# 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'

'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`

*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.

## 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 th |
| 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

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

## Value

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

## Service syntax

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

## Operations

| | |
|---|---|
| create_application | 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 an 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 togeth |
| describe_component_configuration | Describes the monitoring configuration of the component |
| describe_component_configuration_recommendation | Describes the recommended monitoring configuration of the compone |
| 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 app |
| list_configuration_history | Lists the INFO, WARN, and ERROR events for periodic configuration |
| list_log_patterns | Lists the log patterns in the specific log LogPatternSet |
| list_log_pattern_sets | Lists the log pattern sets in the specific application |
| list_problems | Lists the problems with your application |
| list_tags_for_resource | Retrieve a list of the tags (keys and values) that are associated with a s |
| 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 applicatio |
| update_application | Updates the application |
| update_component | Updates the custom component name and/or the list of resources that |
| update_component_configuration | Updates the monitoring configurations for the component |
| update_log_pattern | Adds a log pattern to a LogPatternSet |

## Examples

```
## 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

```
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

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

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

## Operations

| | |
|---|---|
| attach_instances | Attaches one or more EC2 instances to the specified Auto Scaling group |
| attach_load_balancers | To attach an Application Load Balancer, Network Load Balancer, or Gateway L |
| attach_load_balancer_target_groups | Attaches one or more target groups to the specified Auto Scaling group |
| batch_delete_scheduled_action | Deletes one or more scheduled actions for the specified Auto Scaling group |
| batch_put_scheduled_update_group_action | Creates or updates one or more scheduled scaling actions for an Auto Scaling gr |
| cancel_instance_refresh | Cancels an instance refresh operation in progress |
| complete_lifecycle_action | Completes the lifecycle action for the specified token or instance with the specif |
| create_auto_scaling_group | We strongly recommend using a launch template when calling this operation to |
| create_launch_configuration | Creates a launch configuration |
| create_or_update_tags | Creates or updates tags for the specified Auto Scaling group |
| delete_auto_scaling_group | Deletes the specified Auto Scaling group |
| delete_launch_configuration | Deletes the specified launch configuration |
| delete_lifecycle_hook | Deletes the specified lifecycle hook |
| delete_notification_configuration | Deletes the specified notification |
| delete_policy | Deletes the specified scaling policy |
| delete_scheduled_action | Deletes the specified scheduled action |
| delete_tags | Deletes the specified tags |
| describe_account_limits | Describes the current Amazon EC2 Auto Scaling resource quotas for your AWS |
| describe_adjustment_types | Describes the available adjustment types for Amazon EC2 Auto Scaling scaling |
| describe_auto_scaling_groups | Describes one or more Auto Scaling groups |
| describe_auto_scaling_instances | Describes one or more Auto Scaling instances |
| describe_auto_scaling_notification_types | Describes the notification types that are supported by Amazon EC2 Auto Scalin |
| describe_instance_refreshes | Describes one or more instance refreshes |
| describe_launch_configurations | Describes one or more launch configurations |
| describe_lifecycle_hooks | Describes the lifecycle hooks for the specified Auto Scaling group |
| describe_lifecycle_hook_types | Describes the available types of lifecycle hooks |
| describe_load_balancers | Describes the load balancers for the specified Auto Scaling group |
| describe_load_balancer_target_groups | Describes the target groups for the specified Auto Scaling group |
| describe_metric_collection_types | Describes the available CloudWatch metrics for Amazon EC2 Auto Scaling |
| describe_notification_configurations | Describes the notification actions associated with the specified Auto Scaling gro |
| describe_policies | Describes the policies for the specified Auto Scaling group |
| describe_scaling_activities | Describes one or more scaling activities for the specified Auto Scaling group |
| describe_scaling_process_types | Describes the scaling process types for use with the ResumeProcesses and Susp |
| describe_scheduled_actions | Describes the actions scheduled for your Auto Scaling group that haven't run or |
| describe_tags | Describes the specified tags |
| describe_termination_policy_types | Describes the termination policies supported by Amazon EC2 Auto Scaling |
| detach_instances | Removes one or more instances from the specified Auto Scaling group |
| detach_load_balancers | Detaches one or more Classic Load Balancers from the specified Auto Scaling g |
| detach_load_balancer_target_groups | Detaches one or more target groups from the specified Auto Scaling group |
| disable_metrics_collection | Disables group metrics for the specified Auto Scaling group |

| | |
|---|---|
| enable_metrics_collection | Enables group metrics for the specified Auto Scaling group |
| enter_standby | Moves the specified instances into the standby state |
| execute_policy | Executes the specified policy |
| exit_standby | Moves the specified instances out of the standby state |
| put_lifecycle_hook | Creates or updates a lifecycle hook for the specified Auto Scaling group |
| put_notification_configuration | Configures an Auto Scaling group to send notifications when specified events ta |
| 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 o |
| resume_processes | Resumes the specified suspended auto scaling processes, or all suspended proce |
| 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 a |
| suspend_processes | Suspends the specified auto scaling processes, or all processes, for the specified |
| 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 en |

## Examples

```
## 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)
```

---

`autoscalingplans` *AWS Auto Scaling Plans*

---

## Description

AWS Auto Scaling

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

### API Summary

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

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

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

## Usage

```
autoscalingplans(config = list())
```

## Arguments

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

## Value

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

## Service syntax

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

## Operations

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

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

## Examples

```
## 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

```
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

```
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 r |
| create_change_set | Creates a list of changes that will be applied to a stack so that you can review the char |
| 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 |
| describe_change_set | Returns the inputs for the change set and a list of changes that AWS CloudFormation |
| 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 |
| 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 spec |
| describe_stack_resources | Returns AWS resource descriptions for running and deleted stacks |
| describe_stacks | Returns the description for the specified stack; if no stack name was specified, then it |
| 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 a |
| detect_stack_drift | Detects whether a stack's actual configuration differs, or has drifted, from it's expecte |
| detect_stack_resource_drift | Returns information about whether a resource's actual configuration differs, or has dri |
| detect_stack_set_drift | Detect 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 char |
| get_stack_policy | Returns the stack policy for a specified stack |

| | |
|---|---|
| get_template | Returns the template body for a specified stack |
| get_template_summary | Returns information about a new or existing template |
| list_change_sets | Returns the ID and status of each active change set for a stack |
| list_exports | Lists all exported output values in the account and Region in which you call this actio |
| list_imports | Lists all stacks that are importing an exported output value |
| list_stack_instances | Returns summary information about stack instances that are associated with the speci |
| list_stack_resources | Returns descriptions of all resources of the specified stack |
| list_stacks | Returns the summary information for stacks whose status matches the specified Stack |
| list_stack_set_operation_results | Returns summary information about the results of a stack set operation |
| list_stack_set_operations | Returns summary information about operations performed on a stack set |
| list_stack_sets | Returns summary information about stack sets that are associated with the user |
| list_type_registrations | Returns a list of registration tokens for the specified type(s) |
| list_types | Returns summary information about types that have been registered with CloudForma |
| list_type_versions | Returns summary information about the versions of a type |
| record_handler_progress | Reports progress of a resource handler to CloudFormation |
| register_type | Registers a type with the CloudFormation service |
| set_stack_policy | Sets a stack policy for a specified stack |
| set_type_default_version | Specify the default version of a type |
| signal_resource | Sends a signal to the specified resource with a success or failure status |
| stop_stack_set_operation | Stops an in-progress operation on a stack set and its associated stack instances |
| update_stack | Updates a stack as specified in the template |
| update_stack_instances | Updates the parameter values for stack instances for the specified accounts, within the |
| update_stack_set | Updates the stack set, and associated stack instances in the specified accounts and Reg |
| update_termination_protection | Updates termination protection for the specified stack |
| validate_template | Validates a specified template |

## Examples

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

## End(Not run)
```

---

`cloudtrail` *AWS CloudTrail*

---

## Description

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

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

As an alternative to the API, you can use one of the AWS SDKs, which consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWSCloudTrail. For example, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For information about the AWS SDKs, including how to download and install them, see the Tools for Amazon Web Services page.

See the AWS CloudTrail User Guide for information about the data that is included with each AWS API call listed in the log files.

## Usage

```
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

```
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 |

| | |
|---|---|
| delete_trail | Deletes a trail |
| describe_trails | Retrieves settings for one or more trails associated with the current region for your account |
| get_event_selectors | Describes the settings for the event selectors that you configured for your trail |
| get_insight_selectors | Describes the settings for the Insights event selectors that you configured for your trail |
| get_trail | Returns settings information for a specified trail |
| get_trail_status | Returns a JSON-formatted list of information about the specified trail |
| list_public_keys | Returns all public keys whose private keys were used to sign the digest files within the specified time r |
| list_tags | Lists the tags for the trail in the current region |
| list_trails | Lists trails that are in the current account |
| lookup_events | Looks up management events or CloudTrail Insights events that are captured by CloudTrail |
| put_event_selectors | Configures an event selector or advanced event selectors for your trail |
| put_insight_selectors | Lets you enable Insights event logging by specifying the Insights selectors that you want to enable on |
| remove_tags | Removes the specified tags from a trail |
| start_logging | Starts the recording of AWS API calls and log file delivery for a trail |
| stop_logging | Suspends the recording of AWS API calls and log file delivery for the specified trail |
| update_trail | Updates the settings that specify delivery of log files |

## Examples

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

## End(Not run)
```

---

`cloudwatch` *Amazon CloudWatch*

---

## 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 req |

| | |
|---|---|
| get_metric_statistics | Gets statistics for the specified metric |
| get_metric_widget_image | You can use the GetMetricWidgetImage API to retrieve a snapshot graph of one or more Amaz |
| list_dashboards | Returns a list of the dashboards for your account |
| list_metrics | List the specified metrics |
| list_tags_for_resource | Displays the tags associated with a CloudWatch resource |
| put_anomaly_detector | Creates an anomaly detection model for a CloudWatch metric |
| put_composite_alarm | Creates or updates a composite alarm |
| put_dashboard | Creates a dashboard if it does not already exist, or updates an existing dashboard |
| put_insight_rule | Creates a Contributor Insights rule |
| put_metric_alarm | Creates or updates an alarm and associates it with the specified metric, metric math expression, |
| put_metric_data | Publishes metric data points to Amazon CloudWatch |
| set_alarm_state | Temporarily sets the state of an alarm for testing purposes |
| tag_resource | Assigns one or more tags (key-value pairs) to the specified CloudWatch resource |
| untag_resource | Removes one or more tags from the specified resource |

## Examples

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

## End(Not run)
```

---

`cloudwatchevents` *Amazon CloudWatch Events*

---

## Description

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

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

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

## Usage

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

## Arguments

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

## 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 partn |
| delete_archive | Deletes the specified archive |
| delete_event_bus | Deletes the specified custom event bus or partner event bus |
| delete_partner_event_source | This operation is used by SaaS partners to delete a partner event source |
| delete_rule | Deletes the specified rule |
| describe_archive | Retrieves details about an archive |
| describe_event_bus | Displays details about an event bus in your account |
| describe_event_source | This operation lists details about a partner event source that is shared with your account |
| describe_partner_event_source | An SaaS partner can use this operation to list details about a partner event source that th |
| describe_replay | Retrieves details about a replay |
| describe_rule | Describes the specified rule |

| | |
|---|---|
| disable_rule | Disables the specified rule |
| enable_rule | Enables the specified rule |
| list_archives | Lists your archives |
| list_event_buses | Lists all the event buses in your account, including the default event bus, custom event b |
| list_event_sources | You can use this to see all the partner event sources that have been shared with your AW |
| list_partner_event_source_accounts | An SaaS partner can use this operation to display the AWS account ID that a particular |
| list_partner_event_sources | An SaaS partner can use this operation to list all the partner event source names that the |
| list_replays | Lists your replays |
| list_rule_names_by_target | Lists the rules for the specified target |
| list_rules | Lists your Amazon EventBridge rules |
| list_tags_for_resource | Displays the tags associated with an EventBridge resource |
| list_targets_by_rule | Lists the targets assigned to the specified rule |
| put_events | Sends custom events to Amazon EventBridge so that they can be matched to rules |
| put_partner_events | This is used by SaaS partners to write events to a customer's partner event bus |
| put_permission | Running PutPermission permits the specified AWS account or AWS organization to put |
| put_rule | Creates or updates the specified rule |
| put_targets | Adds the specified targets to the specified rule, or updates the targets if they are already |
| remove_permission | Revokes the permission of another AWS account to be able to put events to the specifie |
| remove_targets | Removes the specified targets from the specified rule |
| start_replay | Starts the specified replay |
| tag_resource | Assigns one or more tags (key-value pairs) to the specified EventBridge resource |
| test_event_pattern | Tests whether the specified event pattern matches the provided event |
| untag_resource | Removes one or more tags from the specified EventBridge resource |
| update_archive | Updates the specified archive |

## Examples

```
## 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

```
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

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

## Operations

| | |
|---|---|
| associate_kms_key | Associates the specified AWS Key Management Service (AWS KMS) customer master key (C |
| 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 Ama |
| 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 |
| delete_log_group | Deletes the specified log group and permanently deletes all the archived log events associated |
| delete_log_stream | Deletes the specified log stream and permanently deletes all the archived log events associated |
| 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 |
| describe_query_definitions | This operation returns a paginated list of your saved CloudWatch Logs Insights query definitio |
| 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 ke |
| 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 |
| 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 | Creates or updates a resource policy allowing other AWS services to put log events to this acc |
| put_retention_policy | Sets the retention of the specified log group |
| put_subscription_filter | Creates or updates a subscription filter and associates it with the specified log group |
| start_query | Schedules a query of a log group using CloudWatch Logs Insights |
| stop_query | Stops a CloudWatch Logs Insights query that is in progress |
| tag_log_group | Adds or updates the specified tags for the specified log group |
| test_metric_filter | Tests the filter pattern of a metric filter against a sample of log event messages |
| untag_log_group | Removes the specified tags from the specified log group |

## Examples

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

## End(Not run)
```

---

`configservice` *AWS Config*

---

## 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

```
configservice(config = list())
```

## Arguments

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

## Value

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

## Service syntax

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

## Operations

| | |
|---|---|
| batch_get_aggregate_resource_config | Returns the current configuration items for resources that are present i |
| batch_get_resource_config | Returns the current configuration for one or more requested resources |
| delete_aggregation_authorization | Deletes the authorization granted to the specified configuration aggreg |
| delete_config_rule | Deletes the specified AWS Config rule and all of its evaluation results |
| delete_configuration_aggregator | Deletes the specified configuration aggregator and the aggregated data |
| delete_configuration_recorder | Deletes the configuration recorder |
| delete_conformance_pack | Deletes the specified conformance pack and all the AWS Config rules, |
| delete_delivery_channel | Deletes the delivery channel |
| delete_evaluation_results | Deletes the evaluation results for the specified AWS Config rule |
| delete_organization_config_rule | Deletes the specified organization config rule and all of its evaluation r |
| delete_organization_conformance_pack | Deletes the specified organization conformance pack and all of the cor |
| delete_pending_aggregation_request | Deletes pending authorization requests for a specified aggregator acco |
| delete_remediation_configuration | Deletes the remediation configuration |
| delete_remediation_exceptions | Deletes one or more remediation exceptions mentioned in the resource |
| delete_resource_config | Records the configuration state for a custom resource that has been del |
| delete_retention_configuration | Deletes the retention configuration |
| delete_stored_query | Deletes the stored query for an AWS account in an AWS Region |
| deliver_config_snapshot | Schedules delivery of a configuration snapshot to the Amazon S3 buck |
| describe_aggregate_compliance_by_config_rules | Returns a list of compliant and noncompliant rules with the number of |
| describe_aggregation_authorizations | Returns a list of authorizations granted to various aggregator accounts |
| describe_compliance_by_config_rule | Indicates whether the specified AWS Config rules are compliant |
| describe_compliance_by_resource | Indicates whether the specified AWS resources are compliant |
| describe_config_rule_evaluation_status | Returns status information for each of your AWS managed Config rule |
| describe_config_rules | Returns details about your AWS Config rules |
| describe_configuration_aggregators | Returns the details of one or more configuration aggregators |
| describe_configuration_aggregator_sources_status | Returns status information for sources within an aggregator |
| describe_configuration_recorders | Returns the details for the specified configuration recorders |
| describe_configuration_recorder_status | Returns the current status of the specified configuration recorder |
| describe_conformance_pack_compliance | Returns compliance details for each rule in that conformance pack |
| describe_conformance_packs | Returns a list of one or more conformance packs |

| | |
|---|---|
| describe_conformance_pack_status | Provides one or more conformance packs deployment status |
| describe_delivery_channels | Returns details about the specified delivery channel |
| describe_delivery_channel_status | Returns the current status of the specified delivery channel |
| describe_organization_config_rules | Returns a list of organization config rules |
| describe_organization_config_rule_statuses | Provides organization config rule deployment status for an organizatio |
| describe_organization_conformance_packs | Returns a list of organization conformance packs |
| describe_organization_conformance_pack_statuses | Provides organization conformance pack deployment status for an orga |
| describe_pending_aggregation_requests | Returns a list of all pending aggregation requests |
| describe_remediation_configurations | Returns the details of one or more remediation configurations |
| describe_remediation_exceptions | Returns the details of one or more remediation exceptions |
| describe_remediation_execution_status | Provides a detailed view of a Remediation Execution for a set of resou |
| describe_retention_configurations | Returns the details of one or more retention configurations |
| get_aggregate_compliance_details_by_config_rule | Returns the evaluation results for the specified AWS Config rule for a |
| get_aggregate_config_rule_compliance_summary | Returns the number of compliant and noncompliant rules for one or m |
| get_aggregate_discovered_resource_counts | Returns the resource counts across accounts and regions that are prese |
| get_aggregate_resource_config | Returns configuration item that is aggregated for your specific resourc |
| get_compliance_details_by_config_rule | Returns the evaluation results for the specified AWS Config rule |
| get_compliance_details_by_resource | Returns the evaluation results for the specified AWS resource |
| get_compliance_summary_by_config_rule | Returns the number of AWS Config rules that are compliant and nonco |
| get_compliance_summary_by_resource_type | Returns the number of resources that are compliant and the number tha |
| get_conformance_pack_compliance_details | Returns compliance details of a conformance pack for all AWS resour |
| get_conformance_pack_compliance_summary | Returns compliance details for the conformance pack based on the cur |
| get_discovered_resource_counts | Returns the resource types, the number of each resource type, and the |
| get_organization_config_rule_detailed_status | Returns detailed status for each member account within an organizatio |
| get_organization_conformance_pack_detailed_status | Returns detailed status for each member account within an organizatio |
| get_resource_config_history | Returns a list of configuration items for the specified resource |
| get_stored_query | Returns the details of a specific stored query |
| list_aggregate_discovered_resources | Accepts a resource type and returns a list of resource identifiers that ar |
| list_discovered_resources | Accepts a resource type and returns a list of resource identifiers for the |
| list_stored_queries | List the stored queries for an AWS account in an AWS Region |
| list_tags_for_resource | List the tags for AWS Config resource |
| put_aggregation_authorization | Authorizes the aggregator account and region to collect data from the s |
| put_config_rule | Adds or updates an AWS Config rule for evaluating whether your AW |
| put_configuration_aggregator | Creates and updates the configuration aggregator with the selected sou |
| put_configuration_recorder | Creates a new configuration recorder to record the selected resource co |
| put_conformance_pack | Creates or updates a conformance pack |
| put_delivery_channel | Creates a delivery channel object to deliver configuration information |
| put_evaluations | Used by an AWS Lambda function to deliver evaluation results to AW |
| put_external_evaluation | Put external evaluation |
| put_organization_config_rule | Adds or updates organization config rule for your entire organization e |
| put_organization_conformance_pack | Deploys conformance packs across member accounts in an AWS Orga |
| put_remediation_configurations | Adds or updates the remediation configuration with a specific AWS Co |
| put_remediation_exceptions | A remediation exception is when a specific resource is no longer consi |
| put_resource_config | Records the configuration state for the resource provided in the reques |
| put_retention_configuration | Creates and updates the retention configuration with details about reten |
| put_stored_query | Saves a new query or updates an existing saved query |
| select_aggregate_resource_config | Accepts a structured query language (SQL) SELECT command and an |
| select_resource_config | Accepts a structured query language (SQL) SELECT command, perfo |

| | |
|---|---|
| start_config_rules_evaluation | Runs an on-demand evaluation for the specified AWS Config rules aga |
| start_configuration_recorder | Starts recording configurations of the AWS resources you have selecte |
| start_remediation_execution | Runs an on-demand remediation for the specified AWS Config rules ag |
| stop_configuration_recorder | Stops recording configurations of the AWS resources you have selecte |
| tag_resource | Associates the specified tags to a resource with the specified resourceA |
| untag_resource | Deletes specified tags from a resource |

## Examples

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

## End(Not run)
```

---

`health` *AWS Health APIs and Notifications*

---

## Description

AWS Health

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

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

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

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

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

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

- Use the eventScopeCode parameter to specify whether to return AWS Health events that are public or account-specific.

- Use pagination to view all events from the response. For example, if you call the describe_events_for_organizatior 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

```
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

```
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 tha |
| describe_affected_entities | Returns a list of entities that have been affected by the specified events, bas |
| describe_affected_entities_for_organization | Returns a list of entities that have been affected by one or more events for |
| describe_entity_aggregates | Returns the number of entities that are affected by each of the specified eve |
| describe_event_aggregates | Returns the number of events of each event type (issue, scheduled change, |
| 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 |
| 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 Organi |
| 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 H |
| 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 Organizatio |

## Examples

```
## Not run:
svc <- health()
svc$describe_affected_accounts_for_organization(
  Foo = 123
)

## End(Not run)
```

---

`licensemanager` *AWS License Manager*

---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## Service syntax

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

```
    region = "string"
  )
)
```

## Operations

| | |
|---|---|
| accept_grant | Accepts the specified grant |
| check_in_license | Checks in the specified license |
| checkout_borrow_license | Checks out the specified license for offline use |
| checkout_license | Checks out the specified license |
| create_grant | Creates a grant for the specified license |
| create_grant_version | Creates a new version of the specified grant |
| create_license | Creates a license |
| create_license_configuration | Creates a license configuration |
| create_license_version | Creates a new version of the specified license |
| create_token | Creates a long-lived token |
| delete_grant | Deletes the specified grant |
| delete_license | Deletes the specified license |
| delete_license_configuration | Deletes the specified license configuration |
| delete_token | Deletes the specified token |
| extend_license_consumption | Extends the expiration date for license consumption |
| get_access_token | Gets a temporary access token to use with AssumeRoleWithWebIdentity |
| get_grant | Gets detailed information about the specified grant |
| get_license | Gets detailed information about the specified license |
| get_license_configuration | Gets detailed information about the specified license configuration |
| get_license_usage | Gets detailed information about the usage of the specified license |
| get_service_settings | Gets the License Manager settings for the current Region |
| list_associations_for_license_configuration | Lists the resource associations for the specified license configuration |
| list_distributed_grants | Lists the grants distributed for the specified license |
| list_failures_for_license_configuration_operations | Lists the license configuration operations that failed |
| list_license_configurations | Lists the license configurations for your account |
| list_licenses | Lists the licenses for your account |
| list_license_specifications_for_resource | Describes the license configurations for the specified resource |
| list_license_versions | Lists all versions of the specified license |
| list_received_grants | Lists grants that are received but not accepted |
| list_received_licenses | Lists received licenses |
| list_resource_inventory | Lists resources managed using Systems Manager inventory |
| list_tags_for_resource | Lists the tags for the specified license configuration |
| list_tokens | Lists your tokens |
| list_usage_for_license_configuration | Lists all license usage records for a license configuration, displaying lice |
| reject_grant | Rejects the specified grant |
| tag_resource | Adds the specified tags to the specified license configuration |
| untag_resource | Removes the specified tags from the specified license configuration |
| update_license_configuration | Modifies the attributes of an existing license configuration |
| update_license_specifications_for_resource | Adds or removes the specified license configurations for the specified AV |
| update_service_settings | Updates License Manager settings for the current Region |

## Examples

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

## End(Not run)
```

---

opsworks *AWS OpsWorks*

---

## Description

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

AWS OpsWorks Stacks is an application management service that provides an integrated experience for overseeing the complete application lifecycle. For information about this product, go to the AWS OpsWorks 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 a registered instance to a layer |
| assign_volume | Assigns one of the stack's registered Amazon EBS volumes to a specified instance |
| associate_elastic_ip | Associates one of the stack's registered Elastic IP addresses with a specified instan |
| attach_elastic_load_balancer | Attaches an Elastic Load Balancing load balancer to a specified layer |
| clone_stack | Creates a clone of a specified stack |
| create_app | Creates an app for a specified stack |
| create_deployment | Runs deployment or stack commands |
| create_instance | Creates an instance in a specified stack |
| create_layer | Creates a layer |
| create_stack | Creates a new stack |
| create_user_profile | Creates a new user profile |
| delete_app | Deletes a specified app |
| delete_instance | Deletes a specified instance, which terminates the associated Amazon EC2 instanc |
| delete_layer | Deletes a specified layer |
| delete_stack | Deletes a specified stack |
| delete_user_profile | Deletes a user profile |
| deregister_ecs_cluster | Deregisters a specified Amazon ECS cluster from a stack |
| deregister_elastic_ip | Deregisters a specified Elastic IP address |
| deregister_instance | Deregister a registered Amazon EC2 or on-premises instance |
| deregister_rds_db_instance | Deregisters an Amazon RDS instance |
| deregister_volume | Deregisters an Amazon EBS volume |
| describe_agent_versions | Describes the available AWS OpsWorks Stacks agent versions |
| describe_apps | Requests a description of a specified set of apps |
| describe_commands | Describes the results of specified commands |
| describe_deployments | Requests a description of a specified set of deployments |
| describe_ecs_clusters | Describes Amazon ECS clusters that are registered with a stack |
| describe_elastic_ips | Describes Elastic IP addresses |
| describe_elastic_load_balancers | Describes a stack's Elastic Load Balancing instances |
| describe_instances | Requests a description of a set of instances |
| describe_layers | Requests a description of one or more layers in a specified stack |
| describe_load_based_auto_scaling | Describes load-based auto scaling configurations for specified layers |
| describe_my_user_profile | Describes a user's SSH information |
| describe_operating_systems | Describes the operating systems that are supported by AWS OpsWorks Stacks |
| describe_permissions | Describes the permissions for a specified stack |
| describe_raid_arrays | Describe an instance's RAID arrays |
| describe_rds_db_instances | Describes Amazon RDS instances |
| describe_service_errors | Describes AWS OpsWorks Stacks service errors |
| describe_stack_provisioning_parameters | Requests a description of a stack's provisioning parameters |
| describe_stacks | Requests a description of one or more stacks |
| describe_stack_summary | Describes the number of layers and apps in a specified stack, and the number of ins |
| describe_time_based_auto_scaling | Describes time-based auto scaling configurations for specified instances |
| describe_user_profiles | Describe specified users |

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

## Examples

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

## End(Not run)
```

opsworkscm *AWS OpsWorks CM*

## 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 `ChefAutomate` 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 Operations section.

## Service syntax

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

## Operations

| | |
|---|---|
| associate_node | Associates a new node with the server |
| create_backup | Creates an application-level backup of a server |
| create_server | Creates and immedately 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 |

| | |
|---|---|
| disassociate_node | Disassociates a node from an AWS OpsWorks CM server, and removes the node from the |
| export_server_engine_attribute | Exports a specified server engine attribute as a base64-encoded string |
| list_tags_for_resource | Returns a list of tags that are applied to the specified AWS OpsWorks for Chef Automate |
| restore_server | Restores a backup to a server that is in a CONNECTION_LOST, HEALTHY, RUNNING |
| start_maintenance | Manually starts server maintenance |
| tag_resource | Applies tags to an AWS OpsWorks for Chef Automate or AWS OpsWorks for Puppet Ent |
| untag_resource | Removes specified tags from an AWS OpsWorks-CM server or backup |
| update_server | Updates settings for a server |
| update_server_engine_attributes | Updates engine-specific attributes on a specified server |

## Examples

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

## End(Not run)
```

---

`organizations` *AWS 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

```
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 |
| attach_policy | Attaches a policy to a root, an organizational unit (OU), or an individual account |
| cancel_handshake | Cancels a handshake |
| create_account | Creates an AWS account that is automatically a member of the organization whos |
| create_gov_cloud_account | This action is available if all of the following are true: |
| create_organization | Creates an AWS organization |
| create_organizational_unit | Creates an organizational unit (OU) within a root or parent OU |
| create_policy | Creates a policy of a specified type that you can attach to a root, an organizationa |
| 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 |
| 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_organization | Retrieves information about the organization that the user's account belongs to |
| describe_organizational_unit | Retrieves information about an organizational unit (OU) |
| describe_policy | Retrieves information about a policy |
| detach_policy | Detaches a policy from a target root, organizational unit (OU), or account |
| disable_aws_service_access | Disables the integration of an AWS service (the service that is specified by Servic |
| 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 Service |
| 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 acco |
| 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 roo |

| | |
|---|---|
| list_aws_service_access_for_organization | Returns a list of the AWS services that you enabled to integrate with your organiz |
| list_children | Lists all of the organizational units (OUs) or accounts that are contained in the sp |
| list_create_account_status | Lists the account creation requests that match the specified status that is currently |
| list_delegated_administrators | Lists the AWS accounts that are designated as delegated administrators in this org |
| list_delegated_services_for_account | List the AWS services for which the specified account is a delegated administrato |
| list_handshakes_for_account | Lists the current handshakes that are associated with the account of the requesting |
| list_handshakes_for_organization | Lists the handshakes that are associated with the organization that the requesting |
| list_organizational_units_for_parent | Lists the organizational units (OUs) in a parent organizational unit or root |
| list_parents | Lists the root or organizational units (OUs) that serve as the immediate parent of |
| list_policies | Retrieves the list of all policies in an organization of a specified type |
| list_policies_for_target | Lists the policies that are directly attached to the specified target root, organizatio |
| list_roots | Lists the roots that are defined in the current organization |
| list_tags_for_resource | Lists tags that are attached to the specified resource |
| list_targets_for_policy | Lists all the roots, organizational units (OUs), and accounts that the specified poli |
| move_account | Moves an account from its current source parent root or organizational unit (OU) |
| register_delegated_administrator | Enables the specified member account to administer the Organizations features of |
| remove_account_from_organization | Removes the specified account from the organization |
| tag_resource | Adds one or more tags to the specified resource |
| untag_resource | Removes any tags with the specified keys from the specified resource |
| update_organizational_unit | Renames the specified organizational unit (OU) |
| update_policy | Updates an existing policy with a new name, description, or content |

## Examples

```
## 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

```
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

```
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

```
## 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

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

## Arguments

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

## Value

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

## Service syntax

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

## Operations

| | |
|---|---|
| create_group | Creates a resource group with the specified name and description |
| delete_group | Deletes the specified resource group |
| get_group | Returns information about a specified resource group |
| get_group_configuration | Returns the service configuration associated with the specified resource group |
| get_group_query | Retrieves the resource query associated with the specified resource group |
| get_tags | Returns a list of tags that are associated with a resource group, specified by an ARN |
| group_resources | Adds the specified resources to the specified group |
| list_group_resources | Returns a list of ARNs of the resources that are members of a specified resource group |
| list_groups | Returns a list of existing resource groups in your account |
| put_group_configuration | Attaches a service configuration to the specified group |
| search_resources | Returns a list of AWS resource identifiers that matches the specified query |
| tag | Adds tags to a resource group with the specified ARN |
| ungroup_resources | Removes the specified resources from the specified group |
| untag | Deletes tags from a specified resource group |
| update_group | Updates the description for an existing group |
| update_group_query | Updates the resource query of a group |

## Examples

```
## Not run:
svc <- resourcegroups()
svc$create_group(
```

```
  Foo = 123
)

## End(Not run)
```

---

`resourcegroupstaggingapi` *AWS Resource Groups Tagging API*

---

## 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

- 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

| | |
|---|---|
| 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 t |
| 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 wh |
| tag_resources | Applies one or more tags to the specified resources |
| untag_resources | Removes the specified tags from the specified resources |

## Examples

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

## End(Not run)
```

---

`servicecatalog` *AWS Service Catalog*

---

## Description

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

```
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

```
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 po |
| 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 | Associate the specified TagOption with the specified portfoli |
| batch_associate_service_action_with_provisioning_artifact | Associates multiple self-service actions with provisioning art |
| batch_disassociate_service_action_from_provisioning_artifact | Disassociates a batch of self-service actions from the specifie |
| copy_product | Copies the specified source product to the specified target pro |
| create_constraint | Creates a constraint |
| create_portfolio | Creates a portfolio |
| create_portfolio_share | Shares the specified portfolio with the specified account or o |
| 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 t |
| 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 accou |
| 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 v |
| 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 c |
| describe_portfolio_share_status | Gets the status of the specified portfolio share operation |

| | |
|---|---|
| describe_product | Gets information about the specified product |
| describe_product_as_admin | Gets information about the specified product |
| describe_product_view | Gets information about the specified product |
| describe_provisioned_product | Gets information about the specified provisioned product |
| describe_provisioned_product_plan | Gets information about the resource changes for the specifie |
| describe_provisioning_artifact | Gets information about the specified provisioning artifact (al |
| describe_provisioning_parameters | Gets information about the configuration required to provisio |
| describe_record | Gets information about the specified request operation |
| describe_service_action | Describes a self-service action |
| describe_service_action_execution_parameters | Finds the default parameters for a specific self-service action |
| describe_tag_option | Gets information about the specified TagOption |
| disable_aws_organizations_access | Disable portfolio sharing through AWS Organizations feature |
| disassociate_budget_from_resource | Disassociates the specified budget from the specified resourc |
| disassociate_principal_from_portfolio | Disassociates a previously associated principal ARN from a s |
| disassociate_product_from_portfolio | Disassociates the specified product from the specified portfol |
| disassociate_service_action_from_provisioning_artifact | Disassociates the specified self-service action association fro |
| disassociate_tag_option_from_resource | Disassociates the specified TagOption from the specified reso |
| enable_aws_organizations_access | Enable portfolio sharing feature through AWS Organizations |
| execute_provisioned_product_plan | Provisions or modifies a product based on the resource chang |
| execute_provisioned_product_service_action | Executes a self-service action against a provisioned product |
| get_aws_organizations_access_status | Get the Access Status for AWS Organization portfolio share |
| get_provisioned_product_outputs | This API takes either a ProvisonedProductId or a Provisioned |
| import_as_provisioned_product | Requests the import of a resource as a Service Catalog provis |
| list_accepted_portfolio_shares | Lists all portfolios for which sharing was accepted by this ac |
| list_budgets_for_resource | Lists all the budgets associated to the specified resource |
| list_constraints_for_portfolio | Lists the constraints for the specified portfolio and product |
| list_launch_paths | Lists the paths to the specified product |
| list_organization_portfolio_access | Lists the organization nodes that have access to the specified |
| list_portfolio_access | Lists the account IDs that have access to the specified portfol |
| list_portfolios | Lists all portfolios in the catalog |
| list_portfolios_for_product | Lists all portfolios that the specified product is associated wi |
| list_principals_for_portfolio | Lists all principal ARNs associated with the specified portfol |
| list_provisioned_product_plans | Lists the plans for the specified provisioned product or all pla |
| list_provisioning_artifacts | Lists all provisioning artifacts (also known as versions) for th |
| list_provisioning_artifacts_for_service_action | Lists all provisioning artifacts (also known as versions) for th |
| list_record_history | Lists the specified requests or all performed requests |
| list_resources_for_tag_option | Lists the resources associated with the specified TagOption |
| list_service_actions | Lists all self-service actions |
| list_service_actions_for_provisioning_artifact | Returns a paginated list of self-service actions associated wit |
| list_stack_instances_for_provisioned_product | Returns summary information about stack instances that are s |
| list_tag_options | Lists the specified TagOptions or all TagOptions |
| provision_product | Provisions the specified product |
| reject_portfolio_share | Rejects an offer to share the specified portfolio |
| scan_provisioned_products | Lists the provisioned products that are available (not termina |
| search_products | Gets information about the products to which the caller has a |
| search_products_as_admin | Gets information about the products for the specified portfoli |
| search_provisioned_products | Gets information about the provisioned products that meet th |
| terminate_provisioned_product | Terminates the specified provisioned product |

| | |
|---|---|
| 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 provisi |
| update_provisioned_product_properties | Requests updates to the properties of the specified provisione |
| update_provisioning_artifact | Updates the specified provisioning artifact (also known as a v |
| 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 | Associates your quota request template with your organization |
| delete_service_quota_increase_request_from_template | Deletes the quota increase request for the specified quota from your |
| disassociate_service_quota_template | Disables your quota request template |
| get_association_for_service_quota_template | Retrieves the status of the association for the quota request template |
| get_aws_default_service_quota | Retrieves the default value for the specified quota |
| get_requested_service_quota_change | Retrieves information about the specified quota increase request |
| get_service_quota | Retrieves the applied quota value for the specified quota |
| get_service_quota_increase_request_from_template | Retrieves information about the specified quota increase request in |
| list_aws_default_service_quotas | Lists the default values for the quotas for the specified AWS service |
| list_requested_service_quota_change_history | Retrieves the quota increase requests for the specified service |
| list_requested_service_quota_change_history_by_quota | Retrieves the quota increase requests for the specified quota |
| list_service_quota_increase_requests_in_template | Lists the quota increase requests in the specified quota request temp |
| list_service_quotas | Lists the applied quota values for the specified AWS service |
| list_services | Lists the names and codes for the services integrated with Service C |
| list_tags_for_resource | Returns a list of the tags assigned to the specified applied quota |
| put_service_quota_increase_request_into_template | Adds a quota increase request to your quota request template |
| request_service_quota_increase | Submits a quota increase request for the specified quota |
| tag_resource | Adds tags to the specified applied quota |
| untag_resource | Removes tags from the specified applied quota |

## Examples

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

```
## End(Not run)
```

---

ssm *Amazon Simple Systems Manager (SSM)*

---

## Description

AWS Systems Manager

AWS Systems Manager is a collection of capabilities that helps you automate management tasks such as collecting system inventory, applying operating system (OS) patches, automating the creation of Amazon Machine Images (AMIs), and configuring operating systems (OSs) and applications at scale. Systems Manager lets you remotely and securely manage the configuration of your managed instances. A *managed instance* is any Amazon Elastic Compute Cloud instance (EC2 instance), or any on-premises server or virtual machine (VM) in your hybrid environment that has been configured for Systems Manager.

This reference is intended to be used with the AWS Systems Manager User Guide.

To get started, verify prerequisites and configure managed instances. For more information, see Setting up AWS Systems Manager in the *AWS Systems Manager User Guide*.

For information about other API actions you can perform on EC2 instances, see the Amazon EC2 API Reference. For information about how to use a Query API, see Making API requests.

## 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

```
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 progr |
| create_activation | Generates an activation code and activation ID you can use to r |
| create_association | A State Manager association defines the state that you want to |
| create_association_batch | Associates the specified Systems Manager document with the s |
| 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, Systen |
| create_patch_baseline | Creates a patch baseline |
| create_resource_data_sync | A resource data sync helps you view data from multiple source |
| delete_activation | Deletes an activation |
| delete_association | Disassociates the specified Systems Manager document from th |
| delete_document | Deletes the Systems Manager document and all instance associ |
| delete_inventory | Delete a custom inventory type or the data associated with a cu |
| delete_maintenance_window | Deletes a maintenance window |
| delete_ops_metadata | Delete OpsMetadata related to an application |
| delete_parameter | Delete a parameter from the system |
| delete_parameters | Delete a list of parameters |
| delete_patch_baseline | Deletes a patch baseline |
| delete_resource_data_sync | Deletes a Resource Data Sync configuration |
| deregister_managed_instance | Removes the server or virtual machine from the list of registere |
| deregister_patch_baseline_for_patch_group | Removes a patch group from a 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 |
| 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 associa |
| describe_association_execution_targets | Use this API action to view information about a specific execut |
| describe_automation_executions | Provides details about all active and terminated Automation ex |
| describe_automation_step_executions | Information about all active and terminated step executions in a |
| 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 appro |
| 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 |

| | |
|---|---|
| describe_instance_patches | Retrieves information about the patches on the specified instanc |
| describe_instance_patch_states | Retrieves the high-level patch state of one or more instances |
| describe_instance_patch_states_for_patch_group | Retrieves the high-level patch state for the instances in the spec |
| describe_inventory_deletions | Describes a specific delete inventory operation |
| describe_maintenance_window_executions | Lists the executions of a maintenance window |
| describe_maintenance_window_execution_task_invocations | Retrieves the individual task executions (one per target) for a pa |
| describe_maintenance_window_execution_tasks | For a given maintenance window execution, lists the tasks that |
| describe_maintenance_windows | Retrieves the maintenance windows in an AWS account |
| describe_maintenance_window_schedule | Retrieves information about upcoming executions of a mainten |
| describe_maintenance_windows_for_target | Retrieves information about the maintenance window targets or |
| describe_maintenance_window_targets | Lists the targets registered with the maintenance window |
| describe_maintenance_window_tasks | Lists the tasks in a maintenance window |
| describe_ops_items | Query a set of OpsItems |
| describe_parameters | Get information about a parameter |
| describe_patch_baselines | Lists the patch baselines in your AWS account |
| describe_patch_groups | Lists all patch groups that have been registered with patch base |
| describe_patch_group_state | Returns high-level aggregated patch compliance state for a patc |
| describe_patch_properties | Lists the properties of available patches organized by product, p |
| describe_sessions | Retrieves a list of all active sessions (both connected and disco |
| get_automation_execution | Get detailed information about a particular Automation executi |
| get_calendar_state | Gets the state of the AWS Systems Manager Change Calendar a |
| get_command_invocation | Returns detailed information about command execution for an i |
| get_connection_status | Retrieves the Session Manager connection status for an instanc |
| get_default_patch_baseline | Retrieves the default patch baseline |
| get_deployable_patch_snapshot_for_instance | Retrieves the current snapshot for the patch baseline the instanc |
| get_document | Gets the contents of the specified Systems Manager document |
| get_inventory | Query inventory information |
| get_inventory_schema | Return a list of inventory type names for the account, or return |
| get_maintenance_window | Retrieves a maintenance window |
| get_maintenance_window_execution | Retrieves details about a specific a maintenance window execut |
| get_maintenance_window_execution_task | Retrieves the details about a specific task run as part of a maint |
| get_maintenance_window_execution_task_invocation | Retrieves information about a specific task running on a specifi |
| get_maintenance_window_task | Lists the tasks in a maintenance window |
| get_ops_item | Get information about an OpsItem by using the ID |
| get_ops_metadata | View operational metadata related to an application in Applicat |
| get_ops_summary | View a summary of OpsItems based on specified filters and agg |
| get_parameter | Get information about a parameter by using the parameter nam |
| get_parameter_history | Retrieves the history of all changes to a parameter |
| get_parameters | Get details of a parameter |
| get_parameters_by_path | Retrieve information about one or more parameters in a specifi |
| get_patch_baseline | Retrieves information about a patch baseline |
| get_patch_baseline_for_patch_group | Retrieves the patch baseline that should be used for the specifie |
| get_service_setting | ServiceSetting is an account-level setting for an AWS service |
| label_parameter_version | A parameter label is a user-defined alias to help you manage di |
| list_associations | Returns all State Manager associations in the current AWS acc |
| list_association_versions | Retrieves all versions of an association for a specific associatio |
| list_command_invocations | An invocation is copy of a command sent to a specific instance |
| list_commands | Lists the commands requested by users of the AWS account |

| | |
|---|---|
| list_compliance_items | For a specified resource ID, this API action returns a list of con |
| list_compliance_summaries | Returns a summary count of compliant and non-compliant reso |
| list_document_metadata_history | Information about approval reviews for a version of an SSM do |
| list_documents | Returns all Systems Manager (SSM) documents in the current |
| list_document_versions | List all versions for a document |
| list_inventory_entries | A list of inventory items returned by the request |
| list_ops_item_events | Returns a list of all OpsItem events in the current AWS accoun |
| list_ops_metadata | Systems Manager calls this API action when displaying all Ap |
| list_resource_compliance_summaries | Returns a resource-level summary count |
| list_resource_data_sync | Lists your resource data sync configurations |
| list_tags_for_resource | Returns a list of the tags assigned to the specified resource |
| modify_document_permission | Shares a Systems Manager document publicly or privately |
| put_compliance_items | Registers a compliance type and other compliance details on a |
| put_inventory | Bulk update custom inventory items on one more instance |
| put_parameter | Add a parameter to the system |
| register_default_patch_baseline | Defines the default patch baseline for the relevant operating sys |
| register_patch_baseline_for_patch_group | Registers a patch baseline for a patch group |
| register_target_with_maintenance_window | Registers a target with a maintenance window |
| register_task_with_maintenance_window | Adds a new task to a maintenance window |
| remove_tags_from_resource | Removes tag keys from the specified resource |
| reset_service_setting | ServiceSetting is an account-level setting for an AWS service |
| resume_session | Reconnects a session to an instance after it has been disconnect |
| send_automation_signal | Sends a signal to an Automation execution to change the curren |
| send_command | Runs commands on one or more managed instances |
| start_associations_once | Use this API action to run an association immediately and only |
| start_automation_execution | Initiates execution of an Automation document |
| start_change_request_execution | Creates a change request for Change Manager |
| start_session | Initiates a connection to a target (for example, an instance) for |
| stop_automation_execution | Stop an Automation that is currently running |
| terminate_session | Permanently ends a session and closes the data connection betw |
| update_association | Updates an association |
| update_association_status | Updates the status of the Systems Manager document associate |
| update_document | Updates one or more values for an SSM document |
| update_document_default_version | Set the default version of a document |
| update_document_metadata | Updates information related to approval reviews for a specific v |
| update_maintenance_window | Updates an existing maintenance window |
| update_maintenance_window_target | Modifies the target of an existing maintenance window |
| update_maintenance_window_task | Modifies a task assigned to a maintenance window |
| update_managed_instance_role | Changes the Amazon Identity and Access Management (IAM) |
| update_ops_item | Edit or change an OpsItem |
| update_ops_metadata | Systems Manager calls this API action when you edit OpsMeta |
| update_patch_baseline | Modifies an existing patch baseline |
| update_resource_data_sync | Update a resource data sync |
| update_service_setting | ServiceSetting is an account-level setting for an AWS service |

## Examples

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

## End(Not run)
```

---

`support` *AWS Support*

---

## Description

The AWS Support API reference is intended for programmers who need detailed information about the AWS Support operations and data types. This service enables you to manage your AWS Support cases programmatically. It uses HTTP methods that return results in JSON format.

- You must have a Business or Enterprise support plan to use the AWS Support API.
- If you call the AWS Support API from an account that does not have a Business or Enterprise support plan, the `SubscriptionRequiredException` error message appears. For information about changing your support plan, see AWS Support.

The AWS Support service also exposes a set of AWS Trusted Advisor features. You can retrieve a list of checks and their descriptions, get check results, specify checks to refresh, and get the refresh status of checks.

The following list describes the AWS Support case management operations:

- **Service names, issue categories, and available severity levels.** The `describe_services` and `describe_severity_levels` operations return AWS service names, service codes, service categories, and problem severity levels. You use these values when you call the `create_case` operation.
- **Case creation, case details, and case resolution.** The `create_case`, `describe_cases`, `describe_attachment`, and `resolve_case` operations create AWS Support cases, retrieve information about cases, and resolve cases.
- **Case communication.** The `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

```
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

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

## Operations

| | |
|---|---|
| add_attachments_to_set | Adds one or more attachments to an attachment set |
| add_communication_to_case | Adds additional customer communication to an AWS Support case |
| create_case | Creates a case in the AWS Support Center |
| describe_attachment | Returns the attachment that has the specified ID |
| describe_cases | Returns a list of cases that you specify by passing one or more case IDs |

| | |
|---|---|
| describe_communications | Returns communications and attachments for one or more support cases |
| describe_services | Returns the current list of AWS services and a list of service categories for |
| describe_severity_levels | Returns the list of severity levels that you can assign to an AWS Support c |
| describe_trusted_advisor_check_refresh_statuses | Returns the refresh status of the AWS Trusted Advisor checks that have th |
| describe_trusted_advisor_check_result | Returns the results of the AWS Trusted Advisor check that has the specifie |
| describe_trusted_advisor_checks | Returns information about all available AWS Trusted Advisor checks, incl |
| describe_trusted_advisor_check_summaries | Returns the results for the AWS Trusted Advisor check summaries for the |
| refresh_trusted_advisor_check | Refreshes the AWS Trusted Advisor check that you specify using the chec |
| resolve_case | Resolves a support case |

## Examples

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

## End(Not run)
```

# Index