# Rackspace Cloud Big Data

## Getting Started Guide

API v1.0 Early Access (Sep 16, 2013)

Early Access

rackspace®
the open cloud company

docs.rackspace.com/api

# Rackspace Cloud Big Data Getting Started Guide

API v1.0 Early Access (2013-09-16)

Copyright © 2013 Rackspace US, Inc. All rights reserved.

This document is intended for software developers interested in developing service management applications using the Rackspace Cloud Big Data Application Programming Interface (API). The document is for informational purposes only and is provided “AS IS.”

RACKSPACE MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS DOCUMENT AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT/SERVICES DESCRIPTION AT ANY TIME WITHOUT NOTICE. RACKSPACE SERVICES OFFERINGS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS MUST TAKE FULL RESPONSIBILITY FOR APPLICATION OF ANY SERVICES MENTIONED HEREIN. EXCEPT AS SET FORTH IN RACKSPACE GENERAL TERMS AND CONDITIONS AND/OR CLOUD TERMS OF SERVICE, RACKSPACE ASSUMES NO LIABILITY WHATSOEVER, AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO ITS SERVICES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT.

Except as expressly provided in any written license agreement from Rackspace, the furnishing of this document does not give you any license to patents, trademarks, copyrights, or other intellectual property.

Rackspace®, Rackspace logo and Fanatical Support® are registered service marks of Rackspace US, Inc. All other product names and trademarks used in this document are for identification purposes only and are property of their respective owners.



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# Table of Contents

1. Document Change History ..... 1
2. Overview ..... 2
  2.1. Early Access Program ..... 2
  2.2. Prerequisites for Running Examples ..... 3
3. Service Endpoints ..... 4
4. Using the API Directly ..... 5
  4.1. Sending Requests to the API ..... 5
    4.1.1. Sending API Requests by Using cURL ..... 5
  4.2. Generating an Authentication Token ..... 7
  4.3. Setting Up and Viewing a Profile ..... 13
  4.4. Viewing Resource Limits ..... 16
  4.5. Creating and Managing Hadoop Clusters ..... 17
    4.5.1. Listing Flavors ..... 17
    4.5.2. Creating a Cluster ..... 18
    4.5.3. Viewing Details ..... 19
    4.5.4. Resizing Clusters ..... 22
    4.5.5. Deleting Clusters ..... 23
5. Using the Python Lava Client ..... 25
  5.1. Setting Up ..... 25
    5.1.1. Prerequisites ..... 25
    5.1.2. Installing the Client ..... 25
    5.1.3. Authenticating Your Session ..... 25
    5.1.4. Setting Up a Profile ..... 26
    5.1.5. Viewing Resource Limits ..... 27
  5.2. Creating and Managing Hadoop Clusters ..... 27
    5.2.1. Provisioning a Hadoop Cluster ..... 27
    5.2.2. Viewing Details ..... 28
    5.2.3. Resizing Clusters ..... 29
    5.2.4. Deleting Clusters ..... 30
6. Additional Resources ..... 31



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# List of Tables

4.1. cURL Command-Line Options ..... 5

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# List of Examples

4.1. cURL Authenticate Request: XML ..... 6
4.2. cURL Authenticate Request: XML ..... 8
4.3. cURL Authenticate Request: JSON ..... 8
4.4. Authenticate Response: XML ..... 9
4.5. Authenticate Response: JSON ..... 10
4.6. cURL Create Profile Request: JSON ..... 14
4.7. Create Profile Response: JSON ..... 14
4.8. cURL Get Profile Request: JSON ..... 15
4.9. Get Profile Response: JSON ..... 15
4.10. cURL View Resource Limits Request: JSON ..... 16
4.11. View Resource Limits Response: JSON ..... 16
4.12. cURL List Flavors Request: JSON ..... 17
4.13. List Flavors Response: JSON ..... 17
4.14. cURL Create Cluster Request: JSON ..... 19
4.15. Create Cluster Response: JSON ..... 19
4.16. cURL List Cluster Nodes Request: JSON ..... 20
4.17. List Cluster Nodes Response: JSON ..... 20
4.18. cURL Resize Cluster Request: JSON ..... 22
4.19. Resize Cluster Response: JSON ..... 22
4.20. cURL Delete Cluster Request: JSON ..... 23
4.21. Delete Cluster Response: JSON ..... 23
5.1. Authentication Response Using CLI Utility ..... 26
5.2. Export Environment Variables ..... 26
5.3. Creating a Profile ..... 26
5.4. Retrieve a Profile by Using the get-profile Command ..... 27
5.5. View Resource Limits by Using the quotas Command ..... 27
5.6. Provision a Cluster by Using the boot Command ..... 27
5.7. List Flavors and Associated Resources by Using the flavor-list Command ..... 28
5.8. Query the Status of a Cluster by Using the list Command ..... 28
5.9. Query the Details of a Cluster by Using the show and nodes Commands ..... 28
5.10. Increase Cluster Size by Using the resize Command ..... 29
5.11. Remove Clusters by Using the delete Command ..... 30



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# 1. Document Change History

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

| Revision Date | Summary of Changes |
|---|---|
| Sep 16, 2013 | • Initial Early Access (EA) release. |

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# 2. Overview

Rackspace Cloud Big Data is an on-demand Apache Hadoop service for the Rackspace open cloud. The service supports a RESTful API and alleviates the pain associated with deploying, managing, and scaling Hadoop clusters.

Cloud Big Data is just as flexible and feature-rich as Hadoop. With Cloud Big Data, you benefit from on-demand servers, utility-based pricing, and access to the full set of Hadoop features and APIs. However, you do not have to worry about provisioning, growing, or maintaining your Hadoop infrastructure. The Cloud Big Data service uses an environment that is specifically optimized for Hadoop, which ensures that your jobs run efficiently and reliably. Note that you are still responsible for developing, troubleshooting, and deploying your applications.

The primary use cases for Cloud Big Data are as follows:

- Create on-demand infrastructure for applications in production where physical servers would be too costly and time-consuming to configure and maintain.
- Develop, test, and pilot data analysis applications.

Cloud Big Data provides the following benefits:

- Create or resize Hadoop clusters in minutes and pay only for what you use.
- Access Hortonworks Data Platform (HDP), an enterprise-ready distribution that is 100 percent Apache open source.
- Provision and manage Hadoop through an easy-to-use Control Panel and a RESTful API.
- Seamlessly access data in Cloud Files containers.
- Gain interoperability with any third-party software tool that supports HDP.
- Access Fanatical Support® on a 24x7x365 basis via chat, phone, or ticket.

This guide provides the following ways to use the Cloud Big Data API:

- Using the API directly (Chapter 4, “Using the API Directly” [5])
- Using a Python Lava command-line client (Chapter 5, “Using the Python Lava Client” [25])

Follow the steps described in this guide to start using the Rackspace Cloud Big Data API.

## 2.1. Early Access Program

The Early Access program enables customers and Rackers, by invitation, to test an early version of the product and provide feedback on the product and its capabilities. The Cloud Big Data Early Access program is a preproduction service offering covered by the Beta Services terms (rackspace.com/information/legal/testterms/) in the Rackspace Cloud Terms of Service.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

Early Access users must meet the following criteria:

- A Rackspace Cloud account
- Prior knowledge of HTTP/1.1 conventions
- Basic familiarity with Cloud and RESTful APIs
- A willingness to provide feedback to the Rackspace Product and Product Marketing functions

By using the Cloud Big Data API, you agree to and understand the following conditions:

- There is no charge for this service during the Early Access period.
- If you participate in this Early Access program, you will receive a notification prior to general availability.
- There is no Service Level Agreement.
- The API contract is not locked and might change during the Early Access program. If the contract changes, Rackspace will notify customers in release notes.
- Support is available Monday through Friday from 8 a.m. to 5 p.m. (Central Time).

## 2.2. Prerequisites for Running Examples

To run the examples in this guide, you must have the following prerequisites:

- Rackspace Cloud account
- Rackspace Cloud user name and password, as specified during registration

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# 3. Service Endpoints

During the Early Access program for Cloud Big Data, you use the following endpoint for your API operations: `https://dfw.bigdata.api.rackspacecloud.com/v1.0/1234`.

### Note

All examples in this guide assume that you are operating against the DFW data center.

Replace the sample account ID number, *1234*, with your actual account number, which is returned as part of the authentication response. Use your actual account number wherever you see the field **your_acct_id** specified in this guide. Refer to Section 4.2, "Generating an Authentication Token" [7].

When you perform a Cloud Big Data API operation, place the endpoint at the beginning of the request URL. For example: `https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/`.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# 4. Using the API Directly

This chapter describes how to use Cloud Big Data by using the API directly.

## 4.1. Sending Requests to the API

You have several options for sending requests through an API:

- Developers and testers might prefer to use cURL, the command-line tool from http://curl.haxx.se/. With cURL, you can send HTTP requests and receive responses back from the command line.
- If you like to use a more graphical interface, the REST client for Mozilla Firefox also works well for testing and trying out commands. See https://addons.mozilla.org/en-US/firefox/addon/restclient/.
- You can download and install RESTClient, a Java application used to test RESTful web services, from http://code.google.com/p/rest-client/.

### 4.1.1. Sending API Requests by Using cURL

cURL is a command-line tool that is available in most UNIX system-based environments and Apple Mac OS X systems, and can be downloaded for Microsoft Windows to interact with REST interfaces. For more information about cURL, visit http://curl.haxx.se/.

cURL enables you to transmit and receive HTTP requests and responses from the command line or from within a shell script. As a result, you can work with the REST API directly without using one of the client APIs.

The following cURL command-line options are used in this guide to run the examples:

#### Table 4.1. cURL Command-Line Options

| Option | Description |
|---|---|
| `-d` | Sends the specified data in a POST request to the HTTP server |
| `-i` | Includes the HTTP header in the output |
| `-H HEADER` | Specifies an HTTP header in the request |
| `-X` | Specifies the request operation to use when communicating with the HTTP server. The specified request is used instead of the default operation, which is GET. For example, `-X PUT` specifies to use the PUT request operation. |

#### Note

If you have the tools, you can run the cURL JSON request examples with the following options to format the output from cURL: **<curl JSON request example> | python -mjson.tool**.

#### Procedure 4.1. To run the cURL request examples

To run the cURL request examples shown in this guide on Linux or Mac systems, perform the following actions:

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

1. Copy each example from the HTML version of this guide an paste it into an ASCII text editor (for example, vi or TextEdit).

2. Modify each example with your required account information and other information, as detailed in this guide.

### Note

The carriage returns in the cURL request examples that are part of the cURL syntax are escaped with a backslash (\) to avoid prematurely terminating the command. However, you should not escape carriage returns inside the XML or JSON message within the command.

Consider the following cURL Authentication Request: XML example, which is described in detail in Section 4.2, "Generating an Authentication Token" [7].

### Example 4.1. cURL Authenticate Request: XML

```
curl -i -d \
'<?xml version="1.0" encoding="UTF-8"?>
 <auth>
    <apiKeyCredentials
        xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"

        username="your_username"
        apiKey="your_api_key"/>
 </auth>' \
'https://identity.api.rackspacecloud.com/v2.0/tokens'
```

The lines that are part of the cURL command syntax have been escaped with a backslash (\) to indicate that the command continues on the next line:

```
curl -i -d \



(... lines within the XML portion of the message are not shown in this
 example)
(... the example shows only lines that are part of cURL syntax)



 </auth>' \
'https://identity.api.rackspacecloud.com/v2.0/tokens'
```

However, the lines *within* the XML portion of the message are not escaped with a backslash, to avoid issues with XML processing.

```
'<?xml version="1.0" encoding="UTF-8"?>
 <auth>
    <apiKeyCredentials
        xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"

        username="your_username"
        apiKey="your_api_key"/>
 </auth>' \
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

The final line of the XML message is escaped, but the backslash lies outside the XML message and continues the cURL command to the next line.

3. After you are finished modifying the text for the cURL request example with your information (for example, user name and API key), copy and paste the text into your terminal window.

**Tip**

If you have trouble copying and pasting the examples as described, try typing the entire example on one long line, removing all the backslash line-continuation characters.

4. Press **Enter** to run the cURL command.

## 4.2. Generating an Authentication Token

Whether you use cURL or a REST client to interact with the Cloud Big Data API, you must generate an authentication token. You provide this token in the X-Auth-Token header in each Cloud Big Data API request. Example 4.2, "cURL Authenticate Request: XML" [8] and Example 4.3, "cURL Authenticate Request: JSON" [8] demonstrate how to use cURL to obtain the authentication token as well as your account number. You must provide both when making subsequent Cloud Big Data API requests.

Remember to replace the placeholders in the following authentication request examples with your information:

- **your_username** —Your common Rackspace Cloud user name, as supplied during registration.
- **your_api_key** — Your API access key. You can obtain the key from the Rackspace Cloud Control Panel in the **<Your Account>** / **API Keys** section.

You can use either of the following endpoints to access the Cloud Authentication service:

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

Your account might be based in either the US or the UK. The location of your account is not determined by your physical location but by the location of the Rackspace retail site that 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.

If you are unsure how your account was created, use the Rackspace contact information at either site to ask for help.

Notice that you authenticate by using a special URL for the Cloud Authentication services. For example, for US-based accounts, you use `https://identity.api.rackspacecloud.com/v2.0/tokens`, as shown in the following

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

Authenticate Request examples. Note that the `v2.0` component in the URL indicates that you are using version 2.0 of the Cloud Authentication API.

### Note

For UK-based accounts, you can use `'https://lon.identity.api.rackspacecloud.com/v2.0/tokens'` instead.

### Example 4.2. cURL Authenticate Request: XML

```
curl -i -d \
'<?xml version="1.0" encoding="UTF-8"?>
<auth>
   <apiKeyCredentials
      xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
         username="your_username"
         apiKey="your_api_key"/>
</auth>' \
-H 'Content-Type: application/xml' \
-H 'Accept: application/xml' \
'https://identity.api.rackspacecloud.com/v2.0/tokens'
```

### Example 4.3. cURL Authenticate Request: JSON

```
curl -i -d \
'{
    "auth":
    {
       "RAX-KSKEY:apiKeyCredentials":
       {
          "username": "your_username",
          "apiKey": "your_api_key"}
    }
}' \
-H 'Content-Type: application/json' \
'https://identity.api.rackspacecloud.com/v2.0/tokens'
```

In the authentication response (examples follow), the authentication `token id` is returned with an `expires` attribute that specifies when the token expires. Remember to supply your authentication token wherever you see the placeholder **your_auth_token** in the examples in this guide.

### Notes

- The values that you receive in your responses vary from the examples shown in this document because they are specific to your account.
- The information shown in the authentication response examples is for US-based accounts. If you authenticate against the UK-endpoint for the Authentication service, you see the service catalog information for UK-based accounts.
- The `expires` attribute denotes the time after which the token will automatically become invalid. A token might be manually revoked before the time identified by the `expires` attribute. The attribute 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.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

- Applications should be designed to re-authenticate after receiving a 401 (Unauthorized) response from a service endpoint.

The `publicURL` endpoints for Cloud Big Data (for example `https://dfw.bigdata.api.rackspacecloud.com/v1.0/1100111`) are also returned in the response.

Your actual account number is after the final slash (/) in the `publicURL` field. In the following examples, the account number is 1100111. You must specify *your* account number on most of the Cloud Big Data API operations, wherever you see the placeholder **your_acct_id** specified in the examples in this guide.

After authentication, you can use cURL to perform GET, DELETE, and POST requests for the Cloud Big Data API.

### Example 4.4. Authenticate Response: 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"?>
<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">
  <token id="vvvvvvvv-wwww-xxxx-yyyy-zzzzzzzzzzzz" expires=
"2011-12-08T22:51:02.000-06:00"/>
  <user id="123456" name="jsmith" rax-auth:defaultRegion="DFW">
    <roles>
      <role id="identity:admin" name="identity:admin" description="Admin Role.
"/>
      <role id="identity:default" name="identity:default" description="Default
 Role."/>
    </roles>
  </user>
  <serviceCatalog>
    <service type="rax:bigdata" name="cloudBigData">
      <endpoint region="DFW" tenantId="1100111" publicURL="https://dfw.big.
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/" />
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
      </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 type="object-store" name="cloudFiles">
      <endpoint region="DFW"
        tenantId="MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
        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"/>
      <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 4.5. Authenticate Response: 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": {

        "token": {
            "expires": "2011-12-08T22:51:02.000-06:00",
            "id": "vvvvvvvv-wwww-xxxx-yyyy-zzzzzzzzzzzz"
        },
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -


```
        "user": {
            "id": "123456",
            "name": "jsmith",
            "RAX-AUTH:defaultRegion": "DFW",
            "roles": [
                {
                    "description": "Admin Role.",
                    "id": "identity:admin",
                    "name": "identity:admin"
                },
                {
                    "description": "Default Role.",
                    "id": "identity:default",
                    "name": "identity:default"
                }
            ]
        },
        "serviceCatalog": [
            {
                "endpoints": [
                    {
                        "publicURL": "https://dfw.bigdata.api.rackspacecloud.
com/v1.0/1100111",
                        "region": "DFW",
                        "tenantId": "1100111"
                    },

                ],
                "name": "cloudBigData",
                "type": "rax:bigdata"
            },
            {
                "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/",
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -


```
                        "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": [
                    {
                        "tenantId": "MossoCloudFS_aaaaaaaa-bbbb-cccc-dddd-
eeeeeeee",
                        "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",
                        "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"
                    }
                ],
                "name": "cloudFiles",
                "type": "object-store"
            },
            {
                "endpoints": [
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
                    {
                        "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"
            }
        ]
    }
}
```

## 4.3. Setting Up and Viewing a Profile

Before you can create Hadoop clusters, you must create a profile, which consists of a user name and password.

### Note

Your Cloud Big Data profile is different from your cloud account. Your profile has the following characteristics and requirements:

- A profile is the configuration for the administration and login account for the cluster.
- Only one profile is allowed for each user or account.
- Any updates or additions override the existing profile.

After you create a profile, every Hadoop server that you provision by using the API will have a local user account with the specified user name and password. This allows you to remotely SSH into a server to transfer data, run or troubleshoot jobs, and so on.

A password must be at least eight characters, with a mixture of uppercase, lowercase, numeric, and special characters.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

The following example shows the cURL request to set up a profile.

### Example 4.6. cURL Create Profile Request: JSON

```
curl -i -X POST https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
profile \
-H "Accept: application/json" \
-H "X-Auth-Token: your_auth_token" \
-H "Content-Type: application/json" -d\
'{
    "profile": {
        "username": "myuser",
        "password": "T0pS3cret!",
        "sshkeys": [
            {
                "publicKey": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQDV9lu1CPeRmnTrDigLOU2lM60igHW4LXEgei9ajMcuUnJXy
UoTthHRlXpWG+D3H1Az/
AT1yPqYEubwBWZOJN3r3hvoqYaJYEKAE36vHceEJ89yLuu2PIMheRyhBgmapHCz9d3IGezuuKh/
DJRVl8Q7pICcsRIhQxHijcyfcPsPnTfDQ89LKg6k4uyxZZobQozXH/
vwtAcsGAW6hX6hsxlvBIqndPPMw2FXkGt56W+YuNNNQC+P
+qTyBQ8mWhy1OQJwvn9N/d0wdUAqogLfTasUXeTRjr+Q7ASC/
QZ6ewqgrBzLbbM7SqobhNcD6O2suUGr2uvGzKY4a3oaGO68ur2Z testuser@rackspace.com",
                "name": "mykey"
            }
        ],
        "cloudCredentials": {
            "username": "myuser",
            "apikey": "e96406df8e9a82565deb57f3555054b8"
        }
    }
}'
```

Remember to replace the placeholders in the example with your actual values for all the cURL examples in this guide:

- **your_auth_token** — Your authentication token, which is returned in your authentication response (see the response examples in Section 4.2, "Generating an Authentication Token" [7])
- **your_acct_id** — Your account ID, which is returned in your authentication response (must be replaced in the request URL)

The following example shows the response for the profile request.

### Example 4.7. Create Profile Response: JSON

```
{
    "profile": {
        "username": "myuser",
        "links": [
            {
                "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
681390/profile",
                "rel": "self"
            },
            {
                "href": "https://dfw.bigdata.api.rackspacecloud.com/681390/
profile",
```

- INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
                "rel": "bookmark"
            }
        ],
        "cloudCredentials": {
            "username": "myuser"
        },
        "userId": "194683",
        "sshkeys": [
            {
                "publicKey": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQDV9lu1CPeRmnTrDigLOU2lM60igHW4LXEgei9ajMcuUnJXy
UoTthHRlXpWG+D3H1Az/
AT1yPqYEubwBWZOJN3r3hvoqYaJYEKAE36vHceEJ89yLuu2PIMheRyhBgmapHCz9d3IGezuuKh/
DJRVl8Q7pICcsRIhQxHijcyfcPsPnTfDQ89LKg6k4uyxZZobQozXH/
vwtAcsGAW6hX6hsxlvBIqndPPMw2FXkGt56W+YuNNNQC+P
+qTyBQ8mWhy1OQJwvn9N/d0wdUAqogLfTasUXeTRjr+Q7ASC/
QZ6ewqgrBzLbbM7SqobhNcD6O2suUGr2uvGzKY4a3oaGO68ur2Z testuser@rackspace.com",
                "name": "mykey"
            }
        ],
        "tenantId": "681390"
    }
}
```

Anytime after you create your profile, you can view its details by using the Get Profile operation. The response displays your profile details, including your user name, user ID, and tenant ID.

| Verb | URI | Description |
|---|---|---|
| **GET** | /profile | Returns detailed information for the current user profile. |

This operation does not require a request body.

The following examples show the cURL request and corresponding response for viewing a profile.

### Example 4.8. cURL Get Profile Request: JSON

```
curl -i -X GET https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
profile -d  \
-H "X-Auth-Token: your_auth_token" \
-H "Accept: application/json"\
-H "Content-type: application/json"
```

### Example 4.9. Get Profile Response: JSON

```
 {"profile":
            { "username": "john.doe",
              "user_id" : "12346",
              "tenant_id" : "123456",
              "sshkeys": [{"name": "t@test"}],
              "cloudCredentials": {},
              "links": [
                { "rel": "self",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
123456/profile"
                },
                { "rel": "bookmark",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/123456/
profile"
```

```
                }
            ],
        }
    }
```

# 4.4. Viewing Resource Limits

The use of the Rackspace Cloud Big Data API is subject to resource limits. You can view the limits associated with your account by using the View Resource Limits operation. This operation displays resource limits such as remaining node count, available RAM, and remaining disk space, for the user.

| Verb | URI | Description |
|---|---|---|
| **GET** | /limits | Displays the resource limits for the user. |

This operation does not require a request body.

The following show the cURL request and corresponding response for viewing resource limits.

### Example 4.10. cURL View Resource Limits Request: JSON

```
curl -i -X GET https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
limits -d \
-H "X-Auth-Token: your_auth_token" \
-H "Accept: application/json" \
-H "Content-type: application/json"
```

### Example 4.11. View Resource Limits Response: JSON

```
{
    "limits": {
        "links": [
            {
                "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
681390/limits",
                "rel": "self"
            },
            {
                "href": "https://dfw.bigdata.api.rackspacecloud.com/681390/
limits",
                "rel": "bookmark"
            }
        ],
        "absolute": {
            "nodeCount": {
                "limit": 5,
                "remaining": 5
            },
            "vcpus": {
                "limit": 10,
                "remaining": 10
            },
            "ram": {
                "limit": 40960,
                "remaining": 40960
            },
            "disk": {
```



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
                "limit": 2048,
                "remaining": 2048
            }
        }
    }
}
```

# 4.5. Creating and Managing Hadoop Clusters

Now you are ready to create and manage Hadoop clusters by using the Rackspace Cloud Big Data API.

## 4.5.1. Listing Flavors

A flavor is an available hardware configuration for a cluster. Each flavor has a unique combination of memory capacity and priority for CPU time. The larger the flavor size you use, the larger the amount of RAM and the higher the priority for CPU time your cluster receives.

You use the List Flavors operation to find the available configurations for your cluster, and then you decide which size you need for your cluster.

| Verb | URI | Description |
|---|---|---|
| **GET** | /flavors | Lists all available flavors. |

This operation does not require a request body.

The following examples show the cURL request and the corresponding response for the List Flavors operation.

### Example 4.12. cURL List Flavors Request: JSON

```
curl -i -X GET https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
flavors -d \
-H "X-Auth-Token: your_auth_token" \
-H "Accept: application/json" \
-H "Content-type: application/json"
```

### Example 4.13. List Flavors Response: JSON

```
        { "flavors": [
          {
            "disk": 256,
            "id": "4fba3bca-7c76-11e2-b737-beeffa00040e",
            "links": [
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
flavors/4fba3bca-7c76-11e2-b737-beeffa00040e",
                "rel": "self"
              },
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/1234/
flavors/4fba3bca-7c76-11e2-b737-beeffa00040e",
                "rel": "bookmark"
              }
            ],
            "name": "Tiny Hadoop Instance",
            "ram": 5120,
```



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -


```
            "vcpus": 1
          },
          {
            "disk": 512,
            "id": "518400b2-7c76-11e2-b737-beeffa00040e",
            "links": [
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
flavors/518400b2-7c76-11e2-b737-beeffa00040e",
                "rel": "self"
              },
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/1234/
flavors/518400b2-7c76-11e2-b737-beeffa00040e",
                "rel": "bookmark"
              }
            ],
            "name": "Small Hadoop Instance",
            "ram": 8192,
            "vcpus": 2
          },
          {
            "disk": 1024,
            "id": "5413894c-7c76-11e2-b737-beeffa00040e",
            "links": [
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
flavors/5413894c-7c76-11e2-b737-beeffa00040e",
                "rel": "self"
              },
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/1234/
flavors/5413894c-7c76-11e2-b737-beeffa00040e",
                "rel": "bookmark"
              }
            ],
            "name": "Medium Hadoop Instance",
            "ram": 15360,
            "vcpus": 4
          },
          {
            "disk": 2000,
            "id": "559dd2b8-7c76-11e2-b737-beeffa00040e",
            "links": [
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
flavors/559dd2b8-7c76-11e2-b737-beeffa00040e",
                "rel": "self"
              },
              { "href": "http://dfw.bigdata.api.rackspacecloud.com/1234/
flavors/559dd2b8-7c76-11e2-b737-beeffa00040e",
                "rel": "bookmark"
              }
            ],
            "name": "Large Hadoop Instance",
            "ram": 27648,
            "vcpus": 8
          }
        ] }
```


## 4.5.2. Creating a Cluster

The Create Cluster operation creates a new Hadoop cluster. After you submit the operation, it takes a few minutes for the service to build and configure your cluster.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

| Verb | URI | Description |
|---|---|---|
| **POST** | /clusters | Creates a new cluster. |

The following example creates a 5-data-node Hadoop cluster with the name `slice`.

The value of the `clusterType` attribute must be HADOOP_HDP1_1. The `flavorID` attribute specifies the RAM, virtual CPUs, and disk storage associated with each data node in your Hadoop cluster.

The following examples show the cURL request and corresponding response for Create Cluster.

### Example 4.14. cURL Create Cluster Request: JSON

```
curl -i -X POST https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
clusters -d \
'{"cluster":
      { "name": "slice",
        "clusterType": "HADOOP_HDP1_1",
        "flavorId": "4fba3bca-7c76-11e2-b737-beeffa00040e",
        "nodeCount": 5
      }
    }' \
-H "X-Auth-Token: your_auth_token" \
-H "Accept: application/json" \
-H "Content-type: application/json"
```

### Example 4.15. Create Cluster Response: JSON

```
    {"cluster":
      { "id": "db478fc1-2d86-4597-8010-cbe787bbbc41",
        "created": "2012-12-27T10:10:10Z",
        "updated": "",
        "name": "slice",
        "clusterType": "HADOOP_HDP1_1",
        "flavorId": "4fba3bca-7c76-11e2-b737-beeffa00040e",
        "nodeCount": 5,
        "status": "BUILD",
        "links": [
          { "rel": "self",
            "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41"
          },
          { "rel": "bookmark",
            "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/clusters/
db478fc1-2d86-4597-8010-cbe787bbbc41"
          }
        ],
      }
    }
```

## 4.5.3. Viewing Details

The Get Node Details operation lists all server nodes for the specified cluster.

| Verb | URI | Description |
|---|---|---|
| **GET** | /clusters/{id}/nodes | Lists all nodes for a cluster. |

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

In the following example, the cluster has a name node, a gateway node, and two data nodes. Each node has a private IP address, which is used for backend (Hadoop) data transfer, and a public IP address, which enables you to access it over the public Internet. Use the gateway node to do your work, including using tools such as Apache Pig and Apache Hive. In the example, you can remotely SSH into the gateway node over the IP address by using the user name and password that you added to your profile during setup.

This operation does not have a request body.

The following examples show the cURL request and corresponding response for listing all nodes for a cluster.

### Example 4.16. cURL List Cluster Nodes Request: JSON

```
curl -i -X GET https://dfw.bigdata.api.rackspacecloud.com/v1.0/your_acct_id/
clusters/ac111111-2d86-4597-8010-cbe787bbbc41/nodes -d \
-H "X-Auth-Token: your_auth_token" \
-H "Accept: application/json" \
-H "Content-Type: application/json"
```

### Example 4.17. List Cluster Nodes Response: JSON

```
        {"nodes":
          [
            { "id": "000",
              "created": "2012-12-27T10:10:10Z",
              "role": "NAMENODE",
              "name": "NAMENODE-1",
              "status": "ACTIVE",
              "addresses": {
                "public": [{ "addr": "168.x.x.3", "version": 4 }],
                "private": [{ "addr": "10.x.x.3", "version": 4}]
              },
              "services": [
                { "name": "namenode" },
                { "name": "jobtracker" },
                { "name": "ssh",
                  "uri": "ssh://user@168.x.x.3"
                }
              ],
              "links": [
                { "rel": "self",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
1234/clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/000"
                },
                { "rel": "bookmark",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/000"
                }
              ],
            },
            { "id": "aaa",
              "role": "GATEWAY",
              "name": "GATEWAY-1",
              "status": "ACTIVE",
              "addresses": {
                "public": [{ "addr": "168.x.x.4", "version": 4 }],
                "private": [{ "addr": "10.x.x.4", "version": 4}]
              },
```

- INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -


```
              "services": [
                { "name": "pig" },
                { "name": "hive" },
                { "name": "ssh",
                  "uri": "ssh://user@168.x.x.4"
                },
                { "name": "status",
                  "uri": "http://10.x.x.4"
                },
                { "name": "hdfs-scp",
                  "uri": "scp://user@168.x.x.4:9022"
                }
              ],
              "links": [
                { "rel": "self",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
1234/clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/aaa"
                },
                { "rel": "bookmark",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/aaa"
                }
              ],
            },
            { "id": "bbb",
              "role": "DATANODE",
              "name": "DATANODE-1",
              "status": "ACTIVE",
              "addresses": {
                "public": [{ "addr": "168.x.x.5", "version": 4 }],
                "private": [{ "addr": "10.x.x.5", "version": 4}]
              },
              "services": [
                { "name": "datanode" },
                { "name": "tasktracker" },
                { "name": "ssh",
                  "uri": "ssh://user@168.x.x.5"
                }
              ],
              "links": [
                { "rel": "self",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
1234/clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/bbb"
                },
                { "rel": "bookmark",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/bbb"
                }
              ],
            },
            { "id": "ccc",
              "role": "DATANODE",
              "name": "DATANODE-2",
              "status": "ACTIVE",
              "addresses": {
                "public": [{ "addr": "168.x.x.6", "version": 4 }],
                "private": [{ "addr": "10.x.x.6", "version": 4}]
              },
              "services": [
                { "name": "datanode" },
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
                { "name": "tasktracker" },
                { "name": "ssh",
                  "uri": "ssh://user@168.x.x.6"
                }
              ],
              "links": [
                { "rel": "self",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/
1234/clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/ccc"
                },
                { "rel": "bookmark",
                  "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41/nodes/ccc"
                }
              ],
            }
          ]
        }
```

## 4.5.4. Resizing Clusters

You can increase the size of an existing cluster by using the `resize` command. Downsizing clusters is not currently supported.

| Verb | URI | Description |
|---|---|---|
| **POST** | /clusters/{clusterId}/action | Resizes a cluster. |

The following example resizes the previously created cluster, named `slice`, to include 10 data nodes. When you use the `resize` command, you specify the total number of data nodes that you want the cluster to have, not just the number of nodes to add. In the example, 10 is the total number of nodes that the cluster will have after the command is run. After you initiate the `resize` operation, you can use the List Cluster Nodes operation to confirm that your cluster has been resized.

The following examples show the cURL request and corresponding response to resize a cluster.

### Example 4.18. cURL Resize Cluster Request: JSON

```
curl -i -X POST https://dfw.bigdata.api.rackspacecloud.com/v1.
0/your_account_ID/clusters/your_cluster_ID/action -d \
'{"resize":{ "nodeCount": 10 }} ' \
-H "Accept: application/json" \
-H "X-Auth-Token:your_auth_token" \
-H "Content-Type: application/json"
```

### Example 4.19. Resize Cluster Response: JSON

```
        {"cluster":
          { "id": "db478fc1-2d86-4597-8010-cbe787bbbc41",
            "created": "2012-12-27T10:10:10Z",
            "updated": "2012-12-27T16:20:10Z",
            "name": "slice",
            "clusterType": "HADOOP_HDP1_1",
            "flavorId": "4fba3bca-7c76-11e2-b737-beeffa00040e",
            "nodeCount": 10,
```

```
            "status": "RESIZING",
            "links": [
              { "rel": "self",
                "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41"
              },
              { "rel": "bookmark",
                "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41"
              }
            ],
          }
        }
```

## 4.5.5. Deleting Clusters

Use the Delete Cluster operation to remove unused Hadoop clusters. This operation deletes any servers associated with the cluster and any data stored in the cluster.

You cannot delete clusters that are in the process of being created or resized.

| Verb | URI | Description |
|---|---|---|
| **DELETE** | /clusters/{id} | Deletes a cluster. |

The following examples show the cURL request and corresponding response to delete a cluster.

### Example 4.20. cURL Delete Cluster Request: JSON

```
curl -i -X DELETE https://dfw.bigdata.api.rackspacecloud.com/v1.
0/your_account_ID/clusters/your_cluster_ID -d \
-H "Accept: application/json" \
-H "X-Auth-Token:your_auth_token" \
-H "Content-Type: application/json"
```

### Example 4.21. Delete Cluster Response: JSON

```
Status: 202 Accepted
        Date: Mon, 06 Aug 2012 21:54:21 GMT
        Content-Type: application/json
        {"cluster":
          { "id": "db478fc1-2d86-4597-8010-cbe787bbbc41",
            "created": "2012-12-27T10:10:10Z",
            "updated": "2012-12-27T20:14:10Z",
            "name": "slice",
            "clusterType": "HADOOP_HDP1_1",
            "flavorId": "4fba3bca-7c76-11e2-b737-beeffa00040e",
            "nodeCount": 5,
            "status": "DELETING",
            "links": [
              { "rel": "self",
                "href": "https://dfw.bigdata.api.rackspacecloud.com/v1.0/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41"
              },REST
              { "rel": "bookmark",
                "href": "https://dfw.bigdata.api.rackspacecloud.com/1234/
clusters/db478fc1-2d86-4597-8010-cbe787bbbc41"
              }
```



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

```
            ]
          }
        }
```



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

# 5. Using the Python Lava Client

This chapter explains how to start using the Python Lava client (python-lavaclient), which is a convenient Python binding and command-line client that uses the RESTful API.

## 5.1. Setting Up

This section describes prerequisites, installation, and setup for the python-lavaclient.

### 5.1.1. Prerequisites

Following are the requirements for using python-lavaclient:

- Computer running Linux or Mac OS X
- Python 2.7.2 or later
- Rackspace Cloud account and access to Rackspace Cloud Big Data

### 5.1.2. Installing the Client

Perform the following steps to install the client.

1. Download python-lavaclient to your computer from http://bit.ly/cbd-client.
2. Extract the archive by using the following command:

   ```
   $ tar xvzf python-lava-0.8.6.tgz
   ```

   A python-lavaclient folder is created.
3. Open the folder and install the client by using the following commands:

   ```
   $ cd python-lavaclient
   $ sudo python setup.py install
   ```
4. Run the help command to ensure that the client has been installed correctly and note the usage information:

   ```
   $ lava help
   ```

### 5.1.3. Authenticating Your Session

Before you can start using Rackspace Cloud Big Data, you need to authenticate your session. To complete this step, you need your Cloud user name and API key.

1. Run the `auth` command. When prompted, enter your user name and API key.

   ```
   $ lava auth
   Username: <enter your Cloud username>
   API Key: <enter your API Key>
   ```

   If the command runs successfully, your authentication token and access endpoint are displayed, as shown in the following example.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

### Example 5.1. Authentication Response Using CLI Utility

```
AUTH_TOKEN=692c2a14-39ad-4ee0-991d-06cd7331f3ca
LAVA_API_URL=https://dfw.bigdata.api.rackspacecloud.com/v1.0/12345
```

2. Export the AUTH_TOKEN and LAVA_API_URL environment variables as shown in the following example.

### Example 5.2. Export Environment Variables

```
$ export AUTH_TOKEN=692c2a14-39ad-4ee0-991d-06cd7331f3ca
$ export LAVA_API_URL=https://dfw.bigdata.api.rackspacecloud.com/v1.0/12345
```

**Note**

The `export` commands are valid only for the current session. You need to rerun the `export` commands if, for example, you create a new console window.

3. To confirm that the client is running, run the list command. The first time that you run this command, it does not return any output.

```
$ lava list
```

## 5.1.4. Setting Up a Profile

Before you can create Hadoop clusters, you must create a profile that consists of a user name and password. To create a profile, use the `create-profile` command as shown in the following example.

**Note**

Your Cloud Big Data profile is different from your cloud account. Your profile has the following characteristics and requirements:

- A profile is the configuration for the administration and login account for the cluster.
- Only one profile is allowed for each user or account.
- Any updates or additions override the existing profile.

In this example, a profile is created with the user name `joe` and the password `Foo1234!`.

### Example 5.3. Creating a Profile

```
$ lava create-profile joe Foo1234!
```

A password must be at least eight characters, with a mixture of uppercase, lowercase, numeric, and special characters.

After you create your profile, every Hadoop server that you provision by using the API will have a local user account with the specified name and password. This allows you to remotely SSH into a server to transfer data, run or troubleshoot jobs, and so on.

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

To confirm that your profile was successfully created, run the `get-profile` command.

**Example 5.4. Retrieve a Profile by Using the get-profile Command**

```
$ lava get-profile
+-------------+-----------+-------------+
| Username | User ID | Tenant ID |
+-------------+---------+--------------+
| joe | 118091 | 12345 |
+-------------+-----------+-------------+
```

## 5.1.5. Viewing Resource Limits

The use of Rackspace Cloud Big Data is subject to resource limits. You can view the limits associated with your account by using the `quotas` command.

**Example 5.5. View Resource Limits by Using the quotas Command**

```
$ lava quotas
+-------------+-----------------+-------------+-----------------+-----------
+--------------+------------+----------------+
| Nodes Limit | Nodes Remaining | VCPUs Limit | VCPUs Remaining | RAM Limit |
 Ram Remaining | Disk Limit | Disk Remaining |
+-------------+-----------------+-------------+-----------------+-----------
+--------------+------------+----------------+
| 5 | 5 | 5 | 5 | 25600 | 25600 | 2048 | 2048 |
+-------------+-----------------+-------------+-----------------+-----------
+--------------+------------+----------------+
```

# 5.2. Creating and Managing Hadoop Clusters

This section explains how to create and manage Hadoop clusters by using Rackspace Cloud Big Data through the python-lavaclient. After you complete this section, you will have a working Apache Hadoop cluster.

## 5.2.1. Provisioning a Hadoop Cluster

Use the `boot` command to provision a Hadoop cluster. The following example creates a 3-data-node Hadoop cluster with the name `foo`.

**Example 5.6. Provision a Cluster by Using the boot Command**

```
$ lava boot --type=HADOOP_HDP1_1 --flavor=4fba3bca-7c76-11e2-b737-beeffa00040e
 --count=3 foo
+----------+--------------------------------------+
| Property | Value |
+----------+--------------------------------------+
| Id | 4820deb2-6212-44f9-b92f-979fe723ffb8 |
| Name | foo |
| Status | BUILD |
| Nodes | 3 |
| Type | HADOOP_HDP1_1 |
| Flavor | 4fba3bca-7c76-11e2-b737-beeffa00040e |
+----------+--------------------------------------+
```

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

The `type` argument must specify HADOOP_HDP1_1. The `flavor` argument specifies the RAM, virtual CPUs, and disk storage associated with each data node in your Hadoop cluster. You can enumerate the flavors and associated resources by using the `flavor-list` command, as shown in the following example.

### Example 5.7. List Flavors and Associated Resources by Using the flavor-list Command

```
$ lava flavor-list
+--------------------------------------+------------------------+-------
+------+------+
| Id | Name | Mem | Disk | CPUs |
+--------------------------------------+------------------------+-------
+------+------+
| 4fba3bca-7c76-11e2-b737-beeffa00040e | Tiny Hadoop Instance | 5120 | 256 | 1
 |
| 518400b2-7c76-11e2-b737-beeffa00040e | Small Hadoop Instance | 8192 | 512 |
 2 |
| 5413894c-7c76-11e2-b737-beeffa00040e | Medium Hadoop Instance | 15360 | 1024
 | 4 |
| 559dd2b8-7c76-11e2-b737-beeffa00040e | Large Hadoop Instance | 27648 | 2048
 | 8 |
+--------------------------------------+------------------------+-------
+------+------+
```

After you run the `boot` command, it takes a few minutes for the service to build and configure your cluster. Use the `list` command to query the build status of your cluster.

### Example 5.8. Query the Status of a Cluster by Using the list Command

```
$ lava list
+--------------------------------------+------+--------+-------
+---------------+
| Id | Name | Status | Nodes | Type |
+--------------------------------------+------+--------+-------
+---------------+
| 4820deb2-6212-44f9-b92f-979fe723ffb8 | foo | ACTIVE | 3 | HADOOP_HDP1_1 |
+--------------------------------------+------+--------+-------
+---------------+
```

When your cluster is ready, the value of `Status` is `ACTIVE`.

## 5.2.2. Viewing Details

Use the `show` and `nodes` commands to query your cluster, as shown in the following example.

### Example 5.9. Query the Details of a Cluster by Using the show and nodes Commands

```
$ lava show foo
+----------+--------------------------------------+
| Property | Value |
+----------+--------------------------------------+
| Id | 4820deb2-6212-44f9-b92f-979fe723ffb8 |
| Name | foo |
| Status | ACTIVE |
```

```
| Nodes | 3 |
| Type | HADOOP_HDP1_1 |
| Flavor | 4fba3bca-7c76-11e2-b737-beeffa00040e |
+----------+-------------------------------------+
$ lava nodes foo
+-------------------------------------+--------------+----------+--------
+----------------+----------------+
| Id | Name | Role | Status | Public IP | Private IP |
+-------------------------------------+--------------+----------+--------
+----------------+----------------+
| f530a9f1-79a8-4378-bf2a-b7f7e0c2bdd3 | NAMENODE-88 | NAMENODE | ACTIVE |
 166.78.132.85 | 10.190.240.88 |
| 87a1bf57-daa6-4dd5-b711-f428cc44fe74 | DATANODE-182 | DATANODE | ACTIVE |
 166.78.132.228 | 10.190.240.228 |
| 82136cd0-2c98-4c76-9224-54aef9942de3 | DATANODE-183 | DATANODE | ACTIVE |
 166.78.132.84 | 10.190.240.84 |
| 8a9929ed-9ba3-4a22-b5a7-8a36a81179a6 | DATANODE-184 | DATANODE | ACTIVE |
 166.78.132.88 | 10.190.240.85 |
| 55ed5001-fdc3-44f1-a82c-97532007f3b3 | GATEWAY-89 | GATEWAY | ACTIVE | 166.
78.132.227 | 10.190.240.227 |
+-------------------------------------+--------------+----------+--------
+----------------+----------------+
```

The example shows that the cluster has the following nodes:

- One name node
- One gateway node
- Three data nodes

Each server node has the following IP addresses:

- A private IP address that is used for backend (Hadoop) data transfers
- A public IP address that allows you to access the server over the public Internet

Use the gateway node to do your work, including using tools such as Pig and Hive. In the example, you can remotely SSH into the gateway node over IP address 166.78.132.227 by using the user name and password that you used to create your profile during setup.

## 5.2.3. Resizing Clusters

You can increase the size of an existing cluster by using the `resize` command. Downsizing clusters is not currently supported. The following example resizes the previously created cluster `foo` to include 5 data nodes.

### Example 5.10. Increase Cluster Size by Using the resize Command

```
$ lava resize --count=5 foo
```

When you use the `resize` command, you specify the total number of data nodes that you want the cluster to have, not just the number of nodes to add. In the example, 5 is the total number of nodes that the cluster will have after the command is run. After you initiate the `resize` operation, use the `show` or `list` commands to confirm that your cluster has been resized.



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -

## 5.2.4. Deleting Clusters

Use the `delete` command to remove unused Hadoop clusters. This command deletes any servers associated with the cluster and any data stored in the cluster.

### Example 5.11. Remove Clusters by Using the delete Command

```
$ lava delete foo
```

### Note

You cannot delete clusters that are in the process of being created or resized.

# 6. Additional Resources

If you have any questions or concerns about using any of the steps in this guide, send an email to cbdbeta@rackspace.com.

For information about all Cloud Big Data API operations, see the *Cloud Big Data Developer Guide* at http://docs-internal.rackspace.com/. All you need to get started with Cloud Big Data is this getting started guide and the developer guide.

For more details about Rackspace Cloud Big Data, go to http://www.rackspace.com/cloud/cloud_hosting_products/databases/. This site also offers links to official Rackspace support channels, including knowledge center articles, forums, phone, chat, and email.

Visit the Product Feedback Forum and tell us what you think about Cloud Big Data.

This API uses standard HTTP 1.1 response codes as documented at www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.



ess - INTERNAL - Early Access - INTERNAL - Early Access - INTERNAL - Early Access -