Rackspace Autoscale

Developer Guide

API v1.0 (Jul 22, 2013)

DRAFT

PREVIEW

rackspace

*the **open cloud** company*

Rackspace Autoscale Developer Guide

API v1.0 (2013-07-22)
Copyright © 2013
Copyright © 2013 Rackspace US, Inc. All rights reserved.

Table of Contents

List of Tables

List of Examples

1. Overview

The Rackspace Autoscale service responds to events by changing capacity to meet current needs. This ability to expand the configuration in response to increased workload means that you can begin with a minimal cloud configuration and grow only when the cost of that growth is justified.

1.1. Intended Audience

This document is intended for software developers who are interested in developing applications that use the Rackspace Cloud Cloud Autoscale API. To use this API, we assume you have the following skills:

- ReSTful web services

- HTTP/1.1 conventions

- JSON data serialization formats

1.2. Document Change History

This version of the document replaces and obsoletes all previous versions. The most recent changes are described in the following table:

Revision Date	Summary of Changes
Jul 22, 2013	• Initial public draft of full DevGuide for Early Access release.
Jul 2, 2013	• Initial internal draft of API Operations section for Preview release.

1.3. Additional Resources

Rackspace Autoscale is an open-source project. You can see and contribute to source code in our GitHub repository, https://github.com/rackerlabs/otter.

You can see and experiment with all Rackspace Autoscale API calls at http://docs.autoscale.apiary.io/.

To use Rackspace Autoscale, you must be able to set cloud monitoring alarms and create cloud servers and cloud load balancers. You can read API documentation for those and other Rackspace services at http://docs.rackspace.com/.

1.4. API Contract Changes

The API contract is not locked and may change during the Early Access Program.

Rackspace will notify customers in Release Notes when and if the contract does change.

1.5. Pricing and Service Level

For the Early Access period, your use of Rackspace Autoscale is not billed. However, Rackspace Autoscale relies on other services which are part of the Rackspace Cloud; your

use of those services through the API will be billed as per the pricing schedule published for each service:

- For Rackspace Next Gen Cloud Servers, pricing is published at http://www.rackspace.com/cloud/servers/#pricing.

- For Rackspace Cloud Load Balancers, pricing is published at http://www.rackspace.com/cloud/load-balancing/pricing/.

- For Rackspace Cloud Monitoring, pricing is published at http://www.rackspace.com/cloud/monitoring/pricing/.

For the Early Access period, there is no Service Level Agreement (SLA) for Rackspace Autoscale.

2. Concepts

To use the Autoscale API effectively, you should understand several key concepts:

3. General API Information

The Autoscale API is implemented using a ReSTful web service interface. Like other products in the Rackspace Cloud suite, Autoscale shares a common token-based authentication system that allows seamless access between products and services.

Note

All requests to authenticate against and operate the service are performed using SSL over HTTP (HTTPS) on TCP port 443.

3.1. Authentication

Every ReST request for Autoscale requires the inclusion of a specific authorization token, supplied by the `X-Auth-Token` HTTP header. Customers obtain this token by first using the Rackspace Cloud Identity Service and supplying a valid username and API access key.

3.1.1. Geographic Endpoints

The Rackspace Cloud Identity Service serves as the entry point to all Rackspace Cloud APIs and is itself a ReSTful web service.

You can use either of the following endpoints to access the Identity Service, regardless of US or UK identities:

• https://identity.api.rackspacecloud.com/v2.0/
• https://lon.identity.api.rackspacecloud.com/v2.0/

Your account may be based in either the US or the UK; this is not determined by your physical location but by the location of the Rackspace retail site which was used to create your account:

• If your account was created via http://www.rackspacecloud.com, it is a US-based account.
• If your account was created via http://www.rackspace.co.uk, it is a UK-based account.

3.1.2. Retrieving the Authentication Token

POST	v2.0/tokens	Authenticate to receive a token and a service catalog.

Normal Response Code(s): 200, 203

Error Response Code(s): unauthorized (401), userDisabled (403), badRequest (400), authFault (500), serviceUnavailable (503)

The authenticate operation provides clients with an authentication token and a list of regional cloud endpoints. The sample requests and responses in this section illustrate a general case. In your authentication request, use your own credentials rather than the sample values shown here for `username` and `apiKey`. When you authenticate

successfully, the response to your authentication request will include a catalog of the services to which you have subscribed rather than the sample values shown here.

3.2. Service Access/Endpoints

Rackspace Autoscale is a regionalized service. The user of the service is therefore responsible for appropriate replication, caching, and overall maintenance of Rackspace Autoscale data across regional boundaries to other Cloud Queuing servers.

You can find the available service access/endpoints for Autoscale summarized in the table below.

Table 3.1. Regionalized Service Endpoints

Region	Endpoint
Chicago (ORD)	`https://ord.queuing.api.rackspacecloud.com/v1.0/1234/`
Dallas/Ft. Worth (DFW)	`https://dfw.queuing.api.rackspacecloud.com/v1.0/1234/`
London (LON)	`https://lon.queuing.api.rackspacecloud.com/v1.0/1234/`
Sydney (SYD)	`https://syd.queuing.api.rackspacecloud.com/v1.0/1234/`

Replace the sample project ID number, `1234`, with your actual account number returned as part of the authentication service response.

You will find the actual account number after the final '/' in the `publicURL` field returned by the authentication response.

Note

Rackspace Autoscale uses the term project ID rather than account ID, which is used in other Rackspace documentation. Both refer to your Rackspace account number, which you can find in your Cloud Control Panel at mycloud.rackspace.com.

3.3. Contract Versions

The Autoscale version defines the contract and build information for the API.

The contract version denotes the data model and behavior that the API supports. The requested contract version is included in all request URLs. Different contract versions of the API may be available at any given time and are not guaranteed to be compatible with one another.

Example 3.1. Sample Request URL (contract version in bold)

```
https://ord.queuing.api.rackspacecloud.com/v1.0/1234
```

Note

This document pertains to contract version 1.0.

3.4. Request/Response Types

The Rackspace Autoscale API supports JSON data serialization formats. The request format is specified using the `Content-Type` header and is *required* for calls that have a request body. The response format can be specified in requests either by using the `Accept` header or by adding a `.json` extension to the request URI. JSON is also used for the response format.

Table 3.2. Response Format

Format	Accept Header	Query Extension	Default
JSON	application/json	.json	Yes

3.5. Limits

All accounts, by default, have a preconfigured set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: *rate limits* and *absolute limits*. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed.

3.5.1. Rate Limits

Rate limits are specified in terms of both a human-readable wild-card URI and a machine-processable regular expression. The regular expression boundary matcher '^' takes effect after the root URI path. For example, the regular expression ^/v1.0/instances would match the bolded portion of the following URI: https://ord.databases.api.rackspacecloud.com/**v1.0/instances**.

The following table specifies the default rate limits for all API operations for all **GET**, **POST**, **PUT**, and **DELETE** calls for databases and database instances:

Table 3.3. Default Rate Limits

Verb	URI	RegEx	Default
GET changes-since	*/instances/*	^/vd+.d+/instances.*	3/minute
POST	*/instances/*	^/vd+.d+/instances.*	10/minute
POST instances	*/instances/*	^/vd+.d+/instances.*	50/day
PUT	*/instances/*	^/vd+.d+/instances.*	10/minute
DELETE	*/instances/*	^/vd+.d+/instances.*	100/minute

Rate limits are applied in order relative to the verb, going from least to most specific. For example, although the threshold for **POST** to /v1.0/* is 10 per minute, one cannot **POST** to /v1.0/* more than 50 times within a single day.

If you exceed the thresholds established for your account, a 413 (Rate Control) HTTP response will be returned with a `Retry-After` header to notify the client when it can attempt to try again.

3.5.2. Absolute Limits

Refer to the following table for the absolute limits that are set.

Table 3.4. Absolute Limits

Name	Description	Limit
	Maximum number of ??? allowed for your account	
	Maximum volume of ??? size per instance in gigabytes (GB) for your account	

3.6. Date/Time Format

The Autoscale service uses an ISO-8601 compliant date format for the display and consumption of date/time values.

Example 3.2. Service Date/Time Format

```
yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

See the table below for a description of the date/time format codes.

May 19th, 2011 at 8:07:08 AM, GMT-5 would have the following format:

```
2011-05-19T08:07:08-05:00
```

Table 3.5. Explanation of Date/Time Format Codes

Code	Description
yyyy	Four digit year
MM	Two digit month
dd	Two digit day of month
T	Separator for date/time
HH	Two digit hour of day (00-23)
mm	Two digit minutes of hour
ss	Two digit seconds of the minute
SSS	Three digit milliseconds of the second
Z	RFC-822 timezone

3.7. Faults

If any Rackspace Autoscale request results in an error, the queuing service returns an appropriate 4xx or 5xx HTTP status code, as well as the following information in the body:

• Title

• Description

• Internal code

• Link to more information

An example of an error message follows.

Example 3.3. Error Message Example

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "title": "Unsupported limit",
  "description": "The given limit cannot be negative, and cannot be greater
 than 50.",
  "code": 1092,
  "link": {
    "rel": "help",
    "href": "http://docs.example.com/messages#limit",
    "text": "API documentation for the limit parameter"
  }
}
```

Error information for each operation is included with the description in Chapter 4, "API Reference" [10].

4. API Reference

Verb	URI	Description
	Group Resource	
GET	v1.0/{tenantId}/groups	Lists all the autoscaling groups for the specified tenant ID.
POST	v1.0/{tenantId}/groups	Create a new scaling group, given the general scaling group configuration, launch configuration, and optional scaling policies. This data provided in the request body in JSON format. If successful, the created group in JSON format containing id and links is returned.
GET	v1.0/{tenantId}/groups/{groupId}	View manifested view of the scaling group configuration, including the launch configuration, and the scaling policies. This data is returned in the body of the response in JSON format.
DELETE	v1.0/{tenantId}/groups/{groupId}	Delete a scaling group if there are no entities belonging to the scaling group. If successful, no response body will be returned. If there are pending or active entities, a 409 will be returned.
GET	v1.0/{tenantId}/groups/{groupId}/state	Get the current state of the scaling group, including the current set of active entities, the number of pending entities, and the desired number of entities. This data is returned in the body of the response in JSON format.
	Configuration Resource	
GET	v1.0/{tenantId}/groups/{groupId}/config	Get the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data is returned in the body of the response in JSON format.
PUT	v1.0/{tenantId}/groups/{groupId}/config	Edit the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.
GET	v1.0/{tenantId}/groups/{groupId}/launch	Get the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data is returned in the body of the response in JSON format.
PUT	v1.0/{tenantId}/groups/{groupId}/launch	Edit the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.
POST	v1.0/{tenantId}/groups/{groupId}/pause	Pause all execution of scaling policies for this scaling group.
POST	v1.0/{tenantId}/groups/{groupId}/resume	Resume all execution of scaling policies for this scaling group.
	Policies Resource	
GET	v1.0/{tenantId}/groups/{groupId}/policies	Get a list of scaling policies in the group. Each policy describes an id, name, type, adjustment, cooldown, and links. This data is returned in the body of the response in JSON format.
POST	v1.0/{tenantId}/groups/{groupId}/policies	Create one or many new scaling policies. Scaling policies must include a name, type, adjustment, and cooldown. The response header will point to the list policies endpoint. An array of scaling policies is provided in the request body in JSON format.

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}`	Get a scaling policy which describes an id, name, type, adjustment, and cooldown, and links. This data is returned in the body of the response in JSON format.
PUT	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}`	Updates a scaling policy. Scaling policies must include a name, type, adjustment, and cooldown. If successful, no response body will be returned.
DELETE	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}`	Delete a scaling policy. If successful, no response body will be returned.
POST	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/execute`	Execute a scaling policy. If successful, a response body will be returned.
Execute Resource		
POST	`v1.0/execute/{capability_version}/{capability_hash}`	Execute a webhook
Webhooks Resource		
GET	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/webhooks`	Get a list of webhooks and their IDs. This data is returned in the body of the response in JSON format.
POST	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/webhooks`	Create one or more new scaling policy webhooks. Webhooks must have a name. If successful, the created response body will contain the IDs and links to the newly created webhooks. This data provided in the request body in JSON format.
GET	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/webhooks/{webhookId}`	Get information about a specific scaling policy webhook. This data is returned in the body of the response in JSON format.
PUT	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/webhooks/{webhookId}`	Update an existing webhook. WebhookIds not recognized will be ignored with accompanying data. URLs will be ignored if submitted, but that will not invalidate the request. If successful, no response body will be returned.
DELETE	`v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/webhooks/{webhookId}`	Delete a scaling policy webhook. If successful, no response body will be returned.

4.1. Group Resource

Verb	URI	Description
GET	`v1.0/{tenantId}/groups`	Lists all the autoscaling groups for the specified tenant ID.
POST	`v1.0/{tenantId}/groups`	Create a new scaling group, given the general scaling group configuration, launch configuration, and optional scaling policies. This data provided in the request body in JSON format. If successful, the created group in JSON format containing id and links is returned.
GET	`v1.0/{tenantId}/groups/{groupId}`	View manifested view of the scaling group configuration, including the launch configuration, and the scaling policies. This data is returned in the body of the response in JSON format.
DELETE	`v1.0/{tenantId}/groups/{groupId}`	Delete a scaling group if there are no entities belonging to the scaling group. If successful, no response body will be returned. If there are pending or active entities, a 409 will be returned.
GET	`v1.0/{tenantId}/groups/{groupId}/state`	Get the current state of the scaling group, including the current set of active entities, the number of pending entities, and the desired number of entities. This data is returned in the body of the response in JSON format.

4.1.1. List Autoscaling Groups

Verb	URI	Description
GET	`v1.0/{tenantId}/groups`	Lists all the autoscaling groups for the specified tenant ID.

Normal Response Code(s): 200

For the tenant specified by , this operation returns a list of autoscaling groups.

Table 4.1. List Autoscaling Groups Request Parameters

Name	Style	Type	Description
`tenantId`	Template		

Example 4.1. List Autoscaling Groups Request: JSON

Example 4.2. List Autoscaling Groups Response: JSON

```
{
                              "groups": [
                              {
                              "id": "{groupId1}"
                              "links": [
                              {
                              "href": "{url_root}/v1.0/010101/groups/{groupId1}"
                              "rel": "self"
                              }
                              ],
                              "active": [],
                              "activeCapacity": 0,
                              "pendingCapacity": 1,
                              "desiredCapacity": 1,
                              "paused": false
                              },
                              {
                              "id": "{groupId2}"
                              "links": [
                              {
                              "href": "{url_root}/v1.0/010101/groups/
{groupId2}",

                              "rel": "self"
                              }
                              ],
                              "active": [],
                              "activeCapacity": 0,
                              "pendingCapacity": 2,
                              "desiredCapacity": 2,
                              "paused": false
                              }
                              ],
                              "groups_links": []
                              }
```

4.1.2. POST

Verb	URI	Description
POST	`v1.0/{tenantId}/groups`	Create a new scaling group, given the general scaling group configuration, launch configuration, and optional scaling policies. This data provided in the request body in JSON format. If successful, the created group in JSON format containing id and links is returned.

Normal Response Code(s): 201

Table 4.2. POST Request Parameters

Name	Style	Type	Description
`tenantId`	Template		

Example 4.3. Request: JSON

```
{
    "groupConfiguration": {
        "name" : "workers"
        "cooldown": 60,
        "minEntities": 5,
        "maxEntities": 100,
        "metadata": {
            "firstkey" : "this is a string"
            "secondkey" : "1"
            }
        },
    "launchConfiguration": {
        "type" : "launch_server"
        "args": {
            "server": {
                "flavorRef": 3,
                "name" : "webhead"
                "imageRef" : "0d589460-f177-4b0f-81c1-8ab8903ac7d8"
                "OS-DCF:diskConfig" : "AUTO"
                "metadata": {
                    "mykey" : "myvalue"
                    },
                "personality": [
                {
                    "path": '/root/.ssh/authorized_keys',
                    "contents" : "ssh-rsa AAAAB3Nza...LiPk== user@example.net"

                    }
                ],
                "networks": [
                {
                    "uuid" : "11111111-1111-1111-1111-111111111111"
                    }
                ],
                },
            "loadBalancers": [
            {
                "loadBalancerId": 2200,
                "port": 8081
                }
```

```
            ]
        }
    },
    "scalingPolicies": [
    {
        "name" : "scale up by 10"
        "change": 10,
        "cooldown": 5,
        "type" : "webhook"
        },
    {
        "name" : "scale down by 5.5 percent"
        "changePercent": -5.5,
        "cooldown": 6,
        "type" : "webhook"
        },
    {
        "name" : "set number of servers to 10"
        "desiredCapacity": 10,
        "cooldown": 3,
        "type" : "webhook"
        }
    ]
}
```

Example 4.4. Response: JSON

```
{
    "group": {
        "id": "{groupId}",
        "links": [
          {
            "href": "{url_root}/v1.0/010101/groups/{groupId}"
            "rel": "self"
          }
        ],
        "groupConfiguration": {
            "name": "workers",
            "cooldown": 60,
            "minEntities": 5,
            "maxEntities": 100,
            "metadata": {
                "firstkey": "this is a string",
                "secondkey": "1",
            }
        },
        "launchConfiguration": {
            "type": "launch_server",
            "args": {
                "server": {
                    "flavorRef": 3,
                    "name": "webhead",
                    "imageRef": "0d589460-f177-4b0f-81c1-8ab8903ac7d8",
                    "OS-DCF:diskConfig": "AUTO",
                    "metadata": {
                        "mykey": "myvalue"
                    },
                    "personality": [
                        {
                            "path": '/root/.ssh/authorized_keys',
```

```
                                    "contents": "ssh-rsa AAAAB3Nza...LiPk==
user@example.net"
                            }
                        ],
                        "networks": [
                            {
                                "uuid": "11111111-1111-1111-1111-111111111111"
                            }
                        ],
                    },
                    "loadBalancers": [
                        {
                            "loadBalancerId": 2200,
                            "port": 8081
                        }
                    ]
                }
            },
            "scalingPolicies": [
                {
                    "id": "{policyId1}",
                    "links": [
                      {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId1}"
                        "rel": "self"
                      }
                    ],
                    "name": "scale up by 10",
                    "change": 10,
                    "cooldown": 5,
                    "type": "webhook"
                },
                {
                    "id": "{policyId2}",
                    "links": [
                      {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId2}"
                        "rel": "self"
                      }
                    ],
                    "name": "scale down by 5.5 percent",
                    "changePercent": -5.5,
                    "cooldown": 6,
                    "type": "webhook"
                },
                {
                    "id": "{policyId3}",
                    "links": [
                      {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId3}"
                        "rel": "self"
                      }
                    ],
                    "name": "Set number of servers to 10",
                    "desiredCapacity": 10,
                    "cooldown": 6,
                    "type": "webhook"
```

```
                }
            ]
        }
}
```

4.1.3. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}`	View manifested view of the scaling group configuration, including the launch configuration, and the scaling policies. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.3. GET Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.5. Request: JSON

Example 4.6. Response: JSON

```
{
    "group": {
        "id": "{groupId}",
        "links": [
            {
                "href": "{url_root}/v1.0/010101/groups/{groupId}"
                "rel": "self"
            }
        ],
        "groupConfiguration": {
            "name": "workers",
            "cooldown": 60,
            "minEntities": 5,
            "maxEntities": 100,
            "metadata": {
                "firstkey": "this is a string",
                "secondkey": "1",
            }
        },
        "launchConfiguration": {
            "type": "launch_server",
            "args": {
                "server": {
                    "flavorRef": 3,
                    "name": "webhead",
                    "imageRef": "0d589460-f177-4b0f-81c1-8ab8903ac7d8",
                    "OS-DCF:diskConfig": "AUTO",
                    "metadata": {
                        "mykey": "myvalue"
                    },
                    "personality": [
                        {
                            "path": '/root/.ssh/authorized_keys',
                            "contents": "ssh-rsa AAAAB3Nza...LiPk==
user@example.net"
                        }
                    ],
                    "networks": [
```

```
                                      {
                                          "uuid": "11111111-1111-1111-1111-111111111111"
                                      }
                                  ],
                          },
                          "loadBalancers": [
                              {
                                  "loadBalancerId": 2200,
                                  "port": 8081
                              }
                          ]
                      }
              },
          "scalingPolicies": [
              {
                  "id": "{policyId1}",
                  "links": [
                    {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId1}"
                        "rel": "self"
                    }
                  ],
                  "name": "scale up by 10",
                  "change": 10,
                  "cooldown": 5,
                  "type": "webhook"
              }
              {
                  "id": "{policyId2}",
                  "links": [
                    {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId2}"
                        "rel": "self"
                    }
                  ],
                  "name": "scale down by 5.5 percent",
                  "changePercent": -5.5,
                  "cooldown": 6,
                  "type": "webhook"
              },
              {
                  "id": "{policyId3}",
                  "links": [
                    {
                        "href": "{url_root}/v1.0/010101/groups/{groupId}/policies/
{policyId3}"
                        "rel": "self"
                    }
                  ],
                  "name": "set number of servers to 10",
                  "desiredCapacity": 10,
                  "cooldown": 3,
                  "type": "webhook"
              }
          ]
      }
}
```

4.1.4. DELETE

Verb	URI	Description
DELETE	`v1.0/{tenantId}/groups/{groupId}`	Delete a scaling group if there are no entities belonging to the scaling group. If successful, no response body will be returned. If there are pending or active entities, a 409 will be returned.

Normal Response Code(s): 204

Table 4.4. DELETE Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.7. Request: JSON

This operation does not return a response body.

4.1.5. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/state`	Get the current state of the scaling group, including the current set of active entities, the number of pending entities, and the desired number of entities. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.5. GET Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.8. Request: JSON

Example 4.9. Response: JSON

```
{
    "group": {
        "id": "{groupId}",
        "links": [
            {
                "href": "{url_root}/v1.0/010101/groups/{groupId},
                "rel": "self"
            }
        ],
        "active": [
            {
                "id": "{instanceId1}"
                "links": [
                    {
                        "href": "https://dfw.servers.api.rackspacecloud.com/
v2/010101/servers/{instanceId1}",
                        "rel": "self"
                    }
                ]
            },
            {
                "id": "{instanceId2}"
                "links": [
                    {
                        "href": "https://dfw.servers.api.rackspacecloud.com/
v2/010101/servers/{instanceId2}",
                        "rel": "self"
                    }
                ]
            }
        ],
        "activeCapacity": 2,
        "pendingCapacity": 2,
        "desiredCapacity": 4,
        "paused": false
    }
```

```
}
```

4.2. Configuration Resource

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/config`	Get the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data is returned in the body of the response in JSON format.
PUT	`v1.0/{tenantId}/groups/{groupId}/config`	Edit the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.
GET	`v1.0/{tenantId}/groups/{groupId}/launch`	Get the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data is returned in the body of the response in JSON format.
PUT	`v1.0/{tenantId}/groups/{groupId}/launch`	Edit the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.
POST	`v1.0/{tenantId}/groups/{groupId}/pause`	Pause all execution of scaling policies for this scaling group.
POST	`v1.0/{tenantId}/groups/{groupId}/resume`	Resume all execution of scaling policies for this scaling group.

4.2.1. GET

Verb	URI	Description
GET	v1.0/{tenantId}/groups/{groupId}/ config	Get the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.6. GET Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		

Example 4.10. Request: JSON

Example 4.11. Response: JSON

```
{
    "groupConfiguration": {
        "name": "workers",
        "cooldown": 60,
        "minEntities": 5,
        "maxEntities": 100,
        "metadata": {
            "firstkey": "this is a string",
            "secondkey": "1",
        }
    }
}
```

4.2.2. PUT

Verb	URI	Description
PUT	`v1.0/{tenantId}/groups/{groupId}/config`	Edit the configuration for a scaling group, which includes the minimum number of entities, the maximum number of entities, global cooldown, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.7. PUT Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		

Example 4.12. Request: JSON

```
{
    "name" : "workers"
    "cooldown": 60,
    "minEntities": 5,
    "maxEntities": 100,
    "metadata": {
        "firstkey" : "this is a string"
        "secondkey" : "1"
        }
}
```

This operation does not return a response body.

4.2.3. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/` `launch`	Get the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.8. GET Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.13. Request: JSON

Example 4.14. Response: JSON

```
{
    "launchConfiguration": {
        "type": "launch_server",
        "args": {
            "server": {
                "flavorRef": 3,
                "name": "webhead",
                "imageRef": "0d589460-f177-4b0f-81c1-8ab8903ac7d8",
                "OS-DCF:diskConfig": "AUTO",
                "metadata": {
                    "mykey": "myvalue"
                },
                "personality": [
                    {
                        "path": '/root/.ssh/authorized_keys',
                        "contents": "ssh-rsa AAAAB3Nza...LiPk== user@example.
net"
                    }
                ],
                "networks": [
                    {
                        "uuid": "11111111-1111-1111-1111-111111111111"
                    }
                ],
            },
            "loadBalancers": [
                {
                    "loadBalancerId": 2200,
                    "port": 8081
                }
            ]
        }
    }
}
```

4.2.4. PUT

Verb	URI	Description
PUT	`v1.0/{tenantId}/groups/{groupId}/` `launch`	Edit the launch configuration for a scaling group, which includes the details of how to create a server, from what image, which load balancers to join it to, and what networks to add it to, and other metadata. This data provided in the request body in JSON format. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.9. PUT Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.15. Request: JSON

```
{
    "type" : "launch_server"
    "args": {
        "server": {
            "flavorRef": 3,
            "name" : "webhead"
            "imageRef" : "0d589460-f177-4b0f-81c1-8ab8903ac7d8"
            "OS-DCF:diskConfig" : "AUTO"
            "metadata": {
                "mykey" : "myvalue"
                },
            "personality": [
            {
                "path": '/root/.ssh/authorized_keys',
                "contents" : "ssh-rsa AAAAB3Nza...LiPk== user@example.net"
                }
            ],
            "networks": [
            {
                "uuid" : "11111111-1111-1111-1111-111111111111"
                }
            ],
            },
        "loadBalancers": [
        {
            "loadBalancerId": 2200,
            "port": 8081
            }
        ]
        }
    }
}
```

This operation does not return a response body.

4.2.5. POST

Verb	URI	Description
POST	`v1.0/{tenantId}/groups/{groupId}/pause`	Pause all execution of scaling policies for this scaling group.

Normal Response Code(s): 204

Table 4.10. POST Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.16. Request: JSON

This operation does not return a response body.

4.2.6. POST

Verb	URI	Description
POST	v1.0/{tenantId}/groups/{groupId}/resume	Resume all execution of scaling policies for this scaling group.

Normal Response Code(s): 204

Table 4.11. POST Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		

Example 4.17. Request: JSON

This operation does not return a response body.

4.3. Policies Resource

Verb	URI	Description
GET	v1.0/{tenantId}/groups/{groupId}/policies	Get a list of scaling policies in the group. Each policy describes an id, name, type, adjustment, cooldown, and links. This data is returned in the body of the response in JSON format.
POST	v1.0/{tenantId}/groups/{groupId}/policies	Create one or many new scaling policies. Scaling policies must include a name, type, adjustment, and cooldown. The response header will point to the list policies endpoint. An array of scaling policies is provided in the request body in JSON format.
GET	v1.0/{tenantId}/groups/{groupId}/policies/{policyId}	Get a scaling policy which describes an id, name, type, adjustment, and cooldown, and links. This data is returned in the body of the response in JSON format.
PUT	v1.0/{tenantId}/groups/{groupId}/policies/{policyId}	Updates a scaling policy. Scaling policies must include a name, type, adjustment, and cooldown. If successful, no response body will be returned.
DELETE	v1.0/{tenantId}/groups/{groupId}/policies/{policyId}	Delete a scaling policy. If successful, no response body will be returned.
POST	v1.0/{tenantId}/groups/{groupId}/policies/{policyId}/execute	Execute a scaling policy. If successful, a response body will be returned.

4.3.1. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/policies`	Get a list of scaling policies in the group. Each policy describes an id, name, type, adjustment, cooldown, and links. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.12. GET Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.18. Request: JSON

Example 4.19. Response: JSON

```
{
    "policies": [
        {
            "id":"{policyId1}",
            "data": {
                "name": "scale up by one server",
                "change": 1,
                "cooldown": 150,
                "type": "webhook"
            },
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId1}/policy/
{policyId1}"
                    "rel": "self"
                }
            ]
        },
        {
            "id": "{policyId2}",
            "data": {
                "name": "scale up ten percent",
                "changePercent": 10,
                "cooldown": 150,
                "type": "webhook"
            },
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId1}/policy/
{policyId2}"
                    "rel": "self"
                }
            ]
        },
        {
            "id":"{policyId3}",
```

```
            "data": {
                "name": "scale down one server",
                "change": -1,
                "cooldown": 150,
                "type": "webhook"
            },
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId1}/policy/
{policyId3}"
                    "rel": "self"
                }
            ]
        },
        {
            "id": "{policyId4}",
            "data": {
                "name": "scale down ten percent",
                "changePercent": -10,
                "cooldown": 150,
                "type": "webhook"
            },
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId1}/policy/
{policyId4}"
                    "rel": "self"
                }
            ]
        }
    ]
}
```

4.3.2. POST

Verb	URI	Description
POST	`v1.0/{tenantId}/groups/{groupId}/policies`	Create one or many new scaling policies. Scaling policies must include a name, type, adjustment, and cooldown. The response header will point to the list policies endpoint. An array of scaling policies is provided in the request body in JSON format.

Normal Response Code(s): 201

Table 4.13. POST Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		

Example 4.20. Request: JSON

```
[
{
    "name" : "scale up by one server"
    "change": 1,
    "cooldown": 150,
    "type" : "webhook"
    },
{
    "name" : "scale down by 5.5 percent"
    "changePercent": -5.5,
    "cooldown": 6,
    "type" : "webhook"
    }
]
```

Example 4.21. Response: JSON

```
{
    "policies": [
        {
            "id": {policyId1},
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId}/policy/
{policyId1}"
                    "rel": "self"
                }
            ],
            "name": "scale up by one server",
            "change": 1,
            "cooldown": 150,
            "type": "webhook"
        },
        {
            "id": {policyId2},
            "links": [
                {
                    "href": "{url_root}/v1.0/010101/groups/{groupId}/policy/
{policyId2}"
```

```
                    "rel": "self"
                }
            ],
            "name": "scale down by 5.5 percent",
            "changePercent": -5.5,
            "cooldown": 6,
            "type": "webhook"
        }
    ]
}
```

4.3.3. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}`	Get a scaling policy which describes an id, name, type, adjustment, and cooldown, and links. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.14. GET Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		

Example 4.22. Request: JSON

Example 4.23. Response: JSON

```
{
    "policy": {
        "id": {policyId},
        "links": [
            {
                "href": "{url_root}/v1.0/010101/groups/{groupId}/policy/
{policyId}"
                "rel": "self"
            }
        ],
        "name": "scale up by one server",
        "change": 1,
        "cooldown": 150,
        "type": "webhook"
    }
}
```

4.3.4. PUT

Verb	URI	Description
PUT	`v1.0/{tenantId}/groups/{groupId}/ policies/{policyId}`	Updates a scaling policy. Scaling policies must include a name, type, adjustment, and cooldown. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.15. PUT Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		

Example 4.24. Request: JSON

```
{
    "name": 'scale down by 5.5 percent',
    "changePercent": -5.5,
    "cooldown": 6,
    "type" : "webhook"
}
```

This operation does not return a response body.

4.3.5. DELETE

Verb	URI	Description
DELETE	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}`	Delete a scaling policy. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.16. DELETE Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		

Example 4.25. Request: JSON

This operation does not return a response body.

4.3.6. POST

Verb	URI	Description
POST	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}/execute`	Execute a scaling policy. If successful, a response body will be returned.

Normal Response Code(s): 202

Table 4.17. POST Request Parameters

Name	Style	Type	Description
`tenantId`	Template		
`groupId`	Template		
`policyId`	Template		

Example 4.26. Request: JSON

Example 4.27. Response: JSON

```
{}
```

4.4. Execute Resource

Verb	URI	Description
POST	`v1.0/execute/{capability_version}/` `{capability_hash}`	Execute a webhook

4.4.1. POST

Verb	URI	Description
POST	`v1.0/execute/{capability_version}/` `{capability_hash}`	Execute a webhook

Normal Response Code(s): 202

Table 4.18. POST Request Parameters

Name	Style	Type	Description
`capability_version`	Template		
`capability_hash`	Template		

Example 4.28. Request: JSON

Example 4.29. Response: JSON

```
{}
```

4.5. Webhooks Resource

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}/webhooks`	Get a list of webhooks and their IDs. This data is returned in the body of the response in JSON format.
POST	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}/webhooks`	Create one or more new scaling policy webhooks. Webhooks must have a name. If successful, the created response body will contain the IDs and links to the newly created webhooks. This data provided in the request body in JSON format.

GET	`v1.0/{tenantId}/groups/{groupId}/`
	`policies/{policyId}/webhooks/`
	`{webhookId}`
		Get information about a specific scaling policy webhook. This data is returned in the body of the response in JSON format.

PUT	`v1.0/{tenantId}/groups/{groupId}/`
	`policies/{policyId}/webhooks/`
	`{webhookId}`
		Update an existing webhook. WebhookIds not recognized will be ignored with accompanying data. URLs will be ignored if submitted, but that will not invalidate the request. If successful, no response body will be returned.

DELETE	`v1.0/{tenantId}/groups/{groupId}/`
	`policies/{policyId}/webhooks/`
	`{webhookId}`
		Delete a scaling policy webhook. If successful, no response body will be returned.

4.5.1. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}/webhooks`	Get a list of webhooks and their IDs. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.19. GET Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		

Example 4.30. Request: JSON

Example 4.31. Response: JSON

```
{
    "webhooks": [
        {
            "id":"{webhookId1}",
            "name": "alice",
            "metadata": {
                "notes": "this is for Alice"
            },
            "links": [
                {
                    "href": ".../{groupId1}/policies/{policyId1}/webhooks/
{webhookId1}/",
                    "rel": "self"
                },
                {
                    "href": ".../execute/1/{capability_hash1}/,
                    "rel": "capability"
                }
            ]
        },
        {
            "id":"{webhookId2}",
            "name": "alice",
            "metadata": {
                "notes": "this is for Bob"
            },
            "links": [
                {
                    "href": ".../{groupId1}/policies/{policyId1}/webhooks/
{webhookId2}/",
                    "rel": "self"
                },
                {
                    "href": ".../execute/1/{capability_hash2}/,
                    "rel": "capability"
                }
            ]
```

```
            }
    ],
    "webhooks_links": []
}
```

4.5.2. POST

Verb	URI	Description
POST	v1.0/{tenantId}/groups/{groupId}/ policies/{policyId}/webhooks	Create one or more new scaling policy webhooks. Webhooks must have a name. If successful, the created response body will contain the IDs and links to the newly created webhooks. This data provided in the request body in JSON format.

Normal Response Code(s): 201

Table 4.20. POST Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		

Example 4.32. Request: JSON

```
[
{
    "name" : "alice"
    "metadata": {
        "notes" : "this is for Alice"
        }
    },
{
    "name" : "bob"
    }
]
```

Example 4.33. Response: JSON

```
{
    "webhooks": [
        {
            "id":"{webhookId1}",
            "alice",
            "metadata": {
                "notes": "this is for Alice"
            },
            "links": [
                {
                    "href": ".../{groupId1}/policies/{policyId1}/webhooks/
{webhookId1}/",
                    "rel": "self"
                },
                {
                    "href": ".../execute/1/{capability_hash1}/,
                    "rel": "capability"
                }
            ]
        },
        {
            "id":"{webhookId2}",
            "name": "bob",
```

```
            "metadata": {},
            "links": [
                {
                    "href": ".../{groupId1}/policies/{policyId1}/webhooks/
{webhookId2}/",
                    "rel": "self"
                },
                {
                    "href": ".../execute/1/{capability_hash2}/,
                    "rel": "capability"
                }
            ]
        }
    ]
}
```

4.5.3. GET

Verb	URI	Description
GET	`v1.0/{tenantId}/groups/{groupId}/` `policies/{policyId}/webhooks/` `{webhookId}`	Get information about a specific scaling policy webhook. This data is returned in the body of the response in JSON format.

Normal Response Code(s): 200

Table 4.21. GET Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		
webhookId	Template		

Example 4.34. Request: JSON

Example 4.35. Response: JSON

```
{
    "webhook": {
        "id":"{webhookId}",
        "name": "webhook name",
        "metadata": {},
        "links": [
            {
                "href": ".../{groupId1}/policies/{policyId1}/webhooks/
{webhookId}/",
                "rel": "self"
            },
            {
                "href": ".../execute/1/{capability_hash2},
                "rel": "capability"
            }
        ]
    }
}
```

4.5.4. PUT

Verb	URI	Description
PUT	v1.0/{tenantId}/groups/{groupId}/ policies/{policyId}/webhooks/ {webhookId}	Update an existing webhook. WebhookIds not recognized will be ignored with accompanying data. URLs will be ignored if submitted, but that will not invalidate the request. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.22. PUT Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		
webhookId	Template		

Example 4.36. Request: JSON

```
{
    "name" : "alice"
    "metadata": {
        "notes" : "this is for Alice"
        }
}
```

This operation does not return a response body.

4.5.5. DELETE

Verb	URI	Description
DELETE	v1.0/{tenantId}/groups/{groupId}/ policies/{policyId}/webhooks/ {webhookId}	Delete a scaling policy webhook. If successful, no response body will be returned.

Normal Response Code(s): 204

Table 4.23. DELETE Request Parameters

Name	Style	Type	Description
tenantId	Template		
groupId	Template		
policyId	Template		
webhookId	Template		

Example 4.37. Request: JSON

This operation does not return a response body.