280
create_model_package, 280
create_model_package_group, 280
create_model_quality_job_definition, 280
create_monitoring_schedule, 280
create_monitoring_subscription, 44
create_mount_target, 135
create_multipart_upload, 274
create_named_query, 25
create_namespace, 244
create_nat_gateway, 120
create_network_acl, 120
create_network_acl_entry, 120
create_network_insights_path, 120
create_network_interface, 120
create_network_interface_permission, 120
create_nfs_file_share, 317
create_nodegroup, 137
create_notebook_instance, 280
create_notebook_instance_lifecycle_config, 280
create_notification, 36
create_notification_subscription, 337
create_object, 39
create_open_id_connect_provider, 176
create_ops_item, 312
create_ops_metadata, 312
create_option_group, 249
create_or_update_tags, 27
create_organization, 225
create_organizational_unit, 225
create_origin_request_policy, 44
create_outbound_cross_cluster_search_connection, 144
create_package, 144
create_parallel_data, 329
create_parameter_group, 104
create_partition, 167
create_partition_index, 167
create_partner_event_source, 57, 153
create_patch_baseline, 312
create_permission, 7
create_pipeline, 74, 75, 77, 103, 280
create_placement_group, 120
create_platform_application, 308
create_platform_endpoint, 308
create_platform_version, 142
create_policy, 176, 225
create_policy_version, 176
create_portfolio, 294
create_portfolio_share, 294
create_presigned_domain_url, 280
create_presigned_notebook_instance_url, 280
create_private_dns_namespace, 296
create_private_virtual_interface, 106
create_processing_job, 280
create_product, 294
create_project, 61, 63, 79, 258, 280
create_project_version, 258
create_protection, 305
create_protection_group, 305
create_provisioned_product_plan, 294
create_provisioning_artifact, 294
create_public_dns_namespace, 297
create_public_key, 44
create_public_virtual_interface, 106
create_publishing_destination, 171
create_pull_request, 66, 69
create_pull_request_approval_rule, 66, 69
create_push_template, 233
create_qualification_type, 214
create_query_logging_config, 267
create_queue, 310
create_quick_connect, 96
create_rate_based_rule, 331, 334
create_realtime_endpoint, 205
create_realtime_log_config, 44
create_receipt_filter, 300
create_receipt_rule, 300
create_receipt_rule_set, 300
create_recommender_configuration, 233
create_regex_match_set, 331, 334
create_regex_pattern_set, 331, 334
create_registry, 167
create_relational_database, 202
create_relational_database_from_snapshot, 202
create_relational_database_snapshot, 202
create_replication_group, 139
create_report_group, 61, 63
create_repository, 63, 69, 130
create_request_validator, 9
create_reserved_instances_listing, 120
create_resolver_endpoint, 272
create_resolver_query_log_config, 272
create_resolver_rule, 272
create_rest_api, 9
create_reusable_delegation_set, 267
create_role, 176
create_route, 14, 21, 120
create_route_response, 14
create_route_table, 120
create_routing_profile, 96
create_rule, 149, 331, 334
create_rule_group, 331, 334
create_saml_provider, 176
create_sample_findings, 171
create_sampling_rule, 343
create_scaling_plan, 30
create_scheduled_action, 255
create_schema, 39, 167, 227
create_script, 167
create_secret, 287
create_security_configuration, 151, 167
create_security_group, 120
create_segment, 233
create_server, 223
create_service, 133, 297
create_service_action, 294
create_service_linked_role, 176
create_service_specific_credential, 176
create_size_constraint_set, 331, 334
create_slot_type_version, 194
create_smb_file_share, 317
create_sms_template, 233
create_snapshot, 109, 120, 139, 317
create_snapshot_copy_grant, 255
create_snapshot_from_volume_recovery_point, 317
create_snapshot_schedule, 255
create_snapshots, 120
create_solution, 227
create_solution_version, 227
create_spot_datafeed_subscription, 120
create_sql_injection_match_set, 331, 334
create_stack, 23, 42, 219, 220
create_stack_instances, 42
create_stack_set, 42
create_stage, 9, 14
create_state_machine, 303
create_storage_location, 142
create_storedi_scsi_volume, 317
create_stream, 184
create_stream_processor, 258
create_streaming_distribution, 44
create_streaming_distribution_with_tags, 44
create_streaming_url, 23
create_studio, 151
create_studio_session_mapping, 151
create_subnet, 120
create_subnet_group, 104
create_subscriber, 36
create_subscription, 305
create_table, 115, 167
create_tag_option, 294
create_tags, 120, 135, 212, 255, 341
create_tape_pool, 317
create_tape_with_barcode, 317
create_tapes, 317
create_target_group, 149
create_task_set, 133
create_template, 244, 300
create_template_alias, 244
create_theme, 244
create_theme_alias, 244
create_threat_intel_set, 171
create_token, 199
create_topic, 308
create_traffic_mirror_filter, 120
create_traffic_mirror_filter_rule, 120
create_traffic_mirror_session, 120
create_traffic_mirror_target, 120
create_traffic_policy, 267
create_traffic_policy_instance, 267
create_traffic_policy_version, 267
create_trail, 53
create_training_job, 280
create_transform_job, 280
create_transit_gateway, 121
create_transit_gateway_connect, 121
create_transit_gateway_connect_peer, 121
create_transit_gateway_multicast_domain, 121
create_transit_gateway_peering_attachment, 121
create_transit_gateway_prefix_list_reference, 121
create_transit_gateway_route, 121
create_transit_gateway_route_table, 121
create_transit_gateway_vpc_attachment, 121
create_transit_virtual_interface, 106
create_trial, 280
create_trial_component, 280
create_trigger, 167
create_trust, 109
create_TYPED_LINK_FACET, 39
create_UNREFERENCED_MERGE_COMMIT, 65, 69
create_usage_limit, 255
create_usage_plan, 9
create_usage_plan_key, 9
create_usage_report_subscription, 23
create_use_case, 96
create_user, 23, 96, 139, 176, 212, 337
create_user_defined_function, 167
create_user_group, 139
create_user_hierarchy_group, 96
create_user_import_job, 84
create_user_pool, 84
create_user_pool_client, 84
create_user_pool_domain, 84
create_user_profile, 78, 79, 220, 280
create_vault, 160
create_virtual_gateway, 21
create_virtual_mfa_device, 176
create_virtual_node, 21
create_virtual_router, 21
create_virtual_service, 21
create_vocabulary, 327
create_vocabulary_filter, 327
create_voice_template, 233
create_volume, 121
create_vpc, 121
create_vpc_association_authorization, 267
create_vpc_endpoint, 121
create_vpc_endpoint_connection_notification, 121
create_vpc_endpoint_service_configuration, 121
create_vpc_link, 9, 14
create_vpc_peering_connection, 121
create_vpn_connection, 121
create_vpn_connection_route, 121
create_vpn_gateway, 121
create_web_acl, 331, 334
create_web_acl_migration_stack, 331, 334
create_webhook, 61, 63
create_work_group, 25
create_worker_block, 214
create_workflow, 167
create_workforce, 280
create_workspaces, 341
create_workteam, 280
create_xss_match_set, 331, 334
datapipeline, 101
dax, 103
deactivate_event_source, 57, 153
deactivate_key_signing_key, 267
deactivate_mfa_device, 176
defactivate_pipeline, 103
defactivate_user, 337
decline_handshake, 225
decline_invitations, 171, 289
decode_authorization_message, 320
decrease_node_groups_in_global_replication_group, 139
decrease_replica_count, 139
decrease_replication_factor, 104
decrease_stream_retention_period, 184
decrypt, 189, 190
define_analysis_scheme, 50
define_expression, 50
define_index_field, 50
define_suggester, 50
delete_accelerator, 165
delete_access_key, 176
delete_access_log_settings, 14
delete_access_point, 135, 277
delete_access_point_policy, 277
delete_account_alias, 176
delete_account_customization, 244
delete_account_password_policy, 176
delete_account_setting, 133
delete_action, 280
delete_action_target, 289
delete_activation, 312
delete_activity, 303
delete_addon, 137
delete_adm_channel, 233
delete_aggregation_authorization, 93
delete_alarm, 202
delete_alarms, 55
delete_algorithm, 280
delete_alias, 190, 192
delete_analysis, 244
delete_analysis_scheme, 50
delete_anomaly_detector, 55
delete_anomaly_monitor, 101
delete_anomaly_subscription, 101
delete_api, 14
delete_api_key, 9
delete_api_mapping, 14
delete_apns_channel, 233
delete_apns_sandbox_channel, 233
delete_apns_voip_channel, 233
delete_apns_voip_sandbox_channel, 233
delete_app, 220, 233, 280
delete_app_image_config, 280
delete_application, 19, 73, 142, 185, 187, 292
delete_application_cloud_watch_logging_option, 186, 187
delete_application_input_processing_configuration, 186, 187
delete_application_output, 186, 187
delete_application_reference_data_source, 186, 187
delete_application_snapshot, 187
delete_application_version, 142
delete_application_vpc_configuration, 187
delete_approval_rule_template, 67, 69
delete_apps_list, 157
delete_archive, 57, 153, 160
delete_artifact, 280
delete_assessment_run, 180
delete_assessment_target, 180
delete_assessment_template, 180
delete_association, 280, 312
delete_attributes, 133, 307
delete_authorizer, 9, 14
delete_auto_scaling_group, 27
delete_auto_snapshot, 202
delete_auto_tape_creation_policy, 317
delete_backups, 49, 115, 158, 223
delete_backup_plan, 31
delete_backup_selection, 31
delete_backup_vault, 31
delete_backup_vault_access_policy, 31
delete_backup_vault_notifications, 31
delete_baidu_channel, 233
delete_bandwidth_rate_limit, 317
delete_base_path_mapping, 9
delete_batch_prediction, 205
delete_bgp_peer, 106
delete_bot, 194
delete_bot_alias, 195
delete_bot_channel_association, 195
delete_bot_version, 195
delete_branch, 65, 69
delete_broker, 213
delete_bucket, 274, 277
delete_bucket_analytics_configuration, 274
delete_bucket_cors, 274
delete_bucket_encryption, 274
delete_bucket_intelligent_tiering_configuration, 274
delete_bucket_inventory_configuration, 274
delete_bucket_lifecycle, 274
delete_bucket_lifecycle_configuration, 277
delete_bucket_metrics_configuration, 274
delete_bucket_ownership_controls, 274
delete_bucket_policy, 274, 277
delete_bucket_replication, 274
delete_bucket_tagging, 274, 277
delete_bucket_website, 274
delete_budget, 36
delete_budget_action, 36
delete_build_batch, 63
delete_byte_match_set, 331, 334
delete_cache_cluster, 139
delete_cache_parameter_group, 139
delete_cache_policy, 44
delete_cache_security_group, 139
delete_cache_subnet_group, 139
delete_campaign, 227, 233
delete_capacity_provider, 133
delete_carrier_gateway, 121
delete_certificate, 6, 202
delete_certificate_authority, 7
delete_change_set, 42
delete_chap_credentials, 317
delete_classifier, 167
delete_client_certificate, 9
delete_client_vpn_endpoint, 121
delete_client_vpn_route, 121
delete_cloud_front_origin_access_identity, 44
delete_cluster, 49, 104, 133, 137, 182, 255
delete_cluster_parameter_group, 255
delete_cluster_security_group, 255
delete_cluster_snapshot, 255
delete_cluster_subnet_group, 255
delete_code_repository, 280
delete_code_signing_config, 192
delete_collection, 258
delete_column_statistics_for_partition, 167
delete_column_statistics_for_table, 167
delete_comment, 337
delete_comment_content, 68, 69
delete_component, 19
delete_compute_environment, 33
delete_conditional_forwarder, 109
delete_config_rule, 93
delete_configuration, 182
delete_configuration_aggregator, 93
delete_configuration_recorder, 93
delete_configuration_set, 237, 239, 300
delete_configuration_set_event_destination, 237, 239, 300
delete_configuration_set_tracking_options, 300
delete_configuration_template, 142
delete_conformance_pack, 93
delete_connection, 12, 106, 167
delete_connection_alias, 341
delete_constraint, 294
delete_contact_method, 202
delete_container_image, 202
delete_container_service, 202
delete_context, 280
delete_cors_configuration, 14
delete_cost_category_definition, 101
delete_crawler, 167
delete_custom_action_type, 77
delete_custom_available_zone, 249
delete_custom_key_store, 190
delete_custom_metadata, 337
delete_custom_routing_accelerator, 165
delete_custom_routing_endpoint_group, 165
delete_custom_routing_listener, 165
delete_custom_verification_email_template, 300
delete_customer_gateway, 121
delete_dashboard, 244
delete_dashboards, 55
delete_data_catalog, 25
delete_data_quality_job_definition, 280
delete_data_set, 244
delete_data_source, 205, 244
delete_database, 167
delete_dataset, 87, 227
delete_dataset_group, 227
delete_db_cluster, 112, 216, 250
delete_db_cluster_endpoint, 217, 250
delete_db_cluster_parameter_group, 112, 217, 250
delete_db_cluster_snapshot, 112, 217, 250
delete_db_instance, 112, 217, 250
delete_db_instance_automated_backup, 250
delete_db_parameter_group, 217, 250
delete_db_proxy, 250
delete_db_security_group, 250
delete_db_snapshot, 112, 217, 250
delete_dedicated_ip_pool, 237
delete_delivery_channel, 93
delete_delivery_stream, 155
delete_deployment, 9, 14
delete_deployment_config, 73
delete_deployment_group, 73
delete_destination, 60
delete_detector, 171
delete_dev_endpoint, 167
delete_device_fleet, 280
delete_dhcp_options, 121
delete_direct_connect_gateway, 106
delete_direct_connect_gateway_association, 106
delete_direct_connect_gateway_association_proposal, 106
delete_directory, 39, 109
delete_directory_config, 23
delete_disk, 202
delete_disk_snapshot, 202
delete_distribution, 45, 202
delete_document, 312, 337
delete_document_classifier, 89
delete_documentation_part, 9
delete_documentation_version, 9
delete_domain, 50, 202, 280, 307
delete_domain_entry, 202
delete_domain_name, 9, 14
delete_egress_only_internet_gateway, 121
delete_elasticsearch_domain, 144
delete_elasticsearch_service_role, 144
delete_email_channel, 233
delete_email_identity, 237
delete_email_template, 233

delete_endpoint, 89, 233, 280, 308
delete_endpoint_config, 280
delete_endpoint_group, 165
delete_entity_recognizer, 89
delete_environment, 37, 38
delete_environment_configuration, 142
delete_environment_membership, 37, 38
delete_evaluation, 205
delete_evaluation_results, 93
delete_event_bus, 57, 153
delete_event_source_mapping, 192
delete_event_stream, 233
delete_event_subscription, 217, 250, 255
delete_event_tracker, 227
delete_experiment, 280
delete_expression, 50
delete_faces, 258
delete_facet, 39
delete_fargate_profile, 137
delete_feature_group, 280
delete_field_level_encryption_config, 45
delete_field_level_encryption_profile, 45
delete_file, 65, 69
delete_file_share, 317
delete_file_system, 135, 158
delete_file_system_policy, 135
delete_filter, 171, 227
delete_fleet, 23, 339
delete_fleets, 121
delete_flow_definition, 280
delete_flow_logs, 121
delete_folder, 337
delete_folder_contents, 337
delete_fpga_image, 121
delete_function, 192
delete_function_code_signing_config, 192
delete_function_concurrency, 192
delete_function_event_invoke_config, 192
delete_gateway, 317
delete_gateway_response, 9
delete_gateway_route, 21
delete_gateway_route_propagation, 21
delete_gcm_channel, 233
delete_geo_match_set, 331, 334
delete_git_hub_account_token, 73
delete_global_cluster, 250
delete_global_replication_group, 139
delete_grant, 199
delete_group, 84, 176, 244, 260, 343
delete_group_membership, 244
delete_group_policy, 176
delete_hapg, 47
delete_health_check, 267
delete_hit, 214
delete_hosted_zone, 267
delete_hsm, 47, 49
delete_hsm_client_certificate, 255
delete_hsm_configuration, 255
delete_human_task_ui, 280
delete_iam_policy_assignment, 244
delete_identities, 81
delete_identity, 300
delete_identity_policy, 300
delete_identity_pool, 81
delete_identity_provider, 84
delete_image, 23, 280
delete_image_builder, 23
delete_image_permissions, 23
delete_image_version, 280
delete_imported_key_material, 190
delete_inbound_cross_cluster_search_connection, 244
delete_index_field, 50
delete_insight, 289
delete_insight_rules, 55
delete_installation_media, 250
delete_instance, 96, 202, 220
delete_instance_profile, 176
delete_instance_snapshot, 202
delete_integration, 9, 14
delete_integration_association, 96
delete_integration_response, 9, 14
delete_intent, 195
delete_intent_version, 195
delete_interconnect, 106
delete_internet_gateway, 121
delete_inventory, 312
delete_invitations, 171, 289
delete_ip_group, 341
delete_ip_set, 171, 331, 334
delete_item, 115
delete_job, 167
delete_job_queue, 33
delete_job_tagging, 277
delete_job, 234
delete_key_group, 45
delete_key_pair, 121, 202
delete_key_signing_key, 268
delete_known_host_keys, 202
delete_labels, 337
delete_lag, 106
delete_language_model, 327
delete_launch_configuration, 27
delete_launch_template, 121
delete_launch_template_versions, 121
delete_layer, 220
delete_layer_version, 192
delete_lexicon, 240
delete_license, 199
delete_license_configuration, 199
delete_lifecycle_hook, 27
delete_lifecycle_policy, 111, 130
delete_listener, 149, 165
delete_load_balancer, 146, 149, 202
delete_load_balancer_listeners, 146
delete_load_balancer_policy, 146
delete_load_balancer_tls_certificate, 202
delete_local_gateway_route, 121
delete_local_gateway_route_table_vpc_association, 121
delete_logs, 60
delete_log_group, 60
delete_log_pattern, 19
delete_log_stream, 60
delete_log_subscription, 109
delete_logging_configuration, 331, 334
delete_login_profile, 176
delete_luna_client, 47
delete_maintenance_window, 313
delete_managed_prefix_list, 121
delete_medical_transcription_job, 327
delete_medical_vocabulary, 327
delete_members, 171, 289
delete_mesh, 21
delete_message, 310
delete_message_batch, 311
delete_method, 9
delete_method_response, 9
delete_metric_filter, 60
delete_ml_model, 205
delete_ml_transform, 167
delete_model, 9, 14, 280
delete_model_bias_job_definition, 280
delete_model_explainability_job_definition, 280
delete_model_package, 280
delete_model_package_group, 280
delete_model_package_group_policy, 280
delete_model_quality_job_definition, 280
delete_monitoring_schedule, 281
delete_monitoring_subscription, 45
delete_mount_target, 135
delete_named_query, 25
delete_namespace, 244, 297
delete_nat_gateway, 121
delete_network_acl, 121
delete_network_acl_entry, 121
delete_network_insights_analysis, 121
delete_network_insights_path, 121
delete_network_interface, 121
delete_network_interface_permission, 121
delete_nodegroup, 137
delete_notebook_instance, 281
delete_notebook_instance_lifecycle_config, 281
delete_notification, 36
delete_notification_channel, 157
delete_notification_configuration, 27
delete_notification_subscription, 337
delete_object, 40, 274
delete_object_tagging, 274
delete_objects, 274
delete_open_id_connect_provider, 176
delete_ops_metadata, 313
delete_option_group, 250
delete_organization, 225
delete_organization_config_rule, 93
delete_organization_conformance_pack, 93
delete_organizational_unit, 225
delete_origin_request_policy, 45
delete_outbound_cross_cluster_search_connection, 144
delete_package, 144
delete_parallel_data, 329
delete_parameter, 313
delete_parameter_group, 104
delete_parameters, 313
delete_partition, 167
delete_partition_index, 167
delete_partner_event_source, 57, 153
delete_patch_baseline, 313
delete_pending_aggregation_request, 93
delete_permission, 7
delete_permission_policy, 331, 334
delete_pipeline, 74, 77, 103, 281
delete_placement_group, 121
delete_platform_application, 308
delete_platform_version, 142
delete_policy, 7, 27, 157, 176, 225
delete_policy_version, 176
delete_portfolio, 294
delete_portfolio_share, 294
delete_product, 294
delete_project, 61, 63, 78, 79, 258, 281
delete_project_version, 258
delete_protection, 305
delete_protection_group, 305
delete_protocols_list, 157
delete_provisioned_concurrency_config, 192
delete_provisioned_product_plan, 294
delete_provisioning_artifact, 294
delete_public_access_block, 274, 277
delete_public_key, 45
delete_publishing_destination, 171
delete_pull_request_approval_rule, 66, 69
delete_push_template, 234
delete_qualification_type, 214
delete_query_definition, 60
delete_query_logging_config, 268
delete_queue, 311
delete_queued_reserved_instances, 121
delete_quick_connect, 96
delete_rate_based_rule, 331, 334
delete_realtime_endpoint, 205
delete_realtime_log_config, 45
delete_receipt_filter, 300
delete_receipt_rule, 300
delete_receipt_rule_set, 300
delete_recommender_configuration, 234
delete_recovery_point, 31
delete_regex_match_set, 331, 334
delete_regex_pattern_set, 331, 334
delete_registry, 167
delete_registry_policy, 131
delete_relational_database, 202
delete_relational_database_snapshot, 202
delete_remediation_configuration, 93
delete_remediation_exceptions, 93
delete_replication_group, 139
delete_report, 61, 63
delete_report_definition, 99
delete_report_group, 61, 63
delete_repository, 65, 69, 131
delete_repository_policy, 131
delete_request_validator, 9
delete_resolver_endpoint, 272
delete_resolver_query_log_config, 272
delete_resolver_rule, 272
delete_resource, 9
delete_resource_config, 93
delete_resource_data_sync, 313
delete_resource_policy, 60, 61, 63, 167, 287
delete_resource_server, 84
delete_resource_share, 247
delete_resources_by_external_id, 73
delete_rest_api, 9
delete_retention_configuration, 93
delete_retention_policy, 60
delete_reusable_delegation_set, 268
delete_role, 176
delete_role_permissions_boundary, 176
delete_role_policy, 176
delete_route, 14, 21, 121
delete_route_request_parameter, 14
delete_route_response, 14
delete_route_settings, 14
delete_route_table, 121
delete_rule, 57, 149, 153, 331, 334
delete_rule_group, 331, 334
delete_saml_provider, 176
delete_sampling_rule, 343
delete_scaling_plan, 30
delete_scaling_policy, 17
delete_scheduled_action, 17, 27, 255
delete_schema, 40, 167, 227
delete_schema_versions, 167
delete_secret, 287
delete_security_configuration, 151, 167
delete_security_group, 121
delete_segment, 234
delete_server, 223
delete_server_certificate, 176
delete_service, 133, 297
delete_service_action, 294
delete_service_linked_role, 176
delete_service_quota_increase_request_from_template, 298
delete_service_specific_credential, 176
delete_session, 198
delete_signing_certificate, 176
delete_size_constraint_set, 331, 334
delete_slot_type, 195
delete_slot_type_version, 195
delete_sms_channel, 234
delete_sms_template, 234
delete_snapshot, 109, 121, 139
delete_snapshot_copy_grant, 255
delete_snapshot_schedule, 255, 317
delete_solution, 228
delete_source_credentials, 61, 63
delete_spot_datafeed_subscription, 121
delete_sql_injection_match_set, 331, 334
delete_ssh_public_key, 176
delete_stack, 23, 42, 220
delete_stack_instances, 42
delete_stack_set, 42
delete_stage, 9, 14
delete_state_machine, 303
delete_storage_lens_configuration, 277
delete_storage_lens_configuration_tagging, 277
delete_stored_query, 93
delete_stream, 184
delete_stream_processor, 258
delete_streaming_distribution, 45
delete_studio, 151
delete_studio_session_mapping, 151
delete_subnet, 122
delete_subnet_group, 104
delete_subscriber, 36
delete_subscription, 305
delete_subscription_filter, 60
delete_suggester, 50
delete_table, 115, 168
delete_table_version, 168
delete_tag_option, 294
delete_tags, 27, 122, 135, 205, 213, 255, 281, 341
delete_tags_for_domain, 270
delete_tape, 317
delete_tape_archive, 317
delete_tape_pool, 317
delete_target_group, 149
delete_task_set, 133
delete_template, 244, 300
delete_template_alias, 244
delete_terminology, 329
delete_theme, 244
delete_theme_alias, 244
delete_threat_intel_set, 172
delete_token, 199
delete_topic, 308
delete_traffic_mirror_filter, 122
delete_traffic_mirror_filter_rule, 122
delete_traffic_mirror_session, 122
delete_traffic_mirror_target, 122
delete_traffic_policy, 268
delete_traffic_policy_instance, 268
delete_trial, 53
delete_transcription_job, 327
delete_transit_gateway, 122
delete_transit_gateway_connect, 122
delete_transit_gateway_connect_peer, 122
delete_transit_gateway_multicast_domain, 122
delete_transit_gateway_peering_attachment, 122
delete_transit_gateway_prefix_list_reference, delete_vpc_association_authorization, 122
delete_transit_gateway_route, 122
delete_transit_gateway_route_table, 122
delete_transit_gateway_vpc_attachment, 122
delete_trial, 281
delete_trial_component, 281
delete_trigger, 168
delete_trust, 109
delete_typed_link_facet, 40
delete_usage_limit, 255
delete_usage_plan, 9
delete_usage_plan_key, 9
delete_usage_report_subscription, 23
delete_use_case, 96
delete_user, 23, 84, 96, 140, 176, 213, 244, 337
delete_user_attributes, 84
delete_user_by_principal_id, 244
delete_user_defined_function, 168
delete_user_endpoints, 234
delete_user_group, 140
delete_user_hierarchy_group, 96
delete_user_permissions_boundary, 176
delete_user_policy, 176
delete_user_pool, 84
delete_user_pool_client, 84
delete_user_pool_domain, 84
delete_user_profile, 78, 79, 220, 281
delete_utterances, 195
delete_vault, 160
delete_vault_access_policy, 160
delete_vault_notifications, 160
delete_verified_email_address, 300
delete_virtual_gateway, 21
delete_virtual_interface, 107
delete_virtual_mfa_device, 176
delete_virtual_node, 21
delete_virtual_router, 21
delete_virtual_service, 21
delete_vocabulary, 327
delete_vocabulary_filter, 327
delete_voice_channel, 234
delete_voice_template, 234
delete_volume, 122, 317
delete_vpc, 122
delete_vpc_endpoints, 122
delete_vpc_endpoint_connection_notifications, 122
delete_vpc_endpoint_service_configurations, 122
delete_vpc_link, 9, 14
delete_vpc_peering_connection, 122
delete_vpn_connection, 122
delete_vpn_connection_route, 122
delete_vpn_gateway, 122
delete_web_acl, 331, 334
delete_webhook, 62, 63, 77
INDEX

363

delete_work_group, 25
delete_worker_block, 214
delete_workflow, 168
delete_workforce, 281
delete_workspace_image, 341
delete_workteam, 281
delete_xss_match_set, 331, 334
deliver_config_snapshot, 93
deny_custom_routing_traffic, 165
deprecate_activity_type, 324
deprecate_domain, 324
deprecate_workflow_type, 324
deprovision_byoip_cidr, 122, 165
deregister_certificate, 109
deregister_container_instance, 133
deregister_db_proxy_targets, 250
deregister_delegated_administrator, 225
deregister_devices, 281
deregister_ecs_cluster, 220
deregister_elastic_ip, 220
deregister_event_topic, 109
deregister_image, 122
deregister_instance, 220, 297
deregister_instance_event_notification_attributes, 122
deregister_instances_from_load_balancer, 146
deregister_job_definition, 34
deregister_managed_instance, 313
deregister_on_premises_instance, 73
deregister_patch_baseline_for_patch_group, 313
deregister_rds_db_instance, 220
deregister_scalable_target, 17
deregister_stream_consumer, 184
deregister_target_from_maintenance_window, 313
deregister_targets, 149
deregister_task_definition, 133
deregister_task_from_maintenance_window, 313
deregister_transit_gateway_multicast_group_members, 122
deregister_transit_gateway_multicast_group_sources, 122
deregister_type, 42
deregister_volume, 220
deregister_webhook_with_third_party, 77
deregister_workspace_directory, 341
describe_accelerator, 165
describe_accelerator_attributes, 165
describe_access_points, 135
describe_account, 225, 341
describe_account_attributes, 122, 142, 223, 250, 255
describe_account_customization, 244
describe_account_limits, 27, 42, 147, 149
describe_account_modifications, 341
describe_account_settings, 244
describe_action, 281
describe_action_targets, 289
describe_activations, 313
describe_active_receipt_rule_set, 300
describe_activities, 337
describe_activity, 303
describe_activity_type, 324
describe_addon, 137
describe_addon_versions, 137
describe_addresses, 122
describe_adjustment_types, 27
describe_affected_accounts_for_organization, 174
describe_affected_entities, 174
describe_affected_entities_for_organization, 174
describe_agent_versions, 220
describe_aggregate_compliance_by_config_rules, 93
describe_aggregate_id_format, 122
describe_aggregation_authorizations, 93
describe_alarm_history, 55
describe_alarms, 55
describe_alarms_for_metric, 55
describe_algorithm, 228, 281
describe_analysis, 244
describe_analysis_permissions, 244
describe_analysis_schemes, 50
describe_anomaly_detectors, 55
describe_app, 281
describe_app_image_config, 281
describe_application, 19, 186, 187
describe_application_snapshot, 187
describe_application_versions, 142
describe_applications, 142
describe_apps, 220
describe_archive, 57, 153
describe_artifact, 281
describe_assessment_runs, 180
describe_assessment_targets, 180
describe_assessment_templates, 180
describe_association, 313
describe_association_execution_targets, 313
describe_association_executions, 313
describe_attachment, 321, 323
describe_attack, 305
describe_attack_statistics, 305
describe_audit_stream_configuration, 339
describe_auto_ml_job, 281
describe_auto_scaling_groups, 27
describe_auto_scaling_instances, 27
describe_auto_scaling_notification_types, 27
describe_automation_executions, 313
describe_automation_step_executions, 313
describe_availability_monitor_test, 317
describe_availability_options, 50
describe_availability_zones, 122
describe_available_patches, 313
describe_backup, 115
describe_backup_job, 31
describe_backup_policy, 135
describe_backup_vault, 31
describe_backups, 49, 158, 223
describe_bandwidth_rate_limit, 317
describe_bandwidth_rate_limit_schedule, 317
describe_batch_inference_job, 228
describe_batch_predictions, 205
describe_broker, 213
describe_broker_engine_types, 213
describe_broker_instance_options, 213
describe_budget, 36
describe_budget_action, 36
describe_budget_action_histories, 36
describe_budget_actions_for_account, 36
describe_budget_actions_for_budget, 36
describe_budget_performance_history, 36
describe_budgets, 36
describe_bundle_tasks, 122
describe_byoip_cidrs, 122
describe_cache, 317
describe_cache_clusters, 140
describe_cache_engine_versions, 140
describe_cache_parameter_groups, 140
describe_cache_parameters, 140
describe_cache_security_groups, 140
describe_cache_subnet_groups, 140
describe_cachedi_scsi_volumes, 317
describe_campaign, 228
describe_capacity_providers, 133
describe_capacity_reservations, 122
describe_carrier_gateways, 122
describe_cases, 321, 323
describe_certificate, 6, 109
describe_certificate_authority, 7
describe_certificate_authority_audit_report, 7
describe_certificates, 112, 250
describe_change_set, 42
describe_chap_credentials, 317
describe_classic_link_instances, 122
describe_client_properties, 341
describe_client_vpn_authorization_rules, 122
describe_client_vpn_connections, 122
describe_client_vpn_endpoints, 122
describe_client_vpn_routes, 122
describe_client_vpn_target_networks, 122
describe_cluster, 137, 151, 182
describe_cluster_db_revisions, 255
describe_cluster_operation, 182
describe_cluster_parameter_groups, 255
describe_cluster_parameters, 255
describe_cluster_security_groups, 255
describe_cluster_snapshots, 255
describe_cluster_subnet_groups, 255
describe_cluster_tracks, 255
describe_cluster_versions, 255
describe_clusters, 49, 104, 133, 255
describe_code_coverages, 63
describe_code_repository, 281
describe_coip_pools, 122
describe_db_parameters, 217, 250
describe_db_proxies, 250
describe_db_proxy_target_groups, 250
describe_db_proxy_targets, 250
describe_db_security_groups, 250
describe_db_snapshot_attributes, 250
describe_db_snapshots, 250
describe_db_subnet_groups, 113, 217, 250
describe_default_cluster_parameters, 255
describe_default_parameters, 104
describe_delivery_channel_status, 93
describe_delivery_channels, 93
describe_delivery_stream, 155
describe_deployments, 220
describe_destinations, 60
describe_device, 281, 339
describe_device_fleet, 281
describe_device_policy_configuration, 339
describe_dhcp_options, 122
describe_dimension_keys, 232
describe_direct_connect_gateway_association_proposals, 107
describe_direct_connect_gateway_associations, 107
describe_direct_connect_gateway_attachments, 107
describe_direct_connect_gateways, 107
describe_directories, 109
describe_directory_configs, 23
describe_document, 313
describe_document_classification_job, 89
describe_document_classifier, 89
describe_document_permission, 313
describe_document_versions, 337
describe_domain, 281, 324, 339
describe_domain_controllers, 109
describe_domain_endpoint_options, 50
describe_domains, 50
describe_dominant_language_detection_job, 89
describe_drt_access, 305
describe_ecs_clusters, 220
describe_edge_packaging_job, 281
describe_effective_instance_associations, 313

describe_effective_patches_for_patch_baseline, 313
describe_effective_policy, 225
describe_egress_only_internet_gateways, 122
describe_elastic_gpus, 123
describe_elastic_ips, 220
describe_elastic_load_balancers, 220
describe_elasticsearch_domain, 144
describe_elasticsearch_domain_config, 144
describe_elasticsearch_domains, 144
describe_elasticsearch_instance_type_limits, 144
describe_emergency_contact_settings, 305
describe_endpoint, 89, 281
describe_endpoint_config, 281
describe_endpoint_group, 165
describe_endpoints, 115
describe_engine_default_cluster_parameters, 113, 217, 250
describe_engine_default_parameters, 140, 217, 250
describe_entities_detection_job, 89
describe_entities_detection_v2_job, 91
describe_entity_aggregates, 174
describe_entity_recognizer, 89
describe_environment_health, 142
describe_environment_managed_action_history, 142
describe_environment_managed_actions, 142
describe_environment_memberships, 37, 38
describe_environment_resources, 142
describe_environment_status, 37, 38
describe_environments, 37, 38, 142
describe_evaluations, 205
describe_event_aggregates, 174
describe_event_bus, 57, 153
describe_event_categories, 113, 217, 250, 255
describe_event_details, 174
describe_event_details_for_organization, 174
describe_event_source, 57, 153
describe_event_subscriptions, 217, 250,
describe_event_topics, 109
describe_event_tracker, 228
describe_event_types, 174
describe_events, 104, 113, 140, 142, 174, 217, 223, 250, 255
describe_events_detection_job, 89
describe_events_for_organization, 173, 174
describe_exclusions, 180
describe_execution, 303
describe_experiment, 281
describe_export, 115
describe_export_image_tasks, 123
describe_export_tasks, 60, 123, 250
describe_expressions, 50
describe_fargate_profile, 137
describe_fast_snapshot_restores, 123
describe_feature_group, 281
describe_feature_transformation, 228
describe_file_system_aliases, 158
describe_file_system_policy, 135
describe_file_systems, 135, 158
describe_filter, 228
describe_findings, 180
describe_fleet_history, 123
describe_fleet_instances, 123
describe_fleet_metadata, 339
describe_fleets, 23, 123
describe_flow_definition, 281
describe_flow_logs, 123
describe_folder_contents, 337
describe_fpga_image_attribute, 123
describe_fpga_images, 123
describe_gateway_information, 317
describe_gateway_route, 21
describe_global_clusters, 250
describe_global_replication_groups, 140
describe_global_settings, 31
describe_global_table, 115
describe_global_table_settings, 115
describe_group, 244
describe_groups, 337
describe_handshake, 225
describe_hapg, 47
describe_health_service_status_for_organization, 174
describe_host_reservation_offerings, 123
describe_host_reservations, 123
describe_hosted_connections, 107
describe_hosts, 123
describe_hsm, 47
describe_hsm_client_certificates, 255
describe_hsm_configurations, 255
describe_hub, 289
describe_human_task_ui, 281
describe_hyper_parameter_tuning_job, 281
describe_iam_instance_profile_associations, 123
describe_iam_policy_assignment, 244
describe_icd10cm_inference_job, 91
describe_id_format, 123
describe_identity, 81
describe_identity_id_format, 123
describe_identity_pool, 81
describe_identity_pool_usage, 87
describe_identity_provider, 84
describe_identity_provider_configuration, 339
describe_identity_usage, 87
describe_image, 281
describe_image_attribute, 123
describe_image_builders, 23
describe_image_permissions, 23
describe_image_scan_findings, 131
describe_image_version, 281
describe_images, 23, 123, 131
describe_import_image_tasks, 123
describe_import_snapshot_tasks, 123
describe_inbound_cross_cluster_search_connections, 144
describe_index_fields, 50
describe_ingestion, 244
describe_insight_rules, 55
describe_installation_media, 250
describe_instance, 96
describe_instance_associations_status, 313
describe_instance_attribute, 96, 123
describe_instance_credit_specifications, 123
describe_instance_health, 147
describe_instance_information, 313
describe_instance_patch_states, 313
describe_instance_patch_states_for_patch_group, 313
describe_instance_patches, 313
describe_instance_refreshes, 27
describe_instance_status, 123
describe_instance_storage_config, 96
describe_instance_type_offerings, 123
describe_instance_types, 123
describe_instances, 123, 220
describe_instances_health, 142
describe_interconnect_loa, 107
describe_interconnects, 107
describe_interconnects, 107
describe_internet_gateways, 123
describe_inventory_deletions, 313
describe_ip_groups, 341
describe_ipv_6_pools, 123
describe_job, 160, 277
describe_job_definitions, 34
describe_job_flows, 151
describe_job_queues, 34
describe_jobs, 34
describe_key, 190
describe_key_pairs, 123
describe_key_phrases_detection_job, 89
describe_kinesis_streaming_destination, 115
describe_labeling_job, 281
describe_lags, 107
describe_language_model, 327
describe-launch-configurations, 27
describe_launch_template_versions, 123
describe_launch_templates, 123
describe_layers, 220
describe_ldaps_settings, 109
describe_lifecycle_configuration, 135
describe_lifecycle_hook_types, 27
describe_lifecycle_hooks, 27
describe_limits, 115, 184
describe_listener, 165
describe_listener_certificates, 149
describe_listeners, 149
describe_loa, 107
describe_load_balancer_attributes, 147, 149
describe_load_balancer_policies, 147
describe_load_balancer_policy_types, 147
describe_load_balancer_target_groups, 27
describe_load_balancers, 27, 147, 149
describe_load_based_auto_scaling, 220
describe_local_gateway_route_table_virtual_interface_group, 123
describe_local_gateway_route_table_vpc_associations, 123
describe_local_gateway_route_tables, 123
describe_local_gateway_virtual_interface_groups, 123
describe_local_gateway_virtual_interfaces, 123
describe_local_gateways, 123
describe_locations, 107
describe_log_groups, 60
describe_log_pattern, 19
describe_log_streams, 60
describe_logging_status, 256
describe_luna_client, 47
describe_maintenance_start_time, 317
describe_maintenance_window_execution_task_invocations, 313
describe_maintenance_window_execution_tasks, 313
describe_maintenance_window_executions, 313
describe_maintenance_window_schedule, 313
describe_maintenance_window_targets, 313
describe_maintenance_window_tasks, 313
describe_maintenance_windows, 313
describe_maintenance_windows_for_target, 313
describe.managed_prefix_lists, 123
describe_merge_conflicts, 66, 69
describe_mesh, 21
describe_metric_collection_types, 27
describe_metric_filters, 60
describe_ml_models, 205
describe_model, 281
describe_model_bias_job_definition, 281
describe_model_explainability_job_definition,
describe_project, 78, 79, 282
describe_project_versions, 258
describe_projects, 258
describe_protected_resource, 31
describe_protection, 305
describe_protection_group, 305
describe_provisioned_product, 294
describe_provisioned_product_plan, 294
describe_provisioning_artifact, 294
describe_provisioning_parameters, 294
describe_publishing_destination, 172
describe_pull_request_events, 66, 69
describe_queries, 60
describe_query_definitions, 60
describe_quick_connect, 96
describe_rds_arrays, 220
describe_rds_db_instances, 220
describe_receipt_rule, 300
describe_receipt_rule_set, 300
describe_recipe, 228
describe_record, 294
describe_recovery_point, 31
describe_region_settings, 31
describe_regions, 109, 124
describe_registry, 131
describe_remediation_configurations, 94
describe_remediation_exceptions, 94
describe_remediation_execution_status, 94
describe_replay, 57, 153
describe_replication_groups, 140
describe_report_creation, 266
describe_report_definitions, 99
describe_repositories, 131
describe_reserved_cache_nodes, 140
describe_reserved_cache_nodes_offerings, 140
describe_reserved_db_instances, 251
describe_reserved_db_instances_offerings, 251
describe_reserved_elasticsearch_instance_offerings, 144
describe_reserved_elasticsearch_instances, 144
describe_reserved_instances, 124
describe_reserved_instances_modifications, 124
describe_reserved_instances_offerings, 124
describe_reserved_node_offerings, 256
describe_reserved_nodes, 256
describe_resize, 256
describe_resource_groups, 180
describe_resource_permissions, 337
describe_resource_policies, 60
describe_resource_server, 85
describe_restore_job, 31
describe_retention_configurations, 94
describe_risk_configuration, 85
describe_root_folders, 337
describe_route, 21
describe_route_tables, 124
describe_routing_profile, 96
describe_rule, 57, 153
describe_rules, 149
describe_rules_packages, 180
describe_rx_norm_inference_job, 91
describe_scalable_targets, 17
describe_scaling_activities, 17, 28
describe_scaling_parameters, 50
describe_scaling_plan_resources, 30
describe_scaling_plans, 30
describe_scaling_policies, 17
describe_scaling_process_types, 28
describe_scheduled_actions, 17, 28, 256
describe_scheduled_instance_availability, 124
describe_scheduled_instances, 124
describe_schema, 228
describe_secret, 287
describe_security_configuration, 151
describe_security_group_references, 124
describe_security_groups, 124
describe_sentiment_detection_job, 89
describe_servers, 223
describe_service_access_policies, 50
describe_service_action, 294
describe_service_action_execution_parameters, 294
describe_service_errors, 220
describe_service_updates, 140
<table>
<thead>
<tr>
<th>Description</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>describe_services</td>
<td>133, 242, 321, 323</td>
</tr>
<tr>
<td>describe_sessions</td>
<td>23, 313</td>
</tr>
<tr>
<td>describe_severity_levels</td>
<td>321, 323</td>
</tr>
<tr>
<td>describe_shared_directories</td>
<td>109</td>
</tr>
<tr>
<td>describe_smb_file_shares</td>
<td>318</td>
</tr>
<tr>
<td>describe_smb_settings</td>
<td>318</td>
</tr>
<tr>
<td>describe_snapshot_attribute</td>
<td>124</td>
</tr>
<tr>
<td>describe_snapshot_copy_grants</td>
<td>256</td>
</tr>
<tr>
<td>describe_snapshot_schedule</td>
<td>318</td>
</tr>
<tr>
<td>describe_snapshots</td>
<td>23, 318</td>
</tr>
<tr>
<td>describe_solution</td>
<td>228</td>
</tr>
<tr>
<td>describe_solution_version</td>
<td>228</td>
</tr>
<tr>
<td>describe_source_regions</td>
<td>251</td>
</tr>
<tr>
<td>describe_spot_datafeed_subscription</td>
<td>124</td>
</tr>
<tr>
<td>describe_spot_fleetfeed_instances</td>
<td>124</td>
</tr>
<tr>
<td>describe_spot_fleet_feed_request_history</td>
<td>124</td>
</tr>
<tr>
<td>describe_spot_fleet_requests</td>
<td>124</td>
</tr>
<tr>
<td>describe_spot_instance_requests</td>
<td>124</td>
</tr>
<tr>
<td>describe_spot_price_history</td>
<td>124</td>
</tr>
<tr>
<td>describe_ssl_policies</td>
<td>149</td>
</tr>
<tr>
<td>describe_stack_drift_detection_status</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_events</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_instance</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_provisioning_parameters</td>
<td>220</td>
</tr>
<tr>
<td>describe_stack_resource</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_resource_drifts</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_resources</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_set</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_set_operation</td>
<td>42</td>
</tr>
<tr>
<td>describe_stack_summary</td>
<td>220</td>
</tr>
<tr>
<td>describe_stacks</td>
<td>23, 42, 220</td>
</tr>
<tr>
<td>describe_stale_security_groups</td>
<td>124</td>
</tr>
<tr>
<td>describe_standards</td>
<td>290</td>
</tr>
<tr>
<td>describe_standards_controls</td>
<td>290</td>
</tr>
<tr>
<td>describe_state_machine</td>
<td>303</td>
</tr>
<tr>
<td>describe_state_machine_for_execution</td>
<td>303</td>
</tr>
<tr>
<td>describe_step</td>
<td>151</td>
</tr>
<tr>
<td>describe_storage</td>
<td>256</td>
</tr>
<tr>
<td>describestoredi_scsi_volumes</td>
<td>318</td>
</tr>
<tr>
<td>describe_stream</td>
<td>117, 184</td>
</tr>
<tr>
<td>describe_stream_consumer</td>
<td>184</td>
</tr>
<tr>
<td>describe_stream_processor</td>
<td>258</td>
</tr>
<tr>
<td>describe_stream_summary</td>
<td>184</td>
</tr>
<tr>
<td>describe_studio</td>
<td>151</td>
</tr>
<tr>
<td>describe_subnet_groups</td>
<td>104</td>
</tr>
<tr>
<td>describe_subnets</td>
<td>124</td>
</tr>
<tr>
<td>describe_subscribed_workteam</td>
<td>282</td>
</tr>
<tr>
<td>describe_subscribers_for_notification</td>
<td>36</td>
</tr>
<tr>
<td>describe_subscription</td>
<td>305</td>
</tr>
<tr>
<td>describe_subscription_filters</td>
<td>60</td>
</tr>
<tr>
<td>describe_suggesters</td>
<td>50</td>
</tr>
<tr>
<td>describe_table</td>
<td>115</td>
</tr>
<tr>
<td>describe_table_replica_auto_scaling</td>
<td>115</td>
</tr>
<tr>
<td>describe_table_restore_status</td>
<td>256</td>
</tr>
<tr>
<td>describe_tag_option</td>
<td>294</td>
</tr>
<tr>
<td>describe_tags</td>
<td>28, 107, 124, 135, 147, 149, 205, 256, 341</td>
</tr>
<tr>
<td>describe_tape_archives</td>
<td>318</td>
</tr>
<tr>
<td>describe_tape_recovery_points</td>
<td>318</td>
</tr>
<tr>
<td>describe_tapes</td>
<td>318</td>
</tr>
<tr>
<td>describe_target_group_attributes</td>
<td>149</td>
</tr>
<tr>
<td>describe_target_groups</td>
<td>149</td>
</tr>
<tr>
<td>describe_target_health</td>
<td>149</td>
</tr>
<tr>
<td>describe_task_definition</td>
<td>133</td>
</tr>
<tr>
<td>describe_task_sets</td>
<td>133</td>
</tr>
<tr>
<td>describe_tasks</td>
<td>133</td>
</tr>
<tr>
<td>describe_template</td>
<td>244</td>
</tr>
<tr>
<td>describe_template_alias</td>
<td>244</td>
</tr>
<tr>
<td>describe_template_permissions</td>
<td>244</td>
</tr>
<tr>
<td>describe_test_cases</td>
<td>62, 63</td>
</tr>
<tr>
<td>describe_text_translation_job</td>
<td>329</td>
</tr>
<tr>
<td>describe_theme</td>
<td>244</td>
</tr>
<tr>
<td>describe_theme_alias</td>
<td>244</td>
</tr>
<tr>
<td>describe_theme_permissions</td>
<td>244</td>
</tr>
<tr>
<td>describe_time_based_auto_scaling</td>
<td>220</td>
</tr>
<tr>
<td>describe_time_to_live</td>
<td>115</td>
</tr>
<tr>
<td>describe_topics_detection_job</td>
<td>89</td>
</tr>
<tr>
<td>describe_traffic_mirror_filter</td>
<td>124</td>
</tr>
<tr>
<td>describe_traffic_mirror_sessions</td>
<td>124</td>
</tr>
<tr>
<td>describe_traffic_mirror_targets</td>
<td>124</td>
</tr>
<tr>
<td>describe_trails</td>
<td>53</td>
</tr>
<tr>
<td>describe_training_job</td>
<td>282</td>
</tr>
<tr>
<td>describe_transform_job</td>
<td>282</td>
</tr>
<tr>
<td>describe_transit_gateway_attachments</td>
<td>124</td>
</tr>
<tr>
<td>describe_transit_gateway_connect_peers</td>
<td>124</td>
</tr>
</tbody>
</table>
describe_transit_gateway_connects, 124
describe_transit_gateway_multicast_domains, 124
describe_transit_gateway_peering_attachments, 124
describe_transit_gateway_route_tables, 124
describe_transit_gateway_vpc_attachments, 124
describe_transit_gateways, 124
describe_trial, 282
describe_trial_component, 282
describe_trusted_advisor_check_refresh_statuses, 322, 323
describe_trusted_advisor_check_result, 322, 323
describe_trusted_advisor_check_summaries, 322, 323
describe_trusted_advisor_checks, 321–323
describe_trusts, 109
describe_type, 42
describe_type_registration, 42
describe_update, 137
describe_update_actions, 140
describe_upload_buffer, 318
describe_usage_limits, 256
describe_usage_report_subscriptions, 23
describe_user, 96, 213, 244
describe_user_groups, 140
describe_user_hierarchy_group, 96
describe_user_hierarchy_structure, 96
describe_user_import_job, 85
describe_user_pool, 85
describe_user_pool_client, 85
describe_user_pool_domain, 85
describe_user_profile, 78, 79, 282
describe_user_profiles, 221
describe_user_stack_associations, 23
describe_users, 23, 140, 337
describe_valid_db_instance_modifications, 217, 251
describe_vault, 160
describe_virtual_gateway, 21
describe_virtual_gateways, 107
describe_virtual_interfaces, 107
describe_virtual_node, 21
describe_virtual_router, 21
describe_virtual_service, 21
describe_voices, 240
describe_volume_attribute, 124
describe_volume_status, 124
describe_volumes, 124, 221
describe_volumes_modifications, 124
describe_vpc_attribute, 124
describe_vpc_classic_link, 124
describe_vpc_classic_link_dns_support, 124
describe_vpc_endpoint_connection_notifications, 124
describe_vpc_endpoint_connections, 124
describe_vpc_endpoint_service_configurations, 124
describe_vpc_endpoint_service_permissions, 124
describe_vpc_endpoint_services, 124
describe_vpc_endpoints, 124
describe_vpc_peering_connections, 125
describe_vpcs, 125
describe_vpn_connections, 125
describe_vpn_gateways, 125
describe_vtl_devices, 318
describe_website_certificate_authority, 339
describe_workflow_execution, 324
describe_workflow_type, 324
describe_workforce, 282
describe_working_storage, 318
describe_workspace_bundles, 341
describe_workspace_directories, 341
describe_workspace_image_permissions, 341
describe_workspace_images, 341
describe_workspace_snapshots, 341
describe_workspaces, 341
describe_workspaces_connection_status, 341
describe_workteam, 282
detach_certificate_from_distribution, 202
detach_classic_link_vpc, 125
detach_disk, 202
detach_elastic_load_balancer, 221
detach_from_index, 40
detach_group_policy, 176
disconnect_instances, 28
disconnect_instances_from_load_balancer, 202
disconnect_internet_gateway, 125
disconnect_load_balancer_from_subnets, 147
disconnect_load_balancer_target_groups, 28
disconnect_load_balancers, 28
disconnect_network_interface, 125
disconnect_object, 40
disconnect_policy, 40, 225
disconnect_role_policy, 177
disconnect_static_ip, 202
disconnect_typed_link, 40
disconnect_user_policy, 177
disconnect_volume, 125, 318
disconnect_vpn_gateway, 125
detect_custom_labels, 258
detect_document_text, 326
detect_dominant_language, 89
detect_entities, 89, 91
detect_entities_v2, 91
detect_faces, 258
detect_key_phrases, 89
detect_labels, 258
detect_moderation_labels, 258
detect_phi, 91
detect_pii_entities, 89
detect_protective_equipment, 258
detect_sentiment, 89
detect_stack_drift, 42
detect_stack_resource_drift, 42
detect_stack_set_drift, 42
detect_syntax, 89
detect_text, 258
directconnect, 105
directoryservice, 107
disable_addon, 202
disable_alarm_actions, 55
disable_availability_zones_for_load_balancers, 147
disable_aws_organizations_access, 294
disable_aws_service_access, 225
disable_client_authentication, 109
disable_directory, 40
disable_domain_auto_renew, 270
disable_domain_transfer_lock, 270
disable_ebs_encryption_by_default, 125
disable_enhanced_monitoring, 184

disable_fast_snapshot_restores, 125
disable_gateway, 318
disable_health_service_access_for_organization, 174
disable_hosted_zone_dnssec, 268
disable_import_findings_for_product, 290
disable_insight_rules, 55
disable_key, 190
disable_key_rotation, 190
disable_kinesis_streaming_destination, 115
disable_ldaps, 109
disable_logging, 256
disable_metrics_collection, 28
disable_organization_admin_account, 172, 290
disable_policy_type, 225
disable_proactive_engagement, 305
disable_radius, 109
disable_rule, 57, 153
disable_sagemaker_servicecatalog_portfolio, 282
disable_security_hub, 290
disable_snapshot_copy, 256
disable_sso, 109
disable_stage_transition, 75, 77
disable_transit_gateway_route_table_propagation, 125
disable_user, 23
disable_vgw_route_propagation, 125
disable_vpc_classic_link, 125
disable_vpc_classic_link_dns_support, 125
disassociate_address, 125
disassociate_admin_account, 157
disassociate_approval_rule_template_from_repository, 67, 69
disassociate_approved_origin, 96
disassociate_budget_from_resource, 294
disassociate_client_vpn_target_network, 125
disassociate_connection_alias, 341
disassociate_connection_from_lag, 107
disassociate_domain, 339
disassociate_drt_log_bucket, 305
disassociate_drt_role, 305
disassociate_elastic_ip, 221
disassociate_enclave_certificate_iam_role, 125
disassociate_environment_operations_role, 142
disassociate_file_system_aliases, 158
disassociate_fleet, 23
disassociate_from_master_account, 172, 290
disassociate_global_replication_group, 140
disassociate_health_check, 305
disassociate_iam_instance_profile, 125
disassociate_instance_storage_config, 97
disassociate_ip_groups, 341
disassociate_kms_key, 60
disassociate_lambda_function, 97
disassociate_lex_bot, 97
disassociate_member_account, 207
disassociate_members, 172, 290
disassociate_node, 224
disassociate_principal_from_portfolio, 294
disassociate_product_from_portfolio, 294
disassociate_qualification_from_worker, 214
disassociate_resolver_endpoint_ip_address, 272
disassociate_resolver_query_log_config, 272
disassociate_resolver_rule, 272
disassociate_resource_share, 247
disassociate_resource_share_permission, 247
disassociate_route_table, 125
disassociate_routing_profile_queues, 97
disassociate_s3_resources, 207
disassociate_security_key, 97
disassociate_service_action_from_provisioning_artifact, 294
disassociate_service_quota_template, 298
disassociate_subnet_cidr_block, 125
disassociate_tag_option_from_resource, 294
disassociate_team_member, 78, 79

disassociate_transit_gateway_multicast_domain, 125
disassociate_transit_gateway_route_table, 125
disassociate_trial_component, 282
disassociate_vpc_cidr_block, 125
disassociate_vpc_from_hosted_zone, 268
disassociate_web_acl, 334
disassociate_website_authorization_provider, 339
disassociate_website_certificate_authority, 339
disconnect_custom_key_store, 190
discover_input_schema, 186, 187
discover_instances, 297
discover_poll_endpoint, 133
dissociate_package, 144
dlm, 110
docdb, 111
domain_metadata, 307
download_db_log_file_portion, 251
download_default_key_pair, 202
download_file, 274
dynamodb, 113
dynamodbstreams, 116
e2, 118
e2instanceconnect, 128
ecr, 129
ecs, 132
efs, 134
eks, 136
elasticache, 138
elasticbeanstalk, 141
elasticsearchservice, 143
elb, 145
elbv2, 147
emr, 150
enable_add_on, 202
enable_alarm_actions, 55
enable_all_features, 225
enable_available_zones_for_load_balancer, 147
enable_aws_organizations_access, 294
enable_aws_service_access, 225
enable_client_authentication, 109
enable_directory, 40
enable_domain_auto_renew, 270
enable_domain_transfer_lock, 270
INDEX

enable_ebs_encryption_by_default, 125
enable_enhanced_monitoring, 184
enable_fast_snapshot_restores, 125
enable_health_service_access_for_organization, 174
enable_hosted_zone_dnssec, 268
enable_import_findings_for_product, 290
enable_insight_rules, 55
enable_key, 190
enable_key_rotation, 190
enable_kinesis_streaming_destination, 115
enable_ldaps, 109
enable_logging, 256
enable_metrics_collection, 28
enable_mfa_device, 177
enable_organization_admin_account, 172, 290
enable_policy_type, 225
enable_proactive_engagement, 305
enable_radius, 109
enable_rule, 57, 153
enable_sagemaker_servicecatalog_portfolio, 282
enable_security_hub, 290
enable_sharing_with_aws_organization, 247
enable_snapshot_copy, 256
enable_sso, 109
enable_stage_transition, 75, 77
enable_transit_gateway_route_table_propagation, 125
enable_user, 23
enable_vgw_route_propagation, 125
enable_volume_io, 125
enable_vpc_classic_link, 125
enable_vpc_classic_link_dns_support, 125
encrypt, 189, 190
enter_standby, 28
estimate_template_cost, 42
evaluate_expression, 103
evaluate_pull_request_approval_rules, 66, 69
eventbridge, 152
execute_budget_action, 36
execute_change_set, 42
execute_policy, 28
execute_provisioned_product_plan, 294
execute_provisioned_product_service_action, 294
execute_sql, 253
execute_statement, 115, 253
execute_transaction, 115
exit_standby, 28
expire_session, 23
export_api, 14
export_backup_plan_template, 31
export_certificate, 6
export_client_vpn_client_certificate_revocation_list, 125
export_client_vpn_client_configuration, 125
export_image, 125
export_server_engine_attribute, 224
export_snapshot, 202
export_table_to_point_in_time, 115
export_transit_gateway_routes, 125
extend_license_consumption, 199
failover_db_cluster, 113, 217, 251
failover_global_replication_group, 140
filter_log_events, 60
firehose, 154
flush_stage_authorizers_cache, 10
flush_stage_cache, 10
fms, 156
forget_device, 85
forgot_password, 85
fsx, 157
generate_client_certificate, 10
generate_credential_report, 177
generate_data_key, 189, 190
generate_data_key_pair, 190
generate_data_key_pair_without_plaintext, 190
generate_data_key_without_plaintext, 189, 190
generate_data_set, 208
generate_organizations_access_report, 177
generate_random, 190
generate_service_last_accessed_details, 177
generate_service_last_accessed_details, 177
get_access_key_info, 320
INDEX

get_access_key_last_used, 177
get_access_point, 277
get_access_point_policy, 277
get_access_point_policy_status, 277
get_access_token, 199
get_account, 10, 237
get_account_authorization_details, 177
get_account_balance, 214
get_account_limit, 268
get_account_password_policy, 177
get_account_send_enabled, 300
get_account_settings, 192
get_account_summary, 177
get_active_names, 202
get_activity_task, 303
get_adm_channel, 234
get_admin_account, 157
get_aggregate_compliance_details_by_config_rule, 94
get_aggregate_config_rule_compliance_summary, 94
get_aggregate_discovered_resource_counts, 94
get_aggregate_resource_config, 94
get_alarms, 202
get_alias, 192
get_anomalies, 101
get_anomaly_monitors, 101
get_anomaly_subscriptions, 101
get_api, 14
get_api_key, 10
get_api_keys, 10
get_api_mapping, 14
get_api_mappings, 14
get_apis, 14
get_apns_channel, 234
get_apns_sandbox_channel, 234
get_apns_voip_channel, 234
get_apns_voip_sandbox_channel, 234
get_app, 234
get_application, 73, 292
get_application_date_range_kpi, 234
get_application_policy, 292
get_application_revision, 73
get_application_settings, 234
get_applied_schema_version, 40
get_approval_rule_template, 67, 69
get_apps, 234
get_apps_list, 157
get_assessment_report, 180
get_assignment, 214
get_associated_enclave_certificate_iam_roles, 125
get Associated_ipv_6_pool_cidrs, 125
get_association_for_service_quota_template, 298
get_attribute_values, 241, 242
get_attributes, 307
get_authorization_token, 131
get_authorizer, 10, 14
get_authorizers, 10, 14
get_auto_snapshots, 202
get_automation_execution, 313
get_aws_default_service_quota, 298
get_aws_organizations_access_status, 315
get.backup_plan, 31
get.backup_plan_from_json, 31
get.backup_plan_from_template, 31
get.backup_selection, 31
get.backup_vault_access_policy, 31
get.backup_vault_notifications, 31
get.baidu_channel, 234
get_base_path_mapping, 10
get_base_path_mappings, 10
get.batch_prediction, 205
get.blacklist_reports, 237
get.blob, 65, 69
get_block_public_access_configuration, 151
get_blueprints, 202
get_bootstrap_brokers, 182
get_bot, 195
get_bot_alias, 195
get_bot_aliases, 195
get_bot_channel_association, 195
get_bot_channel_associations, 195
get_bot_versions, 195
get.bots, 195
get.branch, 65, 69
get.bucket, 277
get.bucket_accelerate_configuration, 274
get_bucket_acl, 274
get_bucket_analytics_configuration, 274
INDEX

get_connection, 12, 168
get_connection_status, 313
get_connections, 168
get_console_output, 125
get_console_screenshot, 125
get_contact_attributes, 97
get_contact_methods, 202
get_contact_reachability_status, 270
get_container_api_metadata, 202
get_container_images, 202
get_container_log, 202
get_container_service_deployments, 202
get_container_service_metric_data, 202
get_container_service_powers, 202
get_container_services, 202
get_content_moderation, 258
get_context_keys_for_custom_policy, 177
get_context_keys_for_principal_policy, 177
get_cost_and_usage, 101
get_cost_and_usage_with_resources, 101
get_cost_categories, 101
get_cost_forecast, 101
get_crawler, 168
get_crawler_metrics, 168
get_crawlers, 168
get_credential_report, 177
get_credentials_for_identity, 81
get_csv_header, 85
get_current_metric_data, 97
get_current_user, 337
get_custom_verification_email_template, 300
get_dashboard, 55
get_dashboard_embed_url, 244
get_data_catalog, 25
get_data_catalog_encryption_settings, 168
get_data_retrieval_policy, 160
get_data_source, 205
get_database, 25, 168
get_databases, 168
get_dataflow_graph, 168
get_dedicated_ip, 237
get_dedicated_ips, 237
get_default_credit_specification, 125
get_default_patch_baseline, 313
get_deliverability_dashboard_options, 237
get_deliverability_test_report, 237
get_deployable_patch_snapshot_for_instance, 313
get_deployment, 10, 14, 73
get_deployment_config, 73
get_deployment_group, 73
get_deployment_instance, 73
get_deployment_target, 73
get_deployments, 10, 14
get_detector, 172
get_dev_endpoint, 168
get_dev_endpoints, 168
get_device, 85
get_device_fleet_report, 282
get_differences, 65, 70
get_dimension_values, 101
get_directory, 40
get_directory_limits, 109
get_discovered_resource_counts, 94
get_disk, 202
getDiskSnapshot, 203
get_disk_snapshots, 203
get_disks, 203
get_distribution, 45
get_distribution_bundles, 203
get_distribution_config, 45
get_distribution_latest_cache_reset, 203
get_distribution_metric_data, 203
get_distributions, 203
get_dnssec, 268
get_document, 314, 337
get_document_analysis, 326
get_document_path, 337
get_document_text_detection, 326
get_document_version, 338
get.documentation_part, 10
get_documentation_parts, 10
generate_documentation_parts, 10
generate_documentation_versions, 10
generate_documentation_versions, 10
generate_domain_deliverability_campaign, 237
generate_domain_detail, 270
generate_domain_names, 10, 14
get_domain_names, 10, 14
get_domain_statistics_report, 237
get_domain_suggestions, 270
get_domains, 203
get_download_url_for_layer, 131
get_ebs_default_kms_key_id, 125
get_ebs_encryption_by_default, 125
get_email_channel, 234
get_email_identity, 237
get_email_template, 234
gerole_enabled_standards, 290
get_encryption_config, 343
get_endpoint, 234
gateway_endpoint_attributes, 308
get_entitlements, 210
gateway_execution_history, 303
get_export, 10, 195
get_export_job, 234
get_export_jobs, 234
get_export_snapshot_records, 203
get_face_detection, 258
gateway_face_search, 258
gateway_facet, 40
get_federation_token, 97, 320
get_field_level_encryption, 45
get_field_level_encryption_config, 45
get_field_level_encryption_profile, 45
get_field_level_encryption_profile_config, 45
get_file, 65, 70
get_file_upload_url, 214
get_filter, 172
get_findings, 172, 288, 290
get_findings_statistics, 172
get_folder, 65, 70, 338
get_folder_path, 338
get_function, 192
get_function_code_signing_config, 192
get_function_concurrency, 192
get_function_configuration, 192
get_function_event_invoke_config, 192
gateway_gateway_response, 10
gateway_gateway_responses, 10
gateway_gcm_channel, 234
gateway_geo_location, 268
gateway_geo_match_set, 331, 344
gateway_grant, 199
gateway_group, 85, 177, 260, 343
get_function_configuration, 260
gateway_group_policy, 177
gateway_group_query, 260
gateway_groups, 343
get_groups_for_capacity_reservation, 125
gateway_health_check, 268
gateway_health_check_count, 268
gateway_health_check_last_failure_reason, 268
gateway_health_check_status, 268
get_hit, 214
get_host_reservation_purchase_preview, 125
get_hosted_zone, 268
get_hosted_zone_count, 268
get_hosted_zone_limit, 268
get_hostname_suggestion, 221
get_id, 81
get_identity_dkim_attributes, 300
get_identity_mail_from_domain_attributes, 300
get_identity_notification_attributes, 300
get_identity_policies, 300
get_identity_pool_configuration, 87
get_identity_pool_roles, 81
get_identity_provider_by_identifier, 85
get_identity_verification_attributes, 300
get_idp, 195
get_import_job, 234
get_import_jobs, 234
get_insight, 343
get_insight_events, 343
get_insight_impact_graph, 343
get_insight_results, 290
get_insight_rule_report, 55
get_insight_selectors, 53
get_insight_summaries, 343
get_insights, 290
get_instance, 203, 297
get_instance_access_details, 203
get_instance_metric_data, 203
get_instance_port_states, 203
get_instance_profile, 177
get_instance_snapshot, 203
get_instance_snapshots, 203
get_instance_state, 203
get_instances, 203
get_instances_health_status, 297
get_integration, 10, 14
get_integration_response, 10, 14
get_integrations, 14
get_intent, 195
get_intent_versions, 195
get_intents, 195
get_invalidation, 45
get_inventory, 314
get_inventory_schema, 314
get_invitations_count, 172, 290
get_ip_set, 172, 331, 334
get_item, 115
get_job, 168
get_job_bookmark, 168
get_job_details, 76, 77
get_job_output, 160
get_job_run, 168
get_job_runs, 168
get_job_tagging, 277
get_jobs, 168
get_journey, 234
get_journey_date_range_kpi, 234
get_journey_execution_activity_metrics, 234
get_journey_execution_metrics, 234
get_key_group, 45
get_key_group_config, 45
get_key_pair, 203
get_key_pairs, 203
get_key_policy, 190
get_key_rotation_status, 190
get_label_detection, 258
get_launch_template_data, 125
get_layer_version, 192
get_layer_version_by_arn, 192
get_layer_version_policy, 192
get_lexicon, 240
get_license, 199
get_license_configuration, 199
get_license_usage, 199
get_lifecycle_policies, 111
get_lifecycle_policy, 111, 131
get_lifecycle_policy_preview, 131
get_link_attributes, 40
get_load_balancer, 203
get_load_balancer_metric_data, 203
get_load_balancer_tls_certificates, 203
get_load_balancers, 203
get_log_events, 60
get_log_group_fields, 60
get_log_record, 60
get_logging_configuration, 331, 334
get_login_profile, 177
get_maintenance_window, 314
get_maintenance_window_execution, 314
get_maintenance_window_execution_task, 314
get_maintenance_window_execution_task_invocation, 314
get_maintenance_window_task, 314
get_managed_prefix_list_associations, 125
get_managed_prefix_list_entries, 126
get_managed_scaling_policy, 151
get_mapping, 168
get_master_account, 172, 290
get_medical_transcription_job, 327
get_medical_vocabulary, 328
get_member_detectors, 172
get_members, 172, 290
get_merge_commit, 66, 70
get_merge_conflicts, 66, 70
get_merge_options, 66, 70
get_method, 10
get_method_response, 10
get_metric_data, 55, 97
get_metric_statistics, 55
get_metric_widget_image, 55
get_ml_model, 205
get_ml_task_run, 168
get_ml_task_runs, 168
get_ml_transform, 168
get_ml_transforms, 168
get_model, 10, 14
get_model_package_group_policy, 282
INDEX

get_model_template, 10, 14
get_models, 10, 14
get_monitoring_subscription, 45
get_named_query, 25
get_namespace, 297
get_notification_channel, 157
get_object, 275
get_object_acl, 275
get_object_attributes, 40
get_object_information, 40
get_object_legal_hold, 275
get_object_lock_configuration, 275
get_object_tagging, 275
get_object_torrent, 275
get_on_premises_instance, 73
get_open_id_connect_provider, 177
get_open_id_token, 81
get_open_id_token_for_developer_identity, 81
get_operation, 203, 297
get_operation_detail, 270
get_operations, 203
get_operations_for_resource, 203
get_ops_item, 314
get_ops_metadata, 314
get_ops_summary, 314
get_organization_config_rule_detailed_status, 94
get_organization_conformance_pack_detailed_status, 94
get_organizations_access_report, 177
get_origin_request_policy, 45
get_orign_request_policy_config, 45
get_package_version_history, 144
get_parallel_data, 329
get_parameter, 314
get_parameter_history, 314
get_parameters, 314
get_parameters_by_path, 314
get_parameters_for_import, 190
get_partition, 168
get_partitions, 168
get_password_data, 126
get_patch_baseline, 314
get_patch_baseline_for_patch_group, 314
get_permission, 247
get_permission_policy, 331, 334
get_person_tracking, 258
get_personalized_ranking, 230
get_pipeline, 74, 75, 77
get_pipeline_definition, 103
get_pipeline_execution, 74, 77
get_pipeline_state, 74, 75, 77
get_plan, 168
get_platform_application_attributes, 308
get_policy, 7, 157, 177, 192
get_policy_version, 177
get_products, 241, 242
get_protection_status, 157
get_protocols_list, 157
get_provisioned_concurrency_config, 192
get_provisioned_product_outputs, 294
get_public_access_block, 275, 277
get_public_key, 45, 190
get_public_key_config, 45
get_pull_request, 66, 70
get_pull_request_approval_states, 66, 70
get_pull_request_override_state, 66, 70
get_push_template, 234
get_qualification_score, 214
get_qualification_type, 214
get_rate_based_rule, 331
get_rate_based_rule_managed_keys, 331, 334
get_rate_based_rule_execution, 25
get_query_logging_config, 268
get_query_results, 25, 60
get_queue_attributes, 311
get_queue_url, 311
get_random_password, 287
get_realtime_log_config, 45
get_recommendations, 230
get_recommendations, 230
get_recommendation, 230
get_recommendation, 230
get_recommendations, 230
get_records, 117, 184
get_recovery_point_restore_metadata, 32
get_regex_match_set, 331, 334
get_regex_pattern_set, 331, 334
get_regions, 203
get_registry, 168
get_registry_policy, 131
get_relational_database, 203
get_relational_database_blueprints, 203
get_relational_database_bundles, 203
get_relational_database_events, 203
get_relational_database_log_events, 203
get_relational_database_log_streams, 203
get_relational_database_master_user_password, 203
get_relational_database_metric_data, 203
get_relational_database_parameters, 203
get_relational_database_snapshot, 203
get_relational_database_snapshots, 203
get_relational_databases, 203
get_report_group_trend, 63
get_repository, 65, 70
get_repository_policy, 131
get_repository_triggers, 68, 70
get_request_validator, 10
get_requested_service_quota_change, 298
get_reservation_coverage, 101
get_reservation_purchase_recommendation, 101
get_reservation_utilization, 101
get_reserved_instances_exchange_quote, 126
get_reserved_node_exchange_offerings, 256
get_resolved_dnssec_config, 272
get_resolved_endpoint, 272
get_resolved_query_log_config, 272
get_resolved_query_log_config_association, 272
get_resolved_query_log_config_policy, 272
generate_rule, 272
genenerate_rule_association, 272
genenerate_rule_policy, 272
gen_resource, 10
gen_resource_config_history, 94
gen_resource_metrics, 232
gen_resource_policies, 168, 247
get_resource_policy, 62, 63, 168, 287
get_resource_share_associations, 247
get_resource_share_invitations, 247
get_resource_shares, 247
get_resources, 10, 266, 338
get_rest_api, 10
gen_reports, 10
get_reusable_delegation_set, 268
get_reusable_delegation_set_limit, 268
generate_rightsizing_recommendation, 101
gen_role, 177
gen_role_policy, 177
generate_route, 15
gen_route_response, 15
gen_route_responses, 15
gen_routes, 15
gen_rule, 331, 334
gen_rule_group, 331, 334
gen_sagemaker_servicecatalog_portfolio_status, 282
gen_saml_provider, 177
gen_sampled_requests, 331, 334
gen_sampling_rules, 343
gen_sampling_statistic_summaries, 343
gen_sampling_targets, 343
gen_savings_plans_coverage, 101
gen_savings_plans_purchase_recommendation, 101
gen_savings_plans_utilization, 101
gen_savings_plans_utilization_details, 101
gen_scaling_plan_resource_forecast_data, 30
gen_schema, 168
gen_schema_as_json, 40
gen_schema_by_definition, 168
gen_schema_version, 168
gen_schema_versions_diff, 168
gen_sdk, 10
gen_sdk_type, 10
gen_sdk_types, 10
gen_search_suggestions, 282
gen_secret_value, 287
gen_security_configuration, 168
gen_security_configurations, 168
gen_segment, 234
INDEX

get_segment_detection, 258
get_segment_export_jobs, 234
get_segment_import_jobs, 234
get_segment_version, 234
get_segment_versions, 234
get_segments, 234
get_send_quota, 301
get_send_statistics, 301
get_server_certificate, 177
get_service, 297
get_service_graph, 343
get_service_last_accessed_details, 177
get_service_last_accessed_details_with_entities, 177
get_service_linked_role_deletion_status, 177
get_service_quota, 298
get_service_quota_increase_request_from_template, 298
get_service_setting, 314
get_service_settings, 199
get_session, 198
get_session_embed_url, 244
get_session_token, 320
get_shard_iterator, 117, 184
get_signing_certificate, 85
get_size_constraint_set, 331, 334
get_slot_type, 195
get_slot_type_versions, 195
get_slot_types, 195
get_sms_attributes, 308
get_sms_channel, 235
get_sms_template, 235
get_snapshot_limits, 109
get_solution_metrics, 228
get_speech_synthesis_task, 240
get_sql_injection_match_set, 331, 334
get_ssh_public_key, 177
get_stack_policy, 42
get_stage, 10, 15
get_stages, 10, 15
get_static_ip, 203
get_static_ips, 203
get_storage_lens_configuration, 277
get_storage_lens_configuration_tagging, 278
get_stored_query, 94
get_streaming_distribution, 45
get_streaming_distribution_config, 45
get_subscription_attributes, 308
get_subscription_state, 305
get_supported_resource_types, 32
get_table, 168
get_table_metadata, 25
get_table_version, 168
get_table_versions, 168
get_tables, 168
get_tag_keys, 266
get_tag_values, 266
get_tags, 10, 15, 101, 169, 260
get_tags, 314, 331, 334
get_template, 43, 301
get_template_summary, 43
get_terminology, 329
get_text_detection, 258
get_third_party_job_details, 76, 77
get_threat_intel_set, 172
get_time_series_service_statistics, 343
get_topic_attributes, 308
get_trace_graph, 343
get_trace_summaries, 343
get_traffic_policy, 268
get_traffic_policy_instance, 268
get_traffic_policy_instance_count, 268
get_trail, 53
get_trail_status, 54
get_transcription_job, 328
get_transit_gateway_attachment_propagations, 126
get_transit_gateway_multicast_domain_associations, 126
get_transit_gateway_prefix_list_references, 126
get_transit_gateway_route_table_associations, 126
get_transit_gateway_route_table_propagations, 126
get_trigger, 169
get_triggers, 169
get_TYPED_LINK_FACET_INFORMATION, 40
get_ui_customization, 85
get_upgrade_history, 145
get_upgrade_status, 145
get_usage, 10
INDEX

kinesisanalytics, 184
kinesisanalyticsv2, 186
kms, 188

label_parameter_version, 314
lambda, 191
leave_organization, 225
lexmodelbuildingsservice, 194
lexruntimeservice, 196
licensemanager, 198
lightsail, 200
list_accelerators, 165
list_accepted_portfolio_shares, 295
list_access_keys, 177
list_access_points, 278
list_account_aliases, 177
list_account_settings, 133
list_accounts, 225
list_accounts_for_parent, 225
list_action_executions, 75, 77
list_action_types, 77
list_actions, 282
list_activated_rules_in_rule_group, 331, 335
list_activities, 303
list_activity_types, 324
list_addons, 137
list_aggregate_discovered_resources, 94
list_algorithms, 282
list_aliases, 190, 193
list_allowed_node_type_modifications, 140
list_analyses, 244
list_app_image_configs, 282
list_application_dependencies, 292
list_application_revisions, 73
list_application_snapshots, 187
list_application_versions, 292
list_applications, 19, 73, 186, 187, 292
list_applied_schema_arns, 40
list_approval_rule_templates, 23
list_approved_origins, 97
list_associations, 282, 314
list_associations_for_license_configuration, 199
list_attached_group_policies, 177
list_attached_indices, 40
list_attached_role_policies, 177
list_attached_user_policies, 177
list_attacks, 305
list_attributes, 133
list_auto_ml_jobs, 282
list_automatic_tape_creation_policies, 318
list_available_management_cidr_ranges, 342
list_available_solution_stacks, 142
list_available_zones, 47
list_aws_default_service_quotas, 298
list_aws_service_access_for_organization, 226
list_backup_jobs, 32
list_backup_plan_templates, 32
list_backup_plan_versions, 32
list_backup_plans, 32
list_backup_selections, 32
list_backup_vaults, 32
list_backups, 115
list_batch_inference_jobs, 228
list_bonus_payments, 214
list_bootstrap_actions, 151
list_branches, 65, 70
list_brokers, 213
list_bucket_analytics_configurations, 275
list_bucket_intelligent_tiering_configurations, 275
list_bucket_inventory_configurations, 275
list_bucket_metrics_configurations, 275
list_buckets, 275
list_budgets_for_resource, 295
list_build_batches, 63
list_build_batches_for_project, 63
list_builds, 62, 64
list_byoip_cidrs, 165
list_byte_match_sets, 332, 335
list_cache_policies, 45
list_campaigns, 228
list_candidates_for_auto_ml_job, 282
list_certificateAuthorities, 7
list_certificates, 6, 109
list_change_sets, 43
list_children, 226
list_closed_workflow_executions, 324
list_cloud_front_origin_access_ids, 45
list_cluster_operations, 182
list_clusters, 133, 137, 151, 182
list_code_repositories, 282
list_code_signing_configs, 193
list_collections, 258
list_command_invocations, 314
list_commands, 314
list_compilation_jobs, 282
list_compliance_items, 314
list_compliance_status, 157
list_compliance_summaries, 314
list_compontents, 19
list_configuration_history, 19
list_configuration_revisions, 182, 213
list_configuration_sets, 237, 239, 301
list_configurations, 182, 213
list_constraints_for_portfolio, 295
list_contact_flows, 97
list_container_instances, 133
list_contexts, 282
list_contributorInsights, 115
list_copy_jobs, 32
list_cost_category_definitions, 101
list_crawlers, 169
list_create_account_status, 226
list_curated_environment_images, 62, 64
list_custom_routing_accelerators, 165
list_custom_routing_endpoint_groups, 165
list_custom_routing_listeners, 165
list_custom_routing_port_mappings, 165
list_custom_routing_port_mappings_by_destination, 165
list_custom_verification_email_templates, 301
list_dashboard_versions, 244
list_dashboards, 55, 244
list_data_catalogs, 25
list_data_quality_job_definitions, 282
list_data_sets, 245
list_data_sources, 245
list_databases, 25
list_dataset_groups, 228
list_dataset_import_jobs, 228
list_datasets, 87, 228
list_dead_letter_source_queues, 311
list_dedicated_ip_pools, 237
list_delegated_administrators, 226
list_delegated_services_for_account, 226
list_deliveryability_test_reports, 237
list_delivery_streams, 155
list_deployment_configs, 73
list_deployment_groups, 73
list_deployment_instances, 73
list_deployment_targets, 73
list_deployments, 73
list_detectors, 172
list_dev_endpoints, 169
list_development_schema_arns, 40
list_device_fleets, 282
list_devices, 85, 282, 339
list_directories, 40
list_discovered_resources, 94
list_distributed_grants, 199
list_distributions, 45
list_distributions_by_cache_policy_id, 45
list_distributions_by_key_group, 45
list_distributions_by_origin_request_policy_id, 45
list_distributions_by_realtime_log_config, 45
list_distributions_by_web_acl_id, 45
list_document_classification_jobs, 89
list_document_classifiers, 89
list_document_metadata_history, 314
list_document_versions, 314
list_documents, 314
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_domain_deliverability_campaigns</td>
<td>237</td>
</tr>
<tr>
<td>list_domain_names</td>
<td>51, 145</td>
</tr>
<tr>
<td>list_domains</td>
<td>270, 282, 307, 324, 339</td>
</tr>
<tr>
<td>list_domains_for_package</td>
<td>145</td>
</tr>
<tr>
<td>list_dominant_language_detection_jobs</td>
<td>89</td>
</tr>
<tr>
<td>list_edge_packaging_jobs</td>
<td>282</td>
</tr>
<tr>
<td>list_elasticsearch_instance_types</td>
<td>145</td>
</tr>
<tr>
<td>list_elasticsearch_versions</td>
<td>145</td>
</tr>
<tr>
<td>list_email_identities</td>
<td>237</td>
</tr>
<tr>
<td>list_enabled_products_for_import</td>
<td>290</td>
</tr>
<tr>
<td>list_endpoint_configs</td>
<td>282</td>
</tr>
<tr>
<td>list_endpoint_groups</td>
<td>165</td>
</tr>
<tr>
<td>list_endpoints</td>
<td>89, 282</td>
</tr>
<tr>
<td>list_endpoints_by_platform_application</td>
<td>308</td>
</tr>
<tr>
<td>list_entities_detection_jobs</td>
<td>89</td>
</tr>
<tr>
<td>list_entities_detection_v2_jobs</td>
<td>91</td>
</tr>
<tr>
<td>list_entities_for_policy</td>
<td>177</td>
</tr>
<tr>
<td>list_entity_recognizers</td>
<td>89</td>
</tr>
<tr>
<td>list_environments</td>
<td>37, 38</td>
</tr>
<tr>
<td>list_event_buses</td>
<td>57, 153</td>
</tr>
<tr>
<td>list_event_source_mappings</td>
<td>193</td>
</tr>
<tr>
<td>list_event_sources</td>
<td>58, 154</td>
</tr>
<tr>
<td>list_event_subscriptions</td>
<td>180</td>
</tr>
<tr>
<td>list_event_trackers</td>
<td>228</td>
</tr>
<tr>
<td>list_events_detection_jobs</td>
<td>89</td>
</tr>
<tr>
<td>list_exclusions</td>
<td>180</td>
</tr>
<tr>
<td>list_executions</td>
<td>303</td>
</tr>
<tr>
<td>list_experiments</td>
<td>282</td>
</tr>
<tr>
<td>list_exports</td>
<td>43, 115</td>
</tr>
<tr>
<td>list_faces</td>
<td>258</td>
</tr>
<tr>
<td>list_facet_attributes</td>
<td>40</td>
</tr>
<tr>
<td>list_facet_names</td>
<td>40</td>
</tr>
<tr>
<td>list_failures_for_license_configuration_operations</td>
<td>199</td>
</tr>
<tr>
<td>list_fargate_profiles</td>
<td>137</td>
</tr>
<tr>
<td>list_feature_groups</td>
<td>282</td>
</tr>
<tr>
<td>list_field_level_encryption_configs</td>
<td>45</td>
</tr>
<tr>
<td>list_field_level_encryption_profiles</td>
<td>45</td>
</tr>
<tr>
<td>list_file_shares</td>
<td>318</td>
</tr>
<tr>
<td>list_filters</td>
<td>172, 228</td>
</tr>
<tr>
<td>list_findings</td>
<td>172, 180</td>
</tr>
<tr>
<td>list_fleets</td>
<td>339</td>
</tr>
<tr>
<td>list_flow_definitions</td>
<td>282</td>
</tr>
<tr>
<td>list_function_event_invoke_configs</td>
<td>193</td>
</tr>
<tr>
<td>list_functions</td>
<td>193</td>
</tr>
<tr>
<td>list_functions_by_code_signing_config</td>
<td>193</td>
</tr>
<tr>
<td>list_gateway_routes</td>
<td>21</td>
</tr>
<tr>
<td>list_gateways</td>
<td>318</td>
</tr>
<tr>
<td>list_geo_locations</td>
<td>268</td>
</tr>
<tr>
<td>list_geo_match_sets</td>
<td>332, 335</td>
</tr>
<tr>
<td>list_git_hub_account_token_names</td>
<td>73</td>
</tr>
<tr>
<td>list_global_tables</td>
<td>115</td>
</tr>
<tr>
<td>list_grants</td>
<td>190</td>
</tr>
<tr>
<td>list_group_memberships</td>
<td>245</td>
</tr>
<tr>
<td>list_group_policies</td>
<td>177</td>
</tr>
<tr>
<td>list_group_resources</td>
<td>260</td>
</tr>
<tr>
<td>list_groups</td>
<td>85, 177, 245, 260</td>
</tr>
<tr>
<td>list_groups_for_user</td>
<td>177</td>
</tr>
<tr>
<td>list_handshakes_for_account</td>
<td>226</td>
</tr>
<tr>
<td>list_handshakes_for_organization</td>
<td>226</td>
</tr>
<tr>
<td>list_hapgs</td>
<td>47</td>
</tr>
<tr>
<td>list_health_checks</td>
<td>268</td>
</tr>
<tr>
<td>list_hi_ts</td>
<td>214</td>
</tr>
<tr>
<td>list_hi_ts_for_qualification_type</td>
<td>214</td>
</tr>
<tr>
<td>list_hosted_zones</td>
<td>268</td>
</tr>
<tr>
<td>list_hosted_zones_by_name</td>
<td>268</td>
</tr>
<tr>
<td>list_hosted_zones_by_vpc</td>
<td>268</td>
</tr>
<tr>
<td>list_hours_of_operations</td>
<td>97</td>
</tr>
<tr>
<td>list_hsms</td>
<td>47</td>
</tr>
<tr>
<td>list_human_task_uis</td>
<td>282</td>
</tr>
<tr>
<td>list_hyper_parameter_tuning_jobs</td>
<td>282</td>
</tr>
<tr>
<td>list_iam_policy_assignments</td>
<td>245</td>
</tr>
<tr>
<td>list_iam_policy_assignments_for_user</td>
<td>245</td>
</tr>
<tr>
<td>list_icd10cm_inference_jobs</td>
<td>91</td>
</tr>
<tr>
<td>list_identities</td>
<td>81, 301</td>
</tr>
<tr>
<td>list_identity_policies</td>
<td>301</td>
</tr>
<tr>
<td>list_identity_pool_usage</td>
<td>87</td>
</tr>
<tr>
<td>list_identity_pools</td>
<td>81</td>
</tr>
<tr>
<td>list_identity_providers</td>
<td>85</td>
</tr>
<tr>
<td>list_image_versions</td>
<td>282</td>
</tr>
<tr>
<td>list_images</td>
<td>131, 282</td>
</tr>
<tr>
<td>list_imports</td>
<td>43</td>
</tr>
<tr>
<td>list_incoming.TypedLinks</td>
<td>40</td>
</tr>
<tr>
<td>list_index</td>
<td>40</td>
</tr>
<tr>
<td>list_ ingestions</td>
<td>245</td>
</tr>
<tr>
<td>list_instance_attributes</td>
<td>97</td>
</tr>
<tr>
<td>list_instance_fleets</td>
<td>151</td>
</tr>
<tr>
<td>list_instance_groups</td>
<td>151</td>
</tr>
</tbody>
</table>
list_instance_profiles, 177
list_instance_profiles_for_role, 177
list_instance_storage_configs, 97
list_instances, 97, 151, 297
list_integration_associations, 97
list_invalidations, 45
list_inventory_entries, 314
list_invitations, 172, 290
list_ip_routes, 109
list_ip_sets, 172, 332, 335
list_jobs, 34, 161, 169, 278
list_journeys, 235
list_kafka_versions, 182
list_key_groups, 45
list_key_phrases_detection_jobs, 89
list_key_policies, 190
list_keys, 190
list_labeling_jobs, 282
list_labeling_jobs_for_workteam, 282
list_lambda_functions, 97
list_language_models, 328
list_launch_paths, 295
list_layer_versions, 193
list_layers, 193
list_lex_bots, 97
list_lexicons, 240
list_license_configurations, 199
list_license_specifications_for_resource, 199
list_license_versions, 199
list_licenses, 199
list_listeners, 165
list_local_disks, 318
list_log_pattern_sets, 19
list_log_patterns, 19
list_log_subscriptions, 109
list_luna_clients, 47
list_managed_schema_arns, 40
list_medical_transcription_jobs, 328
list_medical_vocabularies, 328
list_member_accounts, 157, 207
list_members, 172, 290
list_meshes, 21
list_metrics, 55
list_mfa_devices, 177
list_ml_transforms, 169
list_model_bias_job_definitions, 282
list_model_explainability_job_definitions, 282
list_model_package_groups, 282
list_model_packages, 283
list_model_quality_job_definitions, 283
list_models, 283
list_monitoring_executions, 283
list_monitoring_schedules, 283
list_multipart_uploads, 161, 275
list_named_queries, 25
list_namespaces, 245, 297
list_nodegroups, 137
list_nodes, 182
list_notebook_executions, 151
list_notebook_instance_lifecycle_configs, 283
list_notebook_instances, 283
list_object_attributes, 40
list_object_children, 40
list_object_parent_paths, 40
list_object_parents, 40
list_object_policies, 40
list_object_versions, 275
list_objects, 275
list_objects_v2, 275
list_on_premises_instances, 73
list_open_id_connect_providers, 177
list_open_workflow_executions, 324
list_operations, 270, 297
list_ops_item_events, 314
list_ops_metadata, 314
list_organization_admin_accounts, 172, 290
list_organization_portfolio_access, 295
list_organizational_units_for_parent, 226
list_origin_request_policies, 45
list_outgoing_typed_links, 40
list_packages_for_domain, 145
list_parallel_data, 329
list_parents, 226
list_partner_event_source_accounts, 58, 154
list_partner_event_sources, 58, 154
list_parts, 161, 275
list_pending_invitation_resources, 247
list_permissions, 7, 247
list_phi_detection_jobs, 91
list_phone_numbers, 97
list_phone_numbers_opted_out, 308
list pii_entities_detection_jobs, 89
list_pipeline_execution_steps, 283
list_pipeline_executions, 75, 77, 283
list_pipeline_parameters_for_execution, 283
list_pipelines, 75, 77, 103, 283
list_platform_applications, 308
list_platform_branches, 142
list_platform_versions, 142
list_policies, 157, 177, 226
list_policies_for_target, 226
list_policies_granting_service_access, 177
list_policy_attachments, 40
list_policy_versions, 177
list_portfolio_access, 295
list_portfolios, 295
list_portfolios_for_product, 295
list_principals, 247
list_principals_for_portfolio, 295
list_problems, 19
list_processing_jobs, 283
list_projects, 62, 64, 78, 79, 283
list_prompts, 97
list_protected_resources, 32
list_protection_groups, 305
list_protections, 305
list_protocols_lists, 157
list_provisioned_capacity, 161
list_provisioned_concurrency_configs, 193
list_provisioned_product_plans, 295
list_provisioning_artifacts, 295
list_provisioning_artifacts_for_service_action, 295
list_public_keys, 45, 54
list_published_schema_arns, 40
list_publishing_destinations, 172
list_pull_requests, 66, 70
list_qualification_requests, 214
list_qualification_types, 215
list_query_executions, 25
list_query_logging_configs, 268
list_queue_tags, 311
list_queues, 97, 311
list_quick_connects, 97
list_rate_based_rules, 332, 335
list_realtime_log_configs, 45
list_receipt_filters, 301
list_receipt_rule_sets, 301
list_received_grants, 199
list_received_licenses, 199
list_recipes, 228
list_record_history, 295
list_records, 87
list_recovery_points_by_backup_vault, 32
list_recovery_points_by_resource, 32
list_regex_match_sets, 332, 335
list_regex_pattern_sets, 332, 335
list_regional_buckets, 278
list_registries, 169
list_replays, 58, 154
list_report_groups, 62, 64
list_reports, 62, 64
list_reports_for_report_group, 62, 64
list_repositories, 65, 70
list_repositories_for_approval_rule_template, 67, 70
list_requested_service_quota_change_history, 298
list_requested_service_quota_change_history_by_quota, 298
list_resolver_dnssec_configs, 272
list_resolver_endpoint_ip_addresses, 272
list_resolver_endpoints, 272
list_resolver_query_log_config_associations, 272
list_resolver_query_log_configs, 273
list_resolver_rule_associations, 273
list_resolver_rules, 273
list_resource_compliance_summaries, 314
list_resource_data_sync, 314
list_resource_inventory, 199
list_resource_record_sets, 268
list_resource_servers, 85
list_resource_share_permissions, 247
list_resource_tags, 190
list_resource_types, 247
list_resources, 78, 79, 247
INDEX

list_resources_for_tag_option, 295
list_resources_for_web_acl, 335
list_resources_in_protection_group, 305
list_restore_jobs, 32
list_retirable_grants, 190
list_reusable_delegation_sets, 268
list_review_policy_results_for_hit, 215
list_reviewable_hi_ts, 215
list_role_policies, 177
list_role_tags, 178
list_roles, 178
list_roots, 226
list_routes, 21
list_routing_profile_queues, 97
list_routing_profiles, 97
list_rule_groups, 332, 335
list_rule_names_by_target, 58, 154
list_rules, 58, 154, 332, 335
list_rules.packages, 180
list_rx_norm_inference_jobs, 91
list_s3_resources, 207
list_saml_providers, 178
list_schema_extensions, 109
list_schema_versions, 169
list_schemas, 169, 228
list_scram_secrets, 182
list_secret_version_ids, 287
list_secrets, 287
list_security_configurations, 151
list_security_keys, 97
list_security_profiles, 97
list_sentiment_detection_jobs, 89
list_server_certificates, 178
list_service_actions, 295
list_service_actions_for_provisioning_artifact, 295
list_service_quote_increase_requests_in_template, 298
list_service_quotas, 298
list_service_specific_credentials, 178
list_services, 133, 297, 298
list_shards, 184
list_shared_projects, 62, 64
list_shared_report_groups, 62, 64
list_signing_certificates, 178
list_size_constraint_sets, 332, 335
list_solution_versions, 228
list_solutions, 228
list_source_credentials, 62, 64
list_speech_synthesis_tasks, 240
list_sql_injection_match_sets, 332, 335
list_ssh_public_keys, 178
list_stack_instances, 43
list_stack_instances_for_provisioned_product, 295
list_stack_resources, 43
list_stack_set_operation_results, 43
list_stack_set_operations, 43
list_stack_sets, 43
list_stacks, 43
list_state_machines, 303
list_steps, 151
list_storage_lens_configurations, 278
list_stored_queries, 94
list_stream_consumers, 184
list_stream_processors, 258
list_streaming_distributions, 45
listStreams, 117, 184
list_studio_session_mappings, 151
list_studios, 151
list_subscribed_rule_groups, 332, 335
list_subscribed_workteams, 283
list_subscriptions, 308
list_subscriptions_by_topic, 308
list_table_metadata, 25
list_tables, 115
list_tag_options, 295
list_tags, 7, 32, 49, 54, 104, 145, 193, 213, 221, 283
list_tags_for_certificate, 6
list_tags_for_delivery_stream, 155
list_tags_for_domain, 270
list_tags_for_solution, 78, 79
list_tags_for_resource, 19, 21, 23, 25, 34, 37, 38, 40, 45, 47, 56, 58, 68, 70, 73, 77, 81, 85, 89, 94, 97, 109, 111, 113, 131, 133, 135, 137, 140, 142, 154, 157, 158, 165, 172, 180, 182, 186, 187, 195, 199, 217, 224, 226, 235, 237, 245, 251, 268, 273, 290, 297, 298, 303, 308, 314, 318, 324, 332, 335, 339, 343
list_tags_for_resources, 268
list_tags_for_stream, 184
list_tags_for_vault, 161
list_tags_log_group, 60
list_tags_of_resource, 115
list_tape_pools, 318
list_tapes, 318
list_targets_by_rule, 58, 154
list_targets_for_policy, 226
list_task_definition_families, 133
list_task_definitions, 133
list_tasks, 133
list_team_members, 78, 79
list_template_aliases, 245
list_template_versions, 235, 245
list_templates, 235, 245, 301
list_terminologies, 329
list_text_translation_jobs, 329
list_theme_aliases, 245
list_theme_versions, 245
list_themes, 245
list_threat_intel_sets, 172
list_tokens, 199
list_topics, 309
list_topics_detection_jobs, 89
list_traffic_policies, 268
list_traffic_policy_instances, 268
list_traffic_policy_instances_by_hosted_zone, 268
list_traffic_policy_instances_by_policy, 268
list_traffic_policy_versions, 268
list_trails, 54
list_training_jobs, 283
list_training_jobs_for_hyper_parameter_tuning_job, 283
list_transcription_jobs, 328
list_transform_jobs, 283
list_trial_components, 283
list_trials, 283
list_triggers, 169
list_type_registrations, 43
list_type_versions, 43
list_typed_link_facet_attributes, 40
list_typed_link_facet_names, 40
list_types, 43
list_updates, 137
list_usage_for_license_configuration, 199
list_use_cases, 97
list_user_groups, 245
list_user_hierarchy_groups, 97
list_user_import_jobs, 85
list_user_policies, 178
list_user_pool_clients, 85
list_user_pools, 85
list_user_profiles, 78, 79, 283
list_user_tags, 178
list_users, 85, 97, 178, 213, 245
list_users_in_group, 85
list_vaults, 161
list_verified_email_addresses, 301
list_versions_by_function, 193
list_virtual_gateways, 21
list_virtual_interface_test_history, 107
list_virtual_mfa_devices, 178
list_virtual_nodes, 21
list_virtual_routers, 21
list_virtual_services, 21
list_vocabulary_filters, 328
list_vocabulary_filters, 328
list_volume_initiators, 318
list_volume_recovery_points, 318
list_volumes, 318
list_vpc_association_authorizations, 268
list_web_ac ls, 332, 335
list_webhooks, 77
list_website_authorization_providers, 339
list_website_certificate_authorities, 339
list_work_groups, 25
list_worker_blocks, 215
list_workers_with qualification_type, 215
list_workflows, 169
list_workflows, 169
list_workforces, 283
list_workteams, 283
list_xss_match_sets, 332, 335
lookup_developer_identity, 81
lookup_events, 54
lookup_policy, 40
machinelearning, 204
macie, 206
marketplacecommerceanalytics, 207
marketplaceentitlementservice, 209
marketplacemetering, 210
merge_branches_by_fast_forward, 66, 70
merge_branches_by_squash, 66, 70
merge_branches_by_three_way, 66, 70
merge_developer_identities, 81
merge_pull_request_by_fast_forward, 66, 70
merge_pull_request_by_squash, 66, 70
merge_pull_request_by_three_way, 66, 70
merge_shards, 184
meter_usage, 211
migrate_workspace, 342
modify_account, 342
modify_availability_zone_group, 126
modify_backup_attributes, 49
modify_cache_cluster, 140
modify_cache_cluster, 217
modify_cache_subnet_group, 140
modify_capacity_reservation, 126
modify_certificates, 251
modify_client_properties, 342
modify_client_vpn_endpoint, 126
modify_cluster, 49, 151, 256
modify_cluster_db_revision, 256
modify_cluster_iam_roles, 256
modify_cluster_maintenance, 256
modify_cluster_parameter_group, 256
modify_cluster_snapshot, 256
modify_cluster_snapshot_schedule, 256
modify_cluster_subnet_group, 256
modify_current_cluster_capacity, 251
modify_db_cluster, 113, 217, 251
modify_db_cluster_endpoint, 217, 251
modify_db_clusterparameter_group, 113, 217, 251
modify_db_cluster_snapshot_attribute, 113, 217, 251
modify_db_instance, 113, 217, 251
modify_db_subnet, 217, 251
modify_db_subnet_group, 113, 217, 251
modify_default_creditSpecification, 126
modify_document_permission, 314
modify_events, 126
modify_event_subscription, 217, 251, 256
modify_fleet, 126
modify_fpga_image_attribute, 126
modify_global_cluster, 251
modify_global_replication_group, 140
modify_hap, 47
modify_host, 126
modify_hsm, 47
modify_id_format, 126
modify_identity_id_format, 126
modify_image_attribute, 126
modify_instance_attribute, 126
modify_instance_capacityReservationAttributes, 126
modify_instance_creditSpecification, 126
modify_instance_event_start_time, 126
modify_instance_fleet, 151
modify_instance_groups, 151
modify_instance_metadataOptions, 126
modify_instance_placement, 126
modify_launch_template, 126
modify_listener, 149
modify_load_balancer_attributes, 147, 149
modify_luna_client, 47
modify_managed_prefix_list, 126
modify_mount_target_security_groups, 135
modify_network_interface_attribute, 126
modify_option_group, 251
modify_replication_group, 140
modify_replication_group_shard_configuration, 140
modify_report_definition, 99
modify_reserved_instances, 126
modify_rule, 149
modify_scheduled_action, 256
modify_selfservice_permissions, 342
modify_snapshot_attribute, 126
modify_snapshot_copy_retention_period, 256
modify_snapshot_schedule, 256
modify_spot_fleet_request, 126
modify_subnet_attribute, 126
purchase_reserved_db_instances_offering, 251
purchase_reserved_elasticsearch_instance_offering, 145
purchase_reserved_instances_offering, 127
purchase_reserved_node_offering, 256
purchase_scheduled_instances, 127
purge_queue, 311
put_access_point_policy, 278
put_account_dedicated_ip_warmup_attributes, 237
put_account_sending_attributes, 237
put_account_setting, 133
put_account_setting_default, 133
put_action_revision, 77
put_aggregation_authorization, 94
putalarm, 203
put_anomaly_detector, 56
put_application_policy, 292
put_approval_result, 77
put_apps_list, 157
put_attributes, 133, 307
put_auto_scaling_policy, 151
put_backup_policy, 135
put_backup_vault_access_policy, 32
put_backup_vault_notifications, 32
put_block_public_access_configuration, 152
put_bot, 195
put_bot_alias, 195
put_bucket_accelerate_configuration, 275
put_bucket_acl, 275
put_bucket_analytics_configuration, 275
put_bucket_cors, 275
put_bucket_encryption, 275
put_bucket_intelligent_tiering_configuration, 275
put_bucket_inventory_configuration, 275
put_bucket_lifecycle, 275
put_bucket_lifecycle_configuration, 275, 278
put_bucket_logging, 275
put_bucket_metrics_configuration, 275
put_bucket_notification, 275
put_bucket_notification_configuration, 275
put_bucket_ownership_controls, 276
put_bucket_policy, 276, 278
put_bucket_replication, 276
put_bucket_request_payment, 276
put_bucket_tagging, 276, 278
put_bucket_versioning, 276
put_bucket_website, 276
put_cluster_capacity_providers, 133
put_comment_reaction, 68, 70
put_compliance_items, 314
put_composite_alarm, 56
put_config_rule, 94
put_configuration_aggregator, 94
put_configuration_recorder, 94
put_configuration_set_delivery_options, 237, 301
put_configuration_set_reputation_options, 237
put_configuration_set_sending_options, 238
put_configuration_set_tracking_options, 238
put_conformance_pack, 94
put_dashboard, 56
put_data_catalog_encryption_settings, 169
put_dedicated_ip_in_pool, 238
put_dedicated_ip_warmup_attributes, 238
put_deliverability_dashboard_option, 238
put_delivery_channel, 94
put_destination, 60
put_destination_policy, 60
put_email_identity_dkim_attributes, 238
put_email_identity_feedback_attributes, 238
put_email_identity_mail_from_attributes, 238
put_email_identity_mail_from_attributes, 238
put_encryption_config, 343
put_evaluations, 94
put_event_selectors, 54
put_event_stream, 235
put_events, 58, 154, 229, 235
put_external_evaluation, 94
put_file, 65, 70
put_file_system_policy, 135
put_function_code_signing_config, 193
put_function_concurrency, 193
put_function_event_invoke_config, 193
put_gateway_response, 11
put_group_configuration, 260
put_group_policy, 178
put_identity_policy, 301
put_image, 131
put_image_scanning_configuration, 131
put_image_tag_mutability, 131
put_insight_rule, 56
put_insight_selectors, 54
put_instance_public_ports, 203
put_integration, 11
put_integration_response, 11
put_intent, 195
put_inventory, 314
put_item, 115
put_items, 229
put_job_failure_result, 76, 77
put_job_success_result, 76, 77
put_job_tagging, 278
put_key_policy, 190
put_lexicon, 240
put_lifecycle_configuration, 135
put_lifecycle_event_hook_execution_status, 73
put_lifecycle_hook, 28
put_lifecycle_policy, 131
put_log_events, 60
put_logging_configuration, 332, 335
put_managed_scaling_policy, 152
put_method, 11
put_method_response, 11
put_metric_alarm, 56
put_metric_data, 56
put_metric_filter, 60
put_model_package_group_policy, 283
put_notification_channel, 157
put_notification_configuration, 28
put_object, 276
put_object_acl, 276
put_object_legal_hold, 276
put_object_lock_configuration, 276
put_object_retention, 276
put_object_tagging, 276
put_organization_config_rule, 94
put_organization_conformance_pack, 94
put_parameter, 314
put_partner_events, 58, 154
put_permission, 58, 154
put_permission_policy, 332, 335
put_pipeline_definition, 103
put_policy, 7, 157
put_protocols_list, 157
put_provisioned_concurrency_config, 193
put_public_access_block, 276, 278
put_query_definition, 60
put_record, 155, 184
put_record_batch, 155
put_records, 184
put_registry_policy, 131
put_remediation_configurations, 94
put_remediation_exceptions, 94
put_replication_configuration, 131
put_report_definition, 99
put_repository_triggers, 68, 70
put_resolver_query_log_config_policy, 273
put_resolver_rule_policy, 273
put_resource_config, 94
put_resource_policy, 60, 62, 64, 169, 287
put_rest_api, 11
put_retention_configuration, 94
put_retention_policy, 60
put_role_permissions_boundary, 178
put_role_policy, 178
put_rule, 58, 154
put_scaling_policy, 17, 28
put_scheduled_action, 17
put_scheduled_update_group_action, 28
put_schema_from_json, 40
put_schema_version_metadata, 169
put_secret_value, 287
put_service_quota_increase_request_into_template, 298
put_session, 198
put_slot_type, 195
put_storage_lens_configuration, 278
put_storage_lens_configuration_tagging, 278
put_stored_query, 94
put_subscription_filter, 60
put_targets, 58, 154
put_telemetry_records, 343
put_third_party_job_failure_result, 76, 77
put_third_party_job_success_result, 76, 77
put_trace_segments, 343
put_user_permissions_boundary, 178
put_user_policy, 178
put_users, 229
put_webhook, 77
put_workflow_run_properties, 169
query, 115
query_objects, 103
query_schema_version_metadata, 169
quicksight, 242
ram, 246
rds, 247
rdsdataservice, 252
re_encrypt, 190
rebalance_slots_in_global_replication_group, 140
reboot_broker, 182, 213
reboot_cache_cluster, 140
reboot_cluster, 256
reboot_db_instance, 113, 217, 251
reboot_instance, 204, 221
reboot_instances, 127
reboot_node, 104
reboot_relnational_database, 204
reboot_workspaces, 342
rebuild_environment, 142
rebuild_workspaces, 342
receive_message, 311
recognize_celebrities, 258
record_activity_task_heartbeat, 325
record_handler_progress, 43
record_lifecycle_action_heartbeat, 28
redshift, 253
refresh_cache, 318
refresh_trusted_advisor_check, 322, 323
register_activity_type, 325
register_application_revision, 74
register_certificate, 109
register_container_image, 204
register_container_instance, 133
register_cross_account_access_role, 180
register_db_proxy_targets, 251
register_default_patch_baseline, 314
register_delegated_administrator, 226
register_device, 87
register_devices, 283
register_domain, 270, 325
register_ecs_cluster, 221
register_elastic_ip, 221
register_event_topic, 109
register_image, 127
register_instance, 221, 297
register_instance_event_notification_attributes, 127
register_instances_with_load_balancer, 145, 147
register_job_definition, 34
register_on_premises_instance, 74
register_patch_baseline_for_patch_group, 314
register_rds_db_instance, 221
register_scalable_target, 16, 17
register_schema_version, 169
register_stream_consumer, 184
register_target_with_maintenance_window, 314
register_targets, 149
register_task_definition, 133
register_task_with_maintenance_window, 314
register_transit_gateway_multicast_group_members, 127
register_transit_gateway_multicast_group_sources, 127
register_type, 43
register_usage, 211
register_user, 245
register_volume, 221
register_webhook_with_third_party, 77
register_workflow_type, 325
register_workspace_directory, 342
reimport_api, 15
reject_assignment, 215
reject_domain_transfer_from_another_aws_account, 270
reject_grant, 199
reject_inbound_cross_cluster_search_connection,
reject_portfolio_share, 295
reject_qualification_request, 215
reject_resource_share_invitation, 247
reject_shared_directory, 109
reject_transit_gateway_multicast_domain_association, 127
reject_transit_gateway_peering_attachment, 127
reject_vpc_endpoint_connections, 127
reject_vpc_peering_connection, 127
rekognition, 257
release_address, 127
release_hosts, 127
release_static_ip, 204
remove_account_from_organization, 226
remove_all_resource_permissions, 338
remove_attributes, 235
remove_attributes_from_findings, 180
remove_auto_scaling_policy, 152
remove_client_id_from_open_id_connect_provider, 178
remove_custom_routing_endpoints, 165
remove_facet_from_object, 40
remove_from_global_cluster, 251
remove_ip_routes, 109
remove_layer_version_permission, 193
remove_listener_certificates, 149
remove.managed_scaling_policy, 152
remove_permission, 58, 154, 193, 309, 311
remove_region, 109
remove_resource_permission, 338
remove_role_from_db_cluster, 217, 251
remove_role_from_db_instance, 251
remove_role_from_instance_profile, 178
remove_schema_version_metadata, 169
remove_source_identifier_from_subscription, 217, 251
remove_tags, 54, 103, 145, 147, 149, 152
remove_tags_from_certificate, 6
remove_tags_from_on_premises_instances, 74
remove_tags_from_resource, 47, 109, 113, 140, 217, 251, 314, 318
remove_tags_from_stream, 184
remove_tags_from_vault, 161
remove_targets, 58, 154
remove_user_from_group, 178
render_ui_template, 283
renew_certificate, 6
renew_domain, 270
add_receipt_rule_set, 301
replace_iam_instance_profile_association, 127
replace_network_acl_association, 127
replace_network_acl_entry, 127
replace_route, 127
replace_route_table_association, 127
replace_transit_gateway_route, 127
report_instance_status, 127
report_task_progress, 103
report_task_runner_heartbeat, 103
request_cancel_workflow_execution, 325
request_certificate, 6
request_environment_info, 142
request_service_quota_increase, 299
request_spot_fleet, 127
request_spot_instances, 127
resend_confirmation_code, 85
resend_contact_reachability_email, 270
resend_validation_email, 6
reset_authorizers_cache, 15
reset_cache, 318
reset_cache_parameter_group, 140
reset_cluster_parameter_group, 256
reset_db_cluster_parameter_group, 113, 217, 251
reset_db_parameter_group, 217, 251
reset_distribution_cache, 204
reset_ebs_default_kms_key_id, 127
reset_fpga_image_attribute, 127
reset_image_attribute, 127
reset_instance_attribute, 127
reset_job_bookmark, 169
reset_network_interface_attribute, 127
reset_service_setting, 314
reset_service_specific_credential, 178
reset_snapshot_attribute, 127
reset_user_password, 109
resize_cluster, 256
resolve_case, 321, 323
resolve_customer, 211
resourcegroups, 259
resourcegroupstaggingapi, 261
respond_activity_task_canceled, 325
respond_activity_task_completed, 325
respond_activity_task_failed, 325
respond_decision_task_completed, 325
respond_to_auth_challenge, 85
restart_app_server, 142
restore_address_to_classic, 127
restore_analysis, 245
restore_backup, 49
restore_certificate_authority, 7
restore_db_cluster_from_s3, 251
restore_db_cluster_from_snapshot, 113, 217, 251
restore_db_cluster_to_point_in_time, 113, 217, 251
restore_db_instance_from_db_snapshot, 251
restore_db_instance_from_s3, 251
restore_db_instance_to_point_in_time, 251
restore_domain_access, 339
restore_from_cluster_snapshot, 256
restore_from_snapshot, 109
restore_managed_prefix_list_version, 127
restore_object, 276
restore_secret, 287
restore_server, 224
restore_table_from_backup, 115
restore_table_from_cluster_snapshot, 256
restore_table_to_point_in_time, 115
restore_workspace, 342
resume_cluster, 256
resume_contact_recording, 97
resume_processes, 28
resume_session, 314
resume_workflow_run, 169
resync_mfa_device, 178
retire_grant, 190
retrieve_domain_auth_code, 270
retrieve_environment_info, 142
retrieve_tape_archive, 318
retrieve_tape_recovery_point, 318
retry_build, 64
retry_build_batch, 64
retry_stage_execution, 77
revoke_cache_security_group_ingress, 140
revoke_certificate, 7
revoke_client_vpn_ingress, 127
revoke_cluster_security_group_ingress, 256
revoke_db_security_group_ingress, 251
revoke_domain_access, 339
revoke_grant, 190
revoke_ip_rules, 342
revoke_security_group_egress, 127
revoke_security_group_ingress, 127
rollback_transaction, 253
rotate_encryption_key, 256
rotate_secret, 287
route53, 266
route53domains, 269
route53resolver, 271
run_instances, 127
run_job_flow, 152
run_scheduled_instances, 128
run_task, 133
s3, 273
s3control, 276
sagemaker, 278
sagemakerruntime, 284
scan, 115
scan_provisioned_products, 295
schedule_key_deletion, 191
search, 51, 52, 283
search_analyses, 245
search_dashboards, 245
search_faces, 258
search_faces_by_image, 258
search_local_gateway_routes, 128
search_products, 295
search_products_as_admin, 295
search_provisioned_products, 295
search_resources, 261
search_tables, 169
search_transit_gateway_multicast_groups, 128
search_transit_gateway_routes, 128
secretsmanager, 285
securityhub, 288
select, 307
select_aggregate_resource_config, 94
select_object_content, 276
select_resource_config, 94
send_automation_signal, 314
send_bonus, 215
send_bounce, 301
send_bulk_templated_email, 301
send_command, 314
send_contact_method_verification, 204
send_custom_verification_email, 301
send_diagnostic_interrupt, 128
send_email, 238, 301
send_message, 311
send_message_batch, 311
send_raw_email, 301
send_ssh_public_key, 129
send_task_failure, 303
send_task_heartbeat, 303
send_task_success, 303
send_test_event_notification, 215
send_users_messages, 235
send_voice_message, 239
set_active_receipt_rule_set, 301
set_alarm_state, 56
set_alarm_events, 87
set_data_retrieval_policy, 161
set_default_policy_version, 178
set.desired_capacity, 28
set.endpoint_attributes, 309
set.identity.default_version, 43
set_identity_feedback_forwarding_enabled, 301
set_identity.mail_from_domain, 301
set_identity_notification_topic, 301
set_identity_pool_configuration, 87
set_identity_pool_roles, 81
set_instance_health, 28
set_instance_protection, 28
set.ip.address_type, 149
set_load_balancer_listener.ssl_certificate, 147
set_load_balancer.policies_for.backend_server, 147
set_load_balancer.policies_of_listener, 147
set_load_based_auto_scaling, 221
set_local_console_password, 318
set.permission, 221
set_platform_application_attributes, 309
set_queue_attributes, 311
set_receipt_rule_position, 301
set_repository_policy, 131
set_risk_configuration, 85
set_rule_priorities, 149
set_security_groups, 149
set_security_token_service_preferences, 178
set_smb_guest_password, 318
set.sms.attributes, 309
set_stack_policy, 43
set_status, 103
set_subnets, 149
set_subscription_attributes, 309
set_tags_for_resource, 180
set.task.status, 103
set_termination_protection, 152
set_topic.attributes, 309
set.type.default_version, 43
set.ui.customization, 85
set.user.mfa_preference, 85
set_user_pool.mfa_config, 85
set_user_settings, 85
set_vault.access_policy, 161
set_vault.notifications, 161
set_visible_to_all_users, 152
sfn, 302
share_directory, 109
shield, 304
shutdown_gateway, 318
sign, 191
sign_out_user, 340
sign_up, 85
signal_resource, 43
signal_workflow_execution, 325
simpledb, 306
simulate_custom_policy, 178
simulate_principal_policy, 178
skip_wait_time_for_instance_termination, 74
sns, 307
split_shard, 184
sqs, 309
ssm, 311
start_activity_stream, 251
start_application, 186, 187
start_assessment_run, 180
start_associations_once, 314
start_automation_execution, 315
start_availability_monitor_test, 318
start_backup_job, 32
start_bgp_failover_test, 107
start_build, 62, 64
start_build_batch, 64
start_celebrity_recognition, 258
start_change_request_execution, 315
start_chat_contact, 97
start_config_rules_evaluation, 94
start_configuration_recorder, 94
start_contact_recording, 97
start_content_moderation, 258
start_copy_job, 32
start_crawler, 169
start_crawler_schedule, 169
start_db_cluster, 113, 217, 251
start_db_instance, 252
start_db_instance_automated_backups_replication, 252
start_delivery_stream_encryption, 155
start_document_analysis, 326
start_document_classification_job, 89
start_document_text_detection, 326
start_dominant_language_detection_job, 89
start_elasticsearch_service_software_update, 145
start_entities_detection_job, 89
start_entities_detection_v2_job, 91
start_events_detection_job, 89
start_execution, 303
start_export_labels_task_run, 169
start_export_task, 252
start_face_detection, 258
start_face_search, 258
start_fleet, 23
start_gateway, 318
start_icd10cm_inference_job, 91
start_image_builder, 24
start_image_scan, 131
start_import, 195
start_import_labels_task_run, 169
start_instance, 204, 221
start_instance_refresh, 28
start_instances, 128
start_job_run, 169
start_key_phrases_detection_job, 89
start_label_detection, 258
start_lifecycle_policy_preview, 131
start_logging, 54
start_maintenance, 224
start_medical_transcription_job, 328
start_migration, 140
start_ml_evaluation_task_run, 169
start_ml_labeling_set_generation_task_run, 169
start_monitoring_members, 172
start_monitoring_schedule, 283
start_network_insights_analysis, 128
start_notebook_execution, 152
start_notebook_instance, 283
start_outbound_voice_contact, 97
start_person_tracking, 258
start_phi_detection_job, 91
start_pii_entities_detection_job, 89
start_pipeline_execution, 75, 77, 283
start_project_version, 258
start_query, 60
start_query_execution, 25
start_remediation_database, 204
start_remediation_execution, 94
start_replay, 58, 154
start_report_creation, 266
start_restore_job, 32
start_rx_norm_inference_job, 91
start_schema_extension, 110
start_segment_detection, 258
start_sentiment_detection_job, 89
start_session, 315
start_speech_synthesis_task, 240
start_stack, 221
start_stream_encryption, 184
start_stream_processor, 258
start_support_data_export, 208
start_sync_execution, 303
tag_delivery_stream, 155
unassign_private_ip_addresses, 128
unassign_activities, 221
unassign_volume, 221
undeprcate_activity_type, 325
undeprcate_domain, 325
undeprcate_workflow_type, 325
ungroup_resources, 261
unlink_developer_identity, 8
unlink_identity, 8
unmonitor_instances, 128
unpeer_vpc, 204
unshare_application, 292
unshare_directory, 110
unsubscribe, 309
unsubscribe_from_dataset, 87
unsubscribe_from_event, 180
untag, 261
untag_certificate_authority, 8
untag_delivery_stream, 155
untag_log_group, 60
untag_project, 78, 79
untag_queue, 311
untag_resource, 11, 15, 19, 21, 24, 26, 32, 34,
37, 38, 40, 45, 49, 56, 58, 68, 70, 74,
77, 81, 85, 90, 94, 97, 105, 107, 111,
115, 131, 134, 135, 137, 154, 157,
158, 165, 169, 172, 182, 186, 188,
191, 193, 195, 199, 204, 221, 224,
226, 235, 238, 245, 247, 273, 287,
290, 297, 299, 303, 309, 325, 332,
335, 340, 343
untag_resources, 266
tag_role, 178
untag_user, 178
terminate_client_vpn_connections, 128
terminate_environment, 142
terminate_instance_in_auto_scaling_group, 28
terminate_instances, 128
terminate_job, 34
terminate_job_flows, 152
terminate_provisioned_product, 295
terminate_session, 315
terminate_workflow_execution, 325
terminate_workspaces, 342
test_alarm, 204
test_dns_answer, 268
test_event_pattern, 58, 154
test_failover, 140
test_invoke_authorizer, 11
test_invoke_method, 11
test_metric_filter, 60
test_render_template, 301
test_repository_triggers, 68, 70
textract, 325
transact_get_items, 115
transact_write_items, 115
transcribeservice, 326
transfer_domain, 270
transfer_domain_to_another_aws_account, 270
translate, 328
translate_text, 329
unarchive_findings, 172
unassign_instance, 221
unassign_ipv_6_addresses, 128
INDEX

update_anomaly_monitor, 101
update_anomaly_subscription, 101
update_api, 15
update_api_key, 11
update_api_mapping, 15
update_apns_channel, 235
update_apns_sandbox_channel, 235
update_apns_voip_channel, 235
update_apns_voip_sandbox_channel, 235
update_app, 221
update_app_image_config, 283
update_application, 19, 74, 143, 186, 188, 292
update_application_resource_lifecycle, 143
update_application_settings, 235
update_application_version, 143
update_approval_rule_template_content, 68, 70
update_approval_rule_template_description, 67, 70
update_approval_rule_template_name, 67, 70
update_archive, 58, 154
update_artifact, 283
update_assessment_target, 180
update_association, 315
update_association_status, 315
update_assume_role_policy, 178
update_audit_stream_configuration, 340
update_auth_event_feedback, 85
update_authorizer, 11, 15
update_auto_scaling_group, 28
update_automatic_tape_creation_policy, 318
update_availability_options, 51
update_backup_plan, 32
update_baidu_channel, 235
update_bandwidth_rate_limit, 318
update_bandwidth_rate_limit_schedule, 318
update_base_path_mapping, 11
update_batch_prediction, 206
update_broker, 213
update_broker_count, 182
update_broker_storage, 182
update_budget, 36
update_budget_action, 36
update_byte_match_set, 332, 335
update_cache_policy, 46
update_campaign, 228, 235
update_capacity_provider, 134
update_certificate_authority, 8
update_certificate_options, 6
update_chap_credentials, 318
update_classifier, 169
update_client_certificate, 11
update_cloud_front_origin_access_identity, 46
update_cluster, 105
update_cluster_config, 137
update_cluster_configuration, 182
update_cluster_kafka_version, 182
update_cluster_settings, 134
update_cluster_version, 137
update_code_repository, 283
update_code_signing_config, 193
update_column_statistics_for_partition, 169
update_column_statistics_for_table, 169
update_comment, 68, 70
update_company_network_configuration, 340
update_component, 19
update_component_configuration, 19
update_compute_environment, 34
update_conditional_forwarder, 110
update_configuration, 182, 213
update_configuration_set_event_destination, 238, 239, 301
update_configuration_set_reputation_metrics_enabled, 301
update_configuration_set_sending_enabled, 301
update_configuration_set_tracking_options, 301
update_configuration_template, 143
update_connection, 169
update_connection_alias_permission, 342
update_constraint, 295
update_contact_attributes, 97
update_contact_flow_content, 97
update_contact_flow_name, 97
update_container_agent, 134
update_container_instances_state, 134
update_container_service, 204
update_context, 283
update_continuous_backups, 115
update_contributor_insights, 115
update_cost_category_definition, 101
update_crawler, 170
update_crawler_schedule, 170
update_custom_key_store, 191
update_custom_routing_accelerator, 165
update_custom_routing_accelerator_attributes, update_email_channel, 235
update_custom_routing_listener, 165
update_custom_verification_email_template, 301
update_dashboard, 245
update_dashboard_permissions, 245
update_dashboard_published_version, 245
update_data_catalog, 26
update_data_set, 245
update_data_set_permissions, 245
update_data_source, 206, 245
update_data_source_permissions, 245
update_database, 170
update_default_branch, 65, 70
update_deployment, 11, 15
update_deployment_group, 74
update_destination, 155
update_detector, 172
update_dev_endpoint, 170
update_device_fleet, 283
update_device_policy_configuration, 340
update_device_status, 85
update_devices, 283
update_direct_connect_gateway_association, 107
update_directory_config, 24
update_distribution, 46, 204
update_distribution_bundle, 204
update_document, 315, 338
update_document_default_version, 315
update_document_metadata, 315
update_document_version, 338
update_documentation_part, 11
update_documentation_version, 11
update_domain, 284
update_domain_contact, 270
update_domain_contact_privacy, 270
update_domain_endpoint_options, 51
update_domain_entry, 204
update_domain_metadata, 340
update_domain_name, 11, 15
update_domain_nameservers, 270
update_elastic_ip, 221
update_elasticsearch_domain_config, 145
update_email_channel, 235
update_email_template, 235
update_emergency_contact_settings, 305
update_endpoint, 90, 235, 284
update_endpoint_group, 165
update_endpoint_weights_and_capacities, 284
update_endpoints_batch, 235
update_environment, 37, 38, 143
update_environment_membership, 37, 38
update_evaluation, 206
update_event_source_mapping, 193
update_experiment, 284
update_expiration_for_hit, 215
update_facet, 40
update_field_level_encryption_config, 46
update_field_level_encryption_profile, 46
update_folder, 338
update_function_code, 193
update_function_configuration, 193
update_function_event_invoke_config, 193
update_gateway_information, 318
update_gateway_response, 11
update_gateway_route, 21
update_gateway_software_now, 318
update_gcm_channel, 235
update_geo_match_set, 332, 335
update_global_settings, 32
update_global_table, 115
update_global_table
update_global_table_settings, 115
update_group, 85, 178, 245, 261, 344
update_group_query, 261
update_health_check, 268
update_hit_review_status, 215
update_hit_type_of_hit, 215
update_hosted_zone_comment, 268
update_iam_policy_assignment, 245
update_identity_pool, 81
update_identity_provider, 85
update_identity_provider_configuration, 340
update_image, 284
update_image_permissions, 24
update_insight, 290
update_instance, 221
update_instance_attribute, 97
update_instance_custom_health_status, 297
update_instance_storage_config, 97
update_integration, 11, 15
update_integration_response, 11, 15
update_ip_set, 172, 322, 335
update_item, 115
update_job, 170
update_job_priority, 278
update_job_queue, 34
update_job_status, 278
update_journey, 235
update_journey_state, 235
update_key_description, 191
update_key_group, 46
update_lag, 107
update_layer, 221
update_license_configuration, 200
update_license_specifications_for_resource, 200
update_lifecycle_policy, 111
update_link_attributes, 40
update_listener, 165
update_load_balancer_attribute, 204
update_log_pattern, 19
update_login_profile, 178
update_maintenance_start_time, 318
update_maintenance_window, 315
update_maintenance_window_target, 315
update_maintenance_window_task, 315
update_managed_instance_role, 315
update_medical_vocabulary, 328
update_member_detectors, 172
update_mesh, 21
update_method, 11
update_method_response, 11
update_ml_model, 206
update_ml_transform, 170
update_model, 11, 15
update_model_package, 284
update_monitoring, 182
update_monitoring_schedule, 284
update_my_user_profile, 221
update_nfs_file_share, 318
update_nodegroup_config, 137
update_nodegroup_version, 137
update_notebook_instance, 284
update_notebook_instance_lifecycle_config, 284
update_notification, 36
update_notification_settings, 215
update_number_of_domain_controllers, 110
update_object_attributes, 40
update_open_id_connect_provider_thumbprint, 178
update_ops_item, 315
update_ops_metadata, 315
update_organization_configuration, 172, 290
update_organizational_unit, 226
update_origin_request_policy, 46
update_package, 145
update_parallel_data, 329
update_parameter_group, 105
update_partition, 170
update_patch_baseline, 315
update_pipeline, 75, 77, 284
update_pipeline_execution, 284
update_policy, 226
update_portfolio, 295
update_portfolio_share, 295
update_product, 295
update_project, 62, 64, 78, 79
update_protection_group, 305
update_provisioned_product, 295
update_provisioned_product_properties, 295
update_provisioning_artifact, 295
update_public_key, 46
update_publishing_destination, 172
update_pull_request_approval_rule_content, 67, 70
update_pull_request_approval_state, 67, 70
update_pull_request_description, 67, 70
update_pull_request_status, 67, 70
update_pull_request_title, 67, 70
update_push_template, 235
update_qualification_type, 215
update_quick_connect_config, 97
update_quick_connect_name, 97
update_radius, 110
update_rate_based_rule, 332, 335
update_rds_db_instance, 221
update_realtime_log_config, 46
update_receipt_rule, 301
update_recommender_configuration, 235
update_records, 87
update_recovery_point_lifecycle, 32
update_regex_match_set, 332, 335
update_regex_pattern_set, 332, 335
update_region_settings, 32
update_registry, 170
update_relational_database, 204
update_relational_database_parameters, 204
update_report_group, 62, 64
update_repository_description, 65, 70
update_repository_name, 65, 70
update_request_validator, 11
update_resolver_dnssec_config, 273
update_resolver_endpoint, 273
update_resolver_rule, 273
update_resource, 11
update_resource_data_sync, 315
update_resource_server, 85
update_resource_share, 247
update_rest_api, 11
update_role, 178
update_role_description, 178
update_route, 15, 21
update_route_response, 15
update_routing_profile_concurrency, 97
update_routing_profile_default_outbound_queue, 98
update_routing_profile_name, 98
update_routing_profile_queues, 98
update_rule, 332, 335
update_rule_group, 332, 335
update_rules_of_ip_group, 342
update_s3_resources, 207
update_saml_provider, 178
update_sampling_rule, 344
update_scaling_parameters, 51
update_scaling_plan, 30
update_schema, 40, 170
update_secret, 287
update_secret_version_stage, 287
update_security_group_rule_descriptions_egress, 128
update_security_group_rule_descriptions_ingress, 128
update_security_hub_configuration, 290
update_segment, 235
update_server, 224
update_server_certificate, 178
update_server_engine_attributes, 224
update_service, 134, 297
update_service_access_policies, 51
update_service_action, 295
update_service_primary_task_set, 134
update_service_setting, 315
update_service_settings, 200
update_service_specific_credential, 178
update_shard_count, 184
update_signing_certificate, 178
update_size_constraint_set, 332, 335
update_smb_file_share, 318
update_smb_file_share_visibility, 318
update_smb_security_strategy, 318
update_sms_channel, 235
update_sms_template, 235
update_snapshot_schedule, 318
update_sql_injection_match_set, 332, 335
update_ssh_public_key, 178
update_stack, 24, 43, 219, 221
update_stack_instances, 43
update_stack_set, 43
update_stage, 11, 15
update_standards_control, 288, 290
update_state_machine, 303
update_streaming_distribution, 46
update_studio_session_mapping, 152
update_subnet_group, 105
update_subscriber, 36
update_subscription, 305
update_table, 115, 170
update_table_replica_auto_scaling, 115
update_tag_option, 295
update_tags_for_domain, 270
update_tags_for_resource, 143
update_task_set, 134
update_team_member, 78, 80
update_template, 245, 301
update_template_active_version, 235
update_template_alias, 245
update_template_permissions, 245
update_termination_protection, 43
update_theme, 245
update_theme_alias, 245
update_theme_permissions, 245
update_threat_intel_set, 172
update_time_to_live, 115
update_traffic_policy_comment, 268
update_traffic_policy_instance, 268
update_traffic_policy, 54
update_training_job, 284
update_trial, 284
update_trial_component, 284
update_trigger, 170
update_trust, 110
update_typed_link_facet, 40
update_usage, 11
update_usage_plan, 11
update_user, 178, 213, 245, 338
update_user_attributes, 85
update_user_defined_function, 170
update_user_hierarchy, 98
update_user_hierarchy_group_name, 98
update_user_hierarchy_structure, 98
update_user_identity_info, 98
update_user_phone_config, 98
update_user_pool, 85
update_user_pool_client, 86
update_user_pool_domain, 86
update_user_profile, 78, 80, 221, 284
update_user_routing_profile, 98
update_user_security_profiles, 98
update_virtaul_gateway, 21
update_virtual_interface_attributes, 107
update_virtual_node, 21
update_virtual_router, 21
update_virtual_service, 21
update_vocabulary, 328
update_vocabulary_filter, 328
update_voice_channel, 235
update_voice_template, 235
update_volume, 221
update_vpc_link, 11, 15
update_vtl_device_type, 318
update_web_acl, 332, 335
update_webhook, 62, 64
update_work_group, 26
update_workflow, 170
update_workforce, 284
update_workspace_image_permission, 342
update_workteam, 284
update_xss_match_set, 332, 335
upgrade_applied_schema, 41
upgrade_elasticsearch_domain, 145
upgrade_published_schema, 41
upload_archive, 161
upload_documents, 51, 52
upload_layer_part, 131
upload_multipart_part, 161
upload_part, 276
upload_part_copy, 276
upload_server_certificate, 178
upload_signing_certificate, 178
upload_ssh_public_key, 178
validate_configuration_settings, 143
validate_pipeline_definition, 103
validate_resource_policy, 287
validate_template, 43
verify, 191
verify_domain_dkim, 301
verify_domain_identity, 301
verify_email_address, 301
verify_email_identity, 301
verify_software_token, 86
verify_trust, 110
verify_user_attribute, 86
view_billing, 270
waf, 330
waf_regional, 333
withdraw_byoip_cidr, 128, 165
workdocs, 336
worklink, 338
workspaces, 340
xray, 342