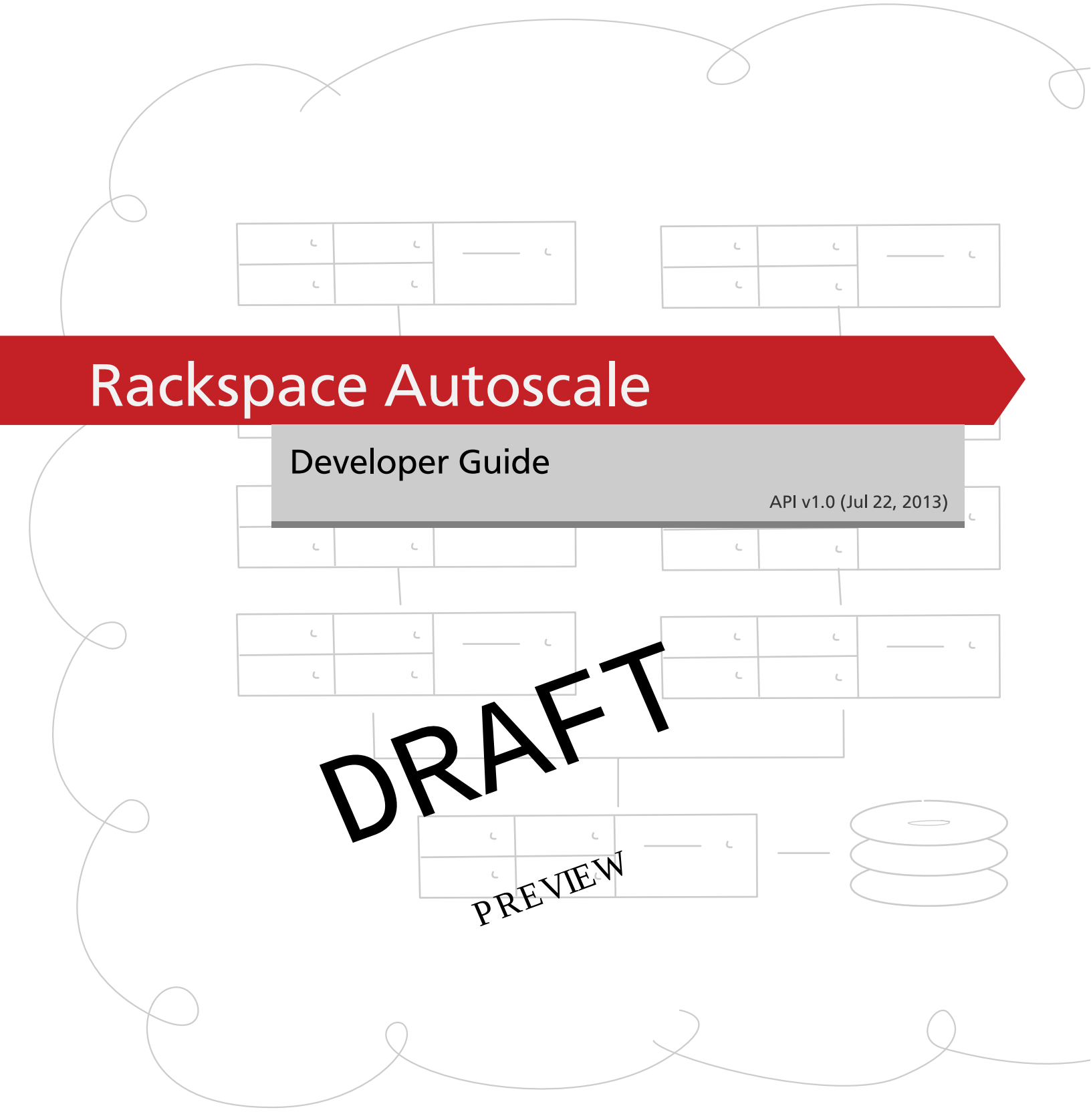
Rackspace Autoscale

Developer Guide

API v1.0 (Jul 22, 2013)



rackspace®
the open cloud company

docs.rackspace.com/api

Rackspace Autoscale Developer Guide

API v1.0 (2013-07-22)

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 chapter 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.

Example 3.1. Authentication Request for US Endpoint: XML

```
POST /v2.0/tokens HTTP/1.1
User-Agent: curl/7.21.0 (x86_64-pc-linux-gnu) libcurl/7.21.0 OpenSSL/0.9.8o
 zlib/1.2.3.4 libidn/1.15 libssh2/1.2.6
Host: identity.api.rackspacecloud.com
Accept: application/xml
Content-Type: application/xml
Content-Length: 88

<?xml version="1.0" encoding="UTF-8"?>
<auth>
   <apiKeyCredentials
       xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
       username=❶"jsmith"
       apiKey=❷"aaaaa-bbbbb-ccccc-12345678"/>
</auth>
```

Example 3.2. Authentication Request for US Endpoint: JSON

```
POST /v2.0/tokens HTTP/1.1
User-Agent: curl/7.21.0 (x86_64-pc-linux-gnu) libcurl/7.21.0 OpenSSL/0.9.8o
 zlib/1.2.3.4 libidn/1.15 libssh2/1.2.6
Host: identity.api.rackspacecloud.com
Accept: application/json
Content-Type: application/json
Content-Length: 54

{
   "auth":
   {
      "RAX-KSKEY:apiKeyCredentials":
      {
         "username": ❶"jsmith",
         "apiKey": ❷"aaaaa-bbbbb-ccccc-12345678"
      }
   }
}
```

❶ The username supplied here is your common Rackspace Cloud username.

❷ The key is your API access key. The key can be obtained from the Rackspace Cloud Control Panel in the **<Your Account>/API Access** section (login here: Control Panel Login).

Example 3.3. Auth Response for US Endpoint: XML

```
<HTTP/1.1 200 OK
    Content-Type: application/xml; charset=UTF-8
    Content-Length: 477
    Date: Thu, 12 Apr 2012 18:50:20 GMT

    <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
    ❶<access xmlns:os-ksadm="http://docs.openstack.org/identity/api/ext/OS-
KSADM/v1.0"
        xmlns="http://docs.openstack.org/identity/api/v2.0"
        xmlns:rax-kskey="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/
v1.0"
        xmlns:rax-ksqa="http://docs.rackspace.com/identity/api/ext/RAX-KSQA/
v1.0"
        xmlns:common="http://docs.openstack.org/common/api/v1.0"
        xmlns:ksgrp="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.
0"
        xmlns:rax-kscatalog="http://docs.openstack.org/identity/api/ext/OS-
KSCATALOG/v1.0"
        xmlns:atom="http://www.w3.org/2005/Atom">
        ❷<token id="vvvvvvvv-wwww-xxxx-yyyy-zzzzzzzzzzzz" expires=
"2011-12-08T22:51:02.000-06:00"/>
        <user id="123456" name="jsmit❸h" rax-auth:defaultRegion="DFW">
            ❹<roles>
                <role id="identity:admin" name="identity:admin" description=
"Admin Role."/>
                <role id="identity:default" name="identity:default"
 description="Default Role."/>
            </roles>
        </user>
        ❺<serviceCatalog>
            <service type="rax:database" name="cloudDatabases">
                <endpoint region="DFW" tenantId="1100111" publicURL="https://
dfw.databases.api.rackspacecloud.com/v1.0/1100111"/>
                <endpoint region="ORD" tenantId="1100111" publicURL="https://
ord.databases.api.rackspacecloud.com/v1.0/1100111"/>
            </service>
            <service type="rax:load-balancer" name="cloudLoadBalancers">
                <endpoint region="DFW" tenantId="1100111" publicURL="https://
dfw.loadbalancers.api.rackspacecloud.com/v1.0/1100111"/>
                <endpoint region="ORD" tenantId="1100111" publicURL="https://
ord.loadbalancers.api.rackspacecloud.com/v1.0/1100111"/>
            </service>
            <service type="compute" name="cloudServersOpenStack">
                <endpoint region="DFW" tenantId="1100111"
                    publicURL="https://dfw.servers.api.rackspacecloud.com/v2/
1100111">
                    <version id="2" info="https://dfw.servers.api.
rackspacecloud.com/v2/"
                        list="https://dfw.servers.api.rackspacecloud.com/" />
                </endpoint>
                <endpoint region="ORD" tenantId="1100111"
                    publicURL="https://ord.servers.api.rackspacecloud.com/v2/
1100111">
                    <version id="2" info="https://ord.servers.api.
rackspacecloud.com/v2/"
                        list="https://ord.servers.api.rackspacecloud.com/" />
                </endpoint>
            </service>
            <service type="compute" name="cloudServers">
                <endpoint tenantId="1100111"
                    publicURL="https://servers.api.rackspacecloud.com/v1.0/
1100111">
                    <version id="1.0"
                        info="https://servers.api.rackspacecloud.com/v1.0/"
                        list="https://servers.api.rackspacecloud.com/"/>
```

```
                </endpoint>
            </service>
            <service ❻type="object-store"❼ name="cloudFiles">
                <endpoint ❽region="DFW"
                    ❾tenantId="MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    ❿publicURL="https://storage101.dfw1.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    internalURL="https://snet-storage101.dfw1.clouddrive.com/
v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"/>
                <endpoint region="ORD"
                    tenantId="MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    publicURL="https://storage101.ord1.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    internalURL="https://snet-storage101.ord1.clouddrive.com/
v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"/>
            </service>
            <service type="rax:object-cdn" name="cloudFilesCDN">
                <endpoint region="DFW"
                    tenantId="MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    publicURL="https://cdn1.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"/>
                <endpoint region="ORD"
                    tenantId="MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
                    publicURL="https://cdn2.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"/>
            </service>
            <service type="rax:dns" name="cloudDNS">
                <endpoint tenantId="1100111"
                    publicURL="https://dns.api.rackspacecloud.com/v1.0/
1100111"/>
            </service>
        </serviceCatalog>
    </access>
```

Example 3.4. Auth Response for US Endpoint: JSON

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 477
Date: Thu, 12 Apr 2012 18:45:13 GMT

{
    "access": {

       ❷ "token": {
            "expires": "2011-12-08T22:51:02.000-06:00",
            "id": "vvvvvvvv-wwww-xxxx-yyyy-zzzzzzzzzzzz"
        },
        "user": {
            "id": "123456",
            "name": "jsmith",
            "RAX-AUTH:defaultRegion": "DFW",❸
          ❹ "roles": [
                {
                    "description": "Admin Role.",
                    "id": "identity:admin",
                    "name": "identity:admin"
                },
                {
```

```
                    "description": "Default Role.",
                    "id": "identity:default",
                    "name": "identity:default"
                }
            ]
        },
      ❺ "serviceCatalog": [
            {
                "endpoints": [
                    {
                        "publicURL": "https://dfw.databases.api.
rackspacecloud.com/v1.0/1100111",
                        "region": "DFW",
                        "tenantId": "1100111"
                    },
                    {
                        "publicURL": "https://ord.databases.api.
rackspacecloud.com/v1.0/1100111",
                        "region": "ORD",
                        "tenantId": "1100111"
                    }
                ],
                "name": "cloudDatabases",
                "type": "rax:database"
            },
            {
                "endpoints": [
                    {
                        "publicURL": "https://dfw.loadbalancers.api.
rackspacecloud.com/v1.0/1100111",
                        "region": "DFW",
                        "tenantId": "1100111"
                    },
                    {
                        "publicURL": "https://ord.loadbalancers.api.
rackspacecloud.com/v1.0/1100111",
                        "region": "ORD",
                        "tenantId": "1100111"
                    }
                ],
                "name": "cloudLoadBalancers",
                "type": "rax:load-balancer"
            },
            {
                "endpoints": [
                    {
                        "tenantId": "1100111",
                        "region": "DFW",
                        "publicURL": "https://dfw.servers.api.rackspacecloud.
com/v2/1100111",
                        "versionId": "2",
                        "versionInfo": "https://dfw.servers.api.
rackspacecloud.com/v2/",
                        "versionList": "https://dfw.servers.api.
rackspacecloud.com/"
                    },
                    {
                        "tenantId": "1100111",
                        "region": "ORD",
```

```
                        "publicURL": "https://ord.servers.api.rackspacecloud.
com/v2/1100111",
                        "versionId": "2",
                        "versionInfo": "https://ord.servers.api.
rackspacecloud.com/v2/",
                        "versionList": "https://ord.servers.api.
rackspacecloud.com/"
                    }
                ],
                "name": "cloudServersOpenStack",
                "type": "compute"
            },
            {
                "endpoints": [
                    {
                        "tenantId": "1100111",
                        "publicURL": "https://servers.api.rackspacecloud.com/
v1.0/1100111",
                        "versionId": "1.0",
                        "versionInfo": "https://servers.api.rackspacecloud.
com/v1.0/",
                        "versionList": "https://servers.api.rackspacecloud.
com/"
                    }
                ],
                "name": "cloudServers",
                "type": "compute"
            },
            {
                "endpoints": [
                    {
                        ❾ "tenantId": "MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-
eeeeeeee",
                        ❿ "publicURL": "https://storage101.dfw1.clouddrive.com/
v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
                        "internalURL": "https://snet-storage101.dfw1.
clouddrive.com/v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
                        ❽ "region": "DFW"
                    },
                    {
                        "tenantId": "MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-
eeeeeeee",
                        "publicURL": "https://storage101.ord1.clouddrive.com/
v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
                        "internalURL": "https://snet-storage101.ord1.
clouddrive.com/v1/MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
                        "region": "ORD"
                    }
                ],
                ❼ "name": "cloudFiles",
                ❻ "type": "object-store"
            },
            {
                "endpoints": [
                    {
                        "tenantId": "MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-
eeeeeeee",
                        "publicURL": "https://cdn1.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
```

```
                        "region": "DFW"
                    },
                    {
                        "tenantId": "MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-
eeeeeeee",
                        "publicURL": "https://cdn2.clouddrive.com/v1/
MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee",
                        "region": "ORD"
                    }
                ],
                "name": "cloudFilesCDN",
                "type": "rax:object-cdn"
            },
            {
                "endpoints": [
                    {
                        "tenantId": "1100111",
                        "publicURL": "https://dns.api.rackspacecloud.com/v1.0/
1100111"
                    }
                ],
                "name": "cloudDNS",
                "type": "rax:dns"
            }
        ]
    }
}
```

Note

The information shown in the Auth Response examples is for US-based accounts. If you authenticate against the UK-endpoint for auth, you will see the service catalog information for UK-based accounts.

❶ In XML responses only, a list of namespaces identifies API extensions that add functionality to the core API.

❷ This token can be presented to a service as evidence of authentication. Tokens are valid for a finite duration; a token's default lifespan is twenty-four hours.

The token's `expires` attribute denotes the time after which the token will automatically become invalid. A token may be manually revoked before the time identified by the `expires` attribute; `expires` predicts a token's maximum possible lifespan but does not guarantee that it will reach that lifespan. Clients are encouraged to cache a token until it expires.

Note

The token's expiration time is formatted differently in the US and UK. These response examples show the US format. For examples of the UK format please refer to http://docs.rackspace.com/auth/api/v2.0/auth-client-devguide/content/POST_authenticate_v2.0_tokens_.html.

❸ Users can be assigned a default region so that, when there is a choice between multiple endpoints associated with a service in the user's catalog, the endpoint for the user's default region will be selected if it is available. In this example, the user's default region is DFW and several of the services in the user's catalog offer endpoints in that region and the ORD region; this user's work will be directed to the DFW region whenever possible.

❹ Users can be assigned multiple roles, with each role providing specific privileges. In this example, `jsmith` is the administrative user for the account, holding the fully-privileged `identity:admin` role. Other users might hold other roles with different privileges. Roles need not be associated with actual job functions such as Administrator, Operator, Developer, Tester, or Trainer.

❺ The service catalog lists the services this user can access. In this example, the user can access one database service, one loadbalancing service, two compute services (Cloud Servers OpenStack and Cloud Servers), two object storage services (Cloud Files Content Distribution Network (CDN), and Cloud Files), and one DNS service. The catalog listing for each service provides at least one endpoint URL for that service. Other information, such as regions, versions, and tenants, is provided if it's relevant to this user's access to this service.

❻ The service type attribute identifies services that perform similar functions, whatever those services might be named. In this example, the services named cloudServers and cloudServersOpenstack are both identified as `type="compute"`, identifying them as compute services even though the word "compute" does not appear in their names.

Important

Use service type as the primary value for locating a service. If multiple endpoints of the same service type exist in the same region, use service name as the tiebreaker.

❼ The service name attribute identifies each unique service in the catalog. Once a service is created, its name does not change. However, new services of the same service type may be added to the catalog with new names.

Important

If you are programmatically parsing an authentication response, use service type rather than service name as the basis for determining whether a user has access to a particular kind of service. Service type is stable across all releases; new service types may be developed, but existing service types are not renamed. In this example, `type="compute"` identifies all the available compute services, one of which is named cloudServers and one of which is named cloudServersOpenStack. New compute service names may be added in future releases; whatever the compute services are named, you can always recognize them by parsing for `type="compute"` in the authentication response's service catalog.

❽ A service may expose endpoints in different regions. Regional endpoints allow clients to provision resources in a manner that provides high availability.

Some services are not region-specific. These services supply a single non-regional endpoint and do not provide access to internal URLs.

9. Some services recognize specification of a tenant. If a service does recognize tenants, the format of the tenant specification is defined only by the service; for details about whether and how to specify a tenant, check the documentation for the service you are using.

10. An endpoint can be assigned public and internal URLs. A public URL is accessible from anywhere. Access to a public URL usually incurs traffic charges. Internal URLs are only accessible to services within the same region. Access to an internal URL is free of charge.

Authentication tokens are typically valid for 24 hours. Applications should be designed to re-authenticate after receiving a 401 (Unauthorized) response from a service endpoint.

Important

If you are programmatically parsing an authentication response, please be aware that service names are stable for the life of the particular service and can be used as keys. You should also be aware that a user's service catalog can include multiple uniquely-named services which perform similar functions. For example, cloudServersOpenStack is the OpenStack version of compute whereas cloudServers is the legacy version of compute; the same user can have access to both services. In Auth 2.0, the service type attribute can be used as a key by which to recognize similar services; see the tip below.

Tip

Beginning with Auth 2.0, the service catalog includes a service type attribute to identify services that perform similar functions but have different names; for example, `type="compute"` identifies compute services such as cloudServers and cloudServersOpenStack. Some developers have found the service type attribute to be useful in parsing the service catalog. For additional information on Auth 2.0 (also known as the Cloud Identity Service), refer to the *Cloud Identity Client Developer Guide* at http://docs.rackspace.com/.

Databases service endpoints are published in the service catalog in the Auth response with the account number, which is a required element of the service endpoints. The examples shown here are for authentication for US customers. Customers with UK-based accounts will see different values in the service catalog. Refer to the next section for more information about service endpoints.

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 3.5. 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 3.6. 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 3.7. 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" [18].

4. API Reference

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

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. Configurations

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

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. Executions

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

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.