Package 'paws.management'

May 16, 2024

Title 'Amazon Web Services' Management & Governance Services

Version 0.6.1

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 (>= 0.6.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.1

Collate 'applicationautoscaling_service.R'
'applicationautoscaling_interfaces.R'
'applicationautoscaling_operations.R'
'applicationcostprofiler_service.R'
'applicationcostprofiler_interfaces.R'
'applicationcostprofiler_operations.R'
'applicationinsights_service.R'
'applicationinsights_interfaces.R'
'applicationinsights_operations.R' 'appregistry_service.R'
'appregistry_interfaces.R' 'appregistry_operations.R'
'auditmanager_service.R' 'auditmanager_interfaces.R'
'auditmanager_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' 'cloudtraildataservice_service.R'
'cloudtraildataservice_interfaces.R'

'cloudtraildataservice_operations.R' 'cloudwatch_service.R'
'cloudwatch_interfaces.R' 'cloudwatch_operations.R'
'cloudwatchevents_service.R' 'cloudwatchevents_interfaces.R'
'cloudwatchevents_operations.R' 'cloudwatchevidently_service.R'
'cloudwatchevidently_interfaces.R'
'cloudwatchevidently_operations.R'
'cloudwatchinternetmonitor_service.R'
'cloudwatchinternetmonitor_interfaces.R'
'cloudwatchinternetmonitor_operations.R'
'cloudwatchlogs_service.R' 'cloudwatchlogs_interfaces.R'
'cloudwatchlogs_operations.R'
'cloudwatchobservabilityaccessmanager_service.R'
'cloudwatchobservabilityaccessmanager_interfaces.R'
'cloudwatchobservabilityaccessmanager_operations.R'
'cloudwatchrum_service.R' 'cloudwatchrum_interfaces.R'
'cloudwatchrum_operations.R' 'configservice_service.R'
'configservice_interfaces.R' 'configservice_operations.R'
'controltower_service.R' 'controltower_interfaces.R'
'controltower_operations.R' 'finspace_service.R'
'finspace_interfaces.R' 'finspace_operations.R'
'health_service.R' 'health_interfaces.R' 'health_operations.R'
'licensemanager_service.R' 'licensemanager_interfaces.R'
'licensemanager_operations.R'
'licensemanagerlinuxsubscriptions_service.R'
'licensemanagerlinuxsubscriptions_interfaces.R'
'licensemanagerlinuxsubscriptions_operations.R'
'licensemanagerusersubscriptions_service.R'
'licensemanagerusersubscriptions_interfaces.R'
'licensemanagerusersubscriptions_operations.R'
'managedgrafana_service.R' 'managedgrafana_interfaces.R'
'managedgrafana_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'
'prometheusservice_service.R' 'prometheusservice_interfaces.R'
'prometheusservice_operations.R' 'reexports_paws.common.R'
'resiliencehub_service.R' 'resiliencehub_interfaces.R'
'resiliencehub_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'

'ssmcontacts_service.R' 'ssmcontacts_interfaces.R'
'ssmcontacts_operations.R' 'ssmincidents_service.R'
'ssmincidents_interfaces.R' 'ssmincidents_operations.R'
'ssmsap_service.R' 'ssmsap_interfaces.R' 'ssmsap_operations.R'
'support_service.R' 'support_interfaces.R'
'support_operations.R' 'supportapp_service.R'
'supportapp_interfaces.R' 'supportapp_operations.R'
'synthetics_service.R' 'synthetics_interfaces.R'
'synthetics_operations.R'

NeedsCompilation no

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

Maintainer Dyfan Jones <dyfan.r.jones@gmail.com>

Repository CRAN

Date/Publication 2024-05-16 08:20:03 UTC

R topics documented:

`applicationautoscaling`

Application Auto Scaling

Description

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

- Amazon AppStream 2.0 fleets
- Amazon Aurora Replicas
- Amazon Comprehend document classification and entity recognizer endpoints
- Amazon DynamoDB tables and global secondary indexes throughput capacity
- Amazon ECS services
- Amazon ElastiCache for Redis clusters (replication groups)
- Amazon EMR clusters
- Amazon Keyspaces (for Apache Cassandra) tables
- Lambda function provisioned concurrency
- Amazon Managed Streaming for Apache Kafka broker storage
- Amazon Neptune clusters
- Amazon SageMaker endpoint variants
- Amazon SageMaker Serverless endpoint provisioned concurrency
- Amazon SageMaker inference components
- Spot Fleets (Amazon EC2)
- Custom resources provided by your own applications or services

To learn more about Application Auto Scaling, see the Application Auto Scaling User Guide.

API Summary

The Application Auto Scaling service API includes three key sets of actions:

- Register and manage scalable targets - Register Amazon Web Services 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.

Usage

```
applicationautoscaling(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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
list_tags_for_resource	Returns all the tags on the specified Application Auto Scaling scalable target
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, which is the resource that you want to scale
tag_resource	Adds or edits tags on an Application Auto Scaling scalable target
untag_resource	Deletes tags from an Application Auto Scaling 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)
```

---

applicationcostprofiler

AWS Application Cost Profiler

Description

This reference provides descriptions of the AWS Application Cost Profiler API.

The AWS Application Cost Profiler API provides programmatic access to view, create, update, and delete application cost report definitions, as well as to import your usage data into the Application Cost Profiler service.

For more information about using this service, see the AWS Application Cost Profiler User Guide.

Usage

```
applicationcostprofiler(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- applicationcostprofiler(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

delete_report_definition	Deletes the specified report definition in AWS Application Cost Profiler
get_report_definition	Retrieves the definition of a report already configured in AWS Application Cost Profiler
import_application_usage	Ingests application usage data from Amazon Simple Storage Service (Amazon S3)
list_report_definitions	Retrieves a list of all reports and their configurations for your AWS account
put_report_definition	Creates the report definition for a report in Application Cost Profiler
update_report_definition	Updates existing report in AWS Application Cost Profiler

Examples

```
## Not run:
svc <- applicationcostprofiler()
svc$delete_report_definition(
```

```
  Foo = 123
)

## 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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.

- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
```

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

add_workload	Adds a workload to a component
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
describe_workload	Describes a workload and its configuration
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
list_workloads	Lists the workloads that are configured on a given component
remove_workload	Remove workload from a component
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
update_problem	Updates the visibility of the problem or specifies the problem as RESO
update_workload	Adds a workload to a component

Examples

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

## End(Not run)
```

`appregistry` *AWS Service Catalog App Registry*

Description

Amazon Web Services Service Catalog AppRegistry enables organizations to understand the application context of their Amazon Web Services resources. AppRegistry provides a repository of your applications, their resources, and the application metadata that you use within your enterprise.

Usage

```
appregistry(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- appregistry(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
```

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

Operations

associate_attribute_group	Associates an attribute group with an application to augment the application's metadat
associate_resource	Associates a resource with an application
create_application	Creates a new application that is the top-level node in a hierarchy of related cloud reso
create_attribute_group	Creates a new attribute group as a container for user-defined attributes
delete_application	Deletes an application that is specified either by its application ID, name, or ARN
delete_attribute_group	Deletes an attribute group, specified either by its attribute group ID, name, or ARN
disassociate_attribute_group	Disassociates an attribute group from an application to remove the extra attributes con
disassociate_resource	Disassociates a resource from application
get_application	Retrieves metadata information about one of your applications
get_associated_resource	Gets the resource associated with the application
get_attribute_group	Retrieves an attribute group by its ARN, ID, or name
get_configuration	Retrieves a TagKey configuration from an account
list_applications	Retrieves a list of all of your applications
list_associated_attribute_groups	Lists all attribute groups that are associated with specified application
list_associated_resources	Lists all of the resources that are associated with the specified application
list_attribute_groups	Lists all attribute groups which you have access to
list_attribute_groups_for_application	Lists the details of all attribute groups associated with a specific application
list_tags_for_resource	Lists all of the tags on the resource
put_configuration	Associates a TagKey configuration to an account
sync_resource	Syncs the resource with current AppRegistry records
tag_resource	Assigns one or more tags (key-value pairs) to the specified resource
untag_resource	Removes tags from a resource
update_application	Updates an existing application with new attributes
update_attribute_group	Updates an existing attribute group with new details

Examples

```
## Not run:
svc <- appregistry()
svc$associate_attribute_group(
  Foo = 123
)

## End(Not run)
```

`auditmanager` *AWS Audit Manager*

Description

Welcome to the Audit Manager API reference. This guide is for developers who need detailed information about the Audit Manager API operations, data types, and errors.

Audit Manager is a service that provides automated evidence collection so that you can continually audit your Amazon Web Services usage. You can use it to assess the effectiveness of your controls, manage risk, and simplify compliance.

Audit Manager provides prebuilt frameworks that structure and automate assessments for a given compliance standard. Frameworks include a prebuilt collection of controls with descriptions and testing procedures. These controls are grouped according to the requirements of the specified compliance standard or regulation. You can also customize frameworks and controls to support internal audits with specific requirements.

Use the following links to get started with the Audit Manager API:

- Actions: An alphabetical list of all Audit Manager API operations.
- Data types: An alphabetical list of all Audit Manager data types.
- Common parameters: Parameters that all operations can use.
- Common errors: Client and server errors that all operations can return.

If you're new to Audit Manager, we recommend that you review the Audit Manager User Guide.

Usage

```
auditmanager(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.

- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- auditmanager(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
```

```
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_assessment_report_evidence_folder	Associates an evidence folder to an assessment report in an Audit Ma
batch_associate_assessment_report_evidence	Associates a list of evidence to an assessment report in an Audit Mana
batch_create_delegation_by_assessment	Creates a batch of delegations for an assessment in Audit Manager
batch_delete_delegation_by_assessment	Deletes a batch of delegations for an assessment in Audit Manager
batch_disassociate_assessment_report_evidence	Disassociates a list of evidence from an assessment report in Audit M
batch_import_evidence_to_assessment_control	Adds one or more pieces of evidence to a control in an Audit Manage
create_assessment	Creates an assessment in Audit Manager
create_assessment_framework	Creates a custom framework in Audit Manager
create_assessment_report	Creates an assessment report for the specified assessment
create_control	Creates a new custom control in Audit Manager
delete_assessment	Deletes an assessment in Audit Manager
delete_assessment_framework	Deletes a custom framework in Audit Manager
delete_assessment_framework_share	Deletes a share request for a custom framework in Audit Manager
delete_assessment_report	Deletes an assessment report in Audit Manager
delete_control	Deletes a custom control in Audit Manager
deregister_account	Deregisters an account in Audit Manager
deregister_organization_admin_account	Removes the specified Amazon Web Services account as a delegated a
disassociate_assessment_report_evidence_folder	Disassociates an evidence folder from the specified assessment report
get_account_status	Gets the registration status of an account in Audit Manager
get_assessment	Gets information about a specified assessment
get_assessment_framework	Gets information about a specified framework
get_assessment_report_url	Gets the URL of an assessment report in Audit Manager
get_change_logs	Gets a list of changelogs from Audit Manager
get_control	Gets information about a specified control
get_delegations	Gets a list of delegations from an audit owner to a delegate
get_evidence	Gets information about a specified evidence item
get_evidence_by_evidence_folder	Gets all evidence from a specified evidence folder in Audit Manager
get_evidence_file_upload_url	Creates a presigned Amazon S3 URL that can be used to upload a file
get_evidence_folder	Gets an evidence folder from a specified assessment in Audit Manage
get_evidence_folders_by_assessment	Gets the evidence folders from a specified assessment in Audit Manag
get_evidence_folders_by_assessment_control	Gets a list of evidence folders that are associated with a specified cont
get_insights	Gets the latest analytics data for all your current active assessments

get_insights_by_assessment	Gets the latest analytics data for a specific active assessment
get_organization_admin_account	Gets the name of the delegated Amazon Web Services administrator a
get_services_in_scope	Gets a list of all of the Amazon Web Services that you can choose to i
get_settings	Gets the settings for a specified Amazon Web Services account
list_assessment_control_insights_by_control_domain	Lists the latest analytics data for controls within a specific control dor
list_assessment_frameworks	Returns a list of the frameworks that are available in the Audit Manag
list_assessment_framework_share_requests	Returns a list of sent or received share requests for custom framework
list_assessment_reports	Returns a list of assessment reports created in Audit Manager
list_assessments	Returns a list of current and past assessments from Audit Manager
list_control_domain_insights	Lists the latest analytics data for control domains across all of your ac
list_control_domain_insights_by_assessment	Lists analytics data for control domains within a specified active asses
list_control_insights_by_control_domain	Lists the latest analytics data for controls within a specific control dor
list_controls	Returns a list of controls from Audit Manager
list_keywords_for_data_source	Returns a list of keywords that are pre-mapped to the specified contro
list_notifications	Returns a list of all Audit Manager notifications
list_tags_for_resource	Returns a list of tags for the specified resource in Audit Manager
register_account	Enables Audit Manager for the specified Amazon Web Services accou
register_organization_admin_account	Enables an Amazon Web Services account within the organization as
start_assessment_framework_share	Creates a share request for a custom framework in Audit Manager
tag_resource	Tags the specified resource in Audit Manager
untag_resource	Removes a tag from a resource in Audit Manager
update_assessment	Edits an Audit Manager assessment
update_assessment_control	Updates a control within an assessment in Audit Manager
update_assessment_control_set_status	Updates the status of a control set in an Audit Manager assessment
update_assessment_framework	Updates a custom framework in Audit Manager
update_assessment_framework_share	Updates a share request for a custom framework in Audit Manager
update_assessment_status	Updates the status of an assessment in Audit Manager
update_control	Updates a custom control in Audit Manager
update_settings	Updates Audit Manager settings for the current account
validate_assessment_report_integrity	Validates the integrity of an assessment report in Audit Manager

Examples

```
## Not run:
svc <- auditmanager()
svc$associate_assessment_report_evidence_folder(
  Foo = 123
)

## End(Not run)
```

`autoscaling` *Auto Scaling*

Description

Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling is designed to automatically launch and terminate EC2 instances based on user-defined scaling policies, scheduled actions, and health checks.

For more information, see the Amazon EC2 Auto Scaling User Guide and the Amazon EC2 Auto Scaling API Reference.

Usage

```
autoscaling(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html`

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.

- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

attach_instances	Attaches one or more EC2 instances to the specified Auto Scaling group
attach_load_balancers	This API operation is superseded by AttachTrafficSources, which can attach mu
attach_load_balancer_target_groups	This API operation is superseded by AttachTrafficSources, which can attach mu

attach_traffic_sources	Attaches one or more traffic sources 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 or rollback that is 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
delete_warm_pool	Deletes the warm pool for the specified Auto Scaling group
describe_account_limits	Describes the current Amazon EC2 Auto Scaling resource quotas for your accou
describe_adjustment_types	Describes the available adjustment types for step scaling and simple scaling poli
describe_auto_scaling_groups	Gets information about the Auto Scaling groups in the account and Region
describe_auto_scaling_instances	Gets information about the Auto Scaling instances in the account and Region
describe_auto_scaling_notification_types	Describes the notification types that are supported by Amazon EC2 Auto Scalin
describe_instance_refreshes	Gets information about the instance refreshes for the specified Auto Scaling gro
describe_launch_configurations	Gets information about the launch configurations in the account and Region
describe_lifecycle_hooks	Gets information about the lifecycle hooks for the specified Auto Scaling group
describe_lifecycle_hook_types	Describes the available types of lifecycle hooks
describe_load_balancers	This API operation is superseded by DescribeTrafficSources, which can describ
describe_load_balancer_target_groups	This API operation is superseded by DescribeTrafficSources, which can describ
describe_metric_collection_types	Describes the available CloudWatch metrics for Amazon EC2 Auto Scaling
describe_notification_configurations	Gets information about the Amazon SNS notifications that are configured for on
describe_policies	Gets information about the scaling policies in the account and Region
describe_scaling_activities	Gets information about the scaling activities in the account and Region
describe_scaling_process_types	Describes the scaling process types for use with the ResumeProcesses and Susp
describe_scheduled_actions	Gets information about the scheduled actions that haven't run or that have not re
describe_tags	Describes the specified tags
describe_termination_policy_types	Describes the termination policies supported by Amazon EC2 Auto Scaling
describe_traffic_sources	Gets information about the traffic sources for the specified Auto Scaling group
describe_warm_pool	Gets information about a warm pool and its instances
detach_instances	Removes one or more instances from the specified Auto Scaling group
detach_load_balancers	This API operation is superseded by DetachTrafficSources, which can detach m
detach_load_balancer_target_groups	This API operation is superseded by DetachTrafficSources, which can detach m
detach_traffic_sources	Detaches one or more traffic sources from the specified Auto Scaling group
disable_metrics_collection	Disables group metrics collection for the specified Auto Scaling group
enable_metrics_collection	Enables group metrics collection 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
get_predictive_scaling_forecast	Retrieves the forecast data for a predictive scaling policy
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
put_warm_pool	Creates or updates a warm pool for the specified 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
rollback_instance_refresh	Cancels an instance refresh that is in progress and rolls back any changes that it
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 an instance refresh
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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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

CloudFormation

CloudFormation allows you to create and manage Amazon Web Services infrastructure deployments predictably and repeatedly. You can use CloudFormation to leverage Amazon Web Services 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 Amazon Web Services infrastructure.

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

For more information about CloudFormation, see the CloudFormation product page.

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

Usage

```
cloudformation(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

activate_organizations_access	Activate trusted access with Organizations
activate_type	Activates a public third-party extension, making it available for use in stack templat
batch_describe_type_configurations	Returns configuration data for the specified CloudFormation extensions, from the C
cancel_update_stack	Cancels an update on the specified stack
continue_update_rollback	For a specified stack that's in the UPDATE_ROLLBACK_FAILED state, continues
create_change_set	Creates a list of changes that will be applied to a stack so that you can review the ch
create_generated_template	Creates a template from existing resources that are not already managed with Cloud
create_stack	Creates a stack as specified in the template
create_stack_instances	Creates stack instances for the specified accounts, within the specified Amazon Wel
create_stack_set	Creates a stack set
deactivate_organizations_access	Deactivates trusted access with Organizations
deactivate_type	Deactivates a public extension that was previously activated in this account and Reg
delete_change_set	Deletes the specified change set
delete_generated_template	Deleted a generated template
delete_stack	Deletes a specified stack
delete_stack_instances	Deletes stack instances for the specified accounts, in the specified Amazon Web Ser
delete_stack_set	Deletes a stack set
deregister_type	Marks an extension or extension version as DEPRECATED in the CloudFormation
describe_account_limits	Retrieves your account's CloudFormation limits, such as the maximum number of s
describe_change_set	Returns the inputs for the change set and a list of changes that CloudFormation will

describe_change_set_hooks	Returns hook-related information for the change set and a list of changes that Cloud
describe_generated_template	Describes a generated template
describe_organizations_access	Retrieves information about the account's OrganizationAccess status
describe_publisher	Returns information about a CloudFormation extension publisher
describe_resource_scan	Describes details of a resource scan
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 StackSet, Amazon W
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 sp
describe_stack_resources	Returns Amazon Web Services resource descriptions for running and deleted stacks
describe_stacks	Returns the description for the specified stack; if no stack name was specified, then
describe_stack_set	Returns the description of the specified StackSet
describe_stack_set_operation	Returns the description of the specified StackSet operation
describe_type	Returns detailed information about an extension that has been registered
describe_type_registration	Returns information about an extension's registration, including its current status ar
detect_stack_drift	Detects whether a stack's actual configuration differs, or has drifted, from its expect
detect_stack_resource_drift	Returns information about whether a resource's actual configuration differs, or has
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 cl
get_generated_template	Retrieves a generated template
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
import_stacks_to_stack_set	Import existing stacks into a new stack sets
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 act
list_generated_templates	Lists your generated templates in this Region
list_imports	Lists all stacks that are importing an exported output value
list_resource_scan_related_resources	Lists the related resources for a list of resources from a resource scan
list_resource_scan_resources	Lists the resources from a resource scan
list_resource_scans	List the resource scans from newest to oldest
list_stack_instance_resource_drifts	Returns drift information for resources in a stack instance
list_stack_instances	Returns summary information about stack instances that are associated with the spe
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 Sta
list_stack_set_auto_deployment_targets	Returns summary information about deployment targets for a stack set
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 extension(s)
list_types	Returns summary information about extension that have been registered with Cloud
list_type_versions	Returns summary information about the versions of an extension
publish_type	Publishes the specified extension to the CloudFormation registry as a public extensi
record_handler_progress	Reports progress of a resource handler to CloudFormation
register_publisher	Registers your account as a publisher of public extensions in the CloudFormation re
register_type	Registers an extension with the CloudFormation service

rollback_stack	When specifying RollbackStack, you preserve the state of previously provisioned re
set_stack_policy	Sets a stack policy for a specified stack
set_type_configuration	Specifies the configuration data for a registered CloudFormation extension, in the g
set_type_default_version	Specify the default version of an extension
signal_resource	Sends a signal to the specified resource with a success or failure status
start_resource_scan	Starts a scan of the resources in this account in this Region
stop_stack_set_operation	Stops an in-progress operation on a stack set and its associated stack instances
test_type	Tests a registered extension to make sure it meets all necessary requirements for bei
update_generated_template	Updates a generated template
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 t
update_stack_set	Updates the stack set, and associated stack instances in the specified accounts and A
update_termination_protection	Updates termination protection for the specified stack
validate_template	Validates a specified template

Examples

```
## Not run:
svc <- cloudformation()
# This example creates a generated template with a resources file.
svc$create_generated_template(
  GeneratedTemplateName = "JazzyTemplate",
  Resources = list(
    list(
      ResourceIdentifier = list(
        BucketName = "jazz-bucket"
      ),
      ResourceType = "AWS::S3::Bucket"
    ),
    list(
      ResourceIdentifier = list(
        DhcpOptionsId = "random-id123"
      ),
      ResourceType = "AWS::EC2::DHCPOptions"
    )
  )
)

## End(Not run)
```

`cloudtrail` *AWS CloudTrail*

Description

CloudTrail

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 Amazon Web Services API calls for your Amazon Web Services account and delivers log files to an Amazon S3 bucket. The recorded information includes the identity of the user, the start time of the Amazon Web Services 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 Amazon Web Services SDKs, which consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .NET, iOS, Android, etc.). The SDKs provide programmatic access to CloudTrail. For example, the SDKs handle cryptographically signing requests, managing errors, and retrying requests automatically. For more information about the Amazon Web Services SDKs, including how to download and install them, see Tools to Build on Amazon Web Services.

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

Usage

```
cloudtrail(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
```

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

Operations

add_tags	Adds one or more tags to a trail, event data store, or channel, up to a limit of 50
cancel_query	Cancels a query if the query is not in a terminated state, such as CANCELLED, F
create_channel	Creates a channel for CloudTrail to ingest events from a partner or external source
create_event_data_store	Creates a new event data store
create_trail	Creates a trail that specifies the settings for delivery of log data to an Amazon S3
delete_channel	Deletes a channel
delete_event_data_store	Disables the event data store specified by EventDataStore, which accepts an event
delete_resource_policy	Deletes the resource-based policy attached to the CloudTrail channel
delete_trail	Deletes a trail
deregister_organization_delegated_admin	Removes CloudTrail delegated administrator permissions from a member account
describe_query	Returns metadata about a query, including query run time in milliseconds, numbe
describe_trails	Retrieves settings for one or more trails associated with the current Region for yo
disable_federation	Disables Lake query federation on the specified event data store
enable_federation	Enables Lake query federation on the specified event data store
get_channel	Returns information about a specific channel
get_event_data_store	Returns information about an event data store specified as either an ARN or the II
get_event_selectors	Describes the settings for the event selectors that you configured for your trail
get_import	Returns information about a specific import
get_insight_selectors	Describes the settings for the Insights event selectors that you configured for your
get_query_results	Gets event data results of a query
get_resource_policy	Retrieves the JSON text of the resource-based policy document attached to the Cl
get_trail	Returns settings information for a specified trail
get_trail_status	Returns a JSON-formatted list of information about the specified trail
list_channels	Lists the channels in the current account, and their source names
list_event_data_stores	Returns information about all event data stores in the account, in the current Regi
list_import_failures	Returns a list of failures for the specified import
list_imports	Returns information on all imports, or a select set of imports by ImportStatus or D
list_insights_metric_data	Returns Insights metrics data for trails that have enabled Insights
list_public_keys	Returns all public keys whose private keys were used to sign the digest files withi
list_queries	Returns a list of queries and query statuses for the past seven days
list_tags	Lists the tags for the specified trails, event data stores, or channels in the current R
list_trails	Lists trails that are in the current account
lookup_events	Looks up management events or CloudTrail Insights events that are captured by C
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 yo
put_resource_policy	Attaches a resource-based permission policy to a CloudTrail channel that is used
register_organization_delegated_admin	Registers an organization's member account as the CloudTrail delegated administ
remove_tags	Removes the specified tags from a trail, event data store, or channel
restore_event_data_store	Restores a deleted event data store specified by EventDataStore, which accepts an
start_event_data_store_ingestion	Starts the ingestion of live events on an event data store specified as either an ARN

start_import	Starts an import of logged trail events from a source S3 bucket to a destination ev
start_logging	Starts the recording of Amazon Web Services API calls and log file delivery for a
start_query	Starts a CloudTrail Lake query
stop_event_data_store_ingestion	Stops the ingestion of live events on an event data store specified as either an ARN
stop_import	Stops a specified import
stop_logging	Suspends the recording of Amazon Web Services API calls and log file delivery f
update_channel	Updates a channel specified by a required channel ARN or UUID
update_event_data_store	Updates an event data store
update_trail	Updates trail settings that control what events you are logging, and how to handle

Examples

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

## End(Not run)
```

`cloudtraildataservice` *AWS CloudTrail Data Service*

Description

The CloudTrail Data Service lets you ingest events into CloudTrail from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers. You can store, access, analyze, troubleshoot and take action on this data without maintaining multiple log aggregators and reporting tools. After you run `put_audit_events` to ingest your application activity into CloudTrail, you can use CloudTrail Lake to search, query, and analyze the data that is logged from your applications.

Usage

```
cloudtraildataservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

put_audit_events	Ingests your application events into CloudTrail Lake

Examples

```
## Not run:
svc <- cloudtraildataservice()
svc$put_audit_events(
  Foo = 123
)

## End(Not run)
```

---

cloudwatch *Amazon CloudWatch*

Description

Amazon CloudWatch monitors your Amazon Web Services (Amazon Web Services) resources and the applications you run on Amazon Web Services 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 Amazon Web Services, 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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:

  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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
delete_metric_stream	Permanently deletes the metric stream that you specify
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 CloudWatch metric values
get_metric_statistics	Gets statistics for the specified metric
get_metric_stream	Returns information about the metric stream that you specify
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_managed_insight_rules	Returns a list that contains the number of managed Contributor Insights rules in your account
list_metrics	List the specified metrics
list_metric_streams	Returns a list of metric streams in this account
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_managed_insight_rules	Creates a managed Contributor Insights rule for a specified Amazon Web Services resource
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
put_metric_stream	Creates or updates a metric stream
set_alarm_state	Temporarily sets the state of an alarm for testing purposes
start_metric_streams	Starts the streaming of metrics for one or more of your metric streams
stop_metric_streams	Stops the streaming of metrics for one or more of your metric streams
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 Amazon Web Services resources. When your resources change state, they automatically send events to 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 Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the running state.
- Direct specific API records from 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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.

- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
```

```
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
  )
```

Operations

activate_event_source	Activates a partner event source that has been deactivated
cancel_replay	Cancels the specified replay
create_api_destination	Creates an API destination, which is an HTTP invocation endpoint configured as a targe
create_archive	Creates an archive of events with the specified settings
create_connection	Creates a connection
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
deauthorize_connection	Removes all authorization parameters from the connection
delete_api_destination	Deletes the specified API destination
delete_archive	Deletes the specified archive
delete_connection	Deletes a connection
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_api_destination	Retrieves details about an API destination
describe_archive	Retrieves details about an archive
describe_connection	Retrieves details about a connection
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_api_destinations	Retrieves a list of API destination in the account in the current Region
list_archives	Lists your archives
list_connections	Retrieves a list of connections from the account
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 Am
list_partner_event_source_accounts	An SaaS partner can use this operation to display the Amazon Web Services account ID
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 Amazon Web Services account or Amazo
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 Amazon Web Services account to be able to put ever
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_api_destination	Updates an API destination
update_archive	Updates the specified archive
update_connection	Updates settings for a connection

Examples

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

## End(Not run)
```

`cloudwatchevidently` *Amazon CloudWatch Evidently*

Description

You can use Amazon CloudWatch Evidently to safely validate new features by serving them to a specified percentage of your users while you roll out the feature. You can monitor the performance of the new feature to help you decide when to ramp up traffic to your users. This helps you reduce risk and identify unintended consequences before you fully launch the feature.

You can also conduct A/B experiments to make feature design decisions based on evidence and data. An experiment can test as many as five variations at once. Evidently collects experiment data and analyzes it using statistical methods. It also provides clear recommendations about which variations perform better. You can test both user-facing features and backend features.

Usage

```
cloudwatchevidently(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- cloudwatchevidently(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_evaluate_feature	This operation assigns feature variation to user sessions
create_experiment	Creates an Evidently experiment
create_feature	Creates an Evidently feature that you want to launch or test
create_launch	Creates a launch of a given feature
create_project	Creates a project, which is the logical object in Evidently that can contain features, launches, a
create_segment	Use this operation to define a segment of your audience
delete_experiment	Deletes an Evidently experiment
delete_feature	Deletes an Evidently feature
delete_launch	Deletes an Evidently launch
delete_project	Deletes an Evidently project
delete_segment	Deletes a segment
evaluate_feature	This operation assigns a feature variation to one given user session
get_experiment	Returns the details about one experiment
get_experiment_results	Retrieves the results of a running or completed experiment

get_feature	Returns the details about one feature
get_launch	Returns the details about one launch
get_project	Returns the details about one launch
get_segment	Returns information about the specified segment
list_experiments	Returns configuration details about all the experiments in the specified project
list_features	Returns configuration details about all the features in the specified project
list_launches	Returns configuration details about all the launches in the specified project
list_projects	Returns configuration details about all the projects in the current Region in your account
list_segment_references	Use this operation to find which experiments or launches are using a specified segment
list_segments	Returns a list of audience segments that you have created in your account in this Region
list_tags_for_resource	Displays the tags associated with an Evidently resource
put_project_events	Sends performance events to Evidently
start_experiment	Starts an existing experiment
start_launch	Starts an existing launch
stop_experiment	Stops an experiment that is currently running
stop_launch	Stops a launch that is currently running
tag_resource	Assigns one or more tags (key-value pairs) to the specified CloudWatch Evidently resource
test_segment_pattern	Use this operation to test a rules pattern that you plan to use to create an audience segment
untag_resource	Removes one or more tags from the specified resource
update_experiment	Updates an Evidently experiment
update_feature	Updates an existing feature
update_launch	Updates a launch of a given feature
update_project	Updates the description of an existing project
update_project_data_delivery	Updates the data storage options for this project

Examples

```
## Not run:
svc <- cloudwatchevidently()
svc$batch_evaluate_feature(
  Foo = 123
)

## End(Not run)
```

`cloudwatchinternetmonitor`

Amazon CloudWatch Internet Monitor

Description

Amazon CloudWatch Internet Monitor provides visibility into how internet issues impact the performance and availability between your applications hosted on Amazon Web Services and your end users. It can reduce the time it takes for you to diagnose internet issues from days to minutes.

Internet Monitor uses the connectivity data that Amazon Web Services captures from its global networking footprint to calculate a baseline of performance and availability for internet traffic. This is the same data that Amazon Web Services uses to monitor internet uptime and availability. With those measurements as a baseline, Internet Monitor raises awareness for you when there are significant problems for your end users in the different geographic locations where your application runs.

Internet Monitor publishes internet measurements to CloudWatch Logs and CloudWatch Metrics, to easily support using CloudWatch tools with health information for geographies and networks specific to your application. Internet Monitor sends health events to Amazon EventBridge so that you can set up notifications. If an issue is caused by the Amazon Web Services network, you also automatically receive an Amazon Web Services Health Dashboard notification with the steps that Amazon Web Services is taking to mitigate the problem.

To use Internet Monitor, you create a *monitor* and associate your application's resources with it - VPCs, NLBs, CloudFront distributions, or WorkSpaces directories - so Internet Monitor can determine where your application's internet traffic is. Internet Monitor then provides internet measurements from Amazon Web Services that are specific to the locations and ASNs (typically, internet service providers or ISPs) that communicate with your application.

For more information, see Using Amazon CloudWatch Internet Monitor in the *Amazon CloudWatch User Guide*.

Usage

```
cloudwatchinternetmonitor(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- cloudwatchinternetmonitor(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
```

```
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_monitor	Creates a monitor in Amazon CloudWatch Internet Monitor
delete_monitor	Deletes a monitor in Amazon CloudWatch Internet Monitor
get_health_event	Gets information that Amazon CloudWatch Internet Monitor has created and stored about a health ev
get_internet_event	Gets information that Amazon CloudWatch Internet Monitor has generated about an internet event
get_monitor	Gets information about a monitor in Amazon CloudWatch Internet Monitor based on a monitor name
get_query_results	Return the data for a query with the Amazon CloudWatch Internet Monitor query interface
get_query_status	Returns the current status of a query for the Amazon CloudWatch Internet Monitor query interface, fo
list_health_events	Lists all health events for a monitor in Amazon CloudWatch Internet Monitor
list_internet_events	Lists internet events that cause performance or availability issues for client locations
list_monitors	Lists all of your monitors for Amazon CloudWatch Internet Monitor and their statuses, along with the
list_tags_for_resource	Lists the tags for a resource
start_query	Start a query to return data for a specific query type for the Amazon CloudWatch Internet Monitor qu
stop_query	Stop a query that is progress for a specific monitor
tag_resource	Adds a tag to a resource
untag_resource	Removes a tag from a resource
update_monitor	Updates a monitor

Examples

```
## Not run:
svc <- cloudwatchinternetmonitor()
svc$create_monitor(
  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, CloudTrail, and other sources. You can then retrieve the associated log data from CloudWatch Logs using the CloudWatch console. Alternatively, you can use CloudWatch Logs commands in the Amazon Web Services 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. Then, it can 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"). You can also 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 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 earlier than this setting are automatically deleted. The CloudWatch Logs agent helps 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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.

- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
```

```
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_kms_key	Associates the specified KMS key with either one log group in the account, or with all st
cancel_export_task	Cancels the specified export task
create_delivery	Creates a delivery
create_export_task	Creates an export task so that you can efficiently export data from a log group to an Ama
create_log_anomaly_detector	Creates an anomaly detector that regularly scans one or more log groups and look for pa
create_log_group	Creates a log group with the specified name
create_log_stream	Creates a log stream for the specified log group
delete_account_policy	Deletes a CloudWatch Logs account policy
delete_data_protection_policy	Deletes the data protection policy from the specified log group
delete_delivery	Deletes s delivery
delete_delivery_destination	Deletes a delivery destination
delete_delivery_destination_policy	Deletes a delivery destination policy
delete_delivery_source	Deletes a delivery source
delete_destination	Deletes the specified destination, and eventually disables all the subscription filters that p
delete_log_anomaly_detector	Deletes the specified CloudWatch Logs anomaly detector
delete_log_group	Deletes the specified log group and permanently deletes all the archived log events assoc
delete_log_stream	Deletes the specified log stream and permanently deletes all the archived log events asso
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_account_policies	Returns a list of all CloudWatch Logs account policies in the account
describe_deliveries	Retrieves a list of the deliveries that have been created in the account
describe_delivery_destinations	Retrieves a list of the delivery destinations that have been created in the account
describe_delivery_sources	Retrieves a list of the delivery sources that have been created in the account
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, running, or have l
describe_query_definitions	This operation returns a paginated list of your saved CloudWatch Logs Insights query de

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 specified KMS key from the specified log group or from all CloudWatc
filter_log_events	Lists log events from the specified log group
get_data_protection_policy	Returns information about a log group data protection policy
get_delivery	Returns complete information about one logical delivery
get_delivery_destination	Retrieves complete information about one delivery destination
get_delivery_destination_policy	Retrieves the delivery destination policy assigned to the delivery destination that you spe
get_delivery_source	Retrieves complete information about one delivery source
get_log_anomaly_detector	Retrieves information about the log anomaly detector that you specify
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
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_anomalies	Returns a list of anomalies that log anomaly detectors have found
list_log_anomaly_detectors	Retrieves a list of the log anomaly detectors in the account
list_tags_for_resource	Displays the tags associated with a CloudWatch Logs resource
list_tags_log_group	The ListTagsLogGroup operation is on the path to deprecation
put_account_policy	Creates an account-level data protection policy or subscription filter policy that applies t
put_data_protection_policy	Creates a data protection policy for the specified log group
put_delivery_destination	Creates or updates a logical delivery destination
put_delivery_destination_policy	Creates and assigns an IAM policy that grants permissions to CloudWatch Logs to deliv
put_delivery_source	Creates or updates a logical delivery source
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 Amazon Web Services services to pu
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_live_tail	Starts a Live Tail streaming session for one or more log groups
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	The TagLogGroup operation is on the path to deprecation
tag_resource	Assigns one or more tags (key-value pairs) to the specified CloudWatch Logs resource
test_metric_filter	Tests the filter pattern of a metric filter against a sample of log event messages
untag_log_group	The UntagLogGroup operation is on the path to deprecation
untag_resource	Removes one or more tags from the specified resource
update_anomaly	Use this operation to suppress anomaly detection for a specified anomaly or pattern
update_log_anomaly_detector	Updates an existing log anomaly detector

Examples

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

```
  Foo = 123
)

## End(Not run)
```

`cloudwatchobservabilityaccessmanager` *CloudWatch Observability Access Manager*

Description

Use Amazon CloudWatch Observability Access Manager to create and manage links between source accounts and monitoring accounts by using *CloudWatch cross-account observability*. With CloudWatch cross-account observability, you can monitor and troubleshoot applications that span multiple accounts within a Region. Seamlessly search, visualize, and analyze your metrics, logs, traces, and Application Insights applications in any of the linked accounts without account boundaries.

Set up one or more Amazon Web Services accounts as *monitoring accounts* and link them with multiple *source accounts*. A monitoring account is a central Amazon Web Services account that can view and interact with observability data generated from source accounts. A source account is an individual Amazon Web Services account that generates observability data for the resources that reside in it. Source accounts share their observability data with the monitoring account. The shared observability data can include metrics in Amazon CloudWatch, logs in Amazon CloudWatch Logs, traces in X-Ray, and applications in Amazon CloudWatch Application Insights.

Usage

```
cloudwatchobservabilityaccessmanager(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.

- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- cloudwatchobservabilityaccessmanager(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
```

```
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_link	Creates a link between a source account and a sink that you have created in a monitoring account
create_sink	Use this to create a sink in the current account, so that it can be used as a monitoring account in Clou
delete_link	Deletes a link between a monitoring account sink and a source account
delete_sink	Deletes a sink
get_link	Returns complete information about one link
get_sink	Returns complete information about one monitoring account sink
get_sink_policy	Returns the current sink policy attached to this sink
list_attached_links	Returns a list of source account links that are linked to this monitoring account sink
list_links	Use this operation in a source account to return a list of links to monitoring account sinks that this so
list_sinks	Use this operation in a monitoring account to return the list of sinks created in that account
list_tags_for_resource	Displays the tags associated with a resource
put_sink_policy	Creates or updates the resource policy that grants permissions to source accounts to link to the monito
tag_resource	Assigns one or more tags (key-value pairs) to the specified resource
untag_resource	Removes one or more tags from the specified resource
update_link	Use this operation to change what types of data are shared from a source account to its linked monito

Examples

```
## Not run:
svc <- cloudwatchobservabilityaccessmanager()
svc$create_link(
  Foo = 123
)

## End(Not run)
```

`cloudwatchrum` *CloudWatch RUM*

Description

With Amazon CloudWatch RUM, you can perform real-user monitoring to collect client-side data about your web application performance from actual user sessions in real time. The data collected includes page load times, client-side errors, and user behavior. When you view this data, you can see it all aggregated together and also see breakdowns by the browsers and devices that your customers use.

You can use the collected data to quickly identify and debug client-side performance issues. CloudWatch RUM helps you visualize anomalies in your application performance and find relevant debugging data such as error messages, stack traces, and user sessions. You can also use RUM to understand the range of end-user impact including the number of users, geolocations, and browsers used.

Usage

```
cloudwatchrum(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- cloudwatchrum(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
```

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

Operations

batch_create_rum_metric_definitions	Specifies the extended metrics and custom metrics that you want a CloudWatch RUM
batch_delete_rum_metric_definitions	Removes the specified metrics from being sent to an extended metrics destination
batch_get_rum_metric_definitions	Retrieves the list of metrics and dimensions that a RUM app monitor is sending to a si
create_app_monitor	Creates a Amazon CloudWatch RUM app monitor, which collects telemetry data from
delete_app_monitor	Deletes an existing app monitor
delete_rum_metrics_destination	Deletes a destination for CloudWatch RUM extended metrics, so that the specified app
get_app_monitor	Retrieves the complete configuration information for one app monitor
get_app_monitor_data	Retrieves the raw performance events that RUM has collected from your web applicat
list_app_monitors	Returns a list of the Amazon CloudWatch RUM app monitors in the account
list_rum_metrics_destinations	Returns a list of destinations that you have created to receive RUM extended metrics,
list_tags_for_resource	Displays the tags associated with a CloudWatch RUM resource
put_rum_events	Sends telemetry events about your application performance and user behavior to Clou
put_rum_metrics_destination	Creates or updates a destination to receive extended metrics from CloudWatch RUM
tag_resource	Assigns one or more tags (key-value pairs) to the specified CloudWatch RUM resourc
untag_resource	Removes one or more tags from the specified resource
update_app_monitor	Updates the configuration of an existing app monitor
update_rum_metric_definition	Modifies one existing metric definition for CloudWatch RUM extended metrics

Examples

```
## Not run:
svc <- cloudwatchrum()
svc$batch_create_rum_metric_definitions(
  Foo = 123
)

## End(Not run)
```

---

configservice *AWS Config*

Description

Config

Config provides a way to keep track of the configurations of all the Amazon Web Services resources associated with your Amazon Web Services account. You can use Config to get the current and

historical configurations of each Amazon Web Services resource and also to get information about the relationship between the resources. An Amazon Web Services 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 Config, see Supported Amazon Web Services resources.

You can access and manage Config through the Amazon Web Services Management Console, the Amazon Web Services Command Line Interface (Amazon Web Services CLI), the Config API, or the Amazon Web Services SDKs for Config. This reference guide contains documentation for the Config API and the Amazon Web Services CLI commands that you can use to manage Config. The 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 Config features and their associated actions or commands, as well as how to work with Amazon Web Services Management Console, see What Is Config in the *Config Developer Guide*.

Usage

```
configservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e`
  `html`

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_get_aggregate_resource_config	Returns the current configuration items for resources that are pres
batch_get_resource_config	Returns the BaseConfigurationItem for one or more requested res
delete_aggregation_authorization	Deletes the authorization granted to the specified configuration ag
delete_config_rule	Deletes the specified Config rule and all of its evaluation results
delete_configuration_aggregator	Deletes the specified configuration aggregator and the aggregated
delete_configuration_recorder	Deletes the configuration recorder
delete_conformance_pack	Deletes the specified conformance pack and all the Config rules, r
delete_delivery_channel	Deletes the delivery channel
delete_evaluation_results	Deletes the evaluation results for the specified Config rule
delete_organization_config_rule	Deletes the specified organization Config rule and all of its evalua
delete_organization_conformance_pack	Deletes the specified organization conformance pack and all of the
delete_pending_aggregation_request	Deletes pending authorization requests for a specified aggregator a
delete_remediation_configuration	Deletes the remediation configuration
delete_remediation_exceptions	Deletes one or more remediation exceptions mentioned in the resc
delete_resource_config	Records the configuration state for a custom resource that has bee
delete_retention_configuration	Deletes the retention configuration
delete_stored_query	Deletes the stored query for a single Amazon Web Services accou
deliver_config_snapshot	Schedules delivery of a configuration snapshot to the Amazon S3
describe_aggregate_compliance_by_config_rules	Returns a list of compliant and noncompliant rules with the numb
describe_aggregate_compliance_by_conformance_packs	Returns a list of the conformance packs and their associated comp
describe_aggregation_authorizations	Returns a list of authorizations granted to various aggregator acco
describe_compliance_by_config_rule	Indicates whether the specified Config rules are compliant
describe_compliance_by_resource	Indicates whether the specified Amazon Web Services resources a
describe_config_rule_evaluation_status	Returns status information for each of your Config managed rules
describe_config_rules	Returns details about your 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 a
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 organi
describe_organization_conformance_packs	Returns a list of organization conformance packs
describe_organization_conformance_pack_statuses	Provides organization conformance pack deployment status for an
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 r
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 Config rule for a sp
get_aggregate_config_rule_compliance_summary	Returns the number of compliant and noncompliant rules for one
get_aggregate_conformance_pack_compliance_summary	Returns the count of compliant and noncompliant conformance pa

get_aggregate_discovered_resource_counts	Returns the resource counts across accounts and regions that are p
get_aggregate_resource_config	Returns configuration item that is aggregated for your specific res
get_compliance_details_by_config_rule	Returns the evaluation results for the specified Config rule
get_compliance_details_by_resource	Returns the evaluation results for the specified Amazon Web Serv
get_compliance_summary_by_config_rule	Returns the number of Config rules that are compliant and noncor
get_compliance_summary_by_resource_type	Returns the number of resources that are compliant and the numb
get_conformance_pack_compliance_details	Returns compliance details of a conformance pack for all Amazon
get_conformance_pack_compliance_summary	Returns compliance details for the conformance pack based on the
get_custom_rule_policy	Returns the policy definition containing the logic for your Config
get_discovered_resource_counts	Returns the resource types, the number of each resource type, and
get_organization_config_rule_detailed_status	Returns detailed status for each member account within an organi
get_organization_conformance_pack_detailed_status	Returns detailed status for each member account within an organi
get_organization_custom_rule_policy	Returns the policy definition containing the logic for your organiz
get_resource_config_history	For accurate reporting on the compliance status, you must record
get_resource_evaluation_summary	Returns a summary of resource evaluation for the specified resour
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 th
list_conformance_pack_compliance_scores	Returns a list of conformance pack compliance scores
list_discovered_resources	Accepts a resource type and returns a list of resource identifiers fo
list_resource_evaluations	Returns a list of proactive resource evaluations
list_stored_queries	Lists the stored queries for a single Amazon Web Services accoun
list_tags_for_resource	List the tags for Config resource
put_aggregation_authorization	Authorizes the aggregator account and region to collect data from
put_config_rule	Adds or updates an Config rule to evaluate if your Amazon Web S
put_configuration_aggregator	Creates and updates the configuration aggregator with the selected
put_configuration_recorder	Creates a new configuration recorder to record configuration chan
put_conformance_pack	Creates or updates a conformance pack
put_delivery_channel	Creates a delivery channel object to deliver configuration informa
put_evaluations	Used by an Lambda function to deliver evaluation results to Confi
put_external_evaluation	Add or updates the evaluations for process checks
put_organization_config_rule	Adds or updates an Config rule for your entire organization to eva
put_organization_conformance_pack	Deploys conformance packs across member accounts in an Amaz
put_remediation_configurations	Adds or updates the remediation configuration with a specific Co
put_remediation_exceptions	A remediation exception is when a specified resource is no longer
put_resource_config	Records the configuration state for the resource provided in the re
put_retention_configuration	Creates and updates the retention configuration with details about
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 an
select_resource_config	Accepts a structured query language (SQL) SELECT command, p
start_config_rules_evaluation	Runs an on-demand evaluation for the specified Config rules again
start_configuration_recorder	Starts recording configurations of the Amazon Web Services resou
start_remediation_execution	Runs an on-demand remediation for the specified Config rules aga
start_resource_evaluation	Runs an on-demand evaluation for the specified resource to deter
stop_configuration_recorder	Stops recording configurations of the Amazon Web Services resou
tag_resource	Associates the specified tags to a resource with the specified resou
untag_resource	Deletes specified tags from a resource

Examples

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

## End(Not run)
```

`controltower` *AWS Control Tower*

Description

These interfaces allow you to apply the Amazon Web Services library of pre-defined *controls* to your organizational units, programmatically. In Amazon Web Services Control Tower, the terms "control" and "guardrail" are synonyms.

To call these APIs, you'll need to know:

- the `controlIdentifier` for the control–or guardrail–you are targeting.
- the ARN associated with the target organizational unit (OU), which we call the `targetIdentifier`.
- the ARN associated with a resource that you wish to tag or untag.

To get the `controlIdentifier` for your Amazon Web Services Control Tower control:

The `controlIdentifier` is an ARN that is specified for each control. You can view the `controlIdentifier` in the console on the **Control details** page, as well as in the documentation.

The `controlIdentifier` is unique in each Amazon Web Services Region for each control. You can find the `controlIdentifier` for each Region and control in the Tables of control metadata in the *Amazon Web Services Control Tower User Guide.*

A quick-reference list of control identifers for the Amazon Web Services Control Tower legacy *Strongly recommended* and *Elective* controls is given in Resource identifiers for APIs and controls in the Controls reference guide section of the *Amazon Web Services Control Tower User Guide*. Remember that *Mandatory* controls cannot be added or removed.

ARN format: `arn:aws:controltower:{REGION}::control/{CONTROL_NAME}`

Example:
`arn:aws:controltower:us-west-2::control/AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED`

To get the `targetIdentifier`:

The `targetIdentifier` is the ARN for an OU.

In the Amazon Web Services Organizations console, you can find the ARN for the OU on the **Organizational unit details** page associated with that OU.

OU ARN format:
`arn:${Partition}:organizations::${MasterAccountId}:ou/o-${OrganizationId}/ou-${OrganizationalUnitId`

Details and examples

- Control API input and output examples with CLI
- Enable controls with CloudFormation
- Control metadata tables
- List of identifiers for legacy controls
- Controls reference guide
- Controls library groupings
- Creating Amazon Web Services Control Tower resources with Amazon Web Services CloudFormation

To view the open source resource repository on GitHub, see aws-cloudformation/aws-cloudformation-resource-providers-controltower

Recording API Requests

Amazon Web Services Control Tower supports Amazon Web Services CloudTrail, a service that records Amazon Web Services API calls for your Amazon Web Services account and delivers log files to an Amazon S3 bucket. By using information collected by CloudTrail, you can determine which requests the Amazon Web Services Control Tower service received, who made the request and when, and so on. For more about Amazon Web Services Control Tower and its support for CloudTrail, see Logging Amazon Web Services Control Tower Actions with Amazon Web Services CloudTrail in the Amazon Web Services Control Tower User Guide. To learn more about CloudTrail, including how to turn it on and find your log files, see the Amazon Web Services CloudTrail User Guide.

Usage

```
controltower(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.

- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- controltower(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
```

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_landing_zone	Creates a new landing zone
delete_landing_zone	Decommissions a landing zone
disable_baseline	Disable an EnabledBaseline resource on the specified Target
disable_control	This API call turns off a control
enable_baseline	Enable (apply) a Baseline to a Target
enable_control	This API call activates a control
get_baseline	Retrieve details about an existing Baseline resource by specifying its identifier
get_baseline_operation	Returns the details of an asynchronous baseline operation, as initiated by any of these APIs: En
get_control_operation	Returns the status of a particular EnableControl or DisableControl operation
get_enabled_baseline	Retrieve details of an EnabledBaseline resource by specifying its identifier
get_enabled_control	Retrieves details about an enabled control
get_landing_zone	Returns details about the landing zone
get_landing_zone_operation	Returns the status of the specified landing zone operation
list_baselines	Returns a summary list of all available baselines
list_enabled_baselines	Returns a list of summaries describing EnabledBaseline resources
list_enabled_controls	Lists the controls enabled by Amazon Web Services Control Tower on the specified organizatio
list_landing_zones	Returns the landing zone ARN for the landing zone deployed in your managed account
list_tags_for_resource	Returns a list of tags associated with the resource
reset_enabled_baseline	Re-enables an EnabledBaseline resource
reset_landing_zone	This API call resets a landing zone
tag_resource	Applies tags to a resource
untag_resource	Removes tags from a resource
update_enabled_baseline	Updates an EnabledBaseline resource's applied parameters or version
update_enabled_control	Updates the configuration of an already enabled control
update_landing_zone	This API call updates the landing zone

Examples

```
## Not run:
svc <- controltower()
svc$create_landing_zone(
  Foo = 123
)

## End(Not run)
```

finspace *FinSpace User Environment Management service*

Description

The FinSpace management service provides the APIs for managing FinSpace environments.

Usage

```
finspace(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- finspace(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_environment	Create a new FinSpace environment
create_kx_changeset	Creates a changeset for a kdb database
create_kx_cluster	Creates a new kdb cluster
create_kx_database	Creates a new kdb database in the environment
create_kx_dataview	Creates a snapshot of kdb database with tiered storage capabilities and a pre-warmed
create_kx_environment	Creates a managed kdb environment for the account
create_kx_scaling_group	Creates a new scaling group
create_kx_user	Creates a user in FinSpace kdb environment with an associated IAM role

create_kx_volume	Creates a new volume with a specific amount of throughput and storage capacity
delete_environment	Delete an FinSpace environment
delete_kx_cluster	Deletes a kdb cluster
delete_kx_cluster_node	Deletes the specified nodes from a cluster
delete_kx_database	Deletes the specified database and all of its associated data
delete_kx_dataview	Deletes the specified dataview
delete_kx_environment	Deletes the kdb environment
delete_kx_scaling_group	Deletes the specified scaling group
delete_kx_user	Deletes a user in the specified kdb environment
delete_kx_volume	Deletes a volume
get_environment	Returns the FinSpace environment object
get_kx_changeset	Returns information about a kdb changeset
get_kx_cluster	Retrieves information about a kdb cluster
get_kx_connection_string	Retrieves a connection string for a user to connect to a kdb cluster
get_kx_database	Returns database information for the specified environment ID
get_kx_dataview	Retrieves details of the dataview
get_kx_environment	Retrieves all the information for the specified kdb environment
get_kx_scaling_group	Retrieves details of a scaling group
get_kx_user	Retrieves information about the specified kdb user
get_kx_volume	Retrieves the information about the volume
list_environments	A list of all of your FinSpace environments
list_kx_changesets	Returns a list of all the changesets for a database
list_kx_cluster_nodes	Lists all the nodes in a kdb cluster
list_kx_clusters	Returns a list of clusters
list_kx_databases	Returns a list of all the databases in the kdb environment
list_kx_dataviews	Returns a list of all the dataviews in the database
list_kx_environments	Returns a list of kdb environments created in an account
list_kx_scaling_groups	Returns a list of scaling groups in a kdb environment
list_kx_users	Lists all the users in a kdb environment
list_kx_volumes	Lists all the volumes in a kdb environment
list_tags_for_resource	A list of all tags for a resource
tag_resource	Adds metadata tags to a FinSpace resource
untag_resource	Removes metadata tags from a FinSpace resource
update_environment	Update your FinSpace environment
update_kx_cluster_code_configuration	Allows you to update code configuration on a running cluster
update_kx_cluster_databases	Updates the databases mounted on a kdb cluster, which includes the changesetId and
update_kx_database	Updates information for the given kdb database
update_kx_dataview	Updates the specified dataview
update_kx_environment	Updates information for the given kdb environment
update_kx_environment_network	Updates environment network to connect to your internal network by using a transit
update_kx_user	Updates the user details
update_kx_volume	Updates the throughput or capacity of a volume

Examples

```
## Not run:
svc <- finspace()
```

```
svc$create_environment(
  Foo = 123
)

## End(Not run)
```

`health` *AWS Health APIs and Notifications*

Description

Health

The Health API provides access to the Health information that appears in the Health Dashboard. You can use the API operations to get information about events that might affect your Amazon Web Services and resources.

You must have a Business, Enterprise On-Ramp, or Enterprise Support plan from Amazon Web Services Support to use the Health API. If you call the Health API from an Amazon Web Services account that doesn't have a Business, Enterprise On-Ramp, or Enterprise Support plan, you receive a `SubscriptionRequiredException` error.

For API access, you need an access key ID and a secret access key. Use temporary credentials instead of long-term access keys when possible. Temporary credentials include an access key ID, a secret access key, and a security token that indicates when the credentials expire. For more information, see Best practices for managing Amazon Web Services access keys in the *Amazon Web Services General Reference*.

You can use the Health endpoint health.us-east-1.amazonaws.com (HTTPS) to call the Health API operations. Health supports a multi-Region application architecture and has two regional endpoints in an active-passive configuration. You can use the high availability endpoint example to determine which Amazon Web Services Region is active, so that you can get the latest information from the API. For more information, see Accessing the Health API in the *Health User Guide*.

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

If your Amazon Web Services account is part of Organizations, you can use the Health organizational view feature. This feature provides a centralized view of Health events across all accounts in your organization. You can aggregate 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 Health events in the *Health User Guide*.

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

- Use the eventScopeCode parameter to specify whether to return 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_organizatio` 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(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • **credentials**: – **creds**: ∗ **access_key_id**: AWS access key ID ∗ **secret_access_key**: AWS secret access key ∗ **session_token**: AWS temporary session token – **profile**: The name of a profile to use. If not given, then the default profile is used. – **anonymous**: Set anonymous credentials. • **endpoint**: The complete URL to use for the constructed client. • **region**: The AWS Region used in instantiating the client. • **close_connection**: Immediately close all HTTP connections. • **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`. • **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
credentials	Optional credentials shorthand for the config parameter • **creds**: – **access_key_id**: AWS access key ID – **secret_access_key**: AWS secret access key – **session_token**: AWS temporary session token • **profile**: The name of a profile to use. If not given, then the default profile is used. • **anonymous**: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

describe_affected_accounts_for_organization	Returns a list of accounts in the organization from Organizations that are a
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_entity_aggregates_for_organization	Returns a list of entity aggregates for your Organizations that are affected l
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 Organization
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 Health
disable_health_service_access_for_organization	Disables Health from working with Organizations
enable_health_service_access_for_organization	Enables Health to work with Organizations

Examples

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

## End(Not run)
```

`licensemanager` *AWS License Manager*

Description

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

Usage

```
licensemanager(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
```

```
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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_conversion_task_for_resource	Creates a new license conversion task
create_license_manager_report_generator	Creates a report generator
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_license_manager_report_generator	Deletes the specified report generator
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_conversion_task	Gets information about the specified license type conversion task
get_license_manager_report_generator	Gets information about the specified report generator
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_license_conversion_tasks	Lists the license type conversion tasks for your account
list_license_manager_report_generators	Lists the report generators 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
list_received_grants_for_organization	Lists the grants received for all accounts in the organization
list_received_licenses	Lists received licenses

list_received_licenses_for_organization	Lists the licenses received for all accounts in the organization
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_manager_report_generator	Updates a report generator
update_license_specifications_for_resource	Adds or removes the specified license configurations for the specified An
update_service_settings	Updates License Manager settings for the current Region

Examples

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

## End(Not run)
```

`licensemanagerlinuxsubscriptions`

AWS License Manager Linux Subscriptions

Description

With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.

Usage

```
licensemanagerlinuxsubscriptions(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

get_service_settings	Lists the Linux subscriptions service settings
list_linux_subscription_instances	Lists the running Amazon EC2 instances that were discovered with commercial Linux sub
list_linux_subscriptions	Lists the Linux subscriptions that have been discovered
update_service_settings	Updates the service settings for Linux subscriptions

Examples

```
## Not run:
svc <- licensemanagerlinuxsubscriptions()
svc$get_service_settings(
  Foo = 123
)

## End(Not run)
```

`licensemanagerusersubscriptions` *AWS License Manager User Subscriptions*

Description

With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.

Usage

```
licensemanagerusersubscriptions(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- licensemanagerusersubscriptions(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_user	Associates the user to an EC2 instance to utilize user-based subscriptions

deregister_identity_provider	Deregisters the identity provider from providing user-based subscriptions
disassociate_user	Disassociates the user from an EC2 instance providing user-based subscriptions
list_identity_providers	Lists the identity providers for user-based subscriptions
list_instances	Lists the EC2 instances providing user-based subscriptions
list_product_subscriptions	Lists the user-based subscription products available from an identity provider
list_user_associations	Lists user associations for an identity provider
register_identity_provider	Registers an identity provider for user-based subscriptions
start_product_subscription	Starts a product subscription for a user with the specified identity provider
stop_product_subscription	Stops a product subscription for a user with the specified identity provider
update_identity_provider_settings	Updates additional product configuration settings for the registered identity provider

Examples

```
## Not run:
svc <- licensemanagerusersubscriptions()
svc$associate_user(
  Foo = 123
)

## End(Not run)
```

`managedgrafana` *Amazon Managed Grafana*

Description

Amazon Managed Grafana is a fully managed and secure data visualization service that you can use to instantly query, correlate, and visualize operational metrics, logs, and traces from multiple sources. Amazon Managed Grafana makes it easy to deploy, operate, and scale Grafana, a widely deployed data visualization tool that is popular for its extensible data support.

With Amazon Managed Grafana, you create logically isolated Grafana servers called *workspaces*. In a workspace, you can create Grafana dashboards and visualizations to analyze your metrics, logs, and traces without having to build, package, or deploy any hardware to run Grafana servers.

Usage

```
managedgrafana(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

associate_license	Assigns a Grafana Enterprise license to a workspace
create_workspace	Creates a workspace
create_workspace_api_key	Creates a Grafana API key for the workspace
delete_workspace	Deletes an Amazon Managed Grafana workspace
delete_workspace_api_key	Deletes a Grafana API key for the workspace
describe_workspace	Displays information about one Amazon Managed Grafana workspace
describe_workspace_authentication	Displays information about the authentication methods used in one Amazon Managed G
describe_workspace_configuration	Gets the current configuration string for the given workspace
disassociate_license	Removes the Grafana Enterprise license from a workspace
list_permissions	Lists the users and groups who have the Grafana Admin and Editor roles in this worksp
list_tags_for_resource	The ListTagsForResource operation returns the tags that are associated with the Amazon
list_versions	Lists available versions of Grafana
list_workspaces	Returns a list of Amazon Managed Grafana workspaces in the account, with some infor
tag_resource	The TagResource operation associates tags with an Amazon Managed Grafana resource
untag_resource	The UntagResource operation removes the association of the tag with the Amazon Man
update_permissions	Updates which users in a workspace have the Grafana Admin or Editor roles
update_workspace	Modifies an existing Amazon Managed Grafana workspace
update_workspace_authentication	Use this operation to define the identity provider (IdP) that this workspace authenticates
update_workspace_configuration	Updates the configuration string for the given workspace

Examples

```
## Not run:
svc <- managedgrafana()
svc$associate_license(
  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(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
```

```
      anonymous = "logical"
    ),
    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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e` `html`

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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

Organizations is a web service that enables you to consolidate your multiple Amazon Web Services accounts into an *organization* and centrally manage your accounts and their resources.

This guide provides descriptions of the Organizations operations. For more information about using this service, see the Organizations User Guide.

Support and feedback for Organizations

We welcome your feedback. Send your comments to feedback-awsorganizations@amazon.com or post your feedback and questions in the Organizations support forum. For more information about the Amazon Web Services support forums, see Forums Help.

Endpoint to call When using the CLI or the Amazon Web Services SDK

For the current release of Organizations, specify the `us-east-1` region for all Amazon Web Services API and CLI calls made from the commercial Amazon Web Services Regions outside of China. If calling from one of the Amazon Web Services Regions in China, then specify `cn-northwest-1`. You can do this in the CLI by using these parameters and commands:

- Use the following parameter with each command to specify both the endpoint and its region:
  `--endpoint-url https://organizations.us-east-1.amazonaws.com` *(from commercial Amazon Web Services Regions outside of China)*
  or
  `--endpoint-url https://organizations.cn-northwest-1.amazonaws.com.cn` *(from Amazon Web Services Regions in China)*
- Use the default endpoint, but configure your default region with this command:
  `aws configure set default.region us-east-1` *(from commercial Amazon Web Services Regions outside of China)*
  or
  `aws configure set default.region cn-northwest-1` *(from Amazon Web Services Regions in China)*
- Use the following parameter with each command to specify the endpoint:
  `--region us-east-1` *(from commercial Amazon Web Services Regions outside of China)*
  or
  `--region cn-northwest-1` *(from Amazon Web Services Regions in China)*

Recording API Requests

Organizations supports CloudTrail, a service that records Amazon Web Services API calls for your Amazon Web Services account and delivers log files to an Amazon S3 bucket. By using information collected by CloudTrail, you can determine which requests the Organizations service received, who made the request and when, and so on. For more about Organizations and its support for CloudTrail, see Logging Organizations API calls with CloudTrail in the *Organizations User Guide*. To learn more about CloudTrail, including how to turn it on and find your log files, see the CloudTrail User Guide.

Usage

```
organizations(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.

- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
```

```
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    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
close_account	Closes an Amazon Web Services member account within an organization
create_account	Creates an Amazon Web Services account that is automatically a member of the c
create_gov_cloud_account	This action is available if all of the following are true:
create_organization	Creates an Amazon Web Services organization
create_organizational_unit	Creates an organizational unit (OU) within a root or parent OU
create_policy	Creates a policy of a specified type that you can attach to a root, an organizational
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
delete_resource_policy	Deletes the resource policy from your organization
deregister_delegated_administrator	Removes the specified member Amazon Web Services account as a delegated adm
describe_account	Retrieves 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
describe_resource_policy	Retrieves information about a resource policy
detach_policy	Detaches a policy from a target root, organizational unit (OU), or account
disable_aws_service_access	Disables the integration of an Amazon Web Services service (the service that is sp
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 Amazon Web Services service (the service that is sp
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 Amazon Web Services services that you enabled to integrate
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 Amazon Web Services accounts that are designated as delegated adminis
list_delegated_services_for_account	List the Amazon Web Services services for which the specified account is a deleg
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)
put_resource_policy	Creates or updates a resource policy
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 Amazon Web Services 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* .
- To learn more about Performance Insights and Amazon DocumentDB clusters, go to the *Amazon DocumentDB Developer Guide* .

Usage

```
pi(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e` `html`

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_performance_analysis_report	Creates a new performance analysis report for a specific time period for the DB instanc
delete_performance_analysis_report	Deletes a performance analysis report
describe_dimension_keys	For a specific time period, retrieve the top N dimension keys for a metric
get_dimension_key_details	Get the attributes of the specified dimension group for a DB instance or data source
get_performance_analysis_report	Retrieves the report including the report ID, status, time details, and the insights with re
get_resource_metadata	Retrieve the metadata for different features
get_resource_metrics	Retrieve Performance Insights metrics for a set of data sources over a time period
list_available_resource_dimensions	Retrieve the dimensions that can be queried for each specified metric type on a specifie
list_available_resource_metrics	Retrieve metrics of the specified types that can be queried for a specified DB instance
list_performance_analysis_reports	Lists all the analysis reports created for the DB instance
list_tags_for_resource	Retrieves all the metadata tags associated with Amazon RDS Performance Insights reso
tag_resource	Adds metadata tags to the Amazon RDS Performance Insights resource
untag_resource	Deletes the metadata tags from the Amazon RDS Performance Insights resource

Examples

```
## Not run:
svc <- pi()
svc$create_performance_analysis_report(
  Foo = 123
)

## End(Not run)
```

---

prometheusservice *Amazon Prometheus Service*

Description

Amazon Managed Service for Prometheus is a serverless, Prometheus-compatible monitoring service for container metrics that makes it easier to securely monitor container environments at scale. With Amazon Managed Service for Prometheus, you can use the same open-source Prometheus data model and query language that you use today to monitor the performance of your containerized workloads, and also enjoy improved scalability, availability, and security without having to manage the underlying infrastructure.

For more information about Amazon Managed Service for Prometheus, see the Amazon Managed Service for Prometheus User Guide.

Amazon Managed Service for Prometheus includes two APIs.

- Use the Amazon Web Services API described in this guide to manage Amazon Managed Service for Prometheus resources, such as workspaces, rule groups, and alert managers.
- Use the Prometheus-compatible API to work within your Prometheus workspace.

Usage

```
prometheusservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- prometheusservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_alert_manager_definition	The CreateAlertManagerDefinition operation creates the alert manager definition in a wo
create_logging_configuration	The CreateLoggingConfiguration operation creates a logging configuration for the works
create_rule_groups_namespace	The CreateRuleGroupsNamespace operation creates a rule groups namespace within a w
create_scraper	The CreateScraper operation creates a scraper to collect metrics
create_workspace	Creates a Prometheus workspace
delete_alert_manager_definition	Deletes the alert manager definition from a workspace
delete_logging_configuration	Deletes the logging configuration for a workspace
delete_rule_groups_namespace	Deletes one rule groups namespace and its associated rule groups definition
delete_scraper	The DeleteScraper operation deletes one scraper, and stops any metrics collection that th
delete_workspace	Deletes an existing workspace
describe_alert_manager_definition	Retrieves the full information about the alert manager definition for a workspace
describe_logging_configuration	Returns complete information about the current logging configuration of the workspace
describe_rule_groups_namespace	Returns complete information about one rule groups namespace
describe_scraper	The DescribeScraper operation displays information about an existing scraper

describe_workspace	Returns information about an existing workspace
get_default_scraper_configuration	The GetDefaultScraperConfiguration operation returns the default scraper configuration
list_rule_groups_namespaces	Returns a list of rule groups namespaces in a workspace
list_scrapers	The ListScrapers operation lists all of the scrapers in your account
list_tags_for_resource	The ListTagsForResource operation returns the tags that are associated with an Amazon
list_workspaces	Lists all of the Amazon Managed Service for Prometheus workspaces in your account
put_alert_manager_definition	Updates an existing alert manager definition in a workspace
put_rule_groups_namespace	Updates an existing rule groups namespace within a workspace
tag_resource	The TagResource operation associates tags with an Amazon Managed Service for Prome
untag_resource	Removes the specified tags from an Amazon Managed Service for Prometheus resource
update_logging_configuration	Updates the log group ARN or the workspace ID of the current logging configuration
update_workspace_alias	Updates the alias of an existing workspace

Examples

```
## Not run:
svc <- prometheusservice()
svc$create_alert_manager_definition(
  Foo = 123
)

## End(Not run)
```

`resiliencehub` *AWS Resilience Hub*

Description

Resilience Hub helps you proactively prepare and protect your Amazon Web Services applications from disruptions. It offers continual resiliency assessment and validation that integrates into your software development lifecycle. This enables you to uncover resiliency weaknesses, ensure recovery time objective (RTO) and recovery point objective (RPO) targets for your applications are met, and resolve issues before they are released into production.

Usage

```
resiliencehub(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

add_draft_app_version_resource_mappings	Adds the source of resource-maps to the draft version of an applicatio
batch_update_recommendation_status	Enables you to include or exclude one or more operational recommend
create_app	Creates an Resilience Hub application
create_app_version_app_component	Creates a new Application Component in the Resilience Hub applicati
create_app_version_resource	Adds a resource to the Resilience Hub application and assigns it to the
create_recommendation_template	Creates a new recommendation template for the Resilience Hub applic
create_resiliency_policy	Creates a resiliency policy for an application
delete_app	Deletes an Resilience Hub application
delete_app_assessment	Deletes an Resilience Hub application assessment
delete_app_input_source	Deletes the input source and all of its imported resources from the Res
delete_app_version_app_component	Deletes an Application Component from the Resilience Hub applicatio
delete_app_version_resource	Deletes a resource from the Resilience Hub application
delete_recommendation_template	Deletes a recommendation template
delete_resiliency_policy	Deletes a resiliency policy
describe_app	Describes an Resilience Hub application
describe_app_assessment	Describes an assessment for an Resilience Hub application
describe_app_version	Describes the Resilience Hub application version
describe_app_version_app_component	Describes an Application Component in the Resilience Hub applicatio
describe_app_version_resource	Describes a resource of the Resilience Hub application
describe_app_version_resources_resolution_status	Returns the resolution status for the specified resolution identifier for a

describe_app_version_template	Describes details about an Resilience Hub application
describe_draft_app_version_resources_import_status	Describes the status of importing resources to an application version
describe_resiliency_policy	Describes a specified resiliency policy for an Resilience Hub applicati
import_resources_to_draft_app_version	Imports resources to Resilience Hub application draft version from dif
list_alarm_recommendations	Lists the alarm recommendations for an Resilience Hub application
list_app_assessment_compliance_drifts	List of compliance drifts that were detected while running an assessm
list_app_assessments	Lists the assessments for an Resilience Hub application
list_app_component_compliances	Lists the compliances for an Resilience Hub Application Component
list_app_component_recommendations	Lists the recommendations for an Resilience Hub Application Compo
list_app_input_sources	Lists all the input sources of the Resilience Hub application
list_apps	Lists your Resilience Hub applications
list_app_version_app_components	Lists all the Application Components in the Resilience Hub applicatio
list_app_version_resource_mappings	Lists how the resources in an application version are mapped/sourced
list_app_version_resources	Lists all the resources in an Resilience Hub application
list_app_versions	Lists the different versions for the Resilience Hub applications
list_recommendation_templates	Lists the recommendation templates for the Resilience Hub applicatio
list_resiliency_policies	Lists the resiliency policies for the Resilience Hub applications
list_sop_recommendations	Lists the standard operating procedure (SOP) recommendations for th
list_suggested_resiliency_policies	Lists the suggested resiliency policies for the Resilience Hub applicati
list_tags_for_resource	Lists the tags for your resources in your Resilience Hub applications
list_test_recommendations	Lists the test recommendations for the Resilience Hub application
list_unsupported_app_version_resources	Lists the resources that are not currently supported in Resilience Hub
publish_app_version	Publishes a new version of a specific Resilience Hub application
put_draft_app_version_template	Adds or updates the app template for an Resilience Hub application dr
remove_draft_app_version_resource_mappings	Removes resource mappings from a draft application version
resolve_app_version_resources	Resolves the resources for an application version
start_app_assessment	Creates a new application assessment for an application
tag_resource	Applies one or more tags to a resource
untag_resource	Removes one or more tags from a resource
update_app	Updates an application
update_app_version	Updates the Resilience Hub application version
update_app_version_app_component	Updates an existing Application Component in the Resilience Hub app
update_app_version_resource	Updates the resource details in the Resilience Hub application
update_resiliency_policy	Updates a resiliency policy

Examples

```
## Not run:
svc <- resiliencehub()
svc$add_draft_app_version_resource_mappings(
  Foo = 123
)

## End(Not run)
```

`resourcegroups` *AWS Resource Groups*

Description

Resource Groups lets you organize Amazon Web Services resources such as Amazon Elastic Compute Cloud instances, Amazon Relational Database Service databases, and Amazon Simple Storage Service 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 Amazon Web Services Systems Manager Automation documents, on tag-related resources in Amazon Web Services Systems Manager. Groups of tagged resources also let you quickly view a custom console in Amazon Web Services Systems Manager that shows 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 Resource Groups User Guide.

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 Amazon Web Services resources based on a resource query

Usage

```
resourcegroups(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key

    * **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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_account_settings	Retrieves the current status of optional features in Resource Groups
get_group	Returns information about a specified resource group
get_group_configuration	Retrieves 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 Amazon Web Services 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_account_settings	Turns on or turns off optional features in Resource Groups
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

Usage

```
resourcegroupstaggingapi(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID

  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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 Amazon W
get_tag_keys	Returns all tag keys currently in use in the specified Amazon Web Services Region for the calling
get_tag_values	Returns all tag values for the specified key that are used in the specified Amazon Web Services R
start_report_creation	Generates a report that lists all tagged resources in the accounts across your organization and tells
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

Service Catalog

Service Catalog enables organizations to create and manage catalogs of IT services that are approved for Amazon Web Services. To get the most out of this documentation, you should be familiar with the terminology discussed in Service Catalog Concepts.

Usage

```
servicecatalog(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID

    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
```

```
      region = "string",
      close_connection = "logical",
      timeout = "numeric",
      s3_force_path_style = "logical",
      sts_regional_endpoint = "string"
    ),
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    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 p
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 portfo
batch_associate_service_action_with_provisioning_artifact	Associates multiple self-service actions with provisioning a
batch_disassociate_service_action_from_provisioning_artifact	Disassociates a batch of self-service actions from the specif
copy_product	Copies the specified source product to the specified target p
create_constraint	Creates a constraint
create_portfolio	Creates a portfolio
create_portfolio_share	Shares the specified portfolio with the specified account or
create_product	Creates a product
create_provisioned_product_plan	Creates a plan
create_provisioning_artifact	Creates a provisioning artifact (also known as a version) fo
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 acco
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
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
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 specifi
describe_provisioning_artifact	Gets information about the specified provisioning artifact (a
describe_provisioning_parameters	Gets information about the configuration required to provis
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 actio
describe_tag_option	Gets information about the specified TagOption
disable_aws_organizations_access	Disable portfolio sharing through the Organizations service
disassociate_budget_from_resource	Disassociates the specified budget from the specified resour
disassociate_principal_from_portfolio	Disassociates a previously associated principal ARN from a
disassociate_product_from_portfolio	Disassociates the specified product from the specified portf
disassociate_service_action_from_provisioning_artifact	Disassociates the specified self-service action association fr
disassociate_tag_option_from_resource	Disassociates the specified TagOption from the specified re
enable_aws_organizations_access	Enable portfolio sharing feature through Organizations
execute_provisioned_product_plan	Provisions or modifies a product based on the resource char
execute_provisioned_product_service_action	Executes a self-service action against a provisioned product
get_aws_organizations_access_status	Get the Access Status for Organizations portfolio share feat
get_provisioned_product_outputs	This API takes either a ProvisonedProductId or a Provision
import_as_provisioned_product	Requests the import of a resource as an Service Catalog pro
list_accepted_portfolio_shares	Lists all imported portfolios for which account-to-account s
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 specifie
list_portfolio_access	Lists the account IDs that have access to the specified portf
list_portfolios	Lists all portfolios in the catalog
list_portfolios_for_product	Lists all portfolios that the specified product is associated w
list_principals_for_portfolio	Lists all PrincipalARNs and corresponding PrincipalTypes
list_provisioned_product_plans	Lists the plans for the specified provisioned product or all p
list_provisioning_artifacts	Lists all provisioning artifacts (also known as versions) for
list_provisioning_artifacts_for_service_action	Lists all provisioning artifacts (also known as versions) for
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 w
list_stack_instances_for_provisioned_product	Returns summary information about stack instances that are
list_tag_options	Lists the specified TagOptions or all TagOptions
notify_provision_product_engine_workflow_result	Notifies the result of the provisioning engine execution
notify_terminate_provisioned_product_engine_workflow_result	Notifies the result of the terminate engine execution
notify_update_provisioned_product_engine_workflow_result	Notifies the result of the update engine execution
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 termin
search_products	Gets information about the products to which the caller has
search_products_as_admin	Gets information about the products for the specified portfo
search_provisioned_products	Gets information about the provisioned products that meet t
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 provi
update_provisioned_product_properties	Requests updates to the properties of the specified provisio
update_provisioning_artifact	Updates the specified provisioning artifact (also known as a
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 Amazon Web Services workloads grow. Quotas, also referred to as limits, are the maximum number of resources that you can create in your Amazon Web Services account. For more information, see the Service Quotas User Guide.

Usage

```
servicequotas(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
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 Amazon We
list_requested_service_quota_change_history	Retrieves the quota increase requests for the specified Amazon Web
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 Amazon Web Servic
list_services	Lists the names and codes for the Amazon Web Services integrated
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

Amazon Web Services Systems Manager is the operations hub for your Amazon Web Services applications and resources and a secure end-to-end management solution for hybrid cloud environments that enables safe and secure operations at scale.

This reference is intended to be used with the Amazon Web Services Systems Manager User Guide. To get started, see Setting up Amazon Web Services Systems Manager.

Related resources

- For information about each of the capabilities that comprise Systems Manager, see Systems Manager capabilities in the *Amazon Web Services Systems Manager User Guide*.
- For details about predefined runbooks for Automation, a capability of Amazon Web Services Systems Manager, see the *Systems Manager Automation runbook reference* .
- For information about AppConfig, a capability of Systems Manager, see the *AppConfig User Guide* and the * AppConfig API Reference* .
- For information about Incident Manager, a capability of Systems Manager, see the *Systems Manager Incident Manager User Guide* and the * Systems Manager Incident Manager API Reference* .

Usage

```
ssm(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key

    * **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

add_tags_to_resource	Adds or overwrites one or more tags for the specified resource
associate_ops_item_related_item	Associates a related item to a Systems Manager OpsCenter Ops
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 Amazon Web Services Systems Manag
create_document	Creates a Amazon Web Services Systems Manager (SSM docu
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, Amazo
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 Amazon Web Services Systems Ma
delete_document	Deletes the Amazon Web Services Systems Manager document
delete_inventory	Delete a custom inventory type or the data associated with a cu
delete_maintenance_window	Deletes a maintenance window
delete_ops_item	Delete an OpsItem
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
delete_resource_policy	Deletes a Systems Manager resource policy
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 managed no
describe_association_executions	Views all executions for a specific association ID
describe_association_execution_targets	Views information about a specific execution of a specific assoc
describe_automation_executions	Provides details about all active and terminated Automation exe
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 Amazon Web Services Systems Manag
describe_document_permission	Describes the permissions for a Amazon Web Services System
describe_effective_instance_associations	All associations for the managed nodes
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 managed nodes
describe_instance_information	Provides information about one or more of your managed node
describe_instance_patches	Retrieves information about the patches on the specified manag
describe_instance_patch_states	Retrieves the high-level patch state of one or more managed no
describe_instance_patch_states_for_patch_group	Retrieves the high-level patch state for the managed nodes in th
describe_instance_properties	An API operation used by the Systems Manager console to disp
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 Amazon Web Service
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	Lists the parameters in your Amazon Web Services account or
describe_patch_baselines	Lists the patch baselines in your Amazon Web Services accoun
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 informati
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
disassociate_ops_item_related_item	Deletes the association between an OpsItem and a related item
get_automation_execution	Get detailed information about a particular Automation executi
get_calendar_state	Gets the state of a Amazon Web Services Systems Manager cha
get_command_invocation	Returns detailed information about command execution for an i
get_connection_status	Retrieves the Session Manager connection status for a managed
get_default_patch_baseline	Retrieves the default patch baseline
get_deployable_patch_snapshot_for_instance	Retrieves the current snapshot for the patch baseline the manag
get_document	Gets the contents of the specified Amazon Web Services Syste
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	Retrieves the details of a maintenance window task
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 operations metadata (OpsData) based on sp
get_parameter	Get information about a single parameter by specifying the par
get_parameter_history	Retrieves the history of all changes to a parameter
get_parameters	Get information about one or more parameters by specifying m
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_resource_policies	Returns an array of the Policy object
get_service_setting	ServiceSetting is an account-level setting for an Amazon Web S
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 Amazon W
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 managed
list_commands	Lists the commands requested by users of the Amazon Web Se
list_compliance_items	For a specified resource ID, this API operation returns a list of
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 a change te
list_documents	Returns all Systems Manager (SSM) documents in the current A
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 Amazon Web
list_ops_item_related_items	Lists all related-item resources associated with a Systems Mana
list_ops_metadata	Amazon Web Services Systems Manager calls this API operati
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 Amazon Web Services Systems Manager document (S
put_compliance_items	Registers a compliance type and other compliance details on a
put_inventory	Bulk update custom inventory items on one or more managed r
put_parameter	Add a parameter to the system
put_resource_policy	Creates or updates a Systems Manager resource policy
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 Amazon Web S
resume_session	Reconnects a session to a managed node after it has been disco
send_automation_signal	Sends a signal to an Automation execution to change the curren
send_command	Runs commands on one or more managed nodes
start_associations_once	Runs an association immediately and only one time
start_automation_execution	Initiates execution of an Automation runbook
start_change_request_execution	Creates a change request for Change Manager
start_session	Initiates a connection to a target (for example, a managed node
stop_automation_execution	Stop an Automation that is currently running

terminate_session	Permanently ends a session and closes the data connection betw
unlabel_parameter_version	Remove a label or labels from a parameter
update_association	Updates an association
update_association_status	Updates the status of the Amazon Web Services Systems Mana
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 Identity and Access Management (IAM) role that
update_ops_item	Edit or change an OpsItem
update_ops_metadata	Amazon Web Services Systems Manager calls this API operati
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 Amazon Web S

Examples

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

## End(Not run)
```

`ssmcontacts` *AWS Systems Manager Incident Manager Contacts*

Description

Systems Manager Incident Manager is an incident management console designed to help users mitigate and recover from incidents affecting their Amazon Web Services-hosted applications. An incident is any unplanned interruption or reduction in quality of services.

Incident Manager increases incident resolution by notifying responders of impact, highlighting relevant troubleshooting data, and providing collaboration tools to get services back up and running. To achieve the primary goal of reducing the time-to-resolution of critical incidents, Incident Manager automates response plans and enables responder team escalation.

Usage

```
ssmcontacts(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- ssmcontacts(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

accept_page	Used to acknowledge an engagement to a contact channel during an incident
activate_contact_channel	Activates a contact's contact channel
create_contact	Contacts are either the contacts that Incident Manager engages during an incident or the escalat
create_contact_channel	A contact channel is the method that Incident Manager uses to engage your contact
create_rotation	Creates a rotation in an on-call schedule
create_rotation_override	Creates an override for a rotation in an on-call schedule
deactivate_contact_channel	To no longer receive Incident Manager engagements to a contact channel, you can deactivate th
delete_contact	To remove a contact from Incident Manager, you can delete the contact
delete_contact_channel	To no longer receive engagements on a contact channel, you can delete the channel from a cont
delete_rotation	Deletes a rotation from the system
delete_rotation_override	Deletes an existing override for an on-call rotation
describe_engagement	Incident Manager uses engagements to engage contacts and escalation plans during an incident
describe_page	Lists details of the engagement to a contact channel
get_contact	Retrieves information about the specified contact or escalation plan

get_contact_channel	List details about a specific contact channel
get_contact_policy	Retrieves the resource policies attached to the specified contact or escalation plan
get_rotation	Retrieves information about an on-call rotation
get_rotation_override	Retrieves information about an override to an on-call rotation
list_contact_channels	Lists all contact channels for the specified contact
list_contacts	Lists all contacts and escalation plans in Incident Manager
list_engagements	Lists all engagements that have happened in an incident
list_page_receipts	Lists all of the engagements to contact channels that have been acknowledged
list_page_resolutions	Returns the resolution path of an engagement
list_pages_by_contact	Lists the engagements to a contact's contact channels
list_pages_by_engagement	Lists the engagements to contact channels that occurred by engaging a contact
list_preview_rotation_shifts	Returns a list of shifts based on rotation configuration parameters
list_rotation_overrides	Retrieves a list of overrides currently specified for an on-call rotation
list_rotations	Retrieves a list of on-call rotations
list_rotation_shifts	Returns a list of shifts generated by an existing rotation in the system
list_tags_for_resource	Lists the tags of an escalation plan or contact
put_contact_policy	Adds a resource policy to the specified contact or escalation plan
send_activation_code	Sends an activation code to a contact channel
start_engagement	Starts an engagement to a contact or escalation plan
stop_engagement	Stops an engagement before it finishes the final stage of the escalation plan or engagement plan
tag_resource	Tags a contact or escalation plan
untag_resource	Removes tags from the specified resource
update_contact	Updates the contact or escalation plan specified
update_contact_channel	Updates a contact's contact channel
update_rotation	Updates the information specified for an on-call rotation

Examples

```
## Not run:
svc <- ssmcontacts()
svc$accept_page(
  Foo = 123
)

## End(Not run)
```

`ssmincidents` *AWS Systems Manager Incident Manager*

Description

Systems Manager Incident Manager is an incident management console designed to help users mitigate and recover from incidents affecting their Amazon Web Services-hosted applications. An incident is any unplanned interruption or reduction in quality of services.

Incident Manager increases incident resolution by notifying responders of impact, highlighting relevant troubleshooting data, and providing collaboration tools to get services back up and running. To achieve the primary goal of reducing the time-to-resolution of critical incidents, Incident Manager automates response plans and enables responder team escalation.

Usage

```
ssmincidents(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- ssmincidents(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_get_incident_findings	Retrieves details about all specified findings for an incident, including descriptive details about
create_replication_set	A replication set replicates and encrypts your data to the provided Regions with the provided K
create_response_plan	Creates a response plan that automates the initial response to incidents
create_timeline_event	Creates a custom timeline event on the incident details page of an incident record
delete_incident_record	Delete an incident record from Incident Manager
delete_replication_set	Deletes all Regions in your replication set
delete_resource_policy	Deletes the resource policy that Resource Access Manager uses to share your Incident Manager
delete_response_plan	Deletes the specified response plan

delete_timeline_event	Deletes a timeline event from an incident
get_incident_record	Returns the details for the specified incident record
get_replication_set	Retrieve your Incident Manager replication set
get_resource_policies	Retrieves the resource policies attached to the specified response plan
get_response_plan	Retrieves the details of the specified response plan
get_timeline_event	Retrieves a timeline event based on its ID and incident record
list_incident_findings	Retrieves a list of the IDs of findings, plus their last modified times, that have been identified fo
list_incident_records	Lists all incident records in your account
list_related_items	List all related items for an incident record
list_replication_sets	Lists details about the replication set configured in your account
list_response_plans	Lists all response plans in your account
list_tags_for_resource	Lists the tags that are attached to the specified response plan or incident
list_timeline_events	Lists timeline events for the specified incident record
put_resource_policy	Adds a resource policy to the specified response plan
start_incident	Used to start an incident from CloudWatch alarms, EventBridge events, or manually
tag_resource	Adds a tag to a response plan
untag_resource	Removes a tag from a resource
update_deletion_protection	Update deletion protection to either allow or deny deletion of the final Region in a replication se
update_incident_record	Update the details of an incident record
update_related_items	Add or remove related items from the related items tab of an incident record
update_replication_set	Add or delete Regions from your replication set
update_response_plan	Updates the specified response plan
update_timeline_event	Updates a timeline event

Examples

```
## Not run:
svc <- ssmincidents()
svc$batch_get_incident_findings(
  Foo = 123
)

## End(Not run)
```

`ssmsap` *AWS Systems Manager for SAP*

Description

This API reference provides descriptions, syntax, and other details about each of the actions and data types for AWS Systems Manager for SAP. The topic for each action shows the API request parameters and responses.

Usage

```
ssmsap(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- ssmsap(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

delete_resource_permission	Removes permissions associated with the target database
deregister_application	Deregister an SAP application with AWS Systems Manager for SAP
get_application	Gets an application registered with AWS Systems Manager for SAP
get_component	Gets the component of an application registered with AWS Systems Manager for SAP
get_database	Gets the SAP HANA database of an application registered with AWS Systems Manager for SA
get_operation	Gets the details of an operation by specifying the operation ID
get_resource_permission	Gets permissions associated with the target database
list_applications	Lists all the applications registered with AWS Systems Manager for SAP
list_components	Lists all the components registered with AWS Systems Manager for SAP
list_databases	Lists the SAP HANA databases of an application registered with AWS Systems Manager for SA
list_operations	Lists the operations performed by AWS Systems Manager for SAP
list_tags_for_resource	Lists all tags on an SAP HANA application and/or database registered with AWS Systems Man
put_resource_permission	Adds permissions to the target database
register_application	Register an SAP application with AWS Systems Manager for SAP

start_application_refresh	Refreshes a registered application
tag_resource	Creates tag for a resource by specifying the ARN
untag_resource	Delete the tags for a resource
update_application_settings	Updates the settings of an application registered with AWS Systems Manager for SAP

Examples

```
## Not run:
svc <- ssmsap()
svc$delete_resource_permission(
  Foo = 123
)

## End(Not run)
```

`support` *AWS Support*

Description

Amazon Web Services Support

The *Amazon Web Services Support API Reference* is intended for programmers who need detailed information about the Amazon Web Services Support operations and data types. You can use the API to manage your support cases programmatically. The Amazon Web Services Support API uses HTTP methods that return results in JSON format.

- You must have a Business, Enterprise On-Ramp, or Enterprise Support plan to use the Amazon Web Services Support API.
- If you call the Amazon Web Services Support API from an account that doesn't have a Business, Enterprise On-Ramp, or Enterprise Support plan, the `SubscriptionRequiredException` error message appears. For information about changing your support plan, see Amazon Web Services Support.

You can also use the Amazon Web Services Support API to access features for Trusted Advisor. You can return a list of checks and their descriptions, get check results, specify checks to refresh, and get the refresh status of checks.

You can manage your support cases with the following Amazon Web Services Support API operations:

- The `create_case`, `describe_cases`, `describe_attachment`, and `resolve_case` operations create Amazon Web Services Support cases, retrieve information about cases, and resolve cases.
- The `describe_communications`, `add_communication_to_case`, and `add_attachments_to_set` operations retrieve and add communications and attachments to Amazon Web Services Support cases.

- The describe_services and describe_severity_levels operations return Amazon Web Service names, service codes, service categories, and problem severity levels. You use these values when you call the create_case operation.

You can also use the Amazon Web Services Support API to call the Trusted Advisor operations. For more information, see Trusted Advisor in the *Amazon Web Services Support User Guide*.

For authentication of requests, Amazon Web Services Support uses Signature Version 4 Signing Process.

For more information about this service and the endpoints to use, see About the Amazon Web Services Support API in the *Amazon Web Services Support User Guide*.

Usage

```
support(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • **credentials**: – **creds**: ∗ **access_key_id**: AWS access key ID ∗ **secret_access_key**: AWS secret access key ∗ **session_token**: AWS temporary session token – **profile**: The name of a profile to use. If not given, then the default profile is used. – **anonymous**: Set anonymous credentials. • **endpoint**: The complete URL to use for the constructed client. • **region**: The AWS Region used in instantiating the client. • **close_connection**: Immediately close all HTTP connections. • **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY. • **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html
credentials	Optional credentials shorthand for the config parameter • **creds**: – **access_key_id**: AWS access key ID – **secret_access_key**: AWS secret access key – **session_token**: AWS temporary session token • **profile**: The name of a profile to use. If not given, then the default profile is used. • **anonymous**: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

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",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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 Amazon Web Services Su
create_case	Creates a case in the Amazon Web Services 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_create_case_options	Returns a list of CreateCaseOption types along with the corresponding sup
describe_services	Returns the current list of Amazon Web Services services and a list of serv

describe_severity_levels	Returns the list of severity levels that you can assign to a support case
describe_supported_languages	Returns a list of supported languages for a specified categoryCode, issueT
describe_trusted_advisor_check_refresh_statuses	Returns the refresh status of the Trusted Advisor checks that have the spec
describe_trusted_advisor_check_result	Returns the results of the Trusted Advisor check that has the specified che
describe_trusted_advisor_checks	Returns information about all available Trusted Advisor checks, including
describe_trusted_advisor_check_summaries	Returns the results for the Trusted Advisor check summaries for the check
refresh_trusted_advisor_check	Refreshes the Trusted Advisor check that you specify using the check ID
resolve_case	Resolves a support case

Examples

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

## End(Not run)
```

`supportapp` *AWS Support App*

Description

Amazon Web Services Support App in Slack

You can use the Amazon Web Services Support App in Slack API to manage your support cases in Slack for your Amazon Web Services account. After you configure your Slack workspace and channel with the Amazon Web Services Support App, you can perform the following tasks directly in your Slack channel:

- Create, search, update, and resolve your support cases
- Request service quota increases for your account
- Invite Amazon Web Services Support agents to your channel so that you can chat directly about your support cases

For more information about how to perform these actions in Slack, see the following documentation in the *Amazon Web Services Support User Guide*:

- Amazon Web Services Support App in Slack
- Joining a live chat session with Amazon Web Services Support
- Requesting service quota increases
- Amazon Web Services Support App commands in Slack

You can also use the Amazon Web Services Management Console instead of the Amazon Web Services Support App API to manage your Slack configurations. For more information, see Authorize a Slack workspace to enable the Amazon Web Services Support App.

- You must have a Business or Enterprise Support plan to use the Amazon Web Services Support App API.
- For more information about the Amazon Web Services Support App endpoints, see the Amazon Web Services Support App in Slack endpoints in the *Amazon Web Services General Reference*.

Usage

```
supportapp(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html`

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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 <- supportapp(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_slack_channel_configuration	Creates a Slack channel configuration for your Amazon Web Services account

delete_account_alias	Deletes an alias for an Amazon Web Services account ID
delete_slack_channel_configuration	Deletes a Slack channel configuration from your Amazon Web Services accoun
delete_slack_workspace_configuration	Deletes a Slack workspace configuration from your Amazon Web Services acco
get_account_alias	Retrieves the alias from an Amazon Web Services account ID
list_slack_channel_configurations	Lists the Slack channel configurations for an Amazon Web Services account
list_slack_workspace_configurations	Lists the Slack workspace configurations for an Amazon Web Services account
put_account_alias	Creates or updates an individual alias for each Amazon Web Services account I
register_slack_workspace_for_organization	Registers a Slack workspace for your Amazon Web Services account
update_slack_channel_configuration	Updates the configuration for a Slack channel, such as case update notifications

Examples

```
## Not run:
svc <- supportapp()
svc$create_slack_channel_configuration(
  Foo = 123
)

## End(Not run)
```

`synthetics` *Synthetics*

Description

Amazon CloudWatch Synthetics

You can use Amazon CloudWatch Synthetics to continually monitor your services. You can create and manage *canaries*, which are modular, lightweight scripts that monitor your endpoints and APIs from the outside-in. You can set up your canaries to run 24 hours a day, once per minute. The canaries help you check the availability and latency of your web services and troubleshoot anomalies by investigating load time data, screenshots of the UI, logs, and metrics. The canaries seamlessly integrate with CloudWatch ServiceLens to help you trace the causes of impacted nodes in your applications. For more information, see Using ServiceLens to Monitor the Health of Your Applications in the *Amazon CloudWatch User Guide*.

Before you create and manage canaries, be aware of the security considerations. For more information, see Security Considerations for Synthetics Canaries.

Usage

```
synthetics(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e html

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

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

```
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string",
  close_connection = "logical",
  timeout = "numeric",
  s3_force_path_style = "logical",
  sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)
```

Operations

associate_resource	Associates a canary with a group
create_canary	Creates a canary
create_group	Creates a group which you can use to associate canaries with each other, including cross-Region
delete_canary	Permanently deletes the specified canary
delete_group	Deletes a group
describe_canaries	This operation returns a list of the canaries in your account, along with full details about each ca
describe_canaries_last_run	Use this operation to see information from the most recent run of each canary that you have crea
describe_runtime_versions	Returns a list of Synthetics canary runtime versions
disassociate_resource	Removes a canary from a group
get_canary	Retrieves complete information about one canary
get_canary_runs	Retrieves a list of runs for a specified canary
get_group	Returns information about one group
list_associated_groups	Returns a list of the groups that the specified canary is associated with
list_group_resources	This operation returns a list of the ARNs of the canaries that are associated with the specified gr
list_groups	Returns a list of all groups in the account, displaying their names, unique IDs, and ARNs
list_tags_for_resource	Displays the tags associated with a canary or group
start_canary	Use this operation to run a canary that has already been created
stop_canary	Stops the canary to prevent all future runs
tag_resource	Assigns one or more tags (key-value pairs) to the specified canary or group
untag_resource	Removes one or more tags from the specified resource

update_canary	Updates the configuration of a canary that has already been created

Examples

```
## Not run:
svc <- synthetics()
svc$associate_resource(
  Foo = 123
)

## End(Not run)
```

Index