Package ‘paws.management’

August 23, 2021

Title 'Amazon Web Services' Management & Governance Services

Version 0.1.12

Description Interface to 'Amazon Web Services' management and governance services, including 'CloudWatch' application and infrastructure monitoring, 'Auto Scaling' for automatically scaling resources, and more <https://aws.amazon.com/>.

License Apache License (>= 2.0)

URL https://github.com/paws-r/paws

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

Imports paws.common (>= 0.3.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.1.1

Collate 'applicationautoscaling_service.R'
  'applicationautoscaling_interfaces.R'
  'applicationautoscaling_operations.R'
  'applicationinsights_service.R'
  'applicationinsights_interfaces.R'
  'applicationinsights_operations.R' 'autoscaling_service.R'
  'autoscaling_interfaces.R' 'autoscaling_operations.R'
  'autoscalingplans_service.R' 'autoscalingplans_interfaces.R'
  'autoscalingplans_operations.R' 'cloudformation_service.R'
  'cloudformation_interfaces.R' 'cloudformation_operations.R'
  'cloudtrail_service.R' 'cloudtrail_interfaces.R'
  'cloudtrail_operations.R' 'cloudwatch_service.R'
  'cloudwatch_interfaces.R' 'cloudwatch_operations.R'
  'cloudwatchevents_service.R' 'cloudwatchevents_interfaces.R'
  'cloudwatchevents_operations.R' 'cloudwatchlogs_service.R'
  'cloudwatchlogs_interfaces.R' 'cloudwatchlogs_operations.R'
  'configservice_service.R' 'configservice_interfaces.R'
  'configservice_operations.R' 'health_service.R'
  'health_interfaces.R' 'health_operations.R'
R topics documented:

'licensemanager_service.R' 'licensemanager_interfaces.R'
'licensemanager_operations.R' 'opsworks_service.R'
'opsworks_interfaces.R' 'opsworks_operations.R'
'opsworkscm_service.R' 'opsworkscm_interfaces.R'
'opsworkscm_operations.R' 'organizations_service.R'
'organizations_interfaces.R' 'organizations_operations.R'
'pi_service.R' 'pi_interfaces.R' 'pi_operations.R'
'resourcegroups_service.R' 'resourcegroups_interfaces.R'
'resourcegroups_operations.R'
'resourcegroupstaggingapi_service.R'
'resourcegroupstaggingapi_interfaces.R'
'resourcegroupstaggingapi_operations.R'
'servicecatalog_service.R' 'servicecatalog_interfaces.R'
'servicecatalog_operations.R' 'servicequotas_service.R'
'servicequotas_interfaces.R' 'servicequotas_operations.R'
'ssm_service.R' 'ssm_interfaces.R' 'ssm_operations.R'
'support_service.R' 'support_interfaces.R'
'support_operations.R'

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2021-08-23 07:10:27 UTC

R topics documented:

applicationautoscaling ........................................ 3
applicationinsights ............................................. 5
autoscaling ....................................................... 7
autoscalingplans ................................................ 9
cloudformation .................................................. 11
cloudtrail ......................................................... 13
cloudwatch ....................................................... 15
cloudwatchevents ............................................... 17
cloudwatchlogs .................................................. 19
configservice .................................................. 22
health ............................................................ 25
licensemanager .................................................. 27
opsworks ......................................................... 29
opsworkscm ...................................................... 33
organizations .................................................... 35
pi ................................................................. 37
resourcegroups .................................................. 39
resourcegroupstaggingapi ...................................... 41
Application Auto Scaling

Description

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

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

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

```
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

```
## 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"
)

## End(Not run)
```

---

**applicationinsights**  
*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**

<table>
<thead>
<tr>
<th>Name</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 <- applicationinsights(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

- `create_application`: Adds an application that is created from a resource group
- `create_component`: Creates a custom component by grouping similar standalone instances
- `create_log_pattern`: Adds a log pattern to a LogPatternSet
- `delete_application`: Removes the specified application from monitoring
- `delete_component`: Ungroups a custom component
- `delete_log_pattern`: Removes the specified log pattern from a LogPatternSet
- `describe_application`: Describes the application
- `describe_component`: Describes a component and lists the resources that are grouped together
- `describe_component_configuration`: Describes the monitoring configuration of the component
- `describe_component_configuration_recommendation`: Describes the recommended monitoring configuration of the component
- `describe_log_pattern`: Describe a specific log pattern from a LogPatternSet
- `describe_observation`: Describes an anomaly or error with the application
- `describe_problem`: Describes an application problem
- `describe_problem_observations`: Describes the anomalies or errors associated with the problem
- `list_applications`: Lists the IDs of the applications that you are monitoring
- `list_components`: Lists the auto-grouped, standalone, and custom components of the application
- `list_configuration_history`: Lists the INFO, WARN, and ERROR events for periodic configuration updates performed by Application Insights
- `list_log_patterns`: Lists the log patterns in the specified log LogPatternSet
- `list_log_pattern_sets`: Lists the log pattern sets in the specified application
- `list_problems`: Lists the problems with your application
- `list_tags_for_resource`: Retrieve a list of the tags (keys and values) that are associated with a specified application
- `tag_resource`: Add one or more tags (keys and values) to a specified application
- `untag_resource`: Remove one or more tags (keys and values) from a specified application
- `update_application`: Updates the application
- `update_component`: Updates the custom component name and/or the list of resources that make up the component
- `update_component_configuration`: Updates the monitoring configurations for the component
- `update_log_pattern`: Adds a log pattern to a LogPatternSet
Examples

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

autoscaling  

Auto Scaling

Description

Amazon EC2 Auto Scaling

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

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

Usage

```r
autoscaling(config = list())
```

Arguments

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

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

attach_instances
attach_load_balancers
attach_load_balancer_target_groups
batch_delete_scheduled_action
batch_put_scheduled_update_group_action
cancel_instance_refresh
complete_lifecycle_action
create_auto_scaling_group
create_launch_configuration
create_or_update_tags
delete_auto_scaling_group
delete_launch_configuration
delete_lifecycle_hook
delete_notification_configuration
delete_policy
delete_scheduled_action
delete_tags
describe_account_limits
describe_adjustment_types
describe_auto_scaling_groups
describe_auto_scaling_instances
describe_auto_scaling_notification_types
describe_instance_refreshes
describe_launch_configurations
describe_lifecycle_hooks
describe_lifecycle_hook_types
describe_load_balancers
describe_load_balancer_target_groups
describe_metric_collection_types
describe_notification_configurations
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

Attaches one or more EC2 instances to the specified Auto Scaling group
To attach an Application Load Balancer, Network Load Balancer, or Gateway Load Balancer, use the AttachLoadBalancerTargetGroups API operation instead
Attaches one or more target groups to the specified Auto Scaling group
Deletes one or more scheduled actions for the specified Auto Scaling group
Creates or updates one or more scheduled scaling actions for an Auto Scaling group
Cancels an instance refresh operation in progress
Completes the lifecycle action for the specified token or instance with the specified result
We strongly recommend using a launch template when calling this operation to create a launch configuration
Creates or updates tags for the specified Auto Scaling group
Deletes the specified Auto Scaling group
Deletes the specified launch configuration
Deletes the specified lifecycle hook
Deletes the specified notification
Deletes the specified scaling policy
Deletes the specified scheduled action
Deletes the specified tags
Describes the current Amazon EC2 Auto Scaling resource quotas for your AWS account
Describes the available adjustment types for Amazon EC2 Auto Scaling scaling policies
Describes one or more Auto Scaling groups
Describes one or more Auto Scaling instances
Describes the notification types that are supported by Amazon EC2 Auto Scaling
Describes one or more instance refreshes
Describes one or more launch configurations
Describes the lifecycle hooks for the specified Auto Scaling group
Describes the available types of lifecycle hooks
Describes the load balancers for the specified Auto Scaling group
Describes the target groups for the specified Auto Scaling group
Describes the available CloudWatch metrics for Amazon EC2 Auto Scaling
Describes the notification actions associated with the specified 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 operations
Describes the actions scheduled for your Auto Scaling group that haven’t run or haven’t reached their end times
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
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

enable_metrics_collection Enables group metrics for the specified Auto Scaling group
enter_standby Moves the specified instances into the standby state
execute_policy Executes the specified policy
exit_standby Moves the specified instances out of the standby state
put_lifecycle_hook Creates or updates a lifecycle hook for the specified Auto Scaling group
put_notification_configuration Configures an Auto Scaling group to send notifications when specified events take place
put_scaling_policy Creates or updates a scaling policy for an Auto Scaling group
put_scheduled_update_group_action Creates or updates a scheduled scaling action for an Auto Scaling group
record_lifecycle_action_heartbeat Records a heartbeat for the lifecycle action associated with the specified token or instance
resume_processes Resumes the specified suspended auto scaling processes, or all suspended processes, for the specified Auto Scaling group
set_desired_capacity Sets the size of the specified Auto Scaling group
set_instance_health Sets the health status of the specified instance
set_instance_protection Updates the instance protection settings of the specified instances
start_instance_refresh Starts a new instance refresh operation, which triggers a rolling replacement of all previously launched instances in the Auto Scaling group
suspend_processes Suspends the specified auto scaling processes, or all processes, for the specified Auto Scaling group
terminate_instance_in_auto_scaling_group Terminates the specified instance and optionally adjusts the desired group size
update_auto_scaling_group We strongly recommend that all Auto Scaling groups use launch templates to ensure full functionality for Amazon EC2 Auto Scaling and Amazon EC2

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 Plans**

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**

```plaintext
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**

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

**Operations**

- `create_scaling_plan` Creates a scaling plan
- `delete_scaling_plan` Deletes the specified scaling plan
- `describe_scaling_plan_resources` Describes the scalable resources in the specified scaling plan
- `describe_scaling_plans` Describes one or more of your scaling plans
cloudformation

get_scaling_plan_resource_forecast_data  Retrieves the forecast data for a scalable resource
update_scaling_plan                  Updates the specified scaling plan

Examples

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

cloudformation  AWS CloudFormation

Description

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

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

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

Usage

```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.
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_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 it’s expected configuration
- `detect_stack_resource_drift` Returns information about whether a resource’s actual configuration differs, or has drifted, from its expected configuration
- `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_type_registrations
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
Reports progress of a resource handler to CloudFormation
Opens 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)
```

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

Operations

- **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
### Examples

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

### End(Not run)
```

---

### Description

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

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

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

cloudwatch(config = list())

Arguments

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

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 <- 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.
**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

ccloudwatchevents(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 <- 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
deactivate_event_source Deletes the specified archive
deactivate_event_source Deletes the specified custom event bus or partner event bus
deactivate_event_source This operation is used by SaaS partners to delete a partner event source
deactivate_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
enable_rule
list_archives
list_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

Disables the specified rule
Enables the specified rule
Lists your archives
Lists all the event buses in your account, including the default event bus, custom event buses, and partner event buses.
You can use this to see all the partner event sources that have been shared with your AWS account.
You 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
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
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
csvc <- cloudwatchlogs(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```
Operations

**associate_kms_key**
Associates the specified AWS Key Management Service (AWS KMS) customer master key (CMK) with the specified log group.

**cancel_export_task**
Cancels the specified export task.

**create_export_task**
Creates an export task, which allows you to efficiently export data from a log group to an Amazon S3 bucket.

**create_log_group**
Creates a log group with the specified name.

**create_log_stream**
Creates a log stream for the specified log group.

**delete_destination**
Deletes the specified destination, and eventually disables all the subscription filters that publish to it.

**delete_log_group**
Deletes the specified log group and permanently deletes all the archived log events associated with the log group.

**delete_log_stream**
Deletes the specified log stream and permanently deletes all the archived log events associated with the log stream.

**delete_metric_filter**
Deletes the specified metric filter.

**delete_query_definition**
Deletes a saved CloudWatch Logs Insights query definition.

**delete_resource_policy**
Deletes a resource policy from this account.

**delete_retention_policy**
Deletes the specified retention policy.

**delete_subscription_filter**
Deletes the specified subscription filter.

**describe_destinations**
Lists all your destinations.

**describe_export_tasks**
Lists the specified export tasks.

**describe_log_groups**
Lists the specified log groups.

**describe_log_streams**
Lists the log streams for the specified log group.

**describe_metric_filters**
Lists the specified metric filters.

**describe_queries**
Returns a list of CloudWatch Logs Insights queries that are scheduled, executing, or have been executed recently in this account.

**describe_query_definitions**
This operation returns a paginated list of your saved CloudWatch Logs Insights query definitions.

**describe_resource_policies**
Lists the resource policies in this account.

**describe_subscription_filters**
Lists the subscription filters for the specified log group.

**disassociate_kms_key**
Disassociates the associated AWS Key Management Service (AWS KMS) customer master key (CMK) from the specified log group.

**filter_log_events**
Lists log events from the specified log group.

**get_log_events**
Lists log events from the specified log stream.

**get_log_group_fields**
Returns a list of the fields that are included in log events in the specified log group, along with the percentage of log events that contain each field.

**get_log_record**
Retrieves all of the fields and values of a single log event.

**get_query_results**
Returns the results from the specified query.

**list_tags_log_group**
Lists the tags for the specified log group.

**put_destination**
Creates or updates a destination.

**put_destination_policy**
Creates or updates an access policy associated with an existing destination.

**put_log_events**
Uploads a batch of log events to the specified log stream.

**put_metric_filter**
Creates or updates a metric filter and associates it with the specified log group.

**put_query_definition**
Creates or updates a query definition for CloudWatch Logs Insights.

**put_resource_policy**
Sets the retention of the specified log group.

**put_retention_policy**
Sets the retention of the specified log group.

**put_subscription_filter**
Creates or updates a subscription filter and associates it with the specified log group.

**start_query**
Schedules a query of a log group using CloudWatch Logs Insights.

**stop_query**
Stops a CloudWatch Logs Insights query that is in progress.

**tag_log_group**
Tests the filter pattern of a metric filter against a sample of log event messages.

**test_metric_filter**
Removes the specified tags from the specified log group.

**untag_log_group**
Examples

```r
## Not run:
svc <- cloudwatchlogs()
svc$associate_kms_key(
  Foo = 123
)

## End(Not run)
```

## 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
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.
configservice

Service syntax

csvc <- 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
describe_configuration_aggregator_sources_status
describe_configuration_recorders
describe_configuration_recorder_status
describe_conformance_pack_compliance
describe_conformance_packs

Returns the current configuration items for resources that are present in
Returns the current configuration for one or more requested resources
Deletes the authorization granted to the specified configuration aggregator
Deletes the specified AWS Config rule and all of its evaluation results
Deletes the specified configuration aggregator and the aggregated data
Deletes the configuration recorder
Deletes the specified conformance pack and all the AWS Config rules,
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 of the conformance
Deletes pending authorization requests for a specified aggregator account
Deletes the remediation configuration
Deletes one or more remediation exceptions mentioned in the resource
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
Returns a list of compliant and noncompliant rules with the number of
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
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
describe_pending_aggregation_requests
describe_remediation_configurations
describe_remediation_exceptions
describe_remediation_execution_status
describeRetentionConfigurations
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
selectAggregateResourceConfig
select_resource_config

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
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 specific resource
Returs the number of compliant and noncompliant rules for one or more resources
Returns resource counts across accounts and regions that are present in the specified retention 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
Returs the number of resources that are compliant and the number that are not
Returns compliance details of a conformance pack for all AWS resources
Returns compliance details for the conformance pack based on the current conformance pack type, the number of each resource type, and the current status
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 or noncompliant
Accepts a resource type and returns a list of resource identifiers for the specified resource
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 specified source account and region
Adds or updates an AWS Config rule for evaluating whether your AWS resources meet specific security and compliance requirements
Creates and updates the configuration aggregator with the selected source and region configuration aggregator
Create a new configuration recorder to record the selected resource configuration aggregator
Creates or updates a conformance pack
Creates a delivery channel object to deliver configuration information and audit trails to an Amazon S3 bucket or Amazon Simple Notification Service (SNS) topic
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 one or more accounts in an AWS Organization
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
Saves a new query or updates an existing saved query
Accepts a structured query language (SQL) SELECT command and an INSERT INTO command
Accepts a structured query language (SQL) SELECT command, performs the corresponding search, and returns resource configurations matching the properties
put_stored_query Saves a new query or updates an existing saved query
put_retention_configuration Creates and updates the retention configuration with details about retention period (number of days) that AWS Config stores your historical information
put_resource_config Records the configuration state for the resource provided in the request
put_remediation_exceptions A remediation exception is when a specific resource is no longer considered for auto-remediation
put_remediation_configurations Adds or updates the remediation configuration with a specific AWS Config rule with the selected target or action
put_organization_conformance_pack Deploys conformance packs across member accounts in an AWS Organization
put_organization_config_rule Adds or updates organization config rule for your entire organization evaluating whether your AWS resources comply with your desired configurations
put_external_evaluation
put_evaluations Used by an AWS Lambda function to deliver evaluation results to AWS Config
put_delivery_channel Creates a delivery channel object to deliver configuration information to an Amazon S3 bucket and Amazon SNS topic
put_conformance_pack Creates or updates a conformance pack
put_configuration_recorder Creates a new configuration recorder to record the selected resource configurations
put_configuration_aggregator Creates and updates the configuration aggregator with the selected source accounts and regions
put_config_rule Adds or updates an AWS Config rule for evaluating whether your AWS resources comply with your desired configurations
put_aggregation_authorization Authorizes the aggregator account and region to collect data from the source account and region
put_aggregation_authorization
put_remediation_exceptions
put_remediation_configurations
put_organization_config_rule
put_organization_conformance_pack
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
selectAggregateResourceConfig
select_resource_config

Describe Retention Configurations
Provides details of one or more retention configurations
Gets a detailed view of a Remediation Execution for a set of resources
Provides the evaluation results for the specified AWS Config rule for a specific resource
Returns the number of compliant and noncompliant rules for one or more resources
Returns resource counts across accounts and regions that are present in the specified retention configuration
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
Returns the number of resources that are compliant and the number that are not compliant
Returns compliance details of a conformance pack for all AWS resources
Returns compliance details for the conformance pack based on the current conformance pack type, the number of each resource type, and the current status
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 or noncompliant
Accepts a resource type and returns a list of resource identifiers for the specified resource
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 specified source account and region
Adds or updates an AWS Config rule for evaluating whether your AWS resources meet specific security and compliance requirements
Creates and updates the configuration aggregator with the selected source and region configuration aggregator
Create a new configuration recorder to record the selected resource configuration aggregator
Creates or updates a conformance pack
Creates a delivery channel object to deliver configuration information and audit trails to an Amazon S3 bucket or Amazon Simple Notification Service (SNS) topic
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 one or more accounts in an AWS Organization
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
Saves a new query or updates an existing saved query
Accepts a structured query language (SQL) SELECT command and an INSERT INTO command
Accepts a structured query language (SQL) SELECT command, performs the corresponding search, and returns resource configurations matching the properties
**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](https://docs.aws.amazon.com/general/latest/gr/signature_v4.html).

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](https://docs.aws.amazon.com/health/latest/APIReference/Welcome.html).

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.

---

**Examples**

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

## End(Not run)
```
• 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

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

Operations

- `describe_affected_accounts_for_organization`: Returns a list of accounts in the organization from AWS Organizations that are affected by the specified 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 your organization.
- `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
)
## End(Not run)
```

### Description

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

### Usage

```r
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 Op-
erations section.

### Service syntax

```r
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_licenseSpecifications_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
untag_resource
update_license_configuration
update_licenseSpecifications_for_resource
update_service_settings

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
Delegates 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
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)
```

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 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"
    ),
    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
describe_user_profiles

Assign a registered instance to a layer
Assigns one of the stack’s registered Amazon EBS volumes to a specified instance
Associates one of the stack’s registered Elastic IP addresses with a specified instance
Attaches an Elastic Load Balancing load balancer to a specified layer
Creates a clone of a specified stack
Creates an app for a specified stack
Runs deployment or stack commands
Creates an instance in a specified stack
Creates a layer
Creates a new stack
Creates a new user profile
Deletes a specified app
Deletes a specified instance, which terminates the associated Amazon EC2 instance
Deletes a specified layer
Deletes a specified stack
Deletes a user profile
Deregisters a specified Amazon ECS cluster from a stack
Deregisters a specified Elastic IP address
Deregister a registered Amazon EC2 or on-premises instance
Deregisters an Amazon RDS instance
Deregisters an Amazon EBS volume
Describes the available AWS OpsWorks Stacks agent versions
Requests a description of a specified set of apps
Describes the results of specified commands
Requests a description of a specified set of deployments
Describes Amazon ECS clusters that are registered with a stack
Describes Elastic IP addresses
Describes a stack’s Elastic Load Balancing instances
Requests a description of a set of instances
Requests a description of one or more layers in a specified stack
Describes load-based auto scaling configurations for specified layers
Describes a user’s SSH information
Describes the operating systems that are supported by AWS OpsWorks Stacks
Describes the permissions for a specified stack
Describe an instance’s RAID arrays
Describes Amazon RDS instances
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 specified users
describe_volumes

detach_elastic_load_balancer

disassociate_elastic_ip

generate_hostname_suggestion

generate_access

list_tags

reboot_instance

register_instance

register_ebs_volume

set_load_based_auto_scaling

set_permission

set_time_based_auto_scaling

start_instance

start_stack

start_stack

stop_instance

stop_stack

tag_resource

unassign_instance

unassign_volume

untag_resource

update_app

update_elastic_ip

update_instance

update_layer

update_my_user_profile

update_rds_instance

update_stack

update_user_profile

update_volume

Describes an instance’s Amazon EBS volumes

Detaches a specified Elastic Load Balancing instance from its layer

Disassociates an Elastic IP address from its instance

Gets a generated host name for the specified layer, based on the current host name’s theme

This action can be used only with Windows stacks

Returns a list of tags that are applied to the specified stack or layer

Reboots a specified instance

Registers a specified Amazon ECS cluster with a stack

Registers an Elastic IP address with a specified stack

Registers instances that were created outside of AWS OpsWorks Stacks with a specified stack

Registers an Amazon RDS instance with a stack

Registers an Amazon EBS volume with a specified stack

Specify the load-based auto scaling configuration for a specified layer

Specifies a user’s permissions

Specify the time-based auto scaling configuration for a specified instance

Starts a specified instance

Starts a stack’s instances

Stops a specified instance

Stops a specified stack

Apply cost-allocation tags to a specified stack or layer in AWS OpsWorks Stacks

Unassigns a registered instance from all layers that are using the instance

Unassigns an assigned Amazon EBS volume

Removes tags from a specified stack or layer

Updates a specified app

Updates a registered Elastic IP address’s name

Updates a specified instance

Updates a specified layer

Updates a user’s SSH public key

Updates an Amazon RDS instance

Updates a specified stack

Updates a specified user profile

Updates an Amazon EBS volume’s name or mount point

Examples

## 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
• 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 Op-
erations 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
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

---

**Description**

AWS Organizations

**Usage**

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

Operations

- **accept_handshake**: 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:
  - Creates an AWS organization.
  - Creates an organizational unit (OU) within a root or parent OU.
  - Creates a policy of a specified type that you can attach to a root, an organizational unit (OU), or 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_policy**: Retrieves information about an organization (OU).
- **describe_policy**: Retrieves information about an organizational unit (OU).
- **describe_policy**: Retrieves information about a policy.
- **disable_aws_service_access**: Disables the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.
- **disable_policy_type**: Disables an organizational policy type in a root.
- **enable_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 OU or root.
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 a member 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 OU or account.
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).
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)
```

---

**pi**  
AWS Performance Insights

**Description**

Amazon RDS Performance Insights

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

```r
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
svc <- 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)
```

---

**resourcegroups**  
**AWS Resource Groups**

**Description**

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

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

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

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

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

**Usage**

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

**Arguments**

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

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

Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_group</td>
<td>Creates a resource group with the specified name and description</td>
</tr>
<tr>
<td>delete_group</td>
<td>Deletes the specified resource group</td>
</tr>
<tr>
<td>get_group</td>
<td>Returns information about a specified resource group</td>
</tr>
<tr>
<td>get_group_configuration</td>
<td>Returns the service configuration associated with the specified resource group</td>
</tr>
<tr>
<td>get_group_query</td>
<td>Retrieves the resource query associated with the specified resource group</td>
</tr>
<tr>
<td>get_tags</td>
<td>Returns a list of tags that are associated with a resource group, specified by an ARN</td>
</tr>
<tr>
<td>group_resources</td>
<td>Adds the specified resources to the specified group</td>
</tr>
<tr>
<td>list_group_resources</td>
<td>Returns a list of ARNs of the resources that are members of a specified resource group</td>
</tr>
<tr>
<td>list_groups</td>
<td>Returns a list of existing resource groups in your account</td>
</tr>
<tr>
<td>put_group_configuration</td>
<td>Attaches a service configuration to the specified group</td>
</tr>
<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(
```
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
• 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
resourcegroupstaggingapi

- 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

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

Operations
servicecatalog

### Describe

- **describe_report_creation**
  - Describes the status of the StartReportCreation operation
- **get_compliance_summary**
  - Returns a table that shows counts of resources that are noncompliant with their tag policies
- **get_resources**
  - Returns all the tagged or previously tagged resources that are located in the specified Region for the AWS account
- **get_tag_keys**
  - Returns all tag keys in the specified Region for the AWS account
- **get_tag_values**
  - Returns all tag values for the specified key in the specified Region for the AWS account
- **start_report_creation**
  - Generates a report that lists all tagged resources in accounts across your organization and tells whether each resource is compliant with the effective tag policy
- **tag_resources**
  - Applies one or more tags to the specified resources
- **untag_resources**
  - Removes the specified tags from the specified resources

### Examples

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

## servicecatalog

**AWS Service Catalog**

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

```r
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

```r
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_tag_option_with_resource` Associates the specified TagOption with the specified resource
- `batch_associate_service_action_with_provisioning_artifact` Associates multiple self-service actions with provisioning artifacts
- `batch_disassociate_service_action_from_provisioning_artifact` Disassociates a batch of self-service actions from the specified provisioning artifact
- `copy_product` Copies the specified source product to the specified target product or a new product
- `create_constraint` Creates a constraint
- `create_portfolio` Creates a portfolio
- `create_portfolio_share` Shares the specified portfolio with the specified account or organization node
- `create_product` Creates a product
- `create_provisioned_product_plan` Creates a plan
- `create_provisioning_artifact` Creates a provisioning artifact (also known as a version) for the specified product
- `create_service_action` Creates a self-service action
- `create_tag_option` Creates a TagOption
- `delete_constraint` Deletes the specified constraint
- `delete_portfolio` Deletes the specified portfolio
- `delete_portfolio_share` Stops sharing the specified portfolio with the specified account or organization node
- `delete_product` Deletes the specified product
- `delete_provisioned_product_plan` Deletes the specified plan
- `delete_provisioning_artifact` Deletes the specified provisioning artifact (also known as a version) for the specified product
- `delete_service_action` Deletes a self-service action
- `delete_tag_option` Deletes the specified TagOption
- `describe_constraint` Gets information about the specified constraint
- `describe_copy_product_status` Gets the status of the specified copy product operation
- `describe_portfolio` Gets information about the specified portfolio
- `describe_portfolio_shares` Returns a summary of each of the portfolio shares that were created for the specified portfolio
- `describe_portfolio_share_status` Gets the status of the specified portfolio share operation
describe_product
describe_product_as_admin
describe_product_view
describe_provisioned_product
describe_provisioned_product_plan
describe_provisioning_artifact
describe_provisioning_parameters
describe_record
describe_service_action
describe_service_action_execution_parameters
describe_tag_option
disable_aws_organizations_access
disassociate_budget_from_resource
disassociate_principal_from_portfolio
disassociate_product_from_portfolio
disassociate_service_action_from_provisioning_artifact
disassociate_tag_option_from_resource
enable_aws_organizations_access
execute_provisioned_product_plan
execute_provisioned_product_service_action
get_aws_organizations_access_status
get_provisioned_product_outputs
import_as_provisioned_product
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

Gets information about the specified product
Gets information about the specified product
Gets information about the specified product
Gets information about the specified provisioned product
Gets information about the resource changes for the specified plan
Gets information about the specified provisioning artifact (also known as a version)
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
Enable portfolio sharing feature through AWS Organizations
Disassociates the specified budget from the specified resource
Disassociates a previously associated principal ARN from a portfolio
Disassociates the specified product from the specified portfolio
Disassociates the specified self-service action association from the specified provisioning artifact
Disassociates the specified TagOption from the specified resource
Disable portfolio sharing through AWS Organizations feature
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
Requests the import of a resource as a Service Catalog provisioned product
Lists all portfolios for which sharing was accepted by this account
Lists all the budgets associated to a specified resource
Lists the constraints for the specified portfolio and 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 all provisioning artifacts (also known as versions) for the specified self-service action
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 a tag option
Returns summary information about stack instances that are associated with a specified tag option
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
servicequotas

update_constraint  Updates the specified constraint
update_portfolio  Updates the specified portfolio
update_portfolio_share  Updates the specified portfolio share
update_product  Updates the specified product
update_provisioned_product  Requests updates to the configuration of the specified provisioned product
update_provisioned_product_properties  Requests updates to the properties of the specified provisioned product
update_provisioning_artifact  Updates the specified provisioning artifact (also known as a version)
update_service_action  Updates a self-service action
update_tag_option  Updates the specified TagOption

Examples

## Not run:
svc <- servicecatalog()
svc$accept_portfolio_share(
    Foo = 123
)

## End(Not run)

servicequotas  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

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

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
delete_service_quota_increase_request_from_template
disable_service_quota_template
get_association_for_service_quota_template
get_aws_default_service_quota
get_requested_service_quota_change
get_service_quota
get_service_quota_increase_request_from_template
list_aws_default_service_quotas
list_requested_service_quota_change_history
list_requested_service_quota_change_history_by_quota
list_service_quota_increase_requests_in_template
list_service_quotas
list_services
list_tags_for_resource
put_service_quota_increase_request_into_template
request_service_quota_increase
tag_resource
untag_resource

Examples

## Not run:
svc <- servicequotas()
svc$associate_service_quota_template(
    Foo = 123
)
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.

Usage

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

```r
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**: A State Manager association defines the state that you want to maintain on your instances.
- **create_association_batch**: Associates the specified Systems Manager document with the specified instance 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**: Deletes a custom inventory type or the data associated with a custom Inventory type.
- **delete_maintenance_window**: Deletes a maintenance window.
- **delete_ops_metadata**: Delete OpsMetadata related to an application.
- **delete_parameter**: Deletes a parameter from the system.
- **delete_parameters**: Deletes a list of parameters.
- **delete_patch_baseline**: Deletes a patch baseline.
- **delete_resource_data_sync**: Deletes a Resource Data Sync configuration.
- **deregister_managed_instance**: Removes the server or virtual machine from the list of registered instances.
- **deregister_patch_baseline_for_patch_group**: Removes a patch group from a specified patch baseline.
- **deregister_target_from_maintenance_window**: Removes a target from a maintenance window.
- **deregister_task_from_maintenance_window**: Removes a task from a maintenance window.
- **describe_activations**: Describes details about the activation, such as the date and time the activation was created, its current state, the IAM role assigned to the instances in the activation, and the number of instances registered by using this activation.
- **describe_association**: Describes the association for the specified target or instance.
- **describe_association_executions**: Use this API action to view all executions for a specific association.
- **describe_association_execution_targets**: Use this API action to view information about a specific execution of a specific association.
- **describe_automation_executions**: Provides details about all active and terminated Automation executions.
- **describe_automation_step_executions**: Information about all active and terminated step executions in an Automation workflow.
- **describe_available_patches**: Lists all patches eligible to be included in a patch baseline.
- **describe_document**: Describes the specified Systems Manager document.
- **describe_document_permission**: Describes the permissions for a Systems Manager document.
- **describe_effective_instance_associations**: All associations for the instance(s).
- **describe_effective_patches_for_patch_baseline**: Retrieves the current effective patches (the patch and the approval state) for the specified patch baseline.
- **describe_instance_associations_status**: The status of the associations for the instance(s).
- **describe_instance_information**: Describes one or more of your instances, including information about the operating system platform, the version of SSM Agent installed on the instance, instance status, and so on.
describe_instance_patches
describe_instance_patch_states
describe_instance_patch_states_for_patch_group
describe_inventory_deletions
describe_maintenance_window_executions
describe_maintenance_window_execution_task_invocations
describe_maintenance_window_execution_tasks
describe_maintenance_window_windows
describe_maintenance_window_schedule
describe_maintenance_windows_for_target
describe_maintenance_window_targets
describe_maintenance_window_tasks
describe_ops_items
describe_parameters
describe_patch_baselines
describe_patch_groups
describe_patch_group_state
describe_patch_properties
describe_sessions
get_automation_execution
get_calendar_state
get_command_invocation
get_connection_status
get_default_patch_baseline
get_deployable_patch_snapshot_for_instance
get_document
get_inventory
get_inventory_schema
get_maintenance_window
get_maintenance_window_execution
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

Retrieves information about the patches on the specified instance
Retrieves the high-level patch state of one or more instances
Retrieves the high-level patch state for the instances in the specified patch group
Describes a specific delete inventory operation
Lists the executions of a maintenance window
Retrieves the individual task executions (one per target) for a patch group
For a given maintenance window execution, lists the tasks that were run
Retrieves the maintenance windows in an AWS account
Retrieves information about upcoming executions of a maintenance window
Retrieves information about the maintenance window targets or tasks an instance is associated with
Lists the targets registered with the maintenance window
Lists the tasks in a maintenance window
Query a set of OpsItems
Get information about a parameter
Lists the patch baselines in your AWS account
Lists all patch groups that have been registered with patch baselines
Returns high-level aggregated patch compliance state for a patch baseline
Lists the properties of available patches organized by product, product family, classification, severity, and other properties of available patches
Retrieves a list of all active sessions (both connected and disconnected)
Get detailed information about a particular Automation execution
Gets the state of the AWS Systems Manager Change Calendar at an optional, specified time
Returns detailed information about command execution for an instance
Retrieves the Session Manager connection status for an instance
Retrieves the current snapshot for the patch baseline the instance uses
Gets the contents of the specified Systems Manager document
Query inventory information
Return a list of inventory type names for the account, or return information about a specific inventory item
Retrieves a maintenance window
Retrieves details about a specific a maintenance window execution
Retrieves the 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
Retrieves the history of all changes to a parameter
Get details of a parameter
Retrieve information about one or more parameters in a specific hierarchy
Retrieves information about a patch baseline
Retrieves the patch baseline that should be used for the specified instance
ServiceSetting is an account-level setting for an AWS service
A parameter label is a user-defined alias to help you manage different versions of a parameter
Returns all State Manager associations in the current AWS account
Retrieves all versions of an association for a specific association
An invocation is copy of a command sent to a specific instance
Lists the commands requested by users of the AWS account
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
start_associations_once
start_automation_execution
start_change_request_execution
start_session
stop_automation_execution
terminate_session
update_association
update_association_status
update_document
update_document_default_version
update_document_metadata
update_maintenance_window
update_maintenance_window_target
update_maintenance_window_task
update_managed_instance_role
update_ops_item
update_ops_metadata
update_patch_baseline
update_resource_data_sync
update_service_setting

For a specified resource ID, this API action returns a list of compliance statuses for different resource types.
Returns a summary count of compliant and non-compliant resources.
Information about approval reviews for a version of an SSM document.
Returns all Systems Manager (SSM) documents in the current AWS account.
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.
Systems Manager calls this API action when displaying all Application Manager OpsMetadata objects or blobs.
Returns a resource-level summary count.
Lists your resource data sync configurations.
Returns a list of the tags assigned to the specified resource.
Shares a Systems Manager document publicly or privately.
Registers a compliance type and other compliance details on a designated resource.
Bulk update custom inventory items on one more instance.
Add a parameter to the system.
Defines the default patch baseline for the relevant operating system.
Registers a patch baseline for a patch group.
Registers a target with a maintenance window.
Adds a new task to a maintenance window.
Removes tag keys from the specified resource.
ServiceSetting is an account-level setting for an AWS service.
Reconnects a session to an instance after it has been disconnected.
Sends a signal to an Automation execution to change the current behavior or status of the execution.
Runs commands on one or more managed instances.
Use this API action to run an association immediately and only once.
Initiates execution of an Automation document.
Creates a change request for Change Manager.
Initiates a connection to a target (for example, an instance) for a Session Manager session.
Stop 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 an association.
Updates the status of the Systems Manager document associated with the system.
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 Systems Manager 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 that is assigned to the on-premises instance or virtual machines (VM).
Edit or change an OpsItem.
Systems Manager calls this API action when you edit OpsMetadata in Application Manager.
Modifies an existing patch baseline.
Update a resource data sync.
ServiceSetting is an account-level setting for an AWS service.
Examples

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

## End(Not run)
```

### 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 `describe_communications`, `add_communication_to_case`, and `add_attachments_to_set` 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**

```python
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**

```python
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
describe_services
describe_severity_levels
describe_trusted_advisor_check_refresh_statuses
describe_trusted_advisor_check_result
describe_trusted_advisor_checks
describe_trusted_advisor_check_summaries
refresh_trusted_advisor_check
resolve_case

Examples

## Not run:
svc <- support()
svc$add_attachments_to_set(
   Foo = 123
)

## End(Not run)
Index

accept_grant, 28
accept_handshake, 36
accept_portfolio_share, 47
activate_event_source, 18
add_attachments_to_set, 55, 56
add_communication_to_case, 55, 56
add_tags, 14
add_tags_to_resource, 52
applicationautoscaling, 3
applicationinsights, 5
assign_instance, 31
assign_volume, 31
associate_budget_with_resource, 47
associate_elastic_ip, 31
associate_kms_key, 21
associate_node, 34
associate_principal_with_portfolio, 47
associate_product_with_portfolio, 47
associate_service_action_with_provisioning_artifact, 47
associate_service_quota_template, 50
associate_tag_option_with_resource, 47
attach_elastic_load_balancer, 31
attach_instances, 8
attach_load_balancer_target_groups, 8
attach_load_balancers, 8
attach_policy, 36
autoscaling, 7
autoscalingsplans, 9
batch_associate_service_action_with_provisioning_artifact, 47
batch_delete_scheduled_action, 8
batch_disassociate_service_action_from_provisioning_artifact, 47
batch_get_aggregate_resource_config, 23
batch_get_resource_config, 23
batch_put_scheduled_update_group_action, 8
cancel_command, 52
cancel_export_task, 21
cancel_handshake, 36
cancel_instance_refresh, 8
cancel_maintenance_window_execution, 52
cancel_replay, 18
cancel_update_stack, 12
check_in_license, 28
checkout_borrow_license, 28
checkout_license, 28
clear_stack, 30, 31
cloudformation, 11
cloudtrail, 13
cloudwatch, 15
cloudwatchevents, 17
cloudwatchlogs, 19
complete_lifecycle_action, 8
copy_product, 47
create_account, 36
create_activation, 52
create_app, 31
create_application, 6
create_archive, 18
create_association, 52
create_association_batch, 52
create_auto_scaling_group, 8
create_backup, 34
create_case, 55, 56
create_change_set, 12
create_change_set, 6
create_constraint, 47
create_deployment, 31
create_document, 52
create_event_bus, 18
create_export_task, 21
create_gov_cloud_account, 36
INDEX

create_grant, 28
create_grant_version, 28
create_group, 40
create_instance, 31
create_launch_configuration, 8
create_layer, 31
create_license, 28
create_license_configuration, 28
create_license_version, 28
create_log_group, 21
create_log_pattern, 6
create_log_stream, 21
create_maintenance_window, 52
create_ops_item, 52
create_ops_metadata, 52
create_or_update_tags, 8
create_organization, 36
create_organizational_unit, 36
create_partner_event_source, 18
create_patch_baseline, 52
create_policy, 36
create_portfolio, 47
create_portfolio_share, 47
create_product, 47
create_provisioned_product_plan, 47
create_provisioning_artifact, 47
create_resource_data_sync, 52
create_scaling_plan, 10
create_server, 34
deactivate_event_source, 18
decline_handshake, 36
delete_activation, 52
delete_aggregation_authorization, 23
delete_alarms, 16
delete_anomaly_detector, 16
delete_app, 31
delete_application, 6
delete_archive, 18
delete_association, 52
delete_auto_scaling_group, 8
delete_backup, 34
delete_change_set, 12
delete_component, 6
delete_config_rule, 23
delete_configuration_aggregator, 23
delete_configuration_recorder, 23
delete_conformance_pack, 23
delete_constraint, 47
delete_dashboards, 16
delete_delivery_channel, 23
delete_destination, 21
delete_document, 52
delete_evaluation_results, 23
delete_event_bus, 18
delete_grant, 28
delete_group, 40
delete_insight_rules, 16
delete_instance, 31
delete_inVENTORY, 52
delete_launch_configuration, 8
delete_layer, 31
delete_license, 28
delete_license_configuration, 28
delete_license_version, 28
delete_log_group, 21
delete_log_pattern, 6
delete_log_stream, 21
delete_maintenance_window, 52
delete_metric_filter, 21
delete_notification_configuration, 8
delete_ops_metadata, 52
delete_organization, 36
delete_organization_config_rule, 23
delete_organization_conformance_pack, 23
delete_organizational_unit, 36
delete_parameter, 52
delete_parameters, 52
delete_partner_event_source, 18
delete_patch_baseline, 52
delete_pending_aggregation_request, 23
delete_policy, 8, 36
delete_portfolio, 47
delete_portfolio_share, 47
delete_product, 47
delete_provisioned_product_plan, 47
delete_provisioning_artifact, 47
delete_query_definition, 21
delete_remediation_configuration, 23
delete_remediation_exceptions, 23
delete_resource_config, 23
delete_resource_data_sync, 52
delete_resource_policy, 21
delete_retention_configuration, 23
delete_retention_policy, 21
delete_rule, 18
delete_scaling_plan, 10
delete_scaling_policy, 4
delete_scheduled_action, 4, 8
delete_server, 34
delete_service_action, 47
delete_service_quota_increase_request_from_template, 50
delete_stack, 12, 31
delete_stack_instances, 12
delete_stack_set, 12
delete_stored_query, 23
delete_subscription_filter, 21
delete_tag_option, 47
delete_tags, 8
delete_token, 28
delete_trail, 15
delete_user_profile, 31
deliver_config_snapshot, 23
deregister_delegated_administrator, 36
deregister_ecs_cluster, 31
deregister_elastic_ip, 31
deregister_instance, 31
deregister_managed_instance, 52
deregister_patch_baseline_for_patch_group, 52
deregister_rds_db_instance, 31
deregister_scalable_target, 4
deregister_target_from_maintenance_window, 52
deregister_task_from_maintenance_window, 52
deregister_type, 12
deregister_volume, 31
describe_account, 36
describe_account_attributes, 34
describe_account_limits, 8, 12
describe_activations, 52
describe_adjustment_types, 8
describe_affected_accounts_for_organization, 26
describe_affected_entities, 26
describe_affected_entities_for_organization, 26
describe_agent_versions, 31
describe_aggregate_compliance_by_config_rules, 23
describe_aggregation_authorizations, 23
describe_alarm_history, 16
describe_alarms, 16
describe_alarms_for_metric, 16
describe_anomaly_detectors, 16
describe_application, 6
describe_apps, 31
describe_archive, 18
describe_association, 52
describe_association_execution_targets, 52
describe_association_executions, 52
describe_attachment, 55, 56
describe_auto_scaling_groups, 8
describe_auto_scaling_instances, 8
describe_auto_scaling_notification_types, 8
describe_automation_executions, 52
describe_automation_step_executions, 52
describe_available_patches, 52
describe_backups, 34
describe_cases, 55, 56
describe_change_set, 12
describe_commands, 31
describe.communications, 55, 57
describe_compliance_by_config_rule, 23
describe_compliance_by_resource, 23
describe_component, 6
describe_component_configuration, 6
describe_component_configuration_recommendation, 6
describe_config_rule_evaluation_status, 23
describe_config_rules, 23
describe_configuration_aggregator_sources_status, 23
describe_configuration_aggregators, 23
describe_configuration_recorder_status, 23
describe_configuration_recorders, 23
describe_conformance_pack_compliance, 23
describe_conformance_pack_status, 24
describe_conformance_packs, 23
describe_constraint, 47
describe_copy_product_status, 47
describe_create_account_status, 36
describe_delivery_channel_status, 24
describe_delivery_channels, 24
describe_deployments, 31
describe_destinations, 21
describe_dimension_keys, 38
describe_document, 52
describe_document_permission, 52
describe_ecs_clusters, 31
describe_effective_instance_associations, 52
describe_effective_patches_for_patch_baseline, 52
describe_effective_policy, 36
describe_elastic_ips, 31
describe_elastic_load_balancers, 31
describe_entity_aggregates, 26
describe_event_aggregates, 26
describe_event_bus, 18
describe_event_details, 26
describe_event_details_for_organization, 26
describe_event_source, 18
describe_event_types, 26
describe_events, 26, 34
describe_events_for_organization, 26
describe_export_tasks, 21
describe_handshake, 36
describe_health_service_status_for_organization, 26
describe_insight_rules, 16
describe_instance_associations_status, 52
describe_instance_information, 52
describe_instance_patch_states, 53
describe_instance_patch_states_for_patch_group, 53
describe_instance_patches, 53
describe_instance_refreshes, 8
describe_instances, 31
describe_inventory_deletions, 53
describe_launch_configurations, 8
describe_layers, 31
describe_lifecycle_hook_types, 8
describe_lifecycle_hooks, 8
describe_load_balancer_target_groups, 8
describe_load_balancers, 8
describe_load_based_auto_scaling, 31
describe_log_groups, 21
describe_log_pattern, 6
describe_log_streams, 21
describe_maintenance_window_execution_task_invocations, 53
describe_maintenance_window_execution_tasks, 53
describe_maintenance_window_executions, 53
describe_maintenance_window_schedule, 53
describe_maintenance_window_targets, 53
describe_maintenance_window_tasks, 53
describe_maintenance_windows, 53
describe_maintenance_windows_for_target, 53
describe_metric_collection_types, 8
describe_metric_filters, 21
describe_my_user_profile, 31
describe_node_association_status, 34
describe_notification_configurations, 8
describe_observation, 6
describe_operating_systems, 31
describe_ops_items, 53
describe_organization, 36
describe_organization_config_rule_statuses, 24
describe_organization_config_rules, 24
describe_organization_conformance_pack_statuses, 24
describe_organization_conformance_packs, 24
describe_organizational_unit, 36
describe_parameters, 53
describe_partner_event_source, 18
describe_patch_baselines, 53
describe_patch_group_state, 53
describe_patch_groups, 53
describe_patch_properties, 53
INDEX

disassociate_kms_key, 21
disassociate_node, 35
disassociate_principal_from_portfolio, 48
disassociate_product_from_portfolio, 48
disassociate_service_action_from_provisioning_artifact, 48
disassociate_service_quota_template, 50
disassociate_tag_option_from_resource, 48
get_aws_organizations_access_status, 48
get_calendar_state, 53
get_command_invocation, 53
get_compliance_details_by_config_rule, 24
get_compliance_details_by_resource, 24
get_compliance_summary, 46
get_compliance_summary_by_config_rule, 24
get_compliance_summary_by_resource_type, 24
get_compliance_summary_by_resource_type, 24
get_conformance_pack_compliance_details, 24
get_conformance_pack_compliance_summary, 24
get_connection_status, 53
get_dashboard, 16
get_default_patch_baseline, 53
get_deployable_patch_snapshot_for_instance, 53
get_discovered_resource_counts, 24
get_document, 53
get_event_selectors, 15
get_grant, 28
get_group, 40
get_group_configuration, 40
get_group_query, 40
get_hostname_suggestion, 32
get_insight_rule_report, 16
get_insight_selectors, 15
get_inventory, 53
get_inventory_schema, 53
get_license, 28
get_license_configuration, 28
get_license_usage, 28
get_log_events, 21
get_log_group_fields, 21
get_log_record, 21
get_maintenance_window, 53
get_maintenance_window_execution, 53
get_maintenance_window_execution_task, 53
get_maintenance_window_execution_task_invocation, 53
get_maintenance_window_task, 53
get_metric_data, 16
get_metric_statistics, 17
get_aws_default_service_quota, 50
get_automation_execution, 53
get_automation_execution, 53
get_metric_widget_image, 17
get_ops_item, 53
get_ops_metadata, 53
get_ops_summary, 53
get_organization_config_rule_detailed_status, 24
get_organization_conformance_pack_detailed_status, 24
get_parameter, 53
get_parameter_history, 53
get_parameters, 53
get_parameters_by_path, 53
get_patch_baseline, 53
get_patch_baseline_for_patch_group, 53
get_provisioned_product_outputs, 48
get_query_results, 21
get_requested_service_quota_change, 50
get_resource_config_history, 24
get_resource_metrics, 38
get_resources, 46
get_scaling_plan_resource_forecast_data, 11
get_service_quota, 50
get_service_quota_increase_request_from_template, 50
get_service_setting, 53
get_service_settings, 28
get_stack_policy, 12
get_stored_query, 24
get_tag_keys, 46
get_tag_values, 46
get_tags, 40
get_template, 13
get_template_summary, 13
get_trail, 15
get_trail_status, 15
grant_access, 32
group_resources, 40
health, 25
import_as_provisioned_product, 48
invite_account_to_organization, 36
label_parameter_version, 53
leave_organization, 36
licensemanager, 27
list_accepted_portfolio_shares, 48
list_accounts, 36
list_accounts_for_parent, 36
list_aggregate_discovered_resources, 24
list_applications, 6
list_archives, 19
list_association_versions, 53
list_associations, 53
list_associations_for_license_configuration, 28
list_aws_default_service_quotas, 50
list_aws_service_access_for_organization, 37
list_budgets_for_resource, 48
list_change_sets, 13
list_children, 37
list_command_invocations, 53
list_commands, 53
list_compliance_items, 54
list_compliance_summaries, 54
list_components, 6
list_configuration_history, 6
list_constraints_for_portfolio, 48
list_create_account_status, 37
list_dashboards, 17
list_delegated_administrators, 37
list_delegated_services_for_account, 37
list_discovered_resources, 24
list_distributed_grants, 28
list_document_metadata_history, 54
list_document_versions, 54
list_documents, 54
list_event_buses, 19
list_event_sources, 19
list_exports, 13
list_failures_for_license_configuration_operations, 28
list_group_resources, 40
list_groups, 40
list_handshakes_for_account, 37
list_handshakes_for_organization, 37
list_imports, 13
list_inventory_entries, 54
list_launch_paths, 48
list_license_configurations, 28
list_license_specifications_for_resource, 28
list_license_versions, 28
list_licenses, 28
list_log_pattern_sets, 6
list_log_patterns, 6
list_metrics, 17
list_ops_item_events, 54
list_ops_metadata, 54
list_organization_portfolio_access, 48
list_organizational_units_for_parent, 37
list_parents, 37
list_partner_event_source_accounts, 19
list_partner_event_sources, 19
list_policies, 37
list_policies_for_target, 37
list_portfolio_access, 48
list_portfolios, 48
list_portfolios_for_product, 48
list_principals_for_portfolio, 48
list_problems, 6
list_provisioned_product_plans, 48
list_provisioning_artifacts, 48
list_provisioning_artifacts_for_service_action, 48
list_public_keys, 15
list_received_grants, 28
list_received_licenses, 28
list_record_history, 48
list_replays, 19
list_requested_service_quota_change_history, 50
list_requested_service_quota_change_history_by_quota, 50
list_resource_compliance_summaries, 54
list_resource_data_sync, 54
list_resource_inventory, 28
list_resources_for_tag_option, 48
list_roots, 37
list_rule_names_by_target, 19
list_rules, 19
list_service_actions, 48
list_service_actions_for_provisioning_artifact, 48
list_service_quota_increase_requests_in_template, 50
list_service_quotas, 50
list_services, 50
list_stack_instances, 13
list_stack_instances_for_provisioned_product, 54
list_stack_resources, 13
list_stack_set_operation_results, 13
list_stack_set_operations, 13
list_stack_sets, 13
list_stacks, 13
list_stored_queries, 24
list_tag_options, 48
list_tags, 15, 32
list_tags_for_resource, 6, 17, 19, 24, 28, 35, 37, 50, 54
list_tags_log_group, 21
list_targets_by_rule, 19
list_targets_for_policy, 37
list_tokens, 28
list_trails, 15
list_typeRegistrations, 13
list_type_versions, 13
list_types, 13
list_usage_for_license_configuration, 28
lookup_events, 15
modify_document_permission, 54
move_account, 37
opsworks, 29
opsworkscm, 33
organizations, 35
pi, 37
provision_product, 48
put_aggregation_authorization, 24
put_anomaly_detector, 17
put_compliance_items, 54
put_composite_alarm, 17
put_config_rule, 24
put_conformance_pack, 24
put_dashboard, 17
put_destination, 21
put_delivery_channel, 24
put_destination_policy, 21
put_evaluations, 24
put_event_selectors, 15
put_events, 19
put_external_evaluation, 24
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>put_insight_rule</td>
<td>17</td>
</tr>
<tr>
<td>put_insight_selectors</td>
<td>15</td>
</tr>
<tr>
<td>put_inventory</td>
<td>54</td>
</tr>
<tr>
<td>put_lifecycle_hook</td>
<td>9</td>
</tr>
<tr>
<td>put_log_events</td>
<td>21</td>
</tr>
<tr>
<td>put_metric_alarm</td>
<td>17</td>
</tr>
<tr>
<td>put_metric_data</td>
<td>17</td>
</tr>
<tr>
<td>put_metric_filter</td>
<td>21</td>
</tr>
<tr>
<td>put_notification_configuration</td>
<td>9</td>
</tr>
<tr>
<td>put_organization_config_rule</td>
<td>24</td>
</tr>
<tr>
<td>put_organization_conformance_pack</td>
<td>24</td>
</tr>
<tr>
<td>put_parameter</td>
<td>54</td>
</tr>
<tr>
<td>put_partner_events</td>
<td>19</td>
</tr>
<tr>
<td>put_permission</td>
<td>19</td>
</tr>
<tr>
<td>put_query_definition</td>
<td>21</td>
</tr>
<tr>
<td>put_remediation_configurations</td>
<td>24</td>
</tr>
<tr>
<td>put_remediation_exceptions</td>
<td>24</td>
</tr>
<tr>
<td>put_resource_config</td>
<td>24</td>
</tr>
<tr>
<td>put_resource_policy</td>
<td>21</td>
</tr>
<tr>
<td>put_retention_configuration</td>
<td>24</td>
</tr>
<tr>
<td>put_retention_policy</td>
<td>21</td>
</tr>
<tr>
<td>put_rule</td>
<td>19</td>
</tr>
<tr>
<td>put_scaling_policy</td>
<td>4, 9</td>
</tr>
<tr>
<td>put_scheduled_action</td>
<td>4</td>
</tr>
<tr>
<td>put_scheduled_update_group_action</td>
<td>9</td>
</tr>
<tr>
<td>put_service_quota_increase_request_into_template</td>
<td>50</td>
</tr>
<tr>
<td>put_stored_query</td>
<td>24</td>
</tr>
<tr>
<td>putSubscription_filter</td>
<td>21</td>
</tr>
<tr>
<td>put_targets</td>
<td>19</td>
</tr>
<tr>
<td>reboot_instance</td>
<td>32</td>
</tr>
<tr>
<td>record_handler_progress</td>
<td>13</td>
</tr>
<tr>
<td>record_lifecycle_action_heartbeat</td>
<td>9</td>
</tr>
<tr>
<td>refresh_trusted_advisor_check</td>
<td>56, 57</td>
</tr>
<tr>
<td>register_default_patch_baseline</td>
<td>54</td>
</tr>
<tr>
<td>register_delegated_administrator</td>
<td>37</td>
</tr>
<tr>
<td>register_ecs_cluster</td>
<td>32</td>
</tr>
<tr>
<td>register_elastic_ip</td>
<td>32</td>
</tr>
<tr>
<td>register_instance</td>
<td>32</td>
</tr>
<tr>
<td>register_patch_baseline_for_patch_group</td>
<td>54</td>
</tr>
<tr>
<td>register_rds_db_instance</td>
<td>32</td>
</tr>
<tr>
<td>register scalable_target</td>
<td>3, 4</td>
</tr>
<tr>
<td>register_target_with_maintenance_window</td>
<td>54</td>
</tr>
<tr>
<td>register_task_with_maintenance_window</td>
<td>54</td>
</tr>
<tr>
<td>register_type</td>
<td>13</td>
</tr>
<tr>
<td>register_volume</td>
<td>32</td>
</tr>
<tr>
<td>reject_grant</td>
<td>28</td>
</tr>
<tr>
<td>reject_portfolio_share</td>
<td>48</td>
</tr>
<tr>
<td>remove_account_from_organization</td>
<td>37</td>
</tr>
<tr>
<td>remove_permission</td>
<td>19</td>
</tr>
<tr>
<td>remove_tags</td>
<td>15</td>
</tr>
<tr>
<td>remove_tags_from_resource</td>
<td>54</td>
</tr>
<tr>
<td>remove_targets</td>
<td>19</td>
</tr>
<tr>
<td>request_service_quota_increase</td>
<td>50</td>
</tr>
<tr>
<td>reset_service_settings</td>
<td>54</td>
</tr>
<tr>
<td>resolve_case</td>
<td>55, 57</td>
</tr>
<tr>
<td>resourcegroups</td>
<td>39</td>
</tr>
<tr>
<td>resourcegroupstaggingapi</td>
<td>41</td>
</tr>
<tr>
<td>restore_server</td>
<td>35</td>
</tr>
<tr>
<td>resume_processes</td>
<td>9</td>
</tr>
<tr>
<td>resume_session</td>
<td>54</td>
</tr>
<tr>
<td>scan_provisioned_products</td>
<td>48</td>
</tr>
<tr>
<td>search_products</td>
<td>48</td>
</tr>
<tr>
<td>search_products_as_admin</td>
<td>48</td>
</tr>
<tr>
<td>search_provisioned_products</td>
<td>48</td>
</tr>
<tr>
<td>search_resources</td>
<td>40</td>
</tr>
<tr>
<td>select_aggregate_resource_config</td>
<td>24</td>
</tr>
<tr>
<td>select_resource_config</td>
<td>24</td>
</tr>
<tr>
<td>send_automation_signal</td>
<td>54</td>
</tr>
<tr>
<td>send_command</td>
<td>54</td>
</tr>
<tr>
<td>servicecatalog</td>
<td>46</td>
</tr>
<tr>
<td>servicequotas</td>
<td>49</td>
</tr>
<tr>
<td>set_alarm_state</td>
<td>17</td>
</tr>
<tr>
<td>set_desired_capacity</td>
<td>9</td>
</tr>
<tr>
<td>set_instance_health</td>
<td>9</td>
</tr>
<tr>
<td>set_instance_protection</td>
<td>9</td>
</tr>
<tr>
<td>set_load_based_auto_scaling</td>
<td>32</td>
</tr>
<tr>
<td>set_permission</td>
<td>32</td>
</tr>
<tr>
<td>set_stack_policy</td>
<td>13</td>
</tr>
<tr>
<td>set_time_based_auto_scaling</td>
<td>32</td>
</tr>
<tr>
<td>set_type_default_version</td>
<td>13</td>
</tr>
<tr>
<td>signal_resource</td>
<td>13</td>
</tr>
<tr>
<td>ssm</td>
<td>51</td>
</tr>
<tr>
<td>start_associations_once</td>
<td>54</td>
</tr>
<tr>
<td>start_automation_execution</td>
<td>54</td>
</tr>
<tr>
<td>start_change_request_execution</td>
<td>54</td>
</tr>
<tr>
<td>start_config_rules_evaluation</td>
<td>25</td>
</tr>
<tr>
<td>start_configuration_recorder</td>
<td>25</td>
</tr>
<tr>
<td>start_instance</td>
<td>32</td>
</tr>
<tr>
<td>start_instance_refresh</td>
<td>9</td>
</tr>
<tr>
<td>start_logging</td>
<td>15</td>
</tr>
<tr>
<td>start_maintenance</td>
<td>35</td>
</tr>
<tr>
<td>start_query</td>
<td>21</td>
</tr>
</tbody>
</table>
INDEX

start_remediation_execution, 25
start_replay, 19
start_report_creation, 46
start_session, 54
start_stack, 32
stop_automation_execution, 54
stop_configuration_recorder, 25
stop_instance, 32
stop_logging, 15
stop_query, 21
stop_stack, 32
stop_stack_set_operation, 13
support, 55
suspend_processes, 9
tag, 40
tag_log_group, 21
tag_resource, 6, 17, 19, 25, 28, 32, 35, 37, 50
ntag_resources, 46
terminate_instance_in_auto_scaling_group, 9
terminate_provisioned_product, 48
terminate_session, 54
test_event_pattern, 19
test_metric_filter, 21
unassign_instance, 32
unassign_volume, 32
ungroup_resources, 40
untag, 40
untag_log_group, 21
untag_resource, 6, 17, 19, 25, 28, 32, 35, 37, 50
untag_resources, 46
update_app, 32
update_application, 6
update_archive, 19
update_association, 54
update_association_status, 54
update_auto_scaling_group, 9
update_component, 6
update_component_configuration, 6
update_constraint, 49
update_document, 54
update_document_default_version, 54
update_document_metadata, 54
update_elastic_ip, 32
update_group, 40
update_group_query, 40
update_instance, 32
update_layer, 32
update_license_configuration, 28
update_license_specifications_for_resource, 28
update_log_pattern, 6
update_maintenance_window, 54
update_maintenance_window_target, 54
update_maintenance_window_task, 54
update_managed_instance_role, 54
update_my_user_profile, 32
update_ops_item, 54
update_ops_metadata, 54
update_organizational_unit, 37
update_patch_baseline, 54
update_policy, 37
update_portfolio, 49
update_portfolio_share, 49
update_product, 49
update_provisioned_product, 49
update_provisioned_product_properties, 49
update_provisioning_artifact, 49
update_rds_db_instance, 32
update_resource_data_sync, 54
update_scaling_plan, 11
update_server, 35
update_server_engine_attributes, 35
update_service_action, 49
update_service_setting, 54
update_service_settings, 28
update_stack, 13, 30, 32
update_stack_instances, 13
update_stack_set, 13
update_tag_option, 49
update_termination_protection, 13
update_trail, 15
update_user_profile, 32
update_volume, 32
validate_template, 13