



Next Generation Cloud Servers™

Developer Guide

API v2 (Mar 26, 2013)

Next Generation Cloud Servers™ Developer Guide

API v2 (2013-03-26)

Copyright © 2009-2013 Rackspace US, Inc. All rights reserved.

This document is intended for software developers who want to develop applications by using the next generation Rackspace Cloud Servers™ powered by OpenStack. In addition to the core features of the OpenStack Compute Application Programming Interface (API) v2, Rackspace has deployed certain extensions as permitted by the OpenStack Compute API contract. 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.

Table of Contents

Preface	ix
1. Intended Audience	ix
2. Pricing and Service Level	x
1. General API Information	1
1.1. Cloud Servers Concepts	1
1.2. How cURL Commands Work	2
1.3. Authenticate through the Rackspace Cloud Identity Service	6
1.4. Request/Response Types	11
1.5. Links and References	15
1.6. Paginated Collections	19
1.7. Efficient Polling with the <i>Changes-Since</i> Parameter	22
1.8. Limits	23
1.9. Versions	27
1.10. Extensions	31
1.11. Faults	38
2. API Operations	45
2.1. Servers	45
2.2. Server Addresses	75
2.3. Server Actions	78
2.4. Volume Attachment Actions	96
2.5. Flavors	101
2.6. Images	107
2.7. Metadata	131
3. Rackspace Extensions	140
3.1. Bandwidth Stats Extension	140
3.2. Disk Configuration Extension	140
3.3. Extended Status Extension	144
3.4. Rescue Mode Extension	147
3.5. Used Limits Extension	148
4. Document Change History	150
5. Resources	152
Glossary	153

List of Tables

1.1. cURL Command-Line Options	3
1.2. Rackspace Cloud Identity Service Endpoints	6
1.3. Default Rate Limits	23
1.4. Absolute Max Limits	24
1.5. Absolute Total Limits	24
1.6. Fault Elements and Error Response Codes	39
2.1. List Server Response Fields	49
2.2. Create Server Request Attributes	55
2.3. Get Server Details Response Fields	64
2.4. Update Server Request Attributes	68
2.5. Change Administrator Password Request Attributes	79
2.6. Reboot Server Request Attributes	80
2.7. Rebuild Server Request Attributes	81
2.8. Resize Server Request Attributes	86
2.9. Create Image Request Attributes	94
2.10. Attach Volume Request Attributes	96
2.11. Set Metadata Request Attributes	132
2.12. Update Metadata Request Attributes	134
2.13. Set Metadata Item Request Attributes	137
3.1. Extended Status Extension	144
3.2. Server Statuses and Corresponding VM and Task Statuses	145

List of Examples

1.1. cURL Command Example: XML Request and Response	5
1.2. Authenticate to US Identity Endpoint – Username and Password: JSON Request.....	7
1.3. Authenticate to US Identity Endpoint – Username and API Key: JSON Request	7
1.4. Pretty Printing cURL Output	7
1.5. Authenticate: JSON Response	7
1.6. Request with Headers: JSON	11
1.7. Response with Headers: XML	13
1.8. JSON Request with XML Query Extension for the Response	13
1.9. Image Reference in Create Server Request: JSON Request	15
1.10. Full Image Reference in Create Server Request: JSON Request	15
1.11. Image Reference in Create Server Request: XML Request	16
1.12. Full Image Reference in Create Server Request: XML Request	16
1.13. Server with Self Links: JSON	17
1.14. Server with Alternate Link: JSON	17
1.15. Server with Self Links: XML	18
1.16. Image with Alternate Link: XML	18
1.17. Images Collection – First Page: JSON	20
1.18. Images Collection – Second Page: JSON	20
1.19. Images Collection – Last Page: JSON	20
1.20. Images Collection – First Page: XML	21
1.21. Images Collection – Second Page: XML	21
1.22. Images Collection – Last Page: XML	21
1.23. Get Limits: JSON Response	25
1.24. Get Limits: XML Response	27
1.25. List Versions: Request	27
1.26. List Versions: cURL with JSON Request	28
1.27. List Versions: JSON Response	28
1.28. List Versions: cURL with XML Request	28
1.29. List Versions: XML Response	28
1.30. Get Version Details: Request	29
1.31. Get Version Details: cURL with JSON Request	29
1.32. Get Version Details: JSON Response	29
1.33. Get Version Details: cURL with XML Request	30
1.34. Get Version Details: XML Response	30
1.35. Get Extensions: cURL with JSON Request	31
1.36. Get Extensions: JSON Response	31
1.37. Get Extensions: cURL with XML Request	33
1.38. Get Extensions: XML Response	33
1.39. Get Extension: cURL with JSON Request	34
1.40. Get Extension: JSON Response	34
1.41. Get Extension: cURL with XML Request	34
1.42. Get Extension: XML Response	34
1.43. Extended Server: XML Response	35
1.44. Extended Server: JSON Response	36
1.45. Extended Action: XML Request	37
1.46. Extended Action: JSON Request	37
1.47. Fault: JSON Response	38
1.48. Fault: XML Response	38

1.49. Fault, Item Not Found: JSON Response	39
1.50. Fault, Item Not Found: XML Response	39
1.51. Fault, Over Limit: JSON Response	40
1.52. Fault, Over Limit: XML Response	40
1.53. Server in Error State: XML	41
1.54. Server in Error State: JSON	42
1.55. Image in Error State: XML	43
1.56. Image in Error State: JSON	44
2.1. List Servers: JSON Request in a cURL Command	50
2.2. List Servers: JSON Response (detail)	50
2.3. List Servers: XML Request in a cURL Command	53
2.4. List Servers: XML Response (detail)	53
2.5. Create Server: JSON Request	57
2.6. Create Server: JSON Response	57
2.7. Create Server: XML Request	58
2.8. Create Server: XML Response	58
2.9. Set Administrator Password in Create Server Request: JSON Request	59
2.10. Create Server with Access IP: JSON Request	60
2.11. Create Server with Access IP: XML Request	61
2.12. Create Server with Multiple Access IPs: JSON Request	62
2.13. Create Server with Multiple Access IPs: XML Request	62
2.14. Get Server Details: JSON Response	66
2.15. Get Server Details: XML Response	67
2.16. Update Server Name: JSON Request	68
2.17. Update Server Name: JSON Response	68
2.18. Update Server Name: XML Request	70
2.19. Update Server Name: XML Response	70
2.20. Update Server Access Address: JSON Request	70
2.21. Update Server Access Address: JSON Response	70
2.22. Update Server Access Address: XML Request	73
2.23. Update Server Access Address: XML Response	73
2.24. List Addresses: JSON Response	76
2.25. List Addresses: XML Response	76
2.26. List Addresses by Network: JSON Response	77
2.27. List Addresses by Network: XML Response	77
2.28. Change Administrator Password: JSON Request	79
2.29. Change Administrator Password: XML Request	79
2.30. Reboot Server: JSON Request	80
2.31. Reboot Server: XML Request	80
2.32. Rebuild Server: JSON Request	83
2.33. Rebuild Server: JSON Response	83
2.34. Rebuild Server: XML Request	85
2.35. Rebuild Server: XML Response	85
2.36. Resize Server: JSON Request	87
2.37. Resize Server: XML Request	87
2.38. Confirm Resize: JSON Request	88
2.39. Confirm Resize: XML Request	88
2.40. Revert Resize: JSON Request	89
2.41. Revert Resize: XML Request	89
2.42. Rescue Server Request: JSON	91
2.43. Rescue Server Response: JSON	91

2.44. Rescue Server Request: XML	91
2.45. Rescue Server Response: XML	91
2.46. Unrescue Server Request: JSON	92
2.47. Unrescue Server Request Response: JSON	92
2.48. Unrescue Server Request: XML	92
2.49. Unrescue Server Response: XML	92
2.50. Create Image: JSON Request	94
2.51. Create Image: XML Request	95
2.52. Attach Volume Request: JSON	97
2.53. Attach Volume Response: JSON	97
2.54. Attach Volume Request: XML	97
2.55. Attach Volume Response: XML	97
2.56. List Volume Attachments Response: JSON	99
2.57. List Volume Attachments Response: XML	99
2.58. Get Volume Attachment Details Response: JSON	100
2.59. Get Volume Attachment Details Response: XML	100
2.60. List Flavors: JSON Response (detail)	102
2.61. List Flavors: XML Response (detail)	103
2.62. Get Flavor Details: JSON Response	105
2.63. Get Flavor Details: XML Response	106
2.64. List Images: JSON Response (detail)	108
2.65. List Images: XML Response (detail)	110
2.66. Get Image Details: JSON Response	110
2.67. Get Image Details: XML Response	112
2.68. List Metadata: JSON Response	131
2.69. List Metadata: XML Response	131
2.70. Set Metadata: JSON Request	132
2.71. Set Metadata: JSON Response	132
2.72. Set Metadata: XML Request	132
2.73. Set Metadata: XML Response	133
2.74. Update Metadata: JSON Request	134
2.75. Update Metadata: JSON Response	134
2.76. Update Metadata: XML Request	134
2.77. Update Metadata: XML Response	135
2.78. Get Metadata Item: JSON Response	136
2.79. Get Metadata Item: XML Response	136
2.80. Set Metadata Item: JSON Request	137
2.81. Set Metadata Item: JSON Response	137
2.82. Set Metadata Item: XML Request	138
2.83. Set Metadata Item: XML Response	138
3.1. Create Server with OS-DCF:diskConfig: JSON Request	141
3.2. Create Server with OS-DCF:diskConfig: XML Request	142
3.3. Rebuild Server with OS-DCF:diskConfig: JSON Request	142
3.4. Rebuild Server with OS-DCF:diskConfig: XML Request	143
3.5. Resize Server with OS-DCF:diskConfig: JSON Request	143
3.6. Resize Server with OS-DCF:diskConfig: XML Request	144
3.7. Place a Server in Rescue Mode: JSON Request	147
3.8. Place a Server in Rescue Mode: JSON Response	147
3.9. Place a Server in Rescue Mode: XML Request	148
3.10. Place a Server in Rescue Mode: XML Response	148
3.11. Unrescue a Server in Rescue Mode: XML Request	148

3.12. Used Limits: JSON Response	148
--	-----

Preface

Next generation Cloud Servers **powered by OpenStack** is a fast, reliable, and scalable cloud compute solution without the risk of proprietary lock-in. It provides the core features of the OpenStack Compute *API* v2 and also deploys certain extensions as permitted by the OpenStack Compute API contract. Some of these extensions are generally available through OpenStack while others implement Rackspace-specific features to meet customers' expectations and for operational compatibility. The OpenStack Compute API and the Rackspace extensions are known collectively as API v2.

This document describes the features available with API v2.

We welcome feedback, comments, and bug reports. Log into the Rackspace customer portal at <http://www.rackspace.com/support/>.

1. Intended Audience

This guide assists software developers who want to develop applications by using next generation Cloud Servers. To use this information, you should have access to an active Rackspace Cloud Servers account, and you should also be familiar with the following concepts:

- Rackspace Cloud Servers service
- *RESTful* web services
- *HTTP/1.1*
- JSON and/or XML data serialization formats

2. Pricing and Service Level

Next generation Cloud Servers is part of the Rackspace Cloud and your use through the API will be billed as per the pricing schedule at <http://www.rackspace.com/cloud/public/servers/pricing>.

The Service Level Agreement (SLA) for Cloud Servers is available at http://www.rackspace.com/cloud/legal/sla/#cloud_servers.

1. General API Information

API v2 is defined as a *RESTful HTTP* service that uses all aspects of the HTTP protocol, including methods, URIs, media types, and response codes. To request next generation Cloud Servers services, you must first issue an *authentication* request to the Rackspace Cloud Identity Service, which is an implementation of the OpenStack Keystone Identity Service v2.0.

API v2 supports both the JSON and XML data serialization request and response formats.

1.1. Cloud Servers Concepts

To use the next generation Cloud Servers service with or without the Cloud Networks extension you should understand these key concepts:

Concept	Description
Server	A virtual machine (VM) instance running on a host. To create a server, you must specify a name, <i>flavor</i> reference, and <i>image</i> reference. See create server .
Host	A physical server running multiple VM instances.
Flavor	A resource configuration for a server. Each flavor is a unique combination of disk space, memory capacity, vCPUs, and network <i>bandwidth</i> . See flavors .
Image	A collection of files for a specific operating system (OS) that you use to create or rebuild a server. Rackspace provides pre-built images. You can also create custom images from servers that you have launched. Custom images can be used for data backups or as "gold" images for additional servers. See images .
Reboot	This action performs either a soft or hard reboot of a server. A soft reboot is a graceful shutdown and restart of the operating system on your server. A hard reboot power cycles your server, which performs an immediate shutdown and restart. See reboot server .
Rebuild	This action removes all data on the server and replaces it with the specified image. Server ID and IP addresses on the server remain the same. See rebuild server .
Resize	This action converts an existing server to a different flavor, which scales the server up or down. The original server is saved for a period of time to allow rollback if a problem occurs. You can confirm or revert a resize. A confirmed resize removes the original server. A reverted resize restores the original server. All resizes are automatically confirmed after 24 hours if you do not explicitly confirm or revert them. See resize server .
CIDR	Classless Inter-Domain Routing (CIDR). A method for allocating IP addresses and routing Internet Protocol packets. When you create an isolated network through Cloud Networks, you specify a CIDR.
isolated network	A virtual Layer 2 network that you create through Cloud Networks and that you can attach to a new Next Generation Cloud Server. Use an isolated network to keep your server separate from the Rackspace network, the Internet, or both. When you create an isolated network, it is associated with your tenant ID.

Concept	Description
PublicNet	Provides access to the Internet, Rackspace services such as Cloud Monitoring, Managed Cloud Support, RackConnect, Cloud Backup, and certain operating system updates. When you list networks through Cloud Networks, PublicNet is labeled <code>public</code> .
ServiceNet	An internal only, multi-tenant network connection within each Rackspace data center. Provides access to Rackspace services, such as Cloud Files, Cloud Databases, Cloud Backup, and to certain packages and patches. ServiceNet IPs are not accessible from the Internet and are local to each data center. You can configure your account resources to use a ServiceNet IP address so that traffic over the internal network is not billed. When you list networks through Cloud Networks, ServiceNet is labeled as <code>private</code> .

1.2. How cURL Commands Work

cURL is a command-line tool that you can use to interact with *REST* interfaces. cURL lets you to transmit and receive *HTTP* requests and responses from the command line or a shell script, which enables you to work with the API directly. It is available for Linux distributions, Mac OS X, and Windows. For information about cURL, see <http://curl.haxx.se/>.

The cURL examples in this guide use JSON requests and responses. To use XML requests and responses, see [Section 1.2.1, "XML Requests and Responses" \[5\]](#).

The cURL examples in this guide show both JSON and XML requests and responses. For information about XML, see [Section 1.2.1, "XML Requests and Responses" \[5\]](#).

To run the cURL request examples shown in this guide, copy each example from the HTML version of this guide directly to the command line or a script.

The following command is an example cURL command that provisions a server with an isolated network:

```
$ curl https://dfw.servers.api.rackspacecloud.com/v2/$account/servers \
-X POST \
-H "X-Auth-Project-Id: $account" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "X-Auth-Token: $token" \
-d '{"server": {"name": "my_server_with_network", "imageRef": "d42f821e-c2d1-4796-9f07-af5ed7912d0e", "flavorRef": "2", "max_count": 1, "min_count": 1, "networks": [{"uuid": "538a112a-34d1-47ff-bf1e-c40639e886e2"}, {"uuid": "00000000-0000-0000-0000-000000000000"}, {"uuid": "11111111-1111-1111-1111-111111111111"}]}}' \
| python -m json.tool
```



Note

The carriage returns in the cURL request examples are escaped with a backslash (\) character. The escape character allows continuation of the command across multiple lines. However, do not include the escape character in the JSON or XML request body within the cURL command.

The cURL examples in this guide use the following command-line options:

Table 1.1. cURL Command-Line Options

Option	Description
-d	Sends the specified data in a POST request to the HTTP server. Use this option to send a JSON or XML request body to the server.
-H	<p>Specifies an extra HTTP header in the request. You can specify any number of extra headers. Precede each header with the <code>-H</code> option.</p> <p>Common headers in Rackspace API requests are:</p> <ul style="list-style-type: none">• <code>Content-Type</code>. Required for operations with a request body. <p>Specifies the format of the request body. The syntax for the <code>Content-Type</code> header is:</p> <pre>Content-Type: application/format</pre> <p>Where <i>format</i> is either <code>json</code> or <code>xml</code>.</p> <ul style="list-style-type: none">• <code>X-Auth-Project-Id</code>. Optional. Specifies the project ID, which can be your account number or another value.• <code>Accept</code>. Optional. <p>Specifies the format of the response body. The syntax for the <code>Accept</code> header is:</p> <pre>Accept: application/format</pre> <p>Where <i>format</i> is either <code>json</code> or <code>xml</code>. Default is <code>json</code>.</p> <ul style="list-style-type: none">• <code>X-Auth-Token</code>. Required. Specifies the authentication token.
-i	Includes the HTTP header in the output.
-s	Silent or quiet mode. Does not show progress or error messages. Makes cURL mute.
-T	Transfers the specified local file to the remote URL.
-X	Specifies the request method to use when communicating with the HTTP server. The specified request is used instead of the default method, which is GET .



json.tool

For commands that return a response, you can append the following code to the command to call the `json.tool` to pretty-print output:

```
| python -m json.tool
```

To use the `json.tool`, import the `json` module. For information about the `json.tool`, see [json — JSON encoder and decoder](#).

If you run a Python version older than 2.6, import the simplejson module and use the simplejson.tool. For information about the simple.json tool, see [simplejson — JSON encoder and decoder](#).

If you do not want to pretty-print JSON output, omit this code.

Output: The cURL command returns information about the new server in XML format, as shown in the following example:

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="ed5c7754-29b6-45fa-96cb-ab64958c8c0a" adminPass="Dd5pNZtpVVQ3"
  OS-DCF:diskConfig="AUTO">
  <metadata/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ed5c7754-29b6-45fa-96cb-
ab64958c8c0a"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/servers/ed5c7754-29b6-45fa-96cb-
ab64958c8c0a"
    rel="bookmark"/>
</server>
```

1.3. Authenticate through the Rackspace Cloud Identity Service

To authenticate access to Rackspace Cloud services, issue an authentication request to a Rackspace Cloud Identity Service endpoint. The Rackspace Cloud Identity Service is an implementation of the OpenStack Keystone Identity Service v2.0.

In response to valid credentials, an authentication request to the Rackspace Cloud Identity Service returns an authentication token and a service catalog that contains a list of all services and endpoints available for this token. Because the authentication token expires after 24 hours, you must generate a token once a day.

The following sections list the Rackspace Cloud Identity Service endpoints, show you how make an authentication request, and describe the authentication response.

For detailed information about the OpenStack Keystone Identity Service v2.0, see [Cloud Identity Client Developer Guide API v2.0](#).

1.3.1. Rackspace Cloud Identity Service Endpoints

Multiple Rackspace Cloud Identity Service endpoints exist, based on your account location.

When you authenticate, use the appropriate endpoint, as follows:

Table 1.2. Rackspace Cloud Identity Service Endpoints

Account location	Rackspace Cloud Identity Service endpoint
US-based account	https://identity.api.rackspacecloud.com/v2.0
UK-based account	https://lon.identity.api.rackspacecloud.com/v2.0

For information about support for legacy identity endpoints, see [Alternate Authentication Endpoints](#).

1.3.2. Authentication Request

To authenticate, issue a **POST /tokens** request to the appropriate Rackspace Cloud Identity Service endpoint. See [Section 1.3.1, "Rackspace Cloud Identity Service Endpoints"](#) [6].

In the request body, supply one of the following sets of credentials:

- username and password
- username and API key

Your username and password are the ones that you use to log into the Rackspace Cloud control panel.

To obtain your API key, log into <http://mycloud.rackspace.com>, click your username, and select **API Keys** to get your key.

The following cURL examples show how to get an authentication token by entering either your username and password, or username and API key.

Example 1.2. Authenticate to US Identity Endpoint – Username and Password: JSON Request

```
$ curl -s https://identity.api.rackspacecloud.com/v2.0/tokens -X 'POST' \
-d '{"auth":{"passwordCredentials":{"username":"MyRackspaceAcct", "password":"MyRackspacePwd"}}}' \
-H 'Content-Type: application/json' | python -m json.tool
```

Example 1.3. Authenticate to US Identity Endpoint – Username and API Key: JSON Request

```
$ curl -s https://identity.api.rackspacecloud.com/v2.0/tokens -X 'POST' \
-d '{ "auth": { "RAX-KEY": apiKeyCredentials, "username": "MyRackspaceAcct", "apiKey": "00000000000000000000" } }' \
-H "Content-Type: application/json" | python -m json.tool
```



Note

In these examples, the following code is appended to the cURL commands to pretty-print the JSON output:

Example 1.4. Pretty Printing cURL Output

```
| python -m json.tool
```

1.3.3. Authentication Response

In response to valid credentials, your request returns an authentication token and a service catalog with the endpoints to request services.

Do not include explicit API endpoints in your scripts and applications. Instead, find the endpoint for your service and region.

The following output shows a partial authentication response in JSON format:

Example 1.5. Authenticate: JSON Response

```
{
  "access": {
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "internalURL": "https://snet-storage101.dfw1.clouddrive.com/v1/MossoCloudFS_530f8649-324c-499c-a075-2195854d52a7",
            "publicURL": "https://storage101.dfw1.clouddrive.com/v1/MossoCloudFS_530f8649-324c-499c-a075-2195854d52a7",
            "region": "DFW",
            "tenantId": "MossoCloudFS_530f8649-324c-499c-a075-2195854d52a7"
          },
          {
            "internalURL": "https://snet-storage101.ord1.clouddrive.com/v1/MossoCloudFS_530f8649-324c-499c-a075-2195854d52a7"
          }
        ]
      }
    ]
  }
}
```

```
a075-2195854d52a7",
    "publicURL": "https://storage101.ord1.clouddrive.com/v1/MossoCloudFS_530f8649-324c-499c-
    "region": "ORD",
    "tenantId": "MossoCloudFS_530f8649-324c-499c-a075-2195854d52a7"
  },
  {
    "name": "cloudFiles",
    "type": "object-store"
  },
  {
    "endpoints": [
      {
        "publicURL": "https://servers.api.rackspacecloud.com/v1.0/010101",
        "tenantId": "010101",
        "versionId": "1.0",
        "versionInfo": "https://servers.api.rackspacecloud.com/v1.0",
        "versionList": "https://servers.api.rackspacecloud.com/"
      }
    ],
    "name": "cloudServers",
    "type": "compute"
  },
  ...
  {
    "endpoints": [ ❶
      {
        "publicURL": "https://dfw.servers.api.rackspacecloud.com/v2/010101",
        "region": "DFW",
        "tenantId": "010101", ❷
        "versionId": "2",
        "versionInfo": "https://dfw.servers.api.rackspacecloud.com/v2",
        "versionList": "https://dfw.servers.api.rackspacecloud.com/"
      },
      {
        "publicURL": "https://ord.servers.api.rackspacecloud.com/v2/010101",
        "region": "ORD",
        "tenantId": "010101",
        "versionId": "2",
        "versionInfo": "https://ord.servers.api.rackspacecloud.com/v2",
        "versionList": "https://ord.servers.api.rackspacecloud.com/"
      }
    ],
    "name": "cloudServersOpenStack", ❸
    "type": "compute"
  },
  ...
  ],
  "token": {
    "expires": "2012-09-14T15:11:57.585-05:00", ❹
    "id": "858fb4c2-bf15-4dac-917d-8ec750ae9baa", ❺
    "tenant": {
      "id": "010101",
      "name": "010101"
    }
  },
  "user": {
    "RAX-AUTH:defaultRegion": "DFW",
    "id": "170454",
    "name": "MyRackspaceAcct",
    "roles": [
      {
        "description": "User Admin Role.",
        "id": "3",
        "name": "identity:user-admin"
      }
    ]
  }
}
```

Successful authentication returns the following information:

- ❶ **Endpoints to request Rackspace Cloud services.** Appears in the `endpoints` element in the `serviceCatalog` element.

Endpoints information includes the public URL, which is the endpoint that you use to access the service, region, tenant ID, and version information.

To access the Cloud Networks or next generation Cloud Servers service, use the endpoint for the `cloudServersOpenStack` service.

- ❷ **Tenant ID.** Appears in the `tenantId` field in the `endpoints` element. Also known as the account number.

You include the tenant ID in the endpoint URL when you call a Cloud service.

In the following example, you export the tenant ID, 010101, to the `account` environment variable and the authentication token to the `token` environment variable. Then, you issue a cURL command, as follows:

```
$ export account="010101"
$ export token="00000000-0000-0000-000000000000"
$ curl -s https://dfw.servers.api.rackspacecloud.com/v2/$account/images/detail \
-H "X-Auth-Token: $token" | python -m json.tool
```

- ③ **The name of the service.** Appears in the `name` field.

Locate the correct service name in the service catalog, as follows:

- **First generation Cloud Servers.** Named `cloudServers` in the catalog.

If you use the authentication token to access this service, you can view and perform first generation Cloud Servers API operations against your first generation Cloud Servers.

- **Cloud Networks or next generation Cloud Servers.** Named `cloudServersOpenStack` in the catalog.

To access the Cloud Networks or next generation Cloud Servers service, use the `publicURL` value for the `cloudServersOpenStack` service.

Might show multiple endpoints to enable regional choice. Select the appropriate endpoint for the region that you want to interact with by examining the `region` field.

If you use the authentication token to access this service, you can view and perform Cloud Networks or next generation Cloud Servers API operations against your next generation Cloud Servers. To complete Cloud Networks API operations, you must also get access to this service. To request access, click [here](#).

- ④ **Expiration date and time for authentication token.** Appears in the `expires` field in the `token` element.

After this date and time, the token is no longer valid.

This field predicts the maximum lifespan for a token, but does not guarantee that the token reaches that lifespan.

Clients are encouraged to cache a token until it expires.

Because the authentication token expires after 24 hours, you must generate a token once a day.

- ⑤ **Authentication token.** Appears in the `id` field in the `token` element.

You pass the authentication token in the `X-Auth-Token` header each time that you send a request to a service.

In the following example, you export the tenant ID, 010101, to the `account` environment variable. You also export the authentication token, 00000000-0000-0000-000000000000, to the `token` environment variable. Then, you issue a cURL command, as follows:

```
$ export account="010101"  
$ export token="00000000-0000-0000-000000000000"  
$ curl -s https://dfw.servers.api.rackspacecloud.com/v2/$account/images/detail \  
  -H "X-Auth-Token: $token" | python -m json.tool
```

1.4. Request/Response Types

API v2 supports both the JSON and XML data serialization request and response formats.

You specify the request format in the `Content-Type` header in the request. This header is required for operations that have a request body. The syntax for the `Content-Type` header is:

```
Content-Type: application/format
```

Where *format* is either `json` or `xml`.

You specify the response format by using one of the following methods:

- **Accept header.** The syntax for the `Accept` header is:

```
Accept: application/format
```

Where *format* is either `json` or `xml`.

Default is `json`.

- **Query extension.** Add an `.xml` or `.json` extension to the request URI. For example, the `.xml` extension in the following URI request specifies that the response body is returned in XML format:

```
POST /v2/010101/servers.xml
```

If you do not specify a response format, JSON is the default.

If you specify conflicting formats in the `Accept` header and the query extension, the format specified in the query extension takes precedence. For example, if the query extension is `.xml` and the `Accept` header specifies `application/json`, the response is returned in XML format.

You can serialize a response in a different format from the request format. [Example 1.6, "Request with Headers: JSON" \[11\]](#) and [Example 1.7, "Response with Headers: XML" \[13\]](#) show a request body in JSON format and a response body in XML format.

Example 1.6. Request with Headers: JSON

```
POST /v2/010101/servers HTTP/1.1
Host: dfw.servers.api.rackspacecloud.com
Content-Type: application/json
Accept: application/xml
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
{
  "server" : {
    "name" : "api-test-server-1",
    "imageRef" : "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "flavorRef" : "2",

    "OS-DCF:diskConfig" : "AUTO",
    "metadata" : {
      "My Server Name" : "API Test Server 1"
    },
  },
}
```

```
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcYBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGl1tcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHROZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRO
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcYwgYW5kIHlv
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCgOKLVJpY2hhcmQgQmFjaA=="
      }
    ],
    "networks": [
      {
        "uuid": "4ebd35cf-bfe7-4d93-b0d8-eb468ce2245a"
      },
      {
        "uuid": "00000000-0000-0000-0000-000000000000"
      },
      {
        "uuid": "11111111-1111-1111-1111-111111111111"
      }
    ]
  }
}
```

[Example 1.7, “Response with Headers: XML” \[13\]](#) shows the headers and XML response returned by the JSON request:

Example 1.7. Response with Headers: XML

```
HTTP/1.1 202 Accepted
Date: Mon, 23 Jul 2012 20:24:48 GMT
Content-Length: 582
Location: https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/06dba123-2c7e-4639-bea0-09f8e219b056
Content-Type: application/xml
X-Compute-Request-Id: req-ab05045a-452f-4b46-be0d-86494da02a2b
Server: Jetty(8.0.y.z-SNAPSHOT)
```

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="ed5c7754-29b6-45fa-96cb-ab64958c8c0a" adminPass="Dd5pNZtpVVQ3"
  OS-DCF:diskConfig="AUTO">
  <metadata/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
ed5c7754-29b6-45fa-96cb-ab64958c8c0a"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/servers/
ed5c7754-29b6-45fa-96cb-ab64958c8c0a"
    rel="bookmark"/>
</server>
```

The following example shows an alternative method of achieving the same result. The following request uses an URI extension of `.xml` instead of an `Accept` header to request an XML response.



Note

The XML response is not shown.

Example 1.8. JSON Request with XML Query Extension for the Response

```
POST /v2/010101/servers.xml HTTP/1.1
Host: dfw.servers.api.rackspacecloud.com
Content-Type: application/json
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
{
  "server" : {
    "name" : "api-test-server-1",
    "imageRef" : "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "flavorRef" : "2",

    "OS-DCF:diskConfig" : "AUTO",
    "metadata" : {
      "My Server Name" : "API Test Server 1"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBBrbm93IHdoeSBp
dCBtb3ZlcyBpb1BqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGV1ZC4uLk10IGZlZlZWxzIGFuIGl1tcHVz"
```

```
c2lvbi4uLnRoXMgaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHROZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRO
ZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcycwYmV5b3Vh
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbG1mdCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA== "

    }
  ],
  "networks": [
    {
      "uuid": "4ebd35cf-bfe7-4d93-b0d8-eb468ce2245a"
    },
    {
      "uuid": "00000000-0000-0000-0000-000000000000"
    },
    {
      "uuid": "11111111-1111-1111-1111-111111111111"
    }
  ]
}
```


1.5. Links and References

Resources often need to refer to other resources. For example, when you create a server, you must specify the image from which to build the server. You can specify the image by providing an ID or a URL to a remote image. When you provide an ID for a resource, it is assumed that the resource exists in the current endpoint.

Example 1.9. Image Reference in Create Server Request: JSON Request

```
{
  "server" : {
    "name" : "api-test-server-1",
    "imageRef" : "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "flavorRef" : "2",

    "OS-DCF:diskConfig" : "AUTO",
    "metadata" : {
      "My Server Name" : "API Test Server 1"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGl1tcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBpZWhpbmQgYWxsIGNsb3VkcycwYmV5b25kIGhvcml6
dSB3aWxsIGtub3csIHRvbywgZ2h1b25kIGtub3V5b25kIGhvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
      }
    ],
    "networks": [
      {
        "uuid": "4ebd35cf-bfe7-4d93-b0d8-eb468ce2245a"
      },
      {
        "uuid": "00000000-0000-0000-0000-000000000000"
      },
      {
        "uuid": "11111111-1111-1111-1111-111111111111"
      }
    ]
  }
}
```

Example 1.10. Full Image Reference in Create Server Request: JSON Request

```
{
  "server" : {
    "name" : "myUbuntuServer",
    "imageRef" : "https://dfw.servers.api.rackspacecloud.com/v2/010101/images/
3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "flavorRef" : "6",
    "metadata" : {
      "My Server Name" : "Ubuntu 11.10 server"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGl1tcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
```

```
        ZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcYwgYW5kIHlv
        dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmCB5b3Vy
        c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
        b25zLiINCg0KLvJpY2hhcmQgQmFjaA==
    }
  ],
  "networks": [
    {
      "uuid": "4ebd35cf-bfe7-4d93-b0d8-eb468ce2245a"
    },
    {
      "uuid": "00000000-0000-0000-0000-000000000000"
    },
    {
      "uuid": "11111111-1111-1111-1111-111111111111"
    }
  ]
}
```

Example 1.11. Image Reference in Create Server Request: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001"
  flavorRef="6"
  diskConfig="AUTO"
  name="new-server-test"
  >
  <metadata>
    <meta key="My Server Name">Server test</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHNlY2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHNlY2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHV5
      c2lubi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcYwgYW5kIHlv
      dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCg0KLvJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>
```

Example 1.12. Full Image Reference in Create Server Request: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="https://dfw.servers.api.rackspacecloud.com/v2/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001"
  flavorRef="6"
  OS-DCF:diskConfig="AUTO"
  name="new-server-test">
  <metadata>
    <meta key="My Server Name">Server test</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHNlY2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHNlY2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHV5
      c2lubi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcYwgYW5kIHlv
      dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCg0KLvJpY2hhcmQgQmFjaA== </file>
    </personality>
</server>
```

```
</server>
```

For convenience, resources contain links to themselves. This allows a client to easily obtain resource URIs rather than to construct them. The following kinds of link relations are associated with resources:

- **self**. Contains a versioned link to the resource. Use these links when the link will be followed immediately.
- **bookmark**. Provides a permanent link to a resource that is appropriate for long-term storage.
- **alternate**. Contains an alternate representation of the resource. For example, a Cloud Servers image might have an alternate representation in the Cloud Servers image service.

In the following examples, the `rel` attribute shows the type of representation to expect when following the link.

Example 1.13. Server with Self Links: JSON

```
{
  "server" : {
    "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
    "name" : "my-server",
    "links": [
      {
        "rel" : "self",
        "href" : "http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "bookmark",
        "href" : "http://dfw.servers.api.rackspacecloud.com/010101/servers/52415800-8b69-11e0-9b19-734f5736d2a2"
      }
    ]
  }
}
```

Example 1.14. Server with Alternate Link: JSON

```
{
  "image" : {
    "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
    "name" : "My Server Backup",
    "links": [
      {
        "rel" : "self",
        "href" : "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "bookmark",
        "href" : "http://dfw.servers.api.rackspacecloud.com/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "alternate",
        "type" : "application/vnd.openstack.image",
        "href" : "http://glance.api.rackspacecloud.com/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      }
    ]
  }
}
```

Example 1.15. Server with Self Links: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f5736d2a2" name="my-server">
  <atom:link rel="self"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
52415800-8b69-11e0-9b19-734f5736d2a2" />
  <atom:link rel="bookmark"
    href="http://dfw.servers.api.rackspacecloud.com/010101/servers/
52415800-8b69-11e0-9b19-734f5736d2a2"
  />
</server>
```

Example 1.16. Image with Alternate Link: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<image xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f5736d2a2" name="My Server Backup">
  <atom:link rel="self"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/
52415800-8b69-11e0-9b19-734f5736d2a2" />
  <atom:link rel="bookmark"
    href="http://dfw.servers.api.rackspacecloud.com/010101/images/
52415800-8b69-11e0-9b19-734f5736d2a2" />
  <atom:link rel="alternate" type="application/vnd.openstack.image"
    href="http://glance.api.rackspacecloud.com/010101/images/
52415800-8b69-11e0-9b19-734f5736d2a2"
  />
</image>
```

1.6. Paginated Collections

To reduce load on the service, list operations return a maximum number of items at a time. The maximum number of items returned is 1000.

To navigate the collection, you can set the *limit* and *marker* parameters in the URI request. For example:

```
?limit=100&marker=1234
```

The *marker* parameter is the ID of the last item in the previous list. Items are sorted by create time in descending order. When a create time is not available, the items are sorted by ID. A marker with an ID that is not valid returns an `itemNotFound` (404) fault.

The *limit* parameter sets the page size. If the client specifies a *limit* value that is greater than the supported limit, an `overLimit` (413) fault might be thrown.

Both parameters are optional.



Note

Paginated collections never return `itemNotFound` (404) faults when the collection is empty — clients should expect an empty collection.

For convenience, collections contain atom "next" links and can optionally contain "previous" links. The last page in the list contains a "next" link.

The following examples show pages in a collection of images.

To get the first page, issue a **GET** request to the following endpoint and set the *limit* parameter to the page size of a single item:

```
http://dfw.servers.api.rackspacecloud.com/v2/010101/images?limit=1
```

Subsequent links honor the initial page size. A client can follow links to traverse a paginated collection.

1.6.1. JSON Collection

In JSON, members in a paginated collection are stored in a JSON array named after the collection. A JSON object can also hold members in cases where using an associative array is more practical. Properties about the collection itself, including links, are contained in an array with the name of the entity an underscore (_) and `links`. The combination of the objects and arrays that start with the name of the collection and an underscore represent the collection in JSON.

This approach allows for extensibility of paginated collections by allowing them to be associated with arbitrary properties. It also allows collections to be embedded in other objects.

Example 1.17. Images Collection – First Page: JSON

```
{
  "images": [
    {
      "id": "52415800-8b69-11e0-9b19-734f6f006e54",
      "name": "CentOS 5.2",
      "links": [
        {
          "rel": "self",
          "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f6f006e54"
        }
      ]
    }
  ],
  "images_links": [
    {
      "rel": "next",
      "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images?limit=1&marker=52415800-8b69-11e0-9b19-734f6f006e54"
    }
  ]
}
```

Example 1.18. Images Collection – Second Page: JSON

```
{
  "images": [
    {
      "id": "52415800-8b69-11e0-9b19-734f5736d2a2",
      "name": "My Server Backup",
      "links": [
        {
          "rel": "self",
          "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
        }
      ]
    }
  ],
  "images_links": [
    {
      "rel": "next",
      "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images?limit=1&marker=52415800-8b69-11e0-9b19-734f5736d2a2"
    }
  ]
}
```

Example 1.19. Images Collection – Last Page: JSON

```
{
```

```
    "images": [
      {
        "id": "52415800-8b69-11e0-9b19-734f6ff7c475",
        "name": "Backup 2",
        "links": [
          {
            "rel": "self",
            "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f6ff7c475"
          }
        ]
      }
    ]
  }
}
```

1.6.2. XML Collection

Example 1.20. Images Collection – First Page: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f6f006e54" name="CentOS 5.2">
    <atom:link rel="self"
      href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f6f006e54"
    />
  </image>
  <atom:link rel="next"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images?limit=1&marker=52415800-8b69-11e0-9b19-734f6f006e54"
  />
</images>
```

Example 1.21. Images Collection – Second Page: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f5736d2a2"
    name="My Server Backup">
    <atom:link rel="self"
      href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
    />
  </image>
  <atom:link rel="next"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images?limit=1&marker=52415800-8b69-11e0-9b19-734f5736d2a2"
  />
</images>
```

Example 1.22. Images Collection – Last Page: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f6ff7c475" name="Backup 2">
    <atom:link rel="self"
      href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f6ff7c475"
    />
  </image>
</images>
```

1.7. Efficient Polling with the *Changes-Since* Parameter

You can poll for the status of certain operations by issuing a **GET** request on various elements. Rather than re-downloading and re-parsing the full status at each polling interval, you can use the *changes-since* parameter to check for changes since a previous request. The *changes-since* time is specified as an [ISO 8601](#) dateTime (2011-01-24T17:08Z).

The format for the timestamp is:

```
CCYY-MM-DDThh:mm:ss
```

Optionally, to return the time zone as an offset from UTC, append the following:

```
±hh:mm
```

If you omit the time zone (2011-01-24T17:08), the UTC time zone is assumed.

If data has changed, only the items changed since the specified time are returned in the response.

If date has not changed since the *changes-since* time, an empty list is returned.

For example, issue a **GET** request against the following endpoint to list all servers that have changed since Mon, 24 Jan 2011 17:08:00 UTC:

```
https://dfw.servers.api.rackspacecloud.com/v2/010101/servers?changes-since=2011-01-24T17:08:00Z
```

To enable you to keep track of changes, the *changes-since* filter also displays images and servers that have been deleted provided that the *changes-since* filter specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed.

1.8. Limits

Accounts are configured with thresholds, or limits, that manage capacity and prevent abuse of the system.

The system recognizes the following types of limits:

- *rate limits*. Control the frequency at which the user can issue specific API requests. See [Section 1.8.1, “Rate Limits” \[23\]](#).
- *absolute limits*. Control the total number of specific objects that the user can possess simultaneously. See [Section 1.8.2, “Absolute Limits” \[24\]](#).

To query the limits programmatically, see [Section 1.8.3, “Get Limits” \[25\]](#).

1.8.1. Rate Limits

Rate limits control the frequency at which the user can issue specific API requests.

Rate limits are reset after a certain amount of time passes. To request a rate limit increase, contact Rackspace.

Rate limits are specified in terms of both a human-readable wild-card URI and a machine-processable regular expression. The human-readable limit is intended for displaying in graphical user interfaces. The machine-processable form is intended to be used directly by client applications.

The regular expression boundary matcher `^` for the rate limit takes effect after the root URI path. For example, the regular expression `^/servers` would match the bolded portion of the following URI:

```
https://dfw.servers.api.rackspacecloud.com/v2/010101/servers
```

The following table lists the default rate limits:

Table 1.3. Default Rate Limits

Verb	URI	Value	Unit
GET	*	100	minute
GET	/os-networksv2	0	minute
GET	/servers	1000	day
POST	*	100	minute
POST	/os-networksv2	0	day
POST	/servers	1000	day

You can also query the limits programmatically. See [Section 1.8.3, “Get Limits” \[25\]](#).

When a request exceeds the limits established for your account, a 413 HTTP response is returned with a `Retry-After` header that indicates when you can attempt the request again.

1.8.2. Absolute Limits

Absolute limits control the total number of specific objects that the user can possess simultaneously.

Specify absolute limits to limit the overall number of items or amount of capacity in the system. Absolute limits also include the amount of resources currently consumed, which allow for programmatic visibility of usage.

Specify absolute limits as name/value pairs.

The following tables describe absolute limits.

The following `max` limits show the maximum amount of a resource that can be used:

Table 1.4. Absolute Max Limits

Name	Value	Description
maxImageMeta	20	The maximum number of metadata key value pairs associated with a particular image.
maxPersonality	6	The maximum number of file path/content pairs that can be supplied on server build and rebuild.
maxPersonalitySize	10240	The maximum size, in bytes, for each personality file.
maxServerMeta	20	The maximum number of metadata key value pairs associated with a particular server.
maxTotalCores	-1	This limit is disabled. No limits exist on the total number of cores.
maxTotalInstances	100	The maximum number of Cloud Servers at any one time.
maxTotalPrivateNetworks	0	Cloud Networks is disabled.
	Greater than 0	Cloud Networks is enabled. The maximum number of isolated networks that you can create.
maxTotalRAMSize	66560	The maximum total amount of RAM (MB) of all Cloud Servers at any one time.

The following `total` limits show current usage:

Table 1.5. Absolute Total Limits

Name	Value
totalCoresUsed	The total number of cores used.
totalInstancesUsed	The total number of Cloud Servers.
totalRAMUsed	The total amount of RAM (GB) used for all Cloud Servers.

1.8.3. Get Limits

Verb	URI	Description
GET	/limits	Gets the current rate and absolute limits for your account.

Normal Response Codes: 200, 203

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), Method Not Allowed (405), overLimit (413)

Applications can programmatically determine current account limits by using this API operation.

This operation does not require a request body.

This operation returns a response body.

Example 1.23. Get Limits: JSON Response

```
{
  "limits": {
    "absolute": {
      "maxImageMeta": 20,
      "maxPersonality": 6,
      "maxPersonalitySize": 10240,
      "maxServerMeta": 20,
      "maxTotalCores": -1,
      "maxTotalFloatingIps": 5,
      "maxTotalInstances": 100,
      "maxTotalKeypairs": 100,
      "maxTotalPrivateNetworks": 0,
      "maxTotalRAMSize": 66560,
      "maxTotalVolumeGigabytes": -1,
      "maxTotalVolumes": 0,
      "totalCoresUsed": 9,
      "totalInstancesUsed": 3,
      "totalKeypairsUsed": 0,
      "totalPrivateNetworksUsed": 0,
      "totalRAMUsed": 16896,
      "totalSecurityGroupsUsed": 0,
      "totalVolumeGigabytesUsed": 0,
      "totalVolumesUsed": 0
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-09-10T20:11:45.146Z",
            "remaining": 0,
            "unit": "DAY",
            "value": 0,
            "verb": "POST"
          }
        ],
        "next-available": "2012-09-10T20:11:45.146Z",
        "remaining": 0,
        "unit": "MINUTE",
        "value": 0,
        "verb": "GET"
      }
    ],
    "regex": "/v[^/](\\d+)/(rax-networks)/?.*",
    "uri": "/rax-networks"
  },
  {
    "limit": [
      {
        "next-available": "2012-09-10T20:11:45.146Z",
        "remaining": 1000,
        "unit": "DAY",
```

```
        "value": 1000,  
        "verb": "POST"  
    },  
    ],  
    "regex": "/v[^/](\\d+)/servers)?.*",  
    "uri": "/servers"  
},  
{  
    "limit": [  
        {  
            "next-available": "2012-09-10T20:11:45.146Z",  
            "remaining": 100,  
            "unit": "MINUTE",  
            "value": 100,  
            "verb": "ALL"  
        }  
    ],  
    "regex": "/v[^/](\\d+)/?.*",  
    "uri": ""  
}  
]  
}
```

Example 1.24. Get Limits: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<limits xmlns:lim="http://docs.openstack.org/common/api/v1.0"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <rates>
    <rate regex="/v[^/](\d+)/(\rax-networks)/?.*"
      uri="/rax-networks">
      <limit next-available="2012-09-10T20:14:17.997Z"
        unit="DAY" remaining="0" value="0" verb="POST"/>
      <limit next-available="2012-09-10T20:14:17.997Z"
        unit="MINUTE" remaining="0" value="0" verb="GET"/>
    </rate>
    <rate regex="/v[^/](\d+)/(\servers)/?.*" uri="/servers">
      <limit next-available="2012-09-10T20:14:17.997Z"
        unit="DAY" remaining="1000" value="1000" verb="POST"/>
    </rate>
    <rate regex="/v[^/](\d+)/?.*" uri="*">
      <limit next-available="2012-09-10T20:14:17.997Z"
        unit="MINUTE" remaining="100" value="100" verb="ALL"/>
    </rate>
  </rates>
  <absolute
    xmlns:os-used-limits="http://docs.openstack.org/compute/ext/used_limits/api/v1.1"
    xmlns:atom="http://www.w3.org/2005/Atom">
    <limit name="maxServerMeta" value="20"/>
    <limit name="maxTotalInstances" value="100"/>
    <limit name="maxPersonality" value="6"/>
    <limit name="totalPrivateNetworksUsed" value="0"/>
    <limit name="maxImageMeta" value="20"/>
    <limit name="maxPersonalitySize" value="10240"/>
    <limit name="totalVolumesUsed" value="0"/>
    <limit name="maxTotalPrivateNetworks" value="0"/>
    <limit name="maxTotalKeypairs" value="100"/>
    <limit name="totalCoresUsed" value="9"/>
    <limit name="maxTotalVolumes" value="0"/>
    <limit name="totalRAMUsed" value="16896"/>
    <limit name="totalInstancesUsed" value="3"/>
    <limit name="totalVolumeGigabytesUsed" value="0"/>
    <limit name="maxTotalCores" value="-1"/>
    <limit name="totalSecurityGroupsUsed" value="0"/>
    <limit name="maxTotalFloatingIps" value="5"/>
    <limit name="totalKeyPairsUsed" value="0"/>
    <limit name="maxTotalVolumeGigabytes" value="-1"/>
    <limit name="maxTotalRAMSize" value="66560"/>
  </absolute>
</limits>
```

1.9. Versions

You can list the API versions that are available for your account or get details for a specified version.

1.9.1. List Versions

You can list which API versions are available for your account.

Example 1.25. List Versions: Request

```
GET HTTP/1.1
Host: dfw.servers.api.rackspacecloud.com/
```

Normal Response Codes: 200, 203

Error Response Codes: 400, 413, 500, 503

Issue a **GET** request to the root endpoint for a service. In the request, truncate the version and everything to the right of it. For example:

```
https://dfw.servers.api.rackspacecloud.com/
```

This operation does not require a request body.

This operation returns a response body.

Example 1.26. List Versions: cURL with JSON Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/ \
-H "Accept: application/json" \
-H "X-Auth-Token: $token" | python -m json.tool
```



Note

For information about the `json.tool`, see [json.tool \[3\]](#).

Example 1.27. List Versions: JSON Response

```
{
  "versions": [
    {
      "id": "v2",
      "links": [
        {
          "href": "http://dfw.servers.api.rackspacecloud.com/v2",
          "rel": "self"
        }
      ],
      "status": "CURRENT"
    }
  ]
}
```

Example 1.28. List Versions: cURL with XML Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/.xml \
-H "X-Auth-Token: $token"
```

Example 1.29. List Versions: XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<versions xmlns="http://docs.openstack.org/common/api/v1.0"
  xmlns:ns2="http://www.w3.org/2005/Atom"
  xmlns:ns3="http://docs.rackspacecloud.com/power-api/versioning/v1.0">
  <version status="CURRENT" id="v2">
    <media-types/>
    <ns2:link
      href="http://dfw.servers.api.rackspacecloud.com/v2"
      rel="self"/>
  </version>
</versions>
```

1.9.2. Get Version Details

You can get details about an API version for a specific version.

Example 1.30. Get Version Details: Request

```
GET HTTP/1.1
Host: dfw.servers.api.rackspacecloud.com/v2/
```

Normal Response Codes: 200, 203

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), Method Not Allowed (405), overLimit (413)

Issue a **GET** request to the root endpoint for a service and include the version in the request.

For example:

```
https://dfw.servers.api.rackspacecloud.com/v2/
```

Always end version request URLs with a trailing slash (/). If you omit the slash, the server might respond with a 302 redirection request.

You can specify a query extension after the slash in the request. For example:

```
https://dfw.servers.api.rackspacecloud.com/v2/.xml
```



Note

This is a special case that does not hold true for other API requests. In general, requests such as `/servers.xml` and `/servers/.xml` are handled equivalently.

This operation does not require a request body.

This operation returns a response body.

Example 1.31. Get Version Details: cURL with JSON Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/v2/ \
-X GET \
-H "Accept: application/json" \
-H "X-Auth-Token: $token" | python -m json.tool
```

Example 1.32. Get Version Details: JSON Response

```
{
  "version": {
    "id": "v2",
    "status": "CURRENT"
  }
}
```

Example 1.33. Get Version Details: cURL with XML Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/v2/.xml \
-H "X-Auth-Token: $token"
```

Example 1.34. Get Version Details: XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<version xmlns="http://docs.openstack.org/common/api/v1.0"
  xmlns:ns2="http://www.w3.org/2005/Atom"
  xmlns:ns3="http://docs.rackspacecloud.com/power-api/versioning/v1.0"
  status="CURRENT" id="v2">
  <media-types/>
</version>
```


1.10. Extensions

The OpenStack Compute API is extensible and Rackspace has implemented several extensions. You can list available extensions and get details for a specific extension.

1.10.1. List Extensions

Verb	URI	Description
GET	/extensions	Lists available extensions.

Normal Response Codes: 200, 203

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), Method Not Allowed (405), overLimit (413)

Applications can programmatically determine which extensions are available by issuing a **GET** on the /extensions URI.

This operation does not require a request body.

This operation returns a response body. In the response body, each extension is identified by two unique identifiers, a namespace and an alias. Additionally an extension contains documentation links in various formats.

Example 1.35. Get Extensions: cURL with JSON Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/v2/$account/extensions \
-H "X-Auth-Token: $token" | python -m json.tool
```

Example 1.36. Get Extensions: JSON Response

```
{
  "extensions": [
    {
      "alias": "OS-DCF",
      "description": "Disk Management Extension",
      "links": [],
      "name": "DiskConfig",
      "namespace": "http://docs.openstack.org/compute/ext/disk_config/api/v1.1",
      "updated": "2011-09-27T00:00:00+00:00"
    },
    {
      "alias": "os-volumes",
      "description": "Volumes support",
      "links": [],
      "name": "Volumes",
      "namespace": "http://docs.openstack.org/compute/ext/volumes/api/v1.1",
      "updated": "2011-03-25T00:00:00+00:00"
    },
    {
      "alias": "rax-bandwidth",
      "description": "Server Bandwidth Extension",
      "links": [],
      "name": "ServerBandwidth",
      "namespace": "http://docs.rackspace.com/servers/api/ext/server_bandwidth/",
      "updated": "2012-01-19T00:00:00+00:00"
    },
    {
      "alias": "os-consoles",
      "description": "Interactive Console support.",
      "links": [],

```

```
    "name": "Consoles",
    "namespace": "http://docs.openstack.org/compute/ext/os-consoles/api/v2",
    "updated": "2011-12-23T00:00:00+00:00"
  },
  {
    "alias": "rax-networks",
    "description": "Admin-only Network Management Extension",
    "links": [],
    "name": "RAXNetworks",
    "namespace": "http://docs.openstack.org/ext/services/api/v1.1",
    "updated": "2012-03-07T09:46:43-05:00"
  },
  {
    "alias": "os-rescue",
    "description": "Instance rescue mode",
    "links": [],
    "name": "Rescue",
    "namespace": "http://docs.openstack.org/compute/ext/rescue/api/v1.1",
    "updated": "2011-08-18T00:00:00+00:00"
  },
  {
    "alias": "os-used-limits",
    "description": "Provide data on limited resources that are being used.",
    "links": [],
    "name": "Used Limits",
    "namespace": "http://docs.openstack.org/compute/ext/used_limits/api/v1.1",
    "updated": "2012-07-13T00:00:00+00:00"
  }
]
```

Example 1.37. Get Extensions: cURL with XML Request

```
curl -i https://dfw.servers.api.rackspacecloud.com/v2/$account/extensions.xml \
-H "X-Auth-Token: $token"
```

Example 1.38. Get Extensions: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<extensions xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <extension alias="OS-DCF" updated="2011-09-27T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
    name="DiskConfig">
    <description>Disk Management Extension</description>
  </extension>
  <extension alias="os-volumes" updated="2011-03-25T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/volumes/api/v1.1"
    name="Volumes">
    <description>Volumes support</description>
  </extension>
  <extension alias="rax-bandwidth"
    updated="2012-01-19T00:00:00+00:00"
    namespace="http://docs.rackspace.com/servers/api/ext/server_bandwidth/"
    name="ServerBandwidth">
    <description>Server Bandwidth Extension</description>
  </extension>
  <extension alias="os-consoles" updated="2011-12-23T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-consoles/api/v2"
    name="Consoles">
    <description>Interactive Console support.</description>
  </extension>
  <extension alias="rax-networks"
    updated="2012-03-07T09:46:43-05:00"
    namespace="http://docs.openstack.org/ext/services/api/v1.1"
    name="RAXNetworks">
    <description>Admin-only Network Management
      Extension</description>
  </extension>
  <extension alias="os-rescue" updated="2011-08-18T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/rescue/api/v1.1"
    name="Rescue">
    <description>Instance rescue mode</description>
  </extension>
  <extension alias="os-used-limits"
    updated="2012-07-13T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/used_limits/api/v1.1"
    name="Used Limits">
    <description>Provide data on limited resources that are being
      used.</description>
  </extension>
</extensions>
```

1.10.2. Get Extension Details

You can also query extensions by their unique alias to determine if an extension is available. An unavailable extension issues an `itemNotFound` (404) response.

Verb	URI	Description
GET	<code>/extensions/<i>alias</i></code>	Gets details about a specific extension.

Normal Response Codes: 200, 203

Error Response Codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `unauthorized` (401), `forbidden` (403), `badRequest` (400), `Method Not Allowed` (405), `overLimit` (413), `itemNotFound` (404)

This operation does not require a request body.

This operation returns a response body.

The following examples show how to query the OS-DCF extension:

Example 1.39. Get Extension: cURL with JSON Request

```
curl -s https://dfw.servers.api.rackspacecloud.com/v2/$account/extensions/OS-DCF \
-H "X-Auth-Token: $token" | python -m json.tool
```

Example 1.40. Get Extension: JSON Response

```
{
  "extension": {
    "alias": "OS-DCF",
    "description": "Disk Management Extension",
    "links": [],
    "name": "DiskConfig",
    "namespace": "http://docs.openstack.org/compute/ext/disk_config/api/v1.1",
    "updated": "2011-09-27T00:00:00+00:00"
  }
}
```

Example 1.41. Get Extension: cURL with XML Request

```
curl -i https://dfw.servers.api.rackspacecloud.com/v2/$account/extensions/OS-DCF.xml \
-H "X-Auth-Token: $token"
```

Example 1.42. Get Extension: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<extension xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0" alias="OS-DCF"
  updated="2011-09-27T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  name="DiskConfig">
  <description>Disk Management Extension</description>
</extension>
```

1.10.3. Extended Responses and Actions

Use extensions to define new data types, parameters, actions, headers, states, and resources.

In XML, you can define additional elements and attributes. Define these elements in the namespace for the extension.

In JSON, you must use the alias. The volumes element in the [Examples 1.43 \[35\]](#) and [1.44 \[36\]](#) is defined in the RS-CBS namespace.

Actions work in exactly the same manner as illustrated in [Examples 1.45 \[37\]](#) and [1.46 \[37\]](#). Extended headers are always prefixed with X- followed by the alias and a dash: (X-RS-CBS-HEADER1). You must prefix states and parameters with the extension alias followed by a colon. For example, an image can be in the RS-PIE:PrepareShare state.



Important

Applications should be prepared to ignore response data that contains extension elements. An extended state should always be treated as an UNKNOWN state if the application does not support the extension. Applications should also verify that an extension is available before submitting an extended request.

Example 1.43. Extended Server: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<servers xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <server id="52415800-8b69-11e0-9b19-734f6af67565" tenant_id="010101"
    user_id="5678" name="sample-server" status="BUILD"
    progress="60" hostId="e4d909c290d0fblca068ffaddf22cbd0"
    updated="2010-10-10T12:00:00Z" created="2010-08-10T12:00:00Z"
    accessIPv4="67.23.10.132" accessIPv6="::babe:67.23.10.132">
    <image id="52415800-8b69-11e0-9b19-734f6f006e54">
      <atom:link rel="self"
        href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/
52415800-8b69-11e0-9b19-734f6f006e54" />
      <atom:link rel="bookmark"
        href="http://dfw.servers.api.rackspacecloud.com/010101/images/
52415800-8b69-11e0-9b19-734f6f006e54"
      />
    </image>
    <flavor id="52415800-8b69-11e0-9b19-734f216543fd">
      <atom:link rel="self"
        href="http://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/
52415800-8b69-11e0-9b19-734f216543fd" />
      <atom:link rel="bookmark"
        href="http://dfw.servers.api.rackspacecloud.com/010101/flavors/
52415800-8b69-11e0-9b19-734f216543fd"
      />
    </flavor>
    <metadata>
      <meta key="Server Label">Web Head 1</meta>
      <meta key="Image Version">2.1</meta>
    </metadata>
    <addresses>
      <network id="public">
        <ip version="4" addr="67.23.10.132"/>
        <ip version="6" addr="::babe:67.23.10.132"/>
        <ip version="4" addr="67.23.10.131"/>
        <ip version="6" addr="::babe:4317:0A83"/>
      </network>
      <network id="private">
        <ip version="4" addr="10.176.42.16"/>
        <ip version="6" addr="::babe:10.176.42.16"/>
      </network>
    </addresses>
    <atom:link rel="self"
```

```
href="http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/  
52415800-8b69-11e0-9b19-734f6af67565"/>  
<atom:link rel="bookmark"  
href="http://dfw.servers.api.rackspacecloud.com/010101/servers/  
52415800-8b69-11e0-9b19-734f6af67565"/>  
<volumes  
xmlns="http://docs.rackspacecloud.com/servers/api/ext/cbs/v1.1">  
<volume name="OS"  
href="https://cbs.api.rackspacecloud.com/12934/volumes/19"/>  
<volume name="Work"  
href="https://cbs.api.rackspacecloud.com/12934/volumes/23"  
/>  
</volumes>  
</server>  
</servers>
```

Example 1.44. Extended Server: JSON Response

```
{  
  "servers": [  
    {  
      "id": "52415800-8b69-11e0-9b19-734f6af67565",  
      "tenant_id": "010101",  
      "user_id": "MyRackspaceAcct",  
      "name": "sample-server",  
      "updated": "2010-10-10T12:00:00Z",  
      "created": "2010-08-10T12:00:00Z",  
      "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",  
      "status": "BUILD",  
      "progress": 60,  
      "accessIPv4": "67.23.10.132",  
      "accessIPv6": "::babe:67.23.10.132",  
      "image": {  
        "id": "52415800-8b69-11e0-9b19-734f6f006e54",  
        "links": [  
          {  
            "rel": "self",  
            "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/  
52415800-8b69-11e0-9b19-734f6f006e54"  
          },  
          {  
            "rel": "bookmark",  
            "href": "http://dfw.servers.api.rackspacecloud.com/010101/images/  
52415800-8b69-11e0-9b19-734f6f006e54"  
          }  
        ]  
      },  
      "flavor": {  
        "id": "52415800-8b69-11e0-9b19-734f216543fd",  
        "links": [  
          {  
            "rel": "self",  
            "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/  
52415800-8b69-11e0-9b19-734f216543fd"  
          },  
          {  
            "rel": "bookmark",  
            "href": "http://dfw.servers.api.rackspacecloud.com/010101/flavors/  
52415800-8b69-11e0-9b19-734f216543fd"  
          }  
        ]  
      },  
      "addresses": {  
        "public": [  
          {  
            "version": 4,  
            "addr": "67.23.10.132"  
          },  
          {  
            "version": 6,  
            "addr": "::babe:67.23.10.132"  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
        "version": 4,
        "addr": "67.23.10.131"
      },
      {
        "version": 6,
        "addr": "::babe:4317:0A83"
      }
    ],
    "private" : [
      {
        "version": 4,
        "addr": "10.176.42.16"
      },
      {
        "version": 6,
        "addr": "::babe:10.176.42.16"
      }
    ]
  },
  "metadata": {
    "Server Label": "Web Head 1",
    "Image Version": "2.1"
  },
  "links": [
    {
      "rel": "self",
      "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/52415800-8b69-11e0-9b19-734f6af67565"
    },
    {
      "rel": "bookmark",
      "href": "http://dfw.servers.api.rackspacecloud.com/010101/servers/52415800-8b69-11e0-9b19-734f6af67565"
    }
  ],
  "RS-CBS:volumes": [
    {
      "name": "OS",
      "href": "https://cbs.api.rackspacecloud.com/12934/volumes/19"
    },
    {
      "name": "Work",
      "href": "https://cbs.api.rackspacecloud.com/12934/volumes/23"
    }
  ]
}
```

Example 1.45. Extended Action: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<attach-volume
  xmlns="http://docs.rackspacecloud.com/servers/api/ext/cbs/v1.0"
  href="https://cbs.api.rackspacecloud.com/12934/volumes/19"/>
```

Example 1.46. Extended Action: JSON Request

```
{
  "RS-CBS:attach-volume" {
    "href" : "https://cbs.api.rackspacecloud.com/12934/volumes/19"
  }
}
```

1.11. Faults

The API v2 handles the following types of faults:

- Synchronous faults occur at request time.
- Asynchronous faults occur in the background while a server or image is being built or a server is executing an action. When an asynchronous fault occurs, the system places the server or image in an `ERROR` state and embeds the fault in the offending server or image.

When a synchronous or asynchronous fault occurs, the fault contains an HTTP status code, a human readable message, and optional details about the error. Additionally, an asynchronous fault might also contain a time stamp that indicates when the fault occurred.

1.11.1. Synchronous Faults

Synchronous faults occur at request time. When a synchronous fault occurs, the fault contains an HTTP error response code, a human readable message, and optional details about the error.

Example 1.47. Fault: JSON Response

```
{
  "computeFault" : {
    "code" : 500,
    "message" : "Fault!",
    "details" : "Error Details..."
  }
}
```

Example 1.48. Fault: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<computeFault
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="500">
  <message>Fault!</message>
  <details>Error Details...</details>
</computeFault>
```

The error response code is returned in the body of the response for convenience. The message section returns a human-readable message that is appropriate for display to the end user. The details section is optional and may contain information—for example, a stack trace—to assist in tracking down an error. The detail section may or may not be appropriate for display to an end user.

The root element of the fault, such as `computeFault`, may change depending on the type of error. The following table lists possible elements with the associated error response codes:

Table 1.6. Fault Elements and Error Response Codes

Fault Element	Associated Error Response Code	Expected in All Requests?
computeFault	500, 400, other codes possible	✓
Client errors		
badRequest	400	✓
unauthorized	401	✓
forbidden	403	✓
resizeNotAllowed	403	
itemNotFound	404	
Method Not Allowed	405	
buildInProgress	409	
backupOrResizeInProgress	409	
overLimit	413	✓
badMediaType	415	
Server errors		
notImplemented	501	
serviceUnavailable	503	✓
serverCapacityUnavailable	503	

Example 1.49. Fault, Item Not Found: JSON Response

```
{
  "itemNotFound" : {
    "code" : 404,
    "message" : "Not Found",
    "details" : "Error Details..."
  }
}
```

Example 1.50. Fault, Item Not Found: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<itemNotFound
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="404">
  <message>Not Found</message>
  <details>Error Details...</details>
</itemNotFound>
```

From an XML schema perspective, all API faults are extensions of the base fault type `ComputeAPIFault`. When working with a system that binds XML to actual classes (such as JAXB), one should be capable of using `ComputeAPIFault` as a “catch-all” if there’s no interest in distinguishing between individual fault types.

The `OverLimit` fault is generated when a rate limit threshold is exceeded. For convenience, the fault adds a `retryAt` attribute that contains the content of the `Retry-After` header in XML Schema 1.0 date/time format.

Example 1.51. Fault, Over Limit: JSON Response

```
{
  "overLimit" : {
    "code" : 413,
    "message" : "OverLimit Retry...",
    "details" : "Error Details...",
    "retryAt" : "2010-08-01T00:00:00Z"
  }
}
```

Example 1.52. Fault, Over Limit: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<overLimit xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="413" retryAt="2010-08-01T00:00:00Z">
  <message>OverLimit Retry...</message>
  <details>Error Details...</details>
</overLimit>
```

1.11.2. Asynchronous Faults

Asynchronous faults occur in the background while a server or image is being built or a server is executing an action. When an asynchronous fault occurs, the system places the server or image in an `ERROR` state and embeds the fault in the offending server or image.

When an asynchronous fault occurs, the fault contains an HTTP error response code, a human readable message, and optional details about the error. An asynchronous fault might also contain a time stamp that indicates when the fault occurred.

Example 1.53. Server in Error State: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f0000ffff"
  tenant_id="1234" user_id="5678"
  name="sample-server" status="ERROR"
  created="2010-08-10T12:00:00Z"
  progress="66" hostId="e4d909c290d0fb1ca068ffa00000000000">
  <image id="52415800-8b69-11e0-9b19-734f6f007777" />
  <flavor id="52415800-8b69-11e0-9b19-734f216543fd" />
  <fault code="404" created="2010-08-10T11:59:59Z">
    <message>Could not find image 52415800-8b69-11e0-9b19-734f6f007777</message>
    <details>Fault details</details>
  </fault>
  <atom:link
    rel="self"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
52415800-8b69-11e0-9b19-734f000004d2"/>
  <atom:link
    rel="bookmark"
    href="http://dfw.servers.api.rackspacecloud.com/010101/servers/
52415800-8b69-11e0-9b19-734f000004d2"/>
</server>
```

Example 1.54. Server in Error State: JSON

```
{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f0000ffff",
    "tenant_id": "1234",
    "user_id": "5678",
    "name": "sample-server",
    "created": "2010-08-10T12:00:00Z",
    "hostId": "e4d909c290d0fb1ca068ffaaff22cbd0",
    "status": "ERROR",
    "progress": 66,
    "image": {
      "id": "52415800-8b69-11e0-9b19-734f6f007777"
    },
    "flavor": {
      "id": "52415800-8b69-11e0-9b19-734f216543fd"
    },
    "fault": {
      "code": 404,
      "created": "2010-08-10T11:59:59Z",
      "message": "Could not find image 52415800-8b69-11e0-9b19-734f6f007777",
      "details": "Fault details"
    },
    "links": [
      {
        "rel": "self",
        "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/52415800-8b69-11e0-9b19-734f000004d2"
      },
      {
        "rel": "bookmark",
        "href": "http://dfw.servers.api.rackspacecloud.com/010101/servers/52415800-8b69-11e0-9b19-734f000004d2"
      }
    ]
  }
}
```

Example 1.55. Image in Error State: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<image xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f5736d2a2" name="My Server Backup"
  created="2010-08-10T12:00:00Z" status="ERROR" progress="89">
  <server id="52415800-8b69-11e0-9b19-734f335aa7b3"/>
  <fault code="500">
    <message>An internal error occurred</message>
    <details>Error details</details>
  </fault>
  <atom:link rel="self"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/
52415800-8b69-11e0-9b19-734f5736d2a2"/>
  <atom:link rel="bookmark"
    href="http://dfw.servers.api.rackspacecloud.com/010101/images/
52415800-8b69-11e0-9b19-734f5736d2a2"
  />
</image>
```

Example 1.56. Image in Error State: JSON

```
{
  "image" : {
    "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
    "name" : "My Server Backup",
    "created" : "2010-08-10T12:00:00Z",
    "status" : "SAVING",
    "progress" : 89,
    "server" : {
      "id": "52415800-8b69-11e0-9b19-734f335aa7b3"
    },
    "fault" : {
      "code" : 500,
      "message" : "An internal error occurred",
      "details" : "Error details"
    },
    "links": [
      {
        "rel" : "self",
        "href" : "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "bookmark",
        "href" : "http://dfw.servers.api.rackspacecloud.com/010101/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      }
    ]
  }
}
```

2. API Operations

2.1. Servers

A server is a virtual machine (VM) instance in the Cloud Servers environment. To create a server, you must specify a name, flavor reference, and image reference.

Verb	URI	Description
GET	<code>/servers?image=imageId&flavor=flavorId&name=serverName&status=serverStatus&marker=markerID&limit=int&changes-since=dateTime</code>	Lists IDs, names, and links for all servers.
GET	<code>/servers/detail?image=imageId&flavor=flavorId&name=serverName&status=serverStatus&marker=markerID&limit=int&changes-since=dateTime</code>	Lists all details for all servers.
POST	<code>/servers</code>	Creates a server.
GET	<code>/servers/id</code>	Lists details for the specified server.
PUT	<code>/servers/id</code>	Updates the editable attributes for the specified server.
DELETE	<code>/servers/id</code>	Deletes the specified server.

2.1.1. List Servers

Verb	URI	Description
GET	<code>/servers?image=<i>imageId</i>&flavor=<i>flavorId</i>& name=<i>serverName</i>&status=<i>serverStatus</i>&marker=<i>markerID</i>& limit=<i>int</i>& changes-since=<i>dateTime</i></code>	Lists IDs, names, and links for all servers.
GET	<code>/servers/detail?image=<i>imageId</i>&flavor=<i>flavorId</i>& name=<i>serverName</i>&status=<i>serverStatus</i>&marker=<i>markerID</i>& limit=<i>int</i>& changes-since=<i>dateTime</i></code>	Lists all details for all servers.

Normal Response Codes: 200, 203, and 300

Error Response Codes: `computeFault` (400, 500, ...), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `Method Not Allowed` (405), `overLimit` (413), and `serviceUnavailable` (503)

This operation does not require a request body.

This operation returns a response body that lists the servers associated with your account. This operation does not show servers with a status of `DELETED`. To list deleted servers, use the `changes-since` parameter. See [Section 1.7, "Efficient Polling with the *Changes-Since* Parameter"](#) [22].

When you list servers, the addresses for any attached networks are displayed. The network addresses include any isolated networks that you have created and Rackspace public and private networks. For more information about networks, see [Cloud Networks Developer Guide](#).

To filter the list of servers returned in the response body, specify one or more of the following optional URI parameters:

List Servers URI Parameters

<code>image=<i>imageId</i></code>	The image ID. For a list of images, see Section 2.6.1, "List Images" [107].
<code>flavor=<i>flavorId</i></code>	The flavor ID. For a list of flavors, see Section 2.5.1, "List Flavors" [101].
<code>name=<i>serverName</i></code>	The server name.
<code>status=<i>serverStatus</i></code>	The server status. Servers contain a status attribute that indicates the current server state. You can filter on the server status when you complete a list servers request, and the server status is returned in the response body. See Server Status Values [48].
<code>marker=<i>markerID</i></code>	The ID of the last item in the previous list. See Section 1.6, "Paginated Collections" [19].
<code>limit=<i>int</i></code>	The page size. See Section 1.6, "Paginated Collections" [19].

`changes-since=dateTime`

The changes-since time. The list contains servers that have been deleted since the changes-since time. See [Section 1.7, "Efficient Polling with the *Changes-Since* Parameter"](#) [22].

The server status is one of the following values:

Server Status Values

- **ACTIVE.** The server is active and ready to use.
- **BUILD.** The server is being built.
- **DELETED.** The server was deleted. The list servers API operation does not show servers with a status of **DELETED**. To list deleted servers, use the *changes-since* parameter. See [Section 1.7, "Efficient Polling with the *Changes-Since* Parameter" \[22\]](#).
- **ERROR.** The requested operation failed and the server is in an error state.
- **HARD_REBOOT.** The server is going through a hard reboot. A hard reboot power cycles your server, which performs an immediate shutdown and restart. See [Section 2.3.2, "Reboot Server" \[80\]](#).
- **MIGRATING.** The server is being moved from one physical node to another physical node.

Server migration is a Rackspace extension.

- **PASSWORD.** The password for the server is being changed. See [Section 2.3.1, "Change Administrator Password" \[79\]](#).
- **REBOOT.** The server is going through a soft reboot. During a soft reboot, the operating system is signaled to restart, which allows for a graceful shutdown and restart of all processes. See [Section 2.3.2, "Reboot Server" \[80\]](#).
- **REBUILD.** The server is being rebuilt from an image. See [Section 2.3.3, "Rebuild Server" \[81\]](#).
- **RESCUE.** The server is in rescue mode.

Rescue mode is a Rackspace extension. See [Section 3.4, "Rescue Mode Extension" \[147\]](#).

- **RESIZE.** The server is being resized and is inactive until the resize operation completes. See [Section 2.3.4, "Resize Server" \[86\]](#).
- **REVERT_RESIZE.** A resized or migrated server is being reverted to its previous size. The destination server is being cleaned up and the original source server is restarting. For a server that was resized, see [Section 2.3.4, "Resize Server" \[86\]](#).

Server migration is a Rackspace extension.

- **SUSPENDED.** The server is inactive, either by request or necessity. Review support tickets or contact Rackspace support to determine why the server is in this state.
- **UNKNOWN.** The server is in an unknown state. Contact Rackspace support.
- **VERIFY_RESIZE.** The server is waiting for the resize operation to be confirmed so that the original server can be removed.

This operation does not require a request body.

This operation returns a response body.

The following table describes the fields that are returned in the response body:

Table 2.1. List Server Response Fields

Name	Description
accessIPv4	The public IP version 4 access address.
accessIPv6	The public IP version 6 access address.
addresses	Address for any attached isolated networks and Rackspace public and private networks. The <code>version</code> field indicates whether the IP address is version 4 or 6. For more information about networks, see the Cloud Networks Developer Guide .
created	The time stamp for the creation date.
flavor	The flavor reference, including the flavor ID and flavor links. For a list of flavors, see Section 2.5.1, "List Flavors" [101] .
hostId	The host ID. The compute provisioning algorithm has an anti-affinity property that attempts to spread customer VMs across hosts. Under certain situations, VMs from the same customer might be placed on the same host. <code>hostId</code> represents the host your server runs on and can be used to determine this scenario if it is relevant to your application. HostId is unique <i>per account</i> and is not globally unique.
id	The server ID.
image	The image reference, including the image ID and image links. For a list of images, see Section 2.6.1, "List Images" [107] .
links	The server links.
metadata	Metadata key and value pairs.
name	The server name.
progress	The build completion progress, as a percentage. Value is from 0 to 100.
status	The server status. See Server Status Values [48] .
tenant_id	The tenant ID.
updated	The time stamp for the last update.
user_id	The user ID.
Extended attributes	
OS-DCF:diskConfig	<p>Extended attribute. The disk configuration value. Valid values are:</p> <ul style="list-style-type: none">• AUTO. The server is built with a single partition the size of the target flavor disk. The file system is automatically adjusted to fit the entire partition. This keeps things simple and automated. AUTO is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.• MANUAL. The server is built using whatever partition scheme and file system is in the source image. If the target flavor disk is larger, the remaining disk space is left unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration. <p>The namespace for this extended attribute is:</p> <pre>xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"</pre> <p>See Section 3.2, "Disk Configuration Extension" [140].</p>
rax-bandwidth:bandwidth	Extended attribute. The amount of bandwidth used for the specified period.

Name	Description
	<p>The namespace for this extended attribute is:</p> <pre>xmlns:rax-bandwidth="http://docs.rackspace.com/servers/api/ext/server_bandwidth/"</pre> <p>See Section 3.1, "Bandwidth Stats Extension" [140].</p>
OS-EXT-STS	<p>Extended attributes. Shows the following extended statuses for servers:</p> <ul style="list-style-type: none">• OS-EXT-STS:vm_state. The VM state.• OS-EXT-STS:task_state. The task state.• OS-EXT-STS:power_state. The power state. <p>The namespace for this extended attribute is:</p> <pre>xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"</pre> <p>See Section 3.3, "Extended Status Extension" [144].</p>

The following examples show a JSON and XML response for this operation.

To issue a list servers request that returns a JSON response, you can embed the request in a cURL command, as follows:

Example 2.1. List Servers: JSON Request in a cURL Command

```
curl -s https://dfw.servers.api.rackspacecloud.com/v2/$account/servers/detail \
-H "X-Auth-Token: $token" | python -m json.tool
```

Where *account* is your tenant ID and *token* is your authentication token.

Example 2.2. List Servers: JSON Response (detail)

```
{
  "servers": [
    {
      "OS-DCF:diskConfig": "AUTO",
      "OS-EXT-STS:power_state": 1,
      "OS-EXT-STS:task_state": null,
      "OS-EXT-STS:vm_state": "active",
      "accessIPv4": "50.56.172.247",
      "accessIPv6": "2001:4800:780e:0510:d87b:9cbc:ff03:bbbd",
      "addresses": {
        "private": [
          {
            "addr": "10.180.1.226",
            "version": 4
          }
        ],
        "public": [
          {
            "addr": "50.56.172.247",
            "version": 4
          },
          {
            "addr": "2001:4800:780e:0510:d87b:9cbc:ff03:bbbd",
            "version": 6
          }
        ]
      },
      "created": "2012-07-28T15:32:25Z",
      "flavor": {
        "id": "6",
        "links": [
          {
            "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/6",
            "rel": "bookmark"
          }
        ]
      }
    }
  ]
}
```

```
    }
  ],
  "hostId": "fb19e2fe4405c9f819c6f574e4954ec4f1a7e58e94782d171dddeb9a",
  "id": "ff671041-b677-4da4-b901-535e689a796d",
  "image": {
    "id": "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
        "rel": "bookmark"
      }
    ]
  },
  "links": [
    {
      "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ff671041-b677-4da4-b901-535e689a796d",
      "rel": "self"
    },
    {
      "href": "https://dfw.servers.api.rackspacecloud.com/010101/servers/ff671041-b677-4da4-b901-535e689a796d",
      "rel": "bookmark"
    }
  ],
  "metadata": {},
  "name": "myUbuntuServer",
  "progress": 100,
  "rax-bandwidth:bandwidth": [
    {
      "audit_period_end": "2012-08-16T14:12:00Z",
      "audit_period_start": "2012-08-16T06:00:00Z",
      "bandwidth_inbound": 39147845,
      "bandwidth_outbound": 13390651,
      "interface": "public"
    },
    {
      "audit_period_end": "2012-08-16T14:12:00Z",
      "audit_period_start": "2012-08-16T06:00:00Z",
      "bandwidth_inbound": 24229191,
      "bandwidth_outbound": 84,
      "interface": "private"
    }
  ],
  "status": "ACTIVE",
  "tenant_id": "010101",
  "updated": "2012-07-28T15:37:09Z",
  "user_id": "170454"
},
{
  "OS-DCF:diskConfig": "AUTO",
  "OS-EXT-STS:power_state": 1,
  "OS-EXT-STS:task_state": null,
  "OS-EXT-STS:vm_state": "active",
  "accessIPv4": "",
  "accessIPv6": "",
  "addresses": {
    "private": [
      {
        "addr": "10.180.13.75",
        "version": 4
      }
    ],
    "public": [
      {
        "addr": "2001:4800:780e:0510:d87b:9cbc:ff04:3e81",
        "version": 6
      },
      {
        "addr": "50.56.186.185",
        "version": 4
      }
    ]
  },
  "created": "2012-05-15T15:47:37Z",
  "flavor": {
    "id": "6",
```

```
        "links": [
          {
            "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/6",
            "rel": "bookmark"
          }
        ],
        "hostId": "1d65b563fc573c2eb544319e0af598f2b2c5f84f75de252db3757cd3",
        "id": "a09e7493-7429-41e1-8d3f-384d7ece09c0",
        "image": {
          "id": "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
          "links": [
            {
              "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
              "rel": "bookmark"
            }
          ]
        },
        "links": [
          {
            "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/a09e7493-7429-41e1-8d3f-384d7ece09c0",
            "rel": "self"
          },
          {
            "href": "https://dfw.servers.api.rackspacecloud.com/010101/servers/a09e7493-7429-41e1-8d3f-384d7ece09c0",
            "rel": "bookmark"
          }
        ],
        "metadata": {},
        "name": "UbuntuDevStack",
        "progress": 100,
        "rax-bandwidth:bandwidth": [
          {
            "audit_period_end": "2012-08-16T14:15:27Z",
            "audit_period_start": "2012-08-16T06:00:00Z",
            "bandwidth_inbound": 31421008,
            "bandwidth_outbound": 350235,
            "interface": "public"
          },
          {
            "audit_period_end": "2012-08-16T14:15:27Z",
            "audit_period_start": "2012-08-16T06:00:00Z",
            "bandwidth_inbound": 24268488,
            "bandwidth_outbound": 140,
            "interface": "private"
          }
        ],
        "status": "ACTIVE",
        "tenant_id": "010101",
        "updated": "2012-05-15T15:55:00Z",
        "user_id": "170454"
      }
    ]
  }
```

To issue a list servers request that returns an XML response, you can embed the request in a cURL command, as follows:

Example 2.3. List Servers: XML Request in a cURL Command

```
curl -i https://dfw.servers.api.rackspacecloud.com/v2/658405/servers/detail.xml \
-H "Content-Type: application/xml" \
-H "Accept: application/xml" \
-H "X-Auth-Token: $token"
```

Where *account* is your tenant ID and *token* is your authentication token.

Example 2.4. List Servers: XML Response (detail)

```
<?xml version='1.0' encoding='UTF-8'?>
<servers
  xmlns:rax-bandwidth="http://docs.rackspace.com/servers/api/ext/server_bandwidth/"
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2012-07-28T15:37:09Z"
    hostId="fb19e2fe4405c9f819c6f574e4954ec4f1a7e58e94782d171dddeb9a"
    name="myUbuntuServer" created="2012-07-28T15:32:25Z"
    userId="170454" tenantId="010101" accessIPv4="50.56.172.247"
    accessIPv6="2001:4800:780e:0510:d87b:9cbc:ff03:bbbd"
    progress="100" id="ff671041-b677-4da4-b901-535e689a796d"
    OS-EXT-STS:vm_state="active" OS-EXT-STS:task_state="None"
    OS-EXT-STS:power_state="1" OS-DCF:diskConfig="AUTO">
    <image id="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001"
        rel="bookmark"/>
    </image>
    <flavor id="6">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/6"
        rel="bookmark"/>
    </flavor>
    <metadata/>
    <addresses>
      <network id="public">
        <ip version="4" addr="50.56.172.247"/>
        <ip version="6"
          addr="2001:4800:780e:0510:d87b:9cbc:ff03:bbbd"/>
      </network>
      <network id="private">
        <ip version="4" addr="10.180.1.226"/>
      </network>
    </addresses>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ff671041-b677-4da4-b901-535e689a796d"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/servers/ff671041-b677-4da4-b901-535e689a796d"
      rel="bookmark"/>
    <rax-bandwidth:bandwidth>
      <rax-bandwidth:interface interface="public"
        bandwidth_outbound="13390651"
        bandwidth_inbound="39147845"
        audit_period_start="2012-08-16T06:00:00Z"
        audit_period_end="2012-08-16T14:12:00Z"/>
      <rax-bandwidth:interface interface="private"
        bandwidth_outbound="84" bandwidth_inbound="24229191"
        audit_period_start="2012-08-16T06:00:00Z"
        audit_period_end="2012-08-16T14:12:00Z"/>
    </rax-bandwidth:bandwidth>
  </server>
  <server status="ACTIVE" updated="2012-05-15T15:55:00Z">
```

```
hostId="1d65b563fc573c2eb544319e0af598f2b2c5f84f75de252db3757cd3"
name="UbuntuDevStack" created="2012-05-15T15:47:37Z"
userId="170454" tenantId="010101" accessIPv4="" accessIPv6=""
progress="100" id="a09e7493-7429-41e1-8d3f-384d7ece09c0"
OS-EXT-STS:vm_state="active" OS-EXT-STS:task_state="None"
OS-EXT-STS:power_state="1" OS-DCF:diskConfig="AUTO">
<image id="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001">
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-
a2cc-a2fc8d80c001"
    rel="bookmark"/>
</image>
<flavor id="6">
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/6"
    rel="bookmark"/>
</flavor>
<metadata/>
<addresses>
  <network id="public">
    <ip version="6"
      addr="2001:4800:780e:0510:d87b:9cbc:ff04:3e81"/>
    <ip version="4" addr="50.56.186.185"/>
  </network>
  <network id="private">
    <ip version="4" addr="10.180.13.75"/>
  </network>
</addresses>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
a09e7493-7429-41e1-8d3f-384d7ece09c0"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/010101/servers/
a09e7493-7429-41e1-8d3f-384d7ece09c0"
  rel="bookmark"/>
<rax-bandwidth:bandwidth>
  <rax-bandwidth:interface interface="public"
    bandwidth_outbound="350235"
    bandwidth_inbound="31421008"
    audit_period_start="2012-08-16T06:00:00Z"
    audit_period_end="2012-08-16T14:15:27Z"/>
  <rax-bandwidth:interface interface="private"
    bandwidth_outbound="140" bandwidth_inbound="24268488"
    audit_period_start="2012-08-16T06:00:00Z"
    audit_period_end="2012-08-16T14:15:27Z"/>
</rax-bandwidth:bandwidth>
</server>
</servers>
```


2.1.2. Create Server

Verb	URI	Description
POST	/servers	Creates a server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), badMediaType (415), serviceUnavailable (503), serverCapacityUnavailable (503)

Status Transition:	BUILD → ACTIVE
	BUILD → ERROR (on error)

This operation asynchronously provisions a new *server*. The progress of this operation depends on the location of the requested image, network i/o, host load, and the selected flavor. You can check the progress of the request can be checked by issuing a **GET** request on `/servers/id`, which returns a progress value of from 0 to 100, which indicates the percentage completion.

The full URL to the newly created server is returned through the `Location` header and is available as a `self` and `bookmark` link in the server representation. See [Links and References](#).

When you create a server, only the server ID, its links, and the administrator password are guaranteed to be returned in the request.

You can get additional attributes by issuing subsequent **GET** requests on the server. See [Section 2.1.3, "Get Server Details" \[63\]](#).

This operation requires a request body.

The following table describes the attributes that you can specify in the request body:

Table 2.2. Create Server Request Attributes

Name	Description	Required
flavorRef	The flavor ID. For a list of flavors, see Section 2.5.1, "List Flavors" [101] .	Yes
name	The server name. The name that you specify in a create request becomes the initial host name of the server. After the server is built, if you change the server name in the API or change the host name directly, the names are not kept in sync. Also, server names are not guaranteed to be unique.	Yes
imageRef	The image ID. For a list of images, see Section 2.6.1, "List Images" [107] .	Yes
OS-DCF:diskConfig	The disk configuration value. Valid values are: <ul style="list-style-type: none">AUTO. The server is built with a single partition the size of the target flavor disk. The file system is	No

Name	Description	Required
	<p>automatically adjusted to fit the entire partition. This keeps things simple and automated. AUTO is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.</p> <ul style="list-style-type: none">MANUAL. The server is built using whatever partition scheme and file system is in the source image. If the target flavor disk is larger, the remaining disk space is left unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration. <p>See Section 3.2, "Disk Configuration Extension" [140].</p>	
metadata	Metadata key and value pairs. See Section 2.1.2.2, "Server Metadata" [59] .	No
personality	File path and contents. See Section 2.1.2.3, "Server Personality" [60] .	No
uuid attribute on the networks element	<p>The networks to which you want to attach the server.</p> <p>This attribute enables you to attach to an isolated network for your tenant ID, the public Internet network, and the private ServiceNet network.</p> <p>If you do not specify any networks, your server is attached to the public Internet and private ServiceNet networks.</p> <p>If you specify one or more networks, your server is attached to only the networks that you specify.</p> <p>If you want to attach to the private ServiceNet or public Internet networks, you must specify them explicitly. The UUID for the private ServiceNet is 11111111-1111-1111-1111-111111111111. The UUID for the public Internet is 00000000-0000-0000-0000-000000000000.</p>	No

This operation returns a response body.

Example 2.7. Create Server: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001" flavorRef="2"
  name="api-test-server-xml2">
  <metadata>
    <meta key="My Server Name">API Test Server XML</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAGICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHNlY2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHNlY2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
      c2lvbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHROZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRO
      ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYm5kIHlv
      dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmdCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYm5b25kIGhvcml6
      b25zLiINCgOKLVJpY2hhcmQgQmFjaA==</file>
    </personality>
  <networks>
    <uuid>0ef47ac7-6797-4e01-8a47-ed26ec3aaa56</uuid>
    <uuid>00000000-0000-0000-0000-000000000000</uuid>
    <uuid>11111111-1111-1111-1111-111111111111</uuid>
  </networks>
</server>
```

Example 2.8. Create Server: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="ed5c7754-29b6-45fa-96cb-ab64958c8c0a" adminPass="Dd5pNZtpVVQ3"
  OS-DCF:diskConfig="AUTO">
  <metadata/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ed5c7754-29b6-45fa-96cb-ab64958c8c0a"
    rel="self" />
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/servers/ed5c7754-29b6-45fa-96cb-ab64958c8c0a"
    rel="bookmark" />
</server>
```

2.1.2.1. Server Passwords

When you create a server, you can specify a password through the optional `adminPass` attribute.

Though Rackspace does not enforce complexity requirements for the password, the operating system might. If the password is not complex enough, the server might enter an `ERROR` state. In this case, a client can issue a change password action to reset the server password. See [Section 2.3.1, “Change Administrator Password” \[79\]](#).

If you do not specify a password, a randomly generated password is assigned and returned in the response body. For security reasons, the password is not returned in subsequent `GET` calls.

The following example shows how to set an administrator password in a JSON request to create a server:

Example 2.9. Set Administrator Password in Create Server Request: JSON Request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "5f68715f-201f-4600-b5a1-0b97e2b1cb31",
    "flavorRef" : "2",
    "adminPass": "GFf1j9aP",
    "OS-DCF:diskConfig" : "AUTO",
    "metadata" : {
      "My Server Name" : "Ubuntu 10.04 LTS"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHNlY2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHNlY2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYmV5b3Vz
dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
      }
    ]
  }
}
```

To change the administrator password for an existing server, see [Section 2.3.1, “Change Administrator Password” \[79\]](#).

2.1.2.2. Server Metadata

You can supply custom server metadata at launch time. See [Section 2.7, “Metadata” \[131\]](#) for details on working with metadata. The maximum size of the metadata key and value is 255 bytes each. You can query the maximum number of key-

value pairs that can be specified per server through the `maxServerMeta` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

2.1.2.3. Server Personality

You can inject data into the file system of the cloud server instance. For example, you might want to insert ssh keys, set configuration files, or store data that you want to retrieve from inside the instance. This feature provides a minimal amount of launch-time personalization. If you require significant customization, create a custom image.

Follow these guidelines when you inject files:

- The maximum size of the file path data is 255 bytes.
- Encode the file contents as a Base64 string.
- You can query the maximum file size value through the `maxPersonalitySize` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

The absolute limit for the file size refers to the number of bytes in the decoded data and not the number of characters in the encoded data.

- You can inject text files only. You cannot inject binary or zip files into a new build.
- You can specify a maximum number of file path/content pairs. You can query this value through the `maxPersonality` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

The file injection might not occur until after the server is built and booted.

During file injection, any existing files that match specified files are renamed to include the `bak` extension appended with a time stamp. For example, if the file `/etc/passwd` exists, it is backed up as `/etc/passwd.bak.1246036261.5785`.

After file injection, all files have root and the root group as the owner and group owner, respectively, and allow user and group read access only (`-r--r-----`).

2.1.2.4. Server Access Addresses

In certain scenarios, such as a hybrid environment leveraging RackConnect, the IPv4 and/or IPv6 addresses used for primary access to your server might differ from those assigned directly to your Cloud Server. The access IP attributes expose the IPs that can be used to reliably access the instance from an external network.

You can specify such addresses by providing additional attributes on the request to create a server. In addition, you can modify the access IPs over the life of the server. See [Section 2.1.4, "Update Server" \[68\]](#).

If you do not specify IPv4 or IPv6 addresses, these are filled in automatically.

Example 2.10. Create Server with Access IP: JSON Request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "52415800-8b69-11e0-9b19-734f6f006e54",
    "flavorRef" : "52415800-8b69-11e0-9b19-734f1195ff37",
    "accessIPv4" : "67.23.10.132"
  }
}
```

Example 2.11. Create Server with Access IP: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="test"
  accessIPv4="67.23.10.132"
  imageRef="52415800-8b69-11e0-9b19-734f6f006e54"
  flavorRef="52415800-8b69-11e0-9b19-734f1195ff37"/>
```

You can use both IPv4 and IPv6 addresses as access addresses and you can assign both addresses simultaneously as illustrated below. You can update access addresses after you create a server. See [Section 2.1.4, "Update Server" \[68\]](#) for more details.

Example 2.12. Create Server with Multiple Access IPs: JSON Request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "52415800-8b69-11e0-9b19-734f6f006e54",
    "flavorRef" : "52415800-8b69-11e0-9b19-734f1195ff37",
    "accessIPv4" : "67.23.10.132",
    "accessIPv6" : "::babe:67.23.10.132"
  }
}
```

Example 2.13. Create Server with Multiple Access IPs: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="test"
  accessIPv4="67.23.10.132"
  accessIPv6 "::babe:67.23.10.132"
  imageRef="52415800-8b69-11e0-9b19-734f6f006e54"
  flavorRef="52415800-8b69-11e0-9b19-734f1195ff37" />
```


2.1.3. Get Server Details

Verb	URI	Description
GET	/servers/ <i>id</i>	Lists details for the specified server.

Normal Response Codes: 200, 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

This operation returns the details of a specified server.



Specify the server ID as *id* in the URI.

This operation does not require a request body.

This operation returns a response body, which contains details for the specified server.

The following table describes the fields that are returned in the response body:

Table 2.3. Get Server Details Response Fields

Name	Description
accessIPv4	The public IP version 4 access address.  Note The accessIPv4 and accessIPv6 are fields that populate once the server is ready to use. Before that, use the information beneath the addresses list to get the IP address.
accessIPv6	The public IP version 6 access address.  Note The accessIPv4 and accessIPv6 are fields that populate once the server is ready to use. Before that, use the information beneath the addresses list to get the IP address.
addresses	Public and private IP addresses, The <code>version</code> field indicates whether the IP address is version 4 or 6.
created	The time stamp for the creation date.
flavor	The flavor ID. For a list of flavors, see Section 2.5.1, “List Flavors” [101] .
hostId	The host ID. The compute provisioning algorithm has an anti-affinity property that attempts to spread customer VMs across hosts. Under certain situations, VMs from the same customer might be placed on the same host. <code>hostId</code> represents the host your server runs on and can be used to determine this scenario if it is relevant to your application. HostId is unique <i>per account</i> and is not globally unique.
id	The server ID.
image	The image ID. For a list of images, see Section 2.6.1, “List Images” [107] .
links	Server links.
metadata	Metadata key and value pairs.
name	The server name.
progress	The build completion progress, as a percentage. Value is from 0 to 100.
status	The server status. See Server Status Values [48] .
tenant_id	The tenant ID.
updated	The time stamp for the last update.
user_id	The user ID.

Name	Description
Extended attributes	
OS-DCF:diskConfig	<p>Extended attribute. The disk configuration value.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• AUTO. The server is built with a single partition the size of the target flavor disk. The file system is automatically adjusted to fit the entire partition. This keeps things simple and automated. AUTO is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.• MANUAL. The server is built using whatever partition scheme and file system is in the source image. If the target flavor disk is larger, the remaining disk space is left unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration. <p>The namespace for this extended attribute is:</p> <pre>xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"</pre> <p>See Section 3.2, "Disk Configuration Extension" [140].</p>
rax-bandwidth:bandwidth	<p>Extended attribute. The amount of bandwidth used for the specified audit period.</p> <p>The namespace for this extended attribute is:</p> <pre>xmlns:rax-bandwidth="http://docs.rackspace.com/servers/api/ext/server_bandwidth/"</pre> <p>See Section 3.1, "Bandwidth Stats Extension" [140].</p>
OS-EXT-STS	<p>Extended attributes. Shows the following extended statuses for servers:</p> <ul style="list-style-type: none">• OS-EXT-STS:vm_state. The VM state.• OS-EXT-STS:task_state. The task state.• OS-EXT-STS:power_state. The power state. <p>The namespace for this extended attribute is:</p> <pre>xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"</pre> <p>See Section 3.3, "Extended Status Extension" [144].</p>

Example 2.14. Get Server Details: JSON Response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "OS-EXT-STS:power_state": 1,
    "OS-EXT-STS:task_state": null,
    "OS-EXT-STS:vm_state": "active",
    "accessIPv4": "198.101.241.238",
    "accessIPv6": "2001:4800:780e:0510:d87b:9cbc:ff04:513a",
    "addresses": {
      "private": [
        {
          "addr": "10.180.3.171",
          "version": 4
        }
      ],
      "public": [
        {
          "addr": "198.101.241.238",
          "version": 4
        },
        {
          "addr": "2001:4800:780e:0510:d87b:9cbc:ff04:513a",
          "version": 6
        }
      ]
    },
    "created": "2012-08-16T18:41:43Z",
    "flavor": {
      "id": "2",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/2",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "33ccb6c82f3625748b6f2338f54d8e9df07cc583251e001355569056",
    "id": "ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
    "image": {
      "id": "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
          "rel": "bookmark"
        }
      ]
    },
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/servers/ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "My Server Name": "API Test Server 2"
    },
    "name": "api-test-server 2",
    "progress": 100,
    "rax-bandwidth:bandwidth": [],
    "status": "ACTIVE",
    "tenant_id": "010101",
    "updated": "2012-08-16T18:50:38Z",
    "user_id": "170454"
  }
}
```

Example 2.15. Get Server Details: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:rax-bandwidth="http://docs.rackspace.com/servers/api/ext/
server_bandwidth/"
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/
api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1" status="BUILD"
  updated="2012-08-16T18:48:41Z"
  hostId="0b88822e7a5db46eb95b69c0608733904b2cd4f18b0c1e418eb06e1c"
  name="api-test-server-xml" created="2012-08-16T18:48:23Z"
  userId="170454" tenantId="010101" accessIPv4="" accessIPv6=""
  progress="25" id="0f828bfb-609c-4042-be41-6dcc1b76228f"
  OS-EXT-STS:vm_state="building" OS-EXT-STS:task_state="spawning"
  OS-EXT-STS:power_state="0" OS-DCF:diskConfig="AUTO">
  <image id="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001">
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/images/
3afe97b2-26dc-49c5-a2cc-a2fc8d80c001"
      rel="bookmark"/>
  </image>
  <flavor id="2">
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/2"
      rel="bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">API Test Server XML</meta>
  </metadata>
  <addresses>
    <network id="public">
      <ip version="6"
        addr="2001:4800:780e:0510:d87b:9cbc:ff04:5146"/>
      <ip version="4" addr="198.101.242.212"/>
    </network>
    <network id="private">
      <ip version="4" addr="10.180.15.103"/>
    </network>
  </addresses>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
0f828bfb-609c-4042-be41-6dcc1b76228f"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/servers/
0f828bfb-609c-4042-be41-6dcc1b76228f"
    rel="bookmark"/>
  <rax-bandwidth:bandwidth/>
</server>
```

2.1.4. Update Server

Verb	URI	Description
PUT	/servers/ <i>id</i>	Updates the editable attributes for the specified server.

Normal Response Code: 200



Note

If you try to update a server by using the server bookmark link, the response code is 300, unless you use the `Accept: application/json;version=1.1` header with the request.

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serviceUnavailable (503)

Status Transition:	ACTIVE → ACTIVE
--------------------	-----------------

This operation updates the editable attributes of a specified server.

Specify the server ID as *id* in the URI.

This operation requires a request body.

The following table describes the attributes that you can set in the request body:

Table 2.4. Update Server Request Attributes

Attribute	Description
name	The name of the server. If you edit the server name, the server host name does not change. Also, server names are not guaranteed to be unique.
accessIPv4	The IP version 4 address.
accessIPv6	The IP version 6 address.

This operation returns a response body.

2.1.4.1. Update Server Name

Example 2.16. Update Server Name: JSON Request

```
{
  "server" :
  {
    "name" : "new-server-test"
  }
}
```

Example 2.17. Update Server Name: JSON Response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "198.101.241.238",
    "accessIPv6": "2001:4800:780e:0510:d87b:9cbc:ff04:513a",
    "addresses": {
```

```
    "private": [
      {
        "addr": "10.180.3.171",
        "version": 4
      }
    ],
    "public": [
      {
        "addr": "198.101.241.238",
        "version": 4
      },
      {
        "addr": "2001:4800:780e:0510:d87b:9cbc:ff04:513a",
        "version": 6
      }
    ]
  },
  "created": "2012-08-16T18:41:43Z",
  "flavor": {
    "id": "2",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/2",
        "rel": "bookmark"
      }
    ]
  },
  "hostId": "33ccb6c82f3625748b6f2338f54d8e9df07cc583251e001355569056",
  "id": "ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
  "image": {
    "id": "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
        "rel": "bookmark"
      }
    ]
  },
  "links": [
    {
      "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
      "rel": "self"
    },
    {
      "href": "https://dfw.servers.api.rackspacecloud.com/010101/servers/ef08aa7a-b5e4-4bb8-86df-5ac56230f841",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "API Test Server 2"
  },
  "name": "new-server-test",
  "progress": 100,
  "status": "ACTIVE",
  "tenant_id": "010101",
  "updated": "2012-08-16T18:50:38Z",
  "user_id": "170454"
}
```

Example 2.18. Update Server Name: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="new-server-test"/>
```

Example 2.19. Update Server Name: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE"
  updated="2012-07-30T14:39:41Z"
  hostId="c0bf446f5e07234f8d3c76af8d25a65f36f77dc3d22360d804cd5293"
  name="new-server-test" created="2012-07-30T14:38:13Z"
  userId="170454" tenantId="010101" accessIPv4="50.56.175.20"
  accessIPv6="2001:4800:780e:0510:d87b:9cbc:ff04:4b96"
  progress="100" id="f56affe0-aecd-4030-81b2-8e8385c897ca"
  OS-DCF:diskConfig="AUTO">
  <image id="5f68715f-201f-4600-b5a1-0b97e2b1cb31">
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/images/5f68715f-201f-4600-
b5a1-0b97e2b1cb31"
      rel="bookmark"/>
    </image>
    <flavor id="2">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/2"
        rel="bookmark"/>
      </flavor>
    <metadata>
      <meta key="My Server Name">API Test Server XML</meta>
    </metadata>
    <addresses>
      <network id="public">
        <ip version="6"
          addr="2001:4800:780e:0510:d87b:9cbc:ff04:4b96"/>
        <ip version="4" addr="50.56.175.20"/>
      </network>
      <network id="private">
        <ip version="4" addr="10.180.5.181"/>
      </network>
    </addresses>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/010101/servers/f56affe0-
aecd-4030-81b2-8e8385c897ca"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/servers/f56affe0-
aecd-4030-81b2-8e8385c897ca"
      rel="bookmark"/>
  </server>
```

2.1.4.2. Update Addresses

You can update multiple access addresses simultaneously, as shown in the following examples:

Example 2.20. Update Server Access Address: JSON Request

```
{
  "server" :
  {
    "accessIPv4" : "67.23.10.132",
    "accessIPv6" : "::babe:67.23.10.132"
  }
}
```

Example 2.21. Update Server Access Address: JSON Response

```
{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f565bc83b",

```



```
"tenant_id": "1234",
"user_id": "5678",
"name": "new-server-test",
"created": "2010-11-11T12:00:00Z",
"updated": "2010-11-12T12:55:55Z",
"hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
"accessIPv4": "67.23.10.132",
"accessIPv6": "::babe:67.23.10.132",
"progress": 0,
"status": "ACTIVE",
"image": {
  "id": "52415800-8b69-11e0-9b19-734f6f006e54",
  "name": "CentOS 5.2",
  "links": [
    {
      "rel": "self",
      "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/images/52415800-8b69-11e0-9b19-734f6f006e54"
    },
    {
      "rel": "bookmark",
      "href": "http://dfw.servers.api.rackspacecloud.com/010101/images/52415800-8b69-11e0-9b19-734f6f006e54"
    }
  ]
},
"flavor": {
  "id": "52415800-8b69-11e0-9b19-734f1195ff37",
  "name": "256 MB Server",
  "links": [
    {
      "rel": "self",
      "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
    },
    {
      "rel": "bookmark",
      "href": "http://dfw.servers.api.rackspacecloud.com/010101/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
    }
  ]
},
"metadata": {
  "My Server Name": "Apache1"
},
"addresses": {
  "public": [
    {
      "version": 4,
      "addr": "67.23.10.138"
    },
    {
      "version": 6,
      "addr": "::babe:67.23.10.138"
    }
  ],
  "private": [
    {
      "version": 4,
      "addr": "10.176.42.19"
    },
    {
      "version": 6,
      "addr": "::babe:10.176.42.19"
    }
  ]
},
"links": [
  {
    "rel": "self",
    "href": "http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/52415800-8b69-11e0-9b19-734f6f006e54"
  },
  {
    "rel": "bookmark",
    "href": "http://dfw.servers.api.rackspacecloud.com/010101/servers/52415800-8b69-11e0-9b19-734f6f006e54"
  }
]
```

```
}  
}
```

Example 2.22. Update Server Access Address: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  accessIPv4="67.23.10.132"
  accessIPv6="::babe:67.23.10.132"
/>
```

Example 2.23. Update Server Access Address: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f565bc83b" tenant_id="010101"
  user_id="5678" name="new-server-test"
  hostId="e4d909c290d0fblca068ffaddf22cbd0" progress="0"
  status="ACTIVE" created="2010-11-11T12:00:00Z"
  updated="2010-11-12T12:55:55Z" accessIPv4="67.23.10.132"
  accessIPv6="::babe:67.23.10.132">
  <image id="52415800-8b69-11e0-9b19-734f6f006e54" name="CentOS 5.2">
    <atom:link rel="self"
      href="http://dfw.servers.api.rackspacecloud.com/v2/010101/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
    <atom:link rel="bookmark"
      href="http://dfw.servers.api.rackspacecloud.com/010101/images/
52415800-8b69-11e0-9b19-734f6f006e54"
    />
  </image>
  <flavor id="52415800-8b69-11e0-9b19-734f1195ff37"
    name="256 MB Server">
    <atom:link rel="self"
      href="http://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
    <atom:link rel="bookmark"
      href="http://dfw.servers.api.rackspacecloud.com/010101/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"
    />
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="public">
      <ip version="4" addr="67.23.10.138"/>
      <ip version="6" addr="::babe:67.23.10.138"/>
    </network>
    <network id="private">
      <ip version="4" addr="10.176.42.19"/>
      <ip version="6" addr="::babe:10.176.42.19"/>
    </network>
  </addresses>
  <atom:link rel="self"
    href="http://dfw.servers.api.rackspacecloud.com/v2/010101/servers/
52415800-8b69-11e0-9b19-734f6f006e54"/>
  <atom:link rel="bookmark"
    href="http://dfw.servers.api.rackspacecloud.com/010101/servers/
52415800-8b69-11e0-9b19-734f6f006e54"
  />
</server>
```

2.1.5. Delete Server

Verb	URI	Description
DELETE	/servers/ <i>id</i>	Deletes the specified server.

Normal Response Code: 204

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), serviceUnavailable (503)

Status Transition:	ACTIVE → DELETED
	ERROR → DELETED

This operation deletes a specified server instance from the system.

Specify the ID for the server as *id* in the URI.

This operation does not require a request body.

This operation does not return a response body.



Note

You can delete a server in any state.

Successful deletion returns a 204 response code.

2.2. Server Addresses

Verb	URI	Description
GET	/servers/ <i>id</i> /ips	Lists all networks and server addresses associated with a specified server.
GET	/servers/ <i>id</i> /ips/ <i>networkID</i>	Lists addresses associated with a specified server and network.

2.2.1. List Addresses

Verb	URI	Description
GET	/servers/ <i>id</i> /ips	Lists all networks and server addresses associated with a specified server.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), serviceUnavailable (503)

This operation lists all networks and addresses associated with a specified server.

Specify the server ID as *id* in the URI.

This operation does not require a request body.

This operation returns a response body.

Example 2.24. List Addresses: JSON Response

```
{
  "addresses": {
    "private": [
      {
        "addr": "10.180.13.75",
        "version": 4
      }
    ],
    "public": [
      {
        "addr": "2001:4800:780e:0510:d87b:9cbc:ff04:3e81",
        "version": 6
      },
      {
        "addr": "50.56.186.185",
        "version": 4
      }
    ]
  }
}
```

Example 2.25. List Addresses: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<addresses xmlns="http://docs.openstack.org/compute/api/v1.1">
  <network id="public">
    <ip version="6" addr="2001:4800:780e:0510:d87b:9cbc:ff04:4b96"/>
    <ip version="4" addr="50.56.175.20"/>
  </network>
  <network id="private">
    <ip version="4" addr="10.180.5.181"/>
  </network>
</addresses>
```

2.2.2. List Addresses by Network

Verb	URI	Description
GET	<code>/servers/<i>id</i>/ips/<i>networkID</i></code>	Lists addresses associated with a specified server and network.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), serviceUnavailable (503)

This operation lists all addresses associated with a specified server and network.

Specify the server ID as *id* and the network ID as *networkID* in the URI.

This operation does not require a request body.

This operation returns a response body.

Example 2.26. List Addresses by Network: JSON Response

```
{
  "network" : {
    "id" : "public",
    "ip" : [
      { "version" : 4, "addr" : "67.23.10.132" },
      { "version" : 6, "addr" : "::babe:67.23.10.132" },
      { "version" : 4, "addr" : "67.23.10.131" },
      { "version" : 6, "addr" : "::babe:4317:0A83" }
    ]
  }
}
```

Example 2.27. List Addresses by Network: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<network xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="public">
  <ip version="4" addr="67.23.10.132"/>
  <ip version="6" addr "::babe:67.23.10.132"/>
  <ip version="4" addr="67.23.10.131"/>
  <ip version="6" addr "::babe:4317:0A83"/>
</network>
```

2.3. Server Actions

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	changePassword	Changes the administrator password for a specified server.
POST	/servers/ <i>id</i> /action	reboot	Performs a soft or hard reboot of the specified server.
POST	/servers/ <i>id</i> /action	rebuild	Rebuilds the specified server.
POST	/servers/ <i>id</i> /action	resize	Resizes the specified server.
POST	/servers/ <i>id</i> /action	confirmResize	Confirms a pending resize action.
POST	/servers/ <i>id</i> /action	revertResize	Cancels and reverts a pending resize action.
POST	/servers/ <i>id</i> /action	rescue	Places a server in rescue mode.
POST	/servers/ <i>id</i> /action	unrescue	Takes a server out of rescue mode.
POST	/servers/ <i>id</i> /action	createImage	Creates a new image for a specified server.



Note

You can shut down a server instance from inside the instance.

On Linux, issue the following command:

```
$ shutdown -h now
```

On Windows, click **Start** → **Shut down**.

When you shut down an instance, you are still charged for it.

To power the instance back on, issue a reboot through the Control Panel or the API.

2.3.1. Change Administrator Password

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	changePassword	Changes the administrator password for a specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serviceUnavailable (503)

Status Transition:	ACTIVE → PASSWORD → ACTIVE
	ACTIVE → PASSWORD → ERROR (on error)

This operation changes the administrator password for a specified server. The administrator password is the root password for the server.

Specify the server ID as *id* in the URI.

In the request body, specify the changePassword action followed by the `adminPass` attribute, as shown in the following table:

Table 2.5. Change Administrator Password Request Attributes

Attribute	Description	Required
adminPass	The administrator password. Though Rackspace does not enforce complexity requirements for the password, the operating system might. If the password is not complex enough, the server might enter an ERROR state.	Yes

This operation does not return a response body.

Example 2.28. Change Administrator Password: JSON Request

```
{
  "changePassword":
  {
    "adminPass": "Test1234"
  }
}
```

Example 2.29. Change Administrator Password: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<changePassword
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  adminPass="ss1293837$%^"/>
```

2.3.2. Reboot Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	reboot	Performs a soft or hard reboot of the specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serviceUnavailable (503)

Status Transition:	ACTIVE → REBOOT → ACTIVE (soft reboot)
	ACTIVE → HARD_REBOOT → ACTIVE (hard reboot)

This operation performs a soft or hard reboot of a specified server. A soft reboot is a graceful shutdown and restart of your server's operating system. A hard reboot power cycles your server, which performs an immediate shutdown and restart.

Specify the server ID as *id* in the URI.

In the request body, specify the `reboot` action followed by the `type` attribute, as shown in the following table:

Table 2.6. Reboot Server Request Attributes

Attribute	Description	Required
type	The type of reboot: <ul style="list-style-type: none">• SOFT. The operating system is signaled to restart, which allows for a graceful shutdown and restart of all processes.• HARD. Power cycles your server, which performs an immediate shutdown and restart.	No. Default is SOFT.

This operation does not return a response body.

Example 2.30. Reboot Server: JSON Request

```
{
  "reboot" : {
    "type" : "HARD"
  }
}
```

Example 2.31. Reboot Server: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reboot
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  type="HARD"/>
```

2.3.3. Rebuild Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	rebuild	Rebuilds the specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Status Transition:	ACTIVE → REBUILD → ACTIVE
	ACTIVE → REBUILD → ERROR (on error)

The rebuild operation removes all data on the server and replaces it with the specified image. The serverRef and all IP addresses remain the same. If you specify name, metadata, accessIPv4, or accessIPv6 in the rebuild request, new values replace existing values. Otherwise, these values do not change.

You can inject data into the file system during the rebuild.

Specify the server ID as *id* in the URI.

In the request body, specify the `rebuild` action followed by attributes. The following table describes the attributes that you can specify in the request body:

Table 2.7. Rebuild Server Request Attributes

Attribute	Description	Required
name	The new name for the server.	Yes
imageRef	The image ID. For a list of images, see Section 2.6.1, "List Images" [107]	Yes
flavorRef	The flavor ID. For a list of flavors, see Section 2.5.1, "List Flavors" [101] .	Yes
accessIPv4	The IP version 4 address.	No
accessIPv6	The IP version 6 address.	No
adminPass	The administrator password.	Yes
metadata	A metadata key and value pair.	No
personality	The file path and file contents.	No
OS-DCF:diskConfig	The disk configuration value. Valid values are: <ul style="list-style-type: none">AUTO. The server is built with a single partition the size of the target flavor disk. The file system is automatically adjusted to fit the entire partition. This keeps things simple and automated. AUTO is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.MANUAL. The server is built using whatever partition scheme and file system is in the source	No

Attribute	Description	Required
	<p>image. If the target flavor disk is larger, the remaining disk space is left unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration.</p> <p>See Section 3.2, "Disk Configuration Extension" [140].</p>	

This operation returns a response body. The full URL to the rebuilt server is returned in the `Location` header.

Example 2.32. Rebuild Server: JSON Request

```
{
  "rebuild" : {
    "name" : "new-server-test",
    "imageRef" : "d42f821e-c2d1-4796-9f07-af5ed7912d0e",
    "flavorRef" : "2",
    "OS-DCF:diskConfig" : "AUTO",
    "adminPass" : "dianel23",
    "metadata" : {
      "My Server Name" : "Apache1"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbiBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5kIGF0IHN1Y2ggYSBzcGVlZC4uLkl0IGZlZWxzIGFuIGltcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHhvIGdvIG5vdy4gQnV0IHRobSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRobSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYmV5b25kIGhvcm16b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
      }
    ]
  }
}
```

Example 2.33. Rebuild Server: JSON Response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "50.56.175.199",
    "accessIPv6": "2001:4800:780e:0510:d87b:9cbc:ff04:35f7",
    "addresses": {
      "private": [
        {
          "addr": "10.180.12.68",
          "version": 4
        }
      ],
      "public": [
        {
          "addr": "2001:4800:780e:0510:d87b:9cbc:ff04:35f7",
          "version": 6
        },
        {
          "addr": "50.56.175.199",
          "version": 4
        }
      ]
    },
    "adminPass": "dianel23",
    "config_drive": "",
    "created": "2012-07-23T20:20:04Z",
    "flavor": {
      "id": "6",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/123456/flavors/6",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "791b847459d001f02f65f23ea82ae32c4b320ad34a3f892b7593c01f",
    "id": "32406068-8539-40ab-bdd3-8140d30823ad",
    "image": {
      "id": "d42f821e-c2d1-4796-9f07-af5ed7912d0e",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/123456/images/d42f821e-c2d1-4796-9f07-af5ed7912d0e",
          "rel": "bookmark"
        }
      ]
    },
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/123456/servers/32406068-8539-40ab-bdd3-8140d30823ad",
        "rel": "self"
      }
    ]
  }
}
```

```
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/123456/servers/32406068-8539-40ab-  
bdd3-8140d30823ad",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "REBUILD",
    "tenant_id": "123456",
    "updated": "2012-07-26T16:09:16Z",
    "user_id": "170454"
  }
}
```

Example 2.34. Rebuild Server: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<rebuild xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="rebuild-server"
  imageRef="d6dd6c70-a122-4391-91a8-decbla356549"
  OS-DCF:diskConfig="AUTO"
  adminPass="diane123">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAGICAgDQoiQSBjbG9lZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBgdXN0IHNLy2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHNLy2ggYSBzcGVlZC4uLkl0IGZlZWxzIGFuIGltcHVz
      c2lubi4uLnRoXMGaXMGdGh1IHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGh1IHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWphbmQgYXxsIGNsbnB3VkcycWgYW5kIHlv
      dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmZCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCgOKLVJpY2hhcmQgQmFjaA== </file>
    </personality>
  </rebuild>
```

Example 2.35. Rebuild Server: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  status="REBUILD" updated="2012-07-26T16:15:58Z"
  hostId="791b847459d001f02f65f23ea82ae32c4b320ad34a3f892b7593c01f"
  name="rebuild-server" created="2012-07-23T20:20:04Z"
  userId="170454" tenantId="123456" accessIPv4="50.56.175.199"
  accessIPv6="2001:4800:780e:0510:d87b:9cbc:ff04:35f7" progress="0"
  id="32406068-8539-40ab-bdd3-8140d30823ad" adminPass="vSunWX8WkPB5"
  OS-DCF:diskConfig="AUTO">
  <image id="d6dd6c70-a122-4391-91a8-decbla356549">
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/123456/images/d6dd6c70-a122-4391-91a8-
      decbla356549"
      rel="bookmark"/>
    </image>
    <flavor id="6">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/123456/flavors/6"
        rel="bookmark"/>
      </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="public">
        <ip version="6"
          addr="2001:4800:780e:0510:d87b:9cbc:ff04:35f7"/>
        <ip version="4" addr="50.56.175.199"/>
      </network>
      <network id="private">
        <ip version="4" addr="10.180.12.68"/>
      </network>
    </addresses>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/123456/servers/32406068-8539-40ab-
      bdd3-8140d30823ad"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/123456/servers/32406068-8539-40ab-
      bdd3-8140d30823ad"
      rel="bookmark"/>
  </server>
```

2.3.4. Resize Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	<code>resize</code>	Resizes the specified server.

Normal Response Code: 202

Error Response Codes: `computeFault` (400, 500, ...), `badRequest` (400), `unauthorized` (401), `resizeNotAllowed` (403), `forbidden` (403), `itemNotFound` (404), `Method Not Allowed` (405), `buildInProgress` (409), `overLimit` (413), `badMediaType` (415), `serverCapacityUnavailable` (503), `serviceUnavailable` (503)

Status Transition:	ACTIVE → RESIZE → VERIFY_RESIZE
	ACTIVE → RESIZE → ACTIVE (on error)

This operation converts an existing server to a different flavor, which scales the server up or down. The original server is saved for a period of time to allow roll back if a problem occurs. You should text and explicitly confirm all resizes. When you do so, the original server is removed. All resizes are automatically confirmed after 24 hours if you do not explicitly confirm or revert the resize.

Specify the server ID as *id* in the URI.

In the request body, specify the `resize` action followed by attributes. The following table describes the attributes that you can specify in the request body:

Table 2.8. Resize Server Request Attributes

Attribute	Description	Required
<code>name</code>	The name for the resized server.	Yes
<code>flavorRef</code>	The flavor ID. For a list of flavors, see Section 2.5.1, "List Flavors" [101] .	Yes
<code>OS-DCF:diskConfig</code>	<p>The disk configuration value.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• AUTO. The server is built with a single partition the size of the target flavor disk. The file system is automatically adjusted to fit the entire partition. This keeps things simple and automated. AUTO is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.• MANUAL. The server is built using whatever partition scheme and file system is in the source image. If the target flavor disk is larger, the remaining disk space is left unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration. <p>See Section 3.2, "Disk Configuration Extension" [140].</p>	No

This operation does not return a response body.

Example 2.36. Resize Server: JSON Request

```
{  
  "resize" : {  
    "flavorRef" : "3"  
  }  
}
```

Example 2.37. Resize Server: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>  
<resize xmlns="http://docs.openstack.org/compute/api/v1.1"  
  flavorRef="3"/>
```

2.3.5. Confirm Resized Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	confirmResize	Confirms a pending resize action.

Normal Response Code: 204

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), resizeNotAllowed (403), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Status Transition:	VERIFY_RESIZE → ACTIVE
	VERIFY_RESIZE → ERROR (on error)

During a resize operation, the original server is saved for a period of time to allow roll back if a problem occurs. After you verify that the newly resized server works properly, use this operation to confirm the resize. After you confirm the resize, the original server is removed and you cannot roll back to that server. All resizes are automatically confirmed after 24 hours if you do not explicitly confirm or revert the resize.

Specify the server ID as *id* in the URI.

In the request body, specify the `confirmResize` action.

This operation does not return a response body.

Example 2.38. Confirm Resize: JSON Request

```
{
  "confirmResize" : null
}
```

Example 2.39. Confirm Resize: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<confirmResize
  xmlns="http://docs.openstack.org/compute/api/v1.1"/>
```

2.3.6. Revert Resized Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	revertResize	Cancels and reverts a pending resize action.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), resizeNotAllowed (403), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Status Transition:	VERIFY_RESIZE → ACTIVE
	VERIFY_RESIZE → ERROR (on error)

During a resize operation, the original server is saved for a period of time to allow for roll back if a problem occurs. If you determine that a problem exists with a newly resized server, use this operation to revert the resize and roll back to the original server. All resizes are automatically confirmed after 24 hours if you do not explicitly confirm or revert the resize.

Specify the server ID as *id* in the URI.

In the request body, specify the `revertResize` action.

This operation does not return a response body.

Example 2.40. Revert Resize: JSON Request

```
{
  "revertResize" : null
}
```

Example 2.41. Revert Resize: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<revertResize
  xmlns="http://docs.openstack.org/compute/api/v1.1"/>
```

2.3.7. Rescue Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	rescue	Places a server in rescue mode.

Normal Response Code: 200

Error Response Codes: cloudServersFault (400, 500), badRequest (400), unauthorized (401), itemNotFound (404), rescueOrUnrescueProcessingInProgress (409), operationNotAllowed (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Status Transition:	ACTIVE → PREP_RESCUE → RESCUE
--------------------	-------------------------------

Enter rescue mode to reboot a virtual machine (VM) in rescue mode so that you can access the VM with a new root password and fix any file system and configuration errors.

Enter rescue mode to debug system issues that prevent you from booting a server to a usable state.

When you place a server in rescue mode, the following events occur:

1. The VM is shut down.
2. A new VM is created with the following images attached:
 - Primary image. Cleanly running VM based on the image from which the original server was created, with a random password. This password is returned to you in a response to issuing the rescue mode API call. Use this clean image to boot the server and fix any problems.
 - Secondary disk. Image of the VM that needs to be rescued.

When you put a server into rescue mode, you cannot use it until its status goes from `ACTIVE` to `RESCUE`. This does not happen immediately.

For a list of server status codes, see [Server Status Values \[48\]](#).



Note

The SSH server key will be different on the rescue image than your server.

A temporary root password is assigned for use during rescue mode. This password is returned in the response body for this call.

Rescue mode is limited to 90 minutes, after which the rescue image is destroyed and the server attempts to reboot. You can exit rescue mode at any time.

This operation requires a request body.

This operation returns a response body.

Example 2.42. Rescue Server Request: JSON

```
{  
  "rescue" : "none"  
}
```

Example 2.43. Rescue Server Response: JSON

```
{ "adminPass": "m7UKdGiKFpqM" }
```

Example 2.44. Rescue Server Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<rescue  
  xmlns="http://docs.openstack.org/compute/ext/rescue/api/v1.1"/>
```

Example 2.45. Rescue Server Response: XML

```
<adminPass>eBHcCgGBVj6Z</adminPass>
```

2.3.8. Unrescue Server

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	unrescue	Takes a server out of rescue mode.

Normal Response Code: 202

Error Response Codes: cloudServersFault (400, 500), badRequest (400), unauthorized (401), itemNotFound (404), rescueOrUnrescueProcessingInProgress (409), operationNotAllowed (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Status Transition:	RESCUE → PREP_UNRESCUE → ACTIVE
--------------------	---------------------------------

After you resolve any problems and reboot a rescued server, you can unrescue the server. Specify the `unrescue` action in the request body. When you unrescue the server, the repaired image is restored to its running state with your original password.

You can exit rescue mode at any time.

This operation requires a request body.

This operation returns a response body.

Example 2.46. Unrescue Server Request: JSON

```
{
  "unrescue" : null
}
```

Example 2.47. Unrescue Server Request Response: JSON

```
{
  "server": {
    "addresses": {
      "private": [
        "10.183.226.17"
      ],
      "public": [
        "10.13.6.18"
      ]
    },
    "flavorId": 1,
    "hostId": "e7cbb4b2eee53894c7b72af2932dd50d",
    "id": 11696,
    "imageId": 103,
    "metadata": {},
    "name": "default-junit-slice-SANITY",
    "progress": 100,
    "region": "ORD",
    "status": "PREP_UNRESCUE"
  }
}
```

Example 2.48. Unrescue Server Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<unrescue
  xmlns="http://docs.rackspacecloud.com/servers/api/v1.1"/>
```

Example 2.49. Unrescue Server Response: XML

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

```
<server xmlns="http://docs.rackspacecloud.com/servers/api/v1.1"
  status="PREP_UNRESCUE" progress="100" region="ORD"
  hostId="e7cbb4b2eee53894c7b72af2932dd50d" flavorId="1"
  imageId="103" id="11696" name="default-junit-slice-SANITY">
  <metadata/>
  <addresses>
    <public>
      <ip addr="10.13.6.18"/>
    </public>
    <private>
      <ip addr="10.183.226.17"/>
    </private>
  </addresses>
</server>
```

2.3.9. Create Image

Verb	URI	Action specified in request body	Description
POST	/servers/ <i>id</i> /action	createImage	Creates a new image for a specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), resizeNotAllowed (403), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), backupOrResizeInProgress (409), buildInProgress (409), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

Image Status Transition:	SAVING → ACTIVE
	SAVING → ERROR (on error)

This operation creates a new image for a specified server. Once complete, a new image is available that you can use to rebuild or create servers. The full URL to the newly created image is returned through the `Location` header. You can retrieve additional attributes for the image including its creation status by issuing a subsequent **GET** on that URL. See [Section 2.6.2, "Get Image Details" \[110\]](#) for details.

When you create an image, you can also specify custom image metadata. For details about working with metadata, see [Section 2.7, "Metadata" \[131\]](#). The maximum size of the metadata key and value is 255 bytes each. You can query the maximum number of key-value pairs that you can specify per image through the `maxImageMeta` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

Specify the server ID as *id* in the URI.

In the request body, specify the `createImage` action followed by attributes. The following table describes the attributes that you can specify in the request body:

Table 2.9. Create Image Request Attributes

Attribute	Description	Required
name	The name for the new image.	Yes
metadata	Key and value pairs for metadata.	No

This operation does not return a response body.



Note

Currently, image creation is an asynchronous operation, so coordinating the creation with data quiescence, and so on, is currently not possible.

Example 2.50. Create Image: JSON Request

```
{
  "createImage" : {
    "name" : "new-image",
    "metadata" : {
      "ImageType": "Gold",
      "ImageVersion": "2.0"
    }
  }
}
```


Example 2.51. Create Image: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<createImage
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="new-image">
  <metadata>
    <meta key="ImageType">Gold</meta>
    <meta key="ImageVersion">2.0</meta>
  </metadata>
</createImage>
```

2.4. Volume Attachment Actions

Use the following API extensions to attach volumes to servers and manage volume attachments:

Verb	URI	Description
POST	/servers/ <i>id</i> /os-volume_attachments	Attaches a volume to the specified server.
GET	/servers/ <i>id</i> /os-volume_attachments	Lists the volume attachments for the specified server.
GET	/servers/ <i>id</i> /os-volume_attachments/ <i>attachment_id</i>	Lists volume details for the specified volume attachment ID.
DELETE	/servers/ <i>id</i> /os-volume_attachments/ <i>attachment_id</i>	Deletes the specified volume attachment from the specified server.

For information about creating and managing volumes, see [Rackspace Cloud Block Storage Developer Guide](#).

2.4.1. Attach Volume to Server

Verb	URI	Description
POST	/servers/ <i>id</i> /os-volume_attachments	Attaches a volume to the specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), resizeNotAllowed (403), unauthorized (401), forbidden (403), itemNotFound (404), badRequest (400), badMethod (405), backupOrResizeInProgress (409), buildInProgress (409), overLimit (413), badMediaType (415), serverCapacityUnavailable (503), serviceUnavailable (503)

This operation attaches one or more volumes to the specified server.

For information about creating volumes, see [Rackspace Cloud Block Storage Developer Guide](#).

Specify the server ID as *id* in the URI.

This operation requires a request body.

The following table describes the attributes that you specify in the request body:

Table 2.10. Attach Volume Request Attributes

Attribute	Description	Required
volumeld	The ID of the volume that you want to attach to the server instance.	Yes
device	The name of the device, such as /dev/xvdb. Specify <code>auto</code> for auto-assignment.	Yes

This operation returns a response body.

Example 2.52. Attach Volume Request: JSON

```
{
  "volumeAttachment": {
    "device": null,
    "volumeId": "4ab50df6-7480-45df-8604-b1ee39fe857c"
  }
}
```

Example 2.53. Attach Volume Response: JSON

```
{
  "volumeAttachment": {
    "device": "/dev/xvdb",
    "serverId": "76ddf257-2771-4097-aab8-b07b52110376",
    "id": "4ab50df6-7480-45df-8604-b1ee39fe857c",
    "volumeId": "4ab50df6-7480-45df-8604-b1ee39fe857c"
  }
}
```

Example 2.54. Attach Volume Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<volumeAttachment
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  volumeId="volume_id"
  device="device"/>
```

Example 2.55. Attach Volume Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/xvdb"
  serverId="76ddf257-2771-4097-aab8-b07b52110376"
  id="4ab50df6-7480-45df-8604-b1ee39fe857c"
  volumeId="4ab50df6-7480-45df-8604-b1ee39fe857c"/>
```

2.4.2. List Volume Attachments

Verb	URI	Description
GET	/servers/ <i>id</i> /os-volume_attachments	Lists the volume attachments for the specified server.

Normal Response Codes: 200, 203, and 300

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), badMethod (405), overLimit (413)

This operation does not require a request body.

This operation returns a response body that lists the volume attachments for the specified server.

For information about creating volumes, see [Rackspace Cloud Block Storage Developer Guide](#).

Specify the server ID as *id* in the URI.

The response body returns the attachment IDs for the attached volumes.

The following examples show JSON and XML response for this operation:

Example 2.56. List Volume Attachments Response: JSON

```
{
  "volumeAttachments": [
    {
      "device": "/dev/xvdb",
      "serverId": "76ddf257-2771-4097-aab8-b07b52110376",
      "id": "4ab50df6-7480-45df-8604-blee39fe857c",
      "volumeId": "4ab50df6-7480-45df-8604-blee39fe857c"
    }
  ]
}
```

Example 2.57. List Volume Attachments Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachments>
  <volumeAttachment device="/dev/xvdb"
    serverId="76ddf257-2771-4097-aab8-b07b52110376"
    id="4ab50df6-7480-45df-8604-blee39fe857c"
    volumeId="4ab50df6-7480-45df-8604-blee39fe857c" />
</volumeAttachments>
```

2.4.3. Get Volume Attachment Details

Verb	URI	Description
GET	/servers/ <i>id</i> /os-volume_attachments/ <i>attachment_id</i>	Lists volume details for the specified volume attachment ID.

Normal Response Codes: 200, 203, and 300

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), badMethod (405), overLimit (413), itemNotFound (404)

This operation returns the volume details of a specified volume attachment ID for a specified server.

For information about creating volumes, see [Rackspace Cloud Block Storage Developer Guide](#).

Specify the server ID as *id* and the volume attachment ID as *attachment_ID* in the URI.

This operation does not require a request body.

This operation returns a response body.

Example 2.58. Get Volume Attachment Details Response: JSON

```
{
  "volumeAttachment": {
    "device": "/dev/xvdb",
    "serverId": "76ddf257-2771-4097-aab8-b07b52110376",
    "id": "4ab50df6-7480-45df-8604-b1ee39fe857c",
    "volumeId": "4ab50df6-7480-45df-8604-b1ee39fe857c"
  }
}
```

Example 2.59. Get Volume Attachment Details Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/xvdb"
  serverId="76ddf257-2771-4097-aab8-b07b52110376"
  id="4ab50df6-7480-45df-8604-b1ee39fe857c"
  volumeId="4ab50df6-7480-45df-8604-b1ee39fe857c" />
```

2.4.4. Delete Volume Attachment

Verb	URI	Description
DELETE	/servers/ <i>id</i> /os-volume_attachments/ <i>attachment_id</i>	Deletes the specified volume attachment from the specified server.

Normal Response Code: 202

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), badMethod (405), overLimit (413), itemNotFound (404), buildInProgress (409)

Status Transition:	ACTIVE → DELETED
	ERROR → DELETED

This operation deletes a specified volume attachment from a specified server instance.

For information about creating and managing volumes, see [Rackspace Cloud Block Storage Developer Guide](#).

Specify the ID for the server as *id* in the URI.

Specify the volume attachment ID as *attachment_id* in the URI.

This operation does not require a request body.

This operation does not return a response body. A normal response code is 202.

2.5. Flavors

A flavor is a resource configuration for a server. Each flavor is a unique combination of disk, memory, vCPUs, and network bandwidth.

Verb	URI	Description
GET	<code>/flavors?minDisk=minDiskInGB&minRam=minRamInMB&marker=markerID&limit=int</code>	Lists IDs, names, and links for all available flavors.
GET	<code>/flavors/detail?minDisk=minDiskInGB&minRam=minRamInMB&marker=markerID&limit=int</code>	Lists all details for all available flavors.
GET	<code>/flavors/id</code>	Lists details of the specified flavor.

2.5.1. List Flavors

Verb	URI	Description
GET	<code>/flavors?minDisk=minDiskInGB&minRam=minRamInMB&marker=markerID&limit=int</code>	Lists IDs, names, and links for all available flavors.
GET	<code>/flavors/detail?minDisk=minDiskInGB&minRam=minRamInMB&marker=markerID&limit=int</code>	Lists all details for all available flavors.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

This operation lists information for all available flavors.

To filter the list of flavors returned in the response body, you can specify the following optional URI parameters:

<code>minDisk=minDiskInGB</code>	Filters the list of flavors to those with the specified minimum number of gigabytes of disk storage.
<code>minRam=minRamInMB</code>	Filters the list of flavors to those with the specified minimum amount of RAM in megabytes.
<code>marker=markerID</code>	The ID of the last item in the previous list. See Section 1.6, "Paginated Collections" [19].
<code>limit=int</code>	Sets the page size. See Section 1.6, "Paginated Collections" [19].

This operation does not require a request body.

This operation returns a response body.

Example 2.60. List Flavors: JSON Response (detail)

```
{
  "flavors": [
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 20,
      "id": "2",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/2",
          "rel": "self"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/2",
          "rel": "bookmark"
        }
      ],
      "name": "512MB Standard Instance",
      "ram": 512,
      "rxtx_factor": 2.0,
      "swap": 512,
      "vcpus": 1
    },
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 40,
      "id": "3",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/3",
          "rel": "self"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/3",
          "rel": "bookmark"
        }
      ],
      "name": "1GB Standard Instance",
      "ram": 1024,
      "rxtx_factor": 3.0,
      "swap": 1024,
      "vcpus": 1
    },
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 80,
      "id": "4",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/4",
          "rel": "self"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/4",
          "rel": "bookmark"
        }
      ],
      "name": "2GB Standard Instance",
      "ram": 2048,
      "rxtx_factor": 6.0,
      "swap": 2048,
      "vcpus": 2
    },
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 160,
      "id": "5",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/5",
          "rel": "self"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/5",
          "rel": "bookmark"
        }
      ]
    }
  ],
}
```



```
    "name": "4GB Standard Instance",
    "ram": 4096,
    "rxtx_factor": 10.0,
    "swap": 2048,
    "vcpus": 2
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 320,
    "id": "6",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/6",
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/6",
        "rel": "bookmark"
      }
    ],
    "name": "8GB Standard Instance",
    "ram": 8192,
    "rxtx_factor": 15.0,
    "swap": 2048,
    "vcpus": 4
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 620,
    "id": "7",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/7",
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/7",
        "rel": "bookmark"
      }
    ],
    "name": "15GB Standard Instance",
    "ram": 15360,
    "rxtx_factor": 20.0,
    "swap": 2048,
    "vcpus": 6
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 1200,
    "id": "8",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/8",
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/8",
        "rel": "bookmark"
      }
    ],
    "name": "30GB Standard Instance",
    "ram": 30720,
    "rxtx_factor": 30.0,
    "swap": 2048,
    "vcpus": 8
  }
]
```

Example 2.61. List Flavors: XML Response (detail)

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1">
  <flavor name="512MB Standard Instance" ram="512" vcpus="1"
    swap="512" rxtx_factor="2.0" disk="20" id="2">
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/2"
```

```
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/2"
        rel="bookmark"/>
    </flavor>
    <flavor name="1GB Standard Instance" ram="1024" vcpus="1"
      swap="1024" rxtx_factor="3.0" disk="40" id="3">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/3"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/3"
        rel="bookmark"/>
    </flavor>
    <flavor name="2GB Standard Instance" ram="2048" vcpus="2"
      swap="2048" rxtx_factor="6.0" disk="80" id="4">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/4"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/4"
        rel="bookmark"/>
    </flavor>
    <flavor name="4GB Standard Instance" ram="4096" vcpus="2"
      swap="2048" rxtx_factor="10.0" disk="160" id="5">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/5"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/5"
        rel="bookmark"/>
    </flavor>
    <flavor name="8GB Standard Instance" ram="8192" vcpus="4"
      swap="2048" rxtx_factor="15.0" disk="320" id="6">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/6"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/6"
        rel="bookmark"/>
    </flavor>
    <flavor name="15GB Standard Instance" ram="15360" vcpus="6"
      swap="2048" rxtx_factor="20.0" disk="620" id="7">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/7"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/7"
        rel="bookmark"/>
    </flavor>
    <flavor name="30GB Standard Instance" ram="30720" vcpus="8"
      swap="2048" rxtx_factor="30.0" disk="1200" id="8">
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/8"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/8"
        rel="bookmark"/>
    </flavor>
  </flavors>
```

2.5.2. Get Flavor Details

Verb	URI	Description
GET	/flavors/ <i>id</i>	Lists details of the specified flavor.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

Specify the flavor ID as *id* in the URI.

This operation does not require a request body.

This operation returns details of the specified flavor in the response body.

Example 2.62. Get Flavor Details: JSON Response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 20,
    "id": "2",
    "links": [
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/2",
        "rel": "self"
      },
      {
        "href": "https://dfw.servers.api.rackspacecloud.com/010101/flavors/2",
        "rel": "bookmark"
      }
    ],
    "name": "512MB Standard Instance",
    "ram": 512,
    "rxtx_factor": 2.0,
    "swap": 512,
    "vcpus": 1
  }
}
```

Example 2.63. Get Flavor Details: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="512MB Standard Instance" ram="512" vcpus="1" swap="512"
  rxtx_factor="2.0" disk="20" id="2">
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/010101/flavors/2"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/010101/flavors/2"
    rel="bookmark"/>
</flavor>
```

2.6. Images

An image is a collection of files for a specific operating system (OS) that you use to create or rebuild a server. Rackspace provides pre-built images. You can also create custom images from servers that you have launched. Custom images can be used for data backups or as "gold" images for additional servers.

Verb	URI	Description
GET	/images?server=serverRef&name=imageName&status=imageStatus&changes-since=dateTime&marker=markerID&limit=int&type=(BASE SNAPSHOT)	Lists IDs, names, and links for all available images.
GET	/images/detail?server=serverRef&name=imageName&status=imageStatus&changes-since=dateTime&marker=markerID&limit=int&type=(BASE SNAPSHOT)	List all details for all available images.
GET	/images/id	Lists details of the specified image.
DELETE	/images/id	Deletes the specified image.

2.6.1. List Images

Verb	URI	Description
GET	/images?server=serverRef&name=imageName&status=imageStatus&changes-since=dateTime&marker=markerID&limit=int&type=(BASE SNAPSHOT)	Lists IDs, names, and links for all available images.
GET	/images/detail?server=serverRef&name=imageName&status=imageStatus&changes-since=dateTime&marker=markerID&limit=int&type=(BASE SNAPSHOT)	List all details for all available images.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

This operation lists all images visible by the account.

To filter the list of images returned in the response body, you can specify the following optional URI parameters:

`server=serverRef&` Filters the list of images by server. Specify the server reference by ID or by full URL.

`name=imageName&` Filters the list of images by image name.

<code>status=imageStatus&</code>	Filters the list of images by status. In-flight images have a status of <code>SAVING</code> and the conditional progress element contains a value from 0 to 100, which indicates the percentage completion. Other possible values for the status attribute include <code>ACTIVE</code> , <code>DELETED</code> , <code>ERROR</code> , <code>SAVING</code> , and <code>UNKNOWN</code> . Images with an <code>ACTIVE</code> status are available for use.
<code>changes-since=dateTime&</code>	Filters the list of images to those that have changed since the changes-since time. See Section 1.7, “Efficient Polling with the <i>Changes-Since</i> Parameter” [22] .
<code>marker=markerID&</code>	The ID of the last item in the previous list. See Section 1.6, “Paginated Collections” [19] .
<code>limit=int</code>	Sets the page size. See Section 1.6, “Paginated Collections” [19] .
<code>type={BASE SNAPSHOT}</code>	Filters base Rackspace images or any custom server images that you have created.

This operation does not require a request body.

The optional `minDisk` and `minRam` attributes set the minimum disk and RAM required to create a server with the image.

This operation returns a response body.

Example 2.64. List Images: JSON Response (detail)

```
{
  "images": [
    {
      "OS-DCF:diskConfig": "AUTO",
      "created": "2012-10-13T16:53:56Z",
      "id": "a3a2c42f-575f-4381-9c6d-fcd3b7d07d17",
      "links": [
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/v2/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17",
          "rel": "self"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17",
          "rel": "bookmark"
        },
        {
          "href": "https://dfw.servers.api.rackspacecloud.com/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ]
    },
    {
      "metadata": {
        "arch": "x86-64",
        "auto_disk_config": "True",
        "com.rackspace__1__build_core": "1",
        "com.rackspace__1__build_managed": "0",
        "com.rackspace__1__build_rackconnect": "1",
        "com.rackspace__1__options": "0",
        "com.rackspace__1__visible_core": "1",
        "com.rackspace__1__visible_managed": "0",
        "com.rackspace__1__visible_rackconnect": "1",
        "image_type": "base",
        "org.openstack__1__architecture": "x64",
        "org.openstack__1__os_distro": "org.centos",
        "org.openstack__1__os_version": "6.0",
        "os_distro": "centos",
        "os_type": "linux",
        "os_version": "6.0",
        "rax_managed": "false",
```

```
        "rax_options": "0"  
      },  
      "minDisk": 10,  
      "minRam": 256,  
      "name": "CentOS 6.0",  
      "progress": 100,  
      "status": "ACTIVE",  
      "updated": "2012-10-13T16:54:55Z"  
    },  
    ...  
  }  
  ]  
}
```

Example 2.65. List Images: XML Response (detail)

```
<?xml version='1.0' encoding='UTF-8'?>
<images
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1">
  <image status="ACTIVE" updated="2012-08-01T00:37:41Z"
    name="Ubuntu 10.04 LTS (Lucid Lynx)"
    created="2012-08-01T00:37:12Z" minDisk="10" progress="100"
    minRam="512" id="d531a2dd-7ae9-4407-bb5a-e5ea03303d98"
    OS-DCF:diskConfig="AUTO">
    <metadata>
      <meta key="os_distro">ubuntu</meta>
      <meta key="com.rackspace__1__visible_core">1</meta>
      <meta key="com.rackspace__1__options">0</meta>
      <meta key="com.rackspace__1__build_rackconnect">1</meta>
      <meta key="image_type">base</meta>
      <meta key="org.openstack__1__os_version">10.04</meta>
      <meta key="auto_disk_config">true</meta>
      <meta key="com.rackspace__1__visible_rackconnect">1</meta>
      <meta key="org.openstack__1__os_distro">org.ubuntu</meta>
      <meta key="com.rackspace__1__visible_managed">1</meta>
      <meta key="com.rackspace__1__build_core">1</meta>
      <meta key="os_type">linux</meta>
      <meta key="org.openstack__1__architecture">x64</meta>
      <meta key="com.rackspace__1__build_managed">1</meta>
    </metadata>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/010101/images/d531a2dd-7ae9-4407-bb5a-e5ea03303d98"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/images/d531a2dd-7ae9-4407-bb5a-e5ea03303d98"
      rel="bookmark"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/010101/images/d531a2dd-7ae9-4407-bb5a-e5ea03303d98"
      type="application/vnd.openstack.image" rel="alternate"/>
  </image>
  ...
</images>
```

2.6.2. Get Image Details

Verb	URI	Description
GET	/images/ <i>id</i>	Lists details of the specified image.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

Specify the image ID as *id* in the URI.

This operation does not require a request body.

This operation returns details of the specified image in the response body.



Note

The response body does not include the `serverId` field. To retrieve the `serverId` field, get details for all images. See [Section 2.6.1, "List Images" \[107\]](#)

Example 2.66. Get Image Details: JSON Response

```
{
  "image": {
    "OS-DCF:diskConfig": "AUTO",
    "created": "2012-02-28T19:38:57Z",
    "id": "3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",
    "links": [
      {

```



```
      "href": "https://dfw.servers.api.rackspacecloud.com/v2/010101/images/  
3afe97b2-26dc-49c5-a2cc-a2fc8d80c001",  
      "rel": "self"  
    },  
    {  
      "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-  
a2cc-a2fc8d80c001",  
      "rel": "bookmark"  
    },  
    {  
      "href": "https://dfw.servers.api.rackspacecloud.com/010101/images/3afe97b2-26dc-49c5-  
a2cc-a2fc8d80c001",  
      "rel": "alternate",  
      "type": "application/vnd.openstack.image"  
    }  
  ],  
  "metadata": {  
    "arch": "x86-64",  
    "auto_disk_config": "True",  
    "com.rackspace__1__build_core": "1",  
    "com.rackspace__1__build_managed": "0",  
    "com.rackspace__1__build_rackconnect": "0",  
    "com.rackspace__1__options": "0",  
    "com.rackspace__1__visible_core": "1",  
    "com.rackspace__1__visible_managed": "0",  
    "com.rackspace__1__visible_rackconnect": "0",  
    "image_type": "base",  
    "org.openstack__1__architecture": "x64",  
    "org.openstack__1__os_distro": "org.ubuntu",  
    "org.openstack__1__os_version": "11.10",  
    "os_distro": "ubuntu",  
    "os_type": "linux",  
    "os_version": "11.10",  
    "rax_managed": "false",  
    "rax_options": "0"  
  },  
  "minDisk": 10,  
  "minRam": 256,  
  "name": "Ubuntu 11.10",  
  "progress": 100,  
  "status": "ACTIVE",  
  "updated": "2012-02-28T19:39:05Z"  
}
```

Example 2.67. Get Image Details: XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<images
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1">
  <image status="ACTIVE" updated="2012-10-13T16:54:55Z"
    name="CentOS 6.0" created="2012-10-13T16:53:56Z" minDisk="10"
    progress="100" minRam="256"
    id="a3a2c42f-575f-4381-9c6d-fcd3b7d07d17"
    OS-DCF:diskConfig="AUTO">
    <metadata>
      <meta key="os_distro">centos</meta>
      <meta key="com.rackspace__1__visible_core">1</meta>
      <meta key="com.rackspace__1__build_rackconnect">1</meta>
      <meta key="auto_disk_config">True</meta>
      <meta key="com.rackspace__1__options">0</meta>
      <meta key="image_type">base</meta>
      <meta key="org.openstack__1__os_version">6.0</meta>
      <meta key="rax_managed">false</meta>
      <meta key="os_version">6.0</meta>
      <meta key="rax_options">0</meta>
      <meta key="com.rackspace__1__visible_rackconnect">1</meta>
      <meta key="org.openstack__1__os_distro">org.centos</meta>
      <meta key="com.rackspace__1__visible_managed">0</meta>
      <meta key="com.rackspace__1__build_core">1</meta>
      <meta key="arch">x86-64</meta>
      <meta key="os_type">linux</meta>
      <meta key="org.openstack__1__architecture">x64</meta>
      <meta key="com.rackspace__1__build_managed">0</meta>
    </metadata>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17"
      rel="bookmark"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/658405/images/a3a2c42f-575f-4381-9c6d-fcd3b7d07d17"
      type="application/vnd.openstack.image" rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2012-10-13T16:45:05Z"
    name="Red Hat Enterprise Linux 6.1"
    created="2012-10-13T16:43:57Z" minDisk="0" progress="100"
    minRam="256" id="d6dd6c70-a122-4391-91a8-decbla356549"
    OS-DCF:diskConfig="AUTO">
    <metadata>
      <meta key="os_distro">rhel</meta>
      <meta key="com.rackspace__1__visible_core">1</meta>
      <meta key="com.rackspace__1__build_rackconnect">1</meta>
      <meta key="auto_disk_config">True</meta>
      <meta key="com.rackspace__1__options">1</meta>
      <meta key="image_type">base</meta>
      <meta key="arch">x86-64</meta>
      <meta key="org.openstack__1__os_version">6.1</meta>
      <meta key="rax_managed">false</meta>
      <meta key="os_version">6.1</meta>
      <meta key="rax_options">1</meta>
      <meta key="com.rackspace__1__visible_rackconnect">1</meta>
      <meta key="org.openstack__1__os_distro">com.redhat</meta>
      <meta key="com.rackspace__1__visible_managed">1</meta>
      <meta key="com.rackspace__1__build_core">1</meta>
      <meta key="rax_activation_profile">redhat6</meta>
      <meta key="os_type">linux</meta>
      <meta key="org.openstack__1__architecture">x64</meta>
      <meta key="com.rackspace__1__build_managed">1</meta>
    </metadata>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/d6dd6c70-a122-4391-91a8-decbla356549"
      rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/658405/images/d6dd6c70-a122-4391-91a8-decbla356549"
```

```
        rel="bookmark"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/d6dd6c70-a122-4391-91a8-
decbla356549"
        type="application/vnd.openstack.image" rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2012-10-12T13:07:15Z"
      name="Ubuntu 12.04 LTS (Precise Pangolin)"
      created="2012-10-12T13:05:57Z" minDisk="10" progress="100"
      minRam="256" id="5cebb13a-f783-4f8c-8058-c4182c724ccd"
      OS-DCF:diskConfig="AUTO">
      <metadata>
        <meta key="os_distro">ubuntu</meta>
        <meta key="com.rackspace__1__visible_core">1</meta>
        <meta key="com.rackspace__1__build_rackconnect">1</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="com.rackspace__1__options">0</meta>
        <meta key="image_type">base</meta>
        <meta key="org.openstack__1__os_version">12.04</meta>
        <meta key="rax_managed">>false</meta>
        <meta key="os_version">12.04</meta>
        <meta key="rax_options">0</meta>
        <meta key="com.rackspace__1__visible_rackconnect">1</meta>
        <meta key="org.openstack__1__os_distro">org.ubuntu</meta>
        <meta key="com.rackspace__1__visible_managed">1</meta>
        <meta key="com.rackspace__1__build_core">1</meta>
        <meta key="arch">x86-64</meta>
        <meta key="os_type">linux</meta>
        <meta key="org.openstack__1__architecture">x64</meta>
        <meta key="com.rackspace__1__build_managed">1</meta>
      </metadata>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/5cebb13a-f783-4f8c-8058-
c4182c724ccd"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/5cebb13a-f783-4f8c-8058-
c4182c724ccd"
        rel="bookmark"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/5cebb13a-f783-4f8c-8058-
c4182c724ccd"
        type="application/vnd.openstack.image" rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2012-10-11T22:16:55Z"
      name="Windows Server 2008 R2 SP1 (with updates)"
      created="2012-10-11T22:03:23Z" minDisk="40" progress="100"
      minRam="1024" id="7957e53d-b3b9-41fe-8e0d-5252bf20a5bf"
      OS-DCF:diskConfig="MANUAL">
      <metadata>
        <meta key="com.rackspace__1__build_rackconnect">1</meta>
        <meta key="com.rackspace__1__visible_core">1</meta>
        <meta key="com.rackspace__1__options">4</meta>
        <meta key="com.rackspace__1__build_core">1</meta>
        <meta key="com.rackspace__1__build_managed">1</meta>
        <meta key="com.rackspace__1__visible_rackconnect">1</meta>
        <meta key="os_distro">windows</meta>
        <meta key="org.openstack__1__os_distro"
          >com.microsoft.server</meta>
        <meta key="source">kickstart</meta>
        <meta key="image_type">base</meta>
        <meta key="org.openstack__1__os_version">2008.2</meta>
        <meta key="rax_managed">>false</meta>
        <meta key="os_version">2008r2</meta>
        <meta key="com.rackspace__1__build_managed">1</meta>
        <meta key="org.openstack__1__architecture">x64</meta>
        <meta key="com.rackspace__1__visible_core">1</meta>
        <meta key="com.rackspace__1__visible_rackconnect">1</meta>
        <meta key="rax_options">4</meta>
        <meta key="com.rackspace__1__visible_managed">1</meta>
        <meta key="arch">x86-64</meta>
        <meta key="com.rackspace__1__build_core">1</meta>
        <meta key="com.rackspace__1__build_rackconnect">1</meta>
        <meta key="com.rackspace__1__visible_managed">1</meta>
        <meta key="auto_disk_config">0</meta>
        <meta key="os_type">windows</meta>
        <meta key="rax_activation_profile">windows</meta>
      </metadata>
      <atom:link
```

```
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/7957e53d-
b3b9-41fe-8e0d-5252bf20a5bf"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/7957e53d-
b3b9-41fe-8e0d-5252bf20a5bf"
rel="bookmark"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/7957e53d-
b3b9-41fe-8e0d-5252bf20a5bf"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-10-09T15:21:12Z"
name="Windows Server 2012 (with updates) + SQL Server 2012 Web"
created="2012-10-09T14:51:22Z" minDisk="80" progress="100"
minRam="2048" id="b762eeld-11b5-4ae7-aa68-dcclb6f6e24a"
OS-DCF:diskConfig="MANUAL">
<metadata>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__options">36</meta>
<meta key="image_type">base</meta>
<meta key="com.rackspace__1__source">kickstart</meta>
<meta key="org.openstack__1__os_distro"
>com.microsoft.server</meta>
<meta key="auto_disk_config">False</meta>
<meta key="os_type">windows</meta>
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="rax_activation_profile">windows</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">1</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/b762eeld-11b5-4ae7-aa68-
dcclb6f6e24a"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/b762eeld-11b5-4ae7-aa68-
dcclb6f6e24a"
rel="bookmark"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/b762eeld-11b5-4ae7-aa68-
dcclb6f6e24a"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-10-09T13:53:33Z"
name="Windows Server 2012 (with updates) + SQL Server 2012 Standard"
created="2012-10-09T13:23:23Z" minDisk="80" progress="100"
minRam="2048" id="f86eae6d-09ea-42e6-a5b2-422649edcfal"
OS-DCF:diskConfig="MANUAL">
<metadata>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__options">12</meta>
<meta key="image_type">base</meta>
<meta key="com.rackspace__1__source">kickstart</meta>
<meta key="org.openstack__1__os_distro"
>com.microsoft.server</meta>
<meta key="auto_disk_config">False</meta>
<meta key="os_type">windows</meta>
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="rax_activation_profile">windows</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">1</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/f86eae6d-09ea-42e6-
a5b2-422649edcfal"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/f86eae6d-09ea-42e6-
a5b2-422649edcfal"
rel="bookmark"/>
```

```
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/f86eae6d-09ea-42e6-
a5b2-422649edcfaf1"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-10-09T12:25:35Z"
  name="Windows Server 2012 + SQL Server 2012 Web"
  created="2012-10-09T11:55:45Z" minDisk="80" progress="100"
  minRam="2048" id="057d2670-68bc-4e28-b7b1-b9bc72245683"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">36</meta>
    <meta key="image_type">base</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="auto_disk_config">False</meta>
    <meta key="os_type">windows</meta>
    <meta key="com.rackspace__1__visible_rackconnect">0</meta>
    <meta key="com.rackspace__1__build_rackconnect">0</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/057d2670-68bc-4e28-b7b1-
b9bc72245683"
    rel="self"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/658405/images/057d2670-68bc-4e28-b7b1-
b9bc72245683"
      rel="bookmark"/>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/658405/images/057d2670-68bc-4e28-b7b1-
b9bc72245683"
      type="application/vnd.openstack.image" rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2012-10-08T23:05:50Z"
    name="Windows Server 2012 + SQL Server 2012 Standard"
    created="2012-10-08T22:37:21Z" minDisk="80" progress="100"
    minRam="2048" id="d226f189-f83f-4569-95b8-622133d71f02"
    OS-DCF:diskConfig="MANUAL">
    <metadata>
      <meta key="com.rackspace__1__visible_core">1</meta>
      <meta key="com.rackspace__1__options">12</meta>
      <meta key="image_type">base</meta>
      <meta key="com.rackspace__1__source">kickstart</meta>
      <meta key="org.openstack__1__os_distro"
        >com.microsoft.server</meta>
      <meta key="auto_disk_config">False</meta>
      <meta key="os_type">windows</meta>
      <meta key="com.rackspace__1__visible_rackconnect">0</meta>
      <meta key="com.rackspace__1__build_rackconnect">0</meta>
      <meta key="com.rackspace__1__visible_managed">0</meta>
      <meta key="com.rackspace__1__build_core">1</meta>
      <meta key="rax_activation_profile">windows</meta>
      <meta key="org.openstack__1__os_version">2008.2</meta>
      <meta key="org.openstack__1__architecture">x64</meta>
      <meta key="com.rackspace__1__build_managed">0</meta>
    </metadata>
    <atom:link
      href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/d226f189-
f83f-4569-95b8-622133d71f02"
      rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/d226f189-
f83f-4569-95b8-622133d71f02"
        rel="bookmark"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/d226f189-
f83f-4569-95b8-622133d71f02"
        type="application/vnd.openstack.image" rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2012-10-08T18:56:29Z"
      name="Windows Server 2012 (with updates)"
```

```
created="2012-10-08T18:40:14Z" minDisk="40" progress="100"
minRam="1024" id="2748ee06-ff35-4518-9759-4acb57bad4c3"
OS-DCF:diskConfig="MANUAL">
<metadata>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__options">4</meta>
  <meta key="image_type">base</meta>
  <meta key="com.rackspace__1__source">kickstart</meta>
  <meta key="org.openstack__1__os_distro"
    >com.microsoft.server</meta>
  <meta key="auto_disk_config">False</meta>
  <meta key="os_type">windows</meta>
  <meta key="com.rackspace__1__visible_rackconnect">1</meta>
  <meta key="com.rackspace__1__build_rackconnect">1</meta>
  <meta key="com.rackspace__1__visible_managed">1</meta>
  <meta key="com.rackspace__1__build_core">1</meta>
  <meta key="rax_activation_profile">windows</meta>
  <meta key="org.openstack__1__os_version">2012.0</meta>
  <meta key="org.openstack__1__architecture">x64</meta>
  <meta key="com.rackspace__1__build_managed">1</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/2748ee06-
ff35-4518-9759-4acb57bad4c3"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/2748ee06-
ff35-4518-9759-4acb57bad4c3"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/2748ee06-
ff35-4518-9759-4acb57bad4c3"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-10-02T16:16:58Z"
  name="CentOS 5.8" created="2012-10-02T16:14:49Z" minDisk="10"
  progress="100" minRam="256"
  id="acf05b3c-5403-4cf0-900c-9b12b0db0644"
  OS-DCF:diskConfig="AUTO">
<metadata>
  <meta key="os_distro">centos</meta>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__build_rackconnect">0</meta>
  <meta key="auto_disk_config">True</meta>
  <meta key="com.rackspace__1__options">0</meta>
  <meta key="image_type">base</meta>
  <meta key="org.openstack__1__os_version">5.8</meta>
  <meta key="rax_managed">false</meta>
  <meta key="os_version">5.8</meta>
  <meta key="rax_options">0</meta>
  <meta key="com.rackspace__1__visible_rackconnect">0</meta>
  <meta key="org.openstack__1__os_distro">org.centos</meta>
  <meta key="com.rackspace__1__visible_managed">0</meta>
  <meta key="com.rackspace__1__build_core">1</meta>
  <meta key="arch">x86-64</meta>
  <meta key="os_type">linux</meta>
  <meta key="org.openstack__1__architecture">x64</meta>
  <meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/
acf05b3c-5403-4cf0-900c-9b12b0db0644"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/
acf05b3c-5403-4cf0-900c-9b12b0db0644"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/
acf05b3c-5403-4cf0-900c-9b12b0db0644"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-09-27T20:57:51Z"
  name="Arch 2012.08" created="2012-09-27T20:55:53Z"
  minDisk="10" progress="100" minRam="256"
  id="c94f5e59-0760-467a-ae70-9a37cfa6b94e"
  OS-DCF:diskConfig="AUTO">
<metadata>
  <meta key="os_distro">arch</meta>
```

```
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">0</meta>
<meta key="auto_disk_config">True</meta>
<meta key="com.rackspace__1__options">0</meta>
<meta key="image_type">base</meta>
<meta key="org.openstack__1__os_version">2012.08</meta>
<meta key="rax_managed">false</meta>
<meta key="os_version">2012.08</meta>
<meta key="rax_options">0</meta>
<meta key="com.rackspace__1__visible_rackconnect">0</meta>
<meta key="org.openstack__1__os_distro"
  >org.archlinux</meta>
<meta key="com.rackspace__1__visible_managed">0</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="arch">x86-64</meta>
<meta key="os_type">linux</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/c94f5e59-0760-467a-
ae70-9a37cfa6b94e"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/c94f5e59-0760-467a-
ae70-9a37cfa6b94e"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/c94f5e59-0760-467a-
ae70-9a37cfa6b94e"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-09-27T15:08:36Z"
  name="Gentoo 12.3" created="2012-09-27T15:05:10Z" minDisk="10"
  progress="100" minRam="256"
  id="110d5bd8-a0dc-4cf5-8e75-149a58c17bbf"
  OS-DCF:diskConfig="AUTO">
<metadata>
  <meta key="os_distro">gentoo</meta>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__options">0</meta>
  <meta key="auto_disk_config">True</meta>
  <meta key="com.rackspace__1__build_rackconnect">0</meta>
  <meta key="image_type">base</meta>
  <meta key="org.openstack__1__os_version">12.3</meta>
  <meta key="os_version">11.0</meta>
  <meta key="com.rackspace__1__visible_rackconnect">0</meta>
  <meta key="org.openstack__1__os_distro">org.gentoo</meta>
  <meta key="com.rackspace__1__visible_managed">0</meta>
  <meta key="com.rackspace__1__build_core">1</meta>
  <meta key="os_type">linux</meta>
  <meta key="org.openstack__1__architecture">x64</meta>
  <meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/110d5bd8-
a0dc-4cf5-8e75-149a58c17bbf"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/110d5bd8-
a0dc-4cf5-8e75-149a58c17bbf"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/110d5bd8-
a0dc-4cf5-8e75-149a58c17bbf"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-09-26T02:17:36Z"
  name="Windows Server 2008 R2 SP1 + SharePoint Foundation 2010 SP1 & SQL Server 2008 R2 SP1
Std"
  created="2012-09-26T01:21:22Z" minDisk="80" progress="100"
  minRam="8192" id="9eb71a23-2c7e-479c-a6b1-b38aa64f172e"
  OS-DCF:diskConfig="MANUAL">
<metadata>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__options">12</meta>
  <meta key="image_type">base</meta>
  <meta key="com.rackspace__1__source">kickstart</meta>
  <meta key="org.openstack__1__os_distro">
```

```
>com.microsoft.server</meta>
<meta key="auto_disk_config">False</meta>
<meta key="os_type">windows</meta>
<meta key="com.rackspace__1__visible_rackconnect">0</meta>
<meta key="com.rackspace__1__build_rackconnect">0</meta>
<meta key="com.rackspace__1__visible_managed">0</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="rax_activation_profile">windows</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/9eb71a23-2c7e-479c-a6b1-
b38aa64f172e"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/9eb71a23-2c7e-479c-a6b1-
b38aa64f172e"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/9eb71a23-2c7e-479c-a6b1-
b38aa64f172e"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-09-26T02:01:04Z"
  name="Windows Server 2008 R2 SP1 + SharePoint Foundation 2010 SP1 & SQL Server 2008 R2 SP1
Express"
  created="2012-09-26T00:54:38Z" minDisk="80" progress="100"
  minRam="4096" id="7f7183b0-856c-4894-afae-9e52839ce197"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">4</meta>
    <meta key="image_type">base</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="auto_disk_config">False</meta>
    <meta key="os_type">windows</meta>
    <meta key="com.rackspace__1__visible_rackconnect">0</meta>
    <meta key="com.rackspace__1__build_rackconnect">0</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/7f7183b0-856c-4894-
afae-9e52839ce197"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/7f7183b0-856c-4894-
afae-9e52839ce197"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/7f7183b0-856c-4894-
afae-9e52839ce197"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-09-17T23:24:45Z"
  name="Windows Server 2012" created="2012-09-17T23:11:25Z"
  minDisk="40" progress="100" minRam="1024"
  id="ae49b64d-9d68-4b36-98ed-blce84944680"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">4</meta>
    <meta key="image_type">base</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="auto_disk_config">False</meta>
    <meta key="os_type">windows</meta>
    <meta key="com.rackspace__1__visible_rackconnect">0</meta>
    <meta key="com.rackspace__1__build_rackconnect">0</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
```



```
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="rax_activation_profile">windows</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">1</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/ae49b64d-9d68-4b36-98ed-
blce84944680"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/ae49b64d-9d68-4b36-98ed-
blce84944680"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/ae49b64d-9d68-4b36-98ed-
blce84944680"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-08-01T00:37:41Z"
  name="Ubuntu 10.04 LTS (Lucid Lynx)"
  created="2012-08-01T00:37:12Z" minDisk="10" progress="100"
  minRam="512" id="d531a2dd-7ae9-4407-bb5a-e5ea03303d98"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">ubuntu</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">10.04</meta>
    <meta key="auto_disk_config">true</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro">org.ubuntu</meta>
    <meta key="com.rackspace__1__visible_managed">1</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/d531a2dd-7ae9-4407-bb5a-
e5ea03303d98"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/d531a2dd-7ae9-4407-bb5a-
e5ea03303d98"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/d531a2dd-7ae9-4407-bb5a-
e5ea03303d98"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T21:19:45Z"
  name="Windows Server 2008 R2 SP1 + SQL Server 2012 Standard"
  created="2012-07-30T20:41:02Z" minDisk="80" progress="100"
  minRam="2048" id="f7d06722-2b30-4c02-b74d-da5a7337f357"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="os_distro">windows</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">0</meta>
    <meta key="com.rackspace__1__options">12</meta>
    <meta key="image_type">base</meta>
    <meta key="arch">x86-64</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="rax_managed">>false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="os_type">windows</meta>
    <meta key="rax_options">12</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
```

```
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/f7d06722-2b30-4c02-b74d-
da5a7337f357"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/f7d06722-2b30-4c02-b74d-
da5a7337f357"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/f7d06722-2b30-4c02-b74d-
da5a7337f357"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T21:18:23Z"
  name="Windows Server 2008 R2 SP1 + SQL Server 2012 Web"
  created="2012-07-30T20:40:26Z" minDisk="80" progress="100"
  minRam="2048" id="e7a1leed-d348-44da-8210-f136d4256e81"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="os_distro">windows</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">0</meta>
    <meta key="com.rackspace__1__options">36</meta>
    <meta key="image_type">base</meta>
    <meta key="arch">x86-64</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="rax_managed">>false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="os_type">windows</meta>
    <meta key="rax_options">12</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/e7a1leed-d348-44da-8210-
f136d4256e81"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/e7a1leed-d348-44da-8210-
f136d4256e81"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/e7a1leed-d348-44da-8210-
f136d4256e81"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T18:36:59Z"
  name="Windows Server 2008 R2 SP1 (with updates) + SQL Server 2012 Standard"
  created="2012-07-30T18:15:22Z" minDisk="80" progress="100"
  minRam="2048" id="e4589dc6-b972-482f-91ef-67feb891b559"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">12</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="os_distro">windows</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="source">kickstart</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="rax_managed">>false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
```

```
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="rax_options">12</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="arch">x86-64</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="auto_disk_config">0</meta>
<meta key="os_type">windows</meta>
<meta key="rax_activation_profile">windows</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/e4589dc6-
b972-482f-91ef-67feb891b559"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/e4589dc6-
b972-482f-91ef-67feb891b559"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/e4589dc6-
b972-482f-91ef-67feb891b559"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T18:11:47Z"
  name="Windows Server 2008 R2 SP1 + SQL Server 2008 R2 Web"
  created="2012-07-30T17:56:50Z" minDisk="80" progress="100"
  minRam="2048" id="d6153e86-f4e0-4053-a711-d35632e512cd"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="os_distro">windows</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">0</meta>
    <meta key="com.rackspace__1__options">36</meta>
    <meta key="image_type">base</meta>
    <meta key="arch">x86-64</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="rax_managed">>false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="os_type">windows</meta>
    <meta key="rax_options">12</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/d6153e86-f4e0-4053-a711-
d35632e512cd"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/d6153e86-f4e0-4053-a711-
d35632e512cd"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/d6153e86-f4e0-4053-a711-
d35632e512cd"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T17:05:36Z"
  name="Windows Server 2008 R2 SP1 (with updates) + SQL Server 2008 R2 SP1 Web"
  created="2012-07-30T16:45:00Z" minDisk="80" progress="100"
  minRam="2048" id="80599479-b5a2-49f2-bb46-2bc75a8be98b"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__options">36</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="os_distro">windows</meta>
    <meta key="org.openstack__1__os_distro">
```

```
>com.microsoft.server</meta>
<meta key="source">kickstart</meta>
<meta key="image_type">base</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="rax_managed">>false</meta>
<meta key="os_version">2008r2</meta>
<meta key="com.rackspace__1__build_managed">1</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="rax_options">12</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="arch">x86-64</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="auto_disk_config">0</meta>
<meta key="os_type">windows</meta>
<meta key="rax_activation_profile">windows</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/80599479-b5a2-49f2-
bb46-2bc75a8be98b"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/80599479-b5a2-49f2-
bb46-2bc75a8be98b"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/80599479-b5a2-49f2-
bb46-2bc75a8be98b"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T14:12:09Z"
  name="Windows Server 2008 R2 SP1 (with updates) + SQL Server 2012 Web"
  created="2012-07-30T13:49:18Z" minDisk="80" progress="100"
  minRam="2048" id="6f8ab5a1-42ff-433b-be40-e17374f2fff4"
  OS-DCF:diskConfig="MANUAL">
<metadata>
  <meta key="com.rackspace__1__build_rackconnect">1</meta>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__options">36</meta>
  <meta key="com.rackspace__1__build_core">1</meta>
  <meta key="com.rackspace__1__build_managed">1</meta>
  <meta key="com.rackspace__1__visible_rackconnect">1</meta>
  <meta key="os_distro">windows</meta>
  <meta key="org.openstack__1__os_distro"
    >com.microsoft.server</meta>
  <meta key="source">kickstart</meta>
  <meta key="image_type">base</meta>
  <meta key="org.openstack__1__os_version">2008.2</meta>
  <meta key="rax_managed">>false</meta>
  <meta key="os_version">2008r2</meta>
  <meta key="com.rackspace__1__build_managed">1</meta>
  <meta key="org.openstack__1__architecture">x64</meta>
  <meta key="com.rackspace__1__visible_core">1</meta>
  <meta key="com.rackspace__1__visible_rackconnect">1</meta>
  <meta key="rax_options">12</meta>
  <meta key="com.rackspace__1__visible_managed">1</meta>
  <meta key="arch">x86-64</meta>
  <meta key="com.rackspace__1__build_core">1</meta>
  <meta key="com.rackspace__1__build_rackconnect">1</meta>
  <meta key="com.rackspace__1__visible_managed">1</meta>
  <meta key="auto_disk_config">0</meta>
  <meta key="os_type">windows</meta>
  <meta key="rax_activation_profile">windows</meta>
</metadata>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/6f8ab5a1-42ff-433b-be40-
e17374f2fff4"
  rel="self"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/6f8ab5a1-42ff-433b-be40-
e17374f2fff4"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/6f8ab5a1-42ff-433b-be40-
e17374f2fff4"
  type="application/vnd.openstack.image" rel="alternate"/>
```

```
</image>
<image status="ACTIVE" updated="2012-07-30T01:33:16Z"
  name="Windows Server 2008 R2 SP1 (with updates) + SQL Server 2008 R2 SP1 Standard"
  created="2012-07-30T01:12:47Z" minDisk="80" progress="100"
  minRam="2048" id="535d5453-79dd-4635-bbd6-d87b1f1cd717"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="os_distro">windows</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">0</meta>
    <meta key="com.rackspace__1__options">12</meta>
    <meta key="source">kickstart</meta>
    <meta key="image_type">base</meta>
    <meta key="arch">x86-64</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="rax_options">12</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="com.rackspace__1__visible_managed">1</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="os_type">windows</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/535d5453-79dd-4635-bbd6-
d87b1f1cd717"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/535d5453-79dd-4635-bbd6-
d87b1f1cd717"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/535d5453-79dd-4635-bbd6-
d87b1f1cd717"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-30T16:32:54Z"
  name="Windows Server 2008 R2 SP1 + SQL Server 2008 R2 Standard"
  created="2012-07-29T23:40:06Z" minDisk="80" progress="100"
  minRam="2048" id="2a4a02aa-523a-4649-9802-3a09de8e5f1b"
  OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="os_distro">windows</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">0</meta>
    <meta key="com.rackspace__1__options">12</meta>
    <meta key="image_type">base</meta>
    <meta key="arch">x86-64</meta>
    <meta key="com.rackspace__1__source">kickstart</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">2008r2</meta>
    <meta key="os_type">windows</meta>
    <meta key="rax_options">12</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >com.microsoft.server</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="rax_activation_profile">windows</meta>
    <meta key="org.openstack__1__os_version">2008.2</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/
2a4a02aa-523a-4649-9802-3a09de8e5f1b"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/
2a4a02aa-523a-4649-9802-3a09de8e5f1b"
    rel="bookmark"/>
  <atom:link
```

```
href="https://dfw.servers.api.rackspacecloud.com/658405/images/
2a4a02aa-523a-4649-9802-3a09de8e5f1b"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-27T19:49:40Z"
name="Windows Server 2008 R2 SP1"
created="2012-07-27T19:41:40Z" minDisk="40" progress="100"
minRam="1024" id="b9ea8426-8f43-4224-a182-7cdb2bb897c8"
OS-DCF:diskConfig="MANUAL">
<metadata>
<meta key="os_distro">windows</meta>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="auto_disk_config">0</meta>
<meta key="com.rackspace__1__options">4</meta>
<meta key="image_type">base</meta>
<meta key="arch">x86-64</meta>
<meta key="com.rackspace__1__source">kickstart</meta>
<meta key="org.openstack__1__os_distro"
>com.microsoft.server</meta>
<meta key="rax_managed">>false</meta>
<meta key="os_version">2008r2</meta>
<meta key="os_type">windows</meta>
<meta key="rax_options">4</meta>
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="org.openstack__1__os_distro"
>com.microsoft.server</meta>
<meta key="com.rackspace__1__visible_managed">0</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="rax_activation_profile">windows</meta>
<meta key="org.openstack__1__os_version">2008.2</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/b9ea8426-8f43-4224-
a182-7cdb2bb897c8"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/b9ea8426-8f43-4224-
a182-7cdb2bb897c8"
rel="bookmark"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/b9ea8426-8f43-4224-
a182-7cdb2bb897c8"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-27T19:07:10Z"
name="FreeBSD 9" created="2012-07-27T19:06:52Z" minDisk="10"
progress="100" minRam="256"
id="c79fecf7-2c37-4c51-a240-e9fa913c90a3"
OS-DCF:diskConfig="MANUAL">
<metadata>
<meta key="vm_mode">hvm</meta>
<meta key="os_distro">freebsd</meta>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">0</meta>
<meta key="auto_disk_config">False</meta>
<meta key="com.rackspace__1__options">0</meta>
<meta key="image_type">base</meta>
<meta key="org.openstack__1__os_version">9</meta>
<meta key="rax_managed">>false</meta>
<meta key="os_version">9</meta>
<meta key="rax_options">0</meta>
<meta key="com.rackspace__1__visible_rackconnect">0</meta>
<meta key="org.openstack__1__os_distro">org.freebsd</meta>
<meta key="com.rackspace__1__visible_managed">0</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="arch">x86-64</meta>
<meta key="os_type">freebsd</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/c79fecf7-2c37-4c51-a240-
e9fa913c90a3"
rel="self"/>
<atom:link
```

```
href="https://dfw.servers.api.rackspacecloud.com/658405/images/c79fecf7-2c37-4c51-a240-e9fa913c90a3"
  rel="bookmark"/>
<atom:link
  href="https://dfw.servers.api.rackspacecloud.com/658405/images/c79fecf7-2c37-4c51-a240-e9fa913c90a3"
  type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-07-25T22:35:49Z"
  name="CentOS 6.3" created="2012-07-25T22:35:36Z" minDisk="10"
  progress="100" minRam="256"
  id="c195ef3b-9195-4474-b6f7-16e5bd86acd0"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">centos</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">6.3</meta>
    <meta key="rax_managed">>false</meta>
    <meta key="os_version">6.3</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro">org.centos</meta>
    <meta key="com.rackspace__1__visible_managed">1</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/c195ef3b-9195-4474-b6f7-16e5bd86acd0"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/c195ef3b-9195-4474-b6f7-16e5bd86acd0"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/c195ef3b-9195-4474-b6f7-16e5bd86acd0"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-06-20T03:21:09Z"
  name="Fedora 17 (Beefy Miracle)"
  created="2012-06-20T03:20:52Z" minDisk="0" progress="100"
  minRam="0" id="d42f821e-c2d1-4796-9f07-af5ed7912d0e"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="vm_mode">pv</meta>
    <meta key="os_distro">fedora</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">17</meta>
    <meta key="os_version">17</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >org.fedoraproject</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/d42f821e-c2d1-4796-9f07-af5ed7912d0e"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/d42f821e-c2d1-4796-9f07-af5ed7912d0e"
```

```
        rel="bookmark"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/d42f821e-c2d1-4796-9f07-af5ed7912d0e"
        type="application/vnd.openstack.image" rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2012-05-31T19:31:23Z"
      name="CentOS 6.2" created="2012-05-31T19:30:45Z" minDisk="0"
      progress="100" minRam="0"
      id="0cab6212-f231-4abd-9c70-608d0d0e04ba"
      OS-DCF:diskConfig="AUTO">
      <metadata>
        <meta key="vm_mode">pv</meta>
        <meta key="os_distro">centos</meta>
        <meta key="com.rackspace__1__visible_core">1</meta>
        <meta key="com.rackspace__1__build_rackconnect">1</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="com.rackspace__1__options">0</meta>
        <meta key="image_type">base</meta>
        <meta key="org.openstack__1__os_version">6.2</meta>
        <meta key="os_version">6.2</meta>
        <meta key="rax_options">0</meta>
        <meta key="com.rackspace__1__visible_rackconnect">1</meta>
        <meta key="org.openstack__1__os_distro">org.centos</meta>
        <meta key="com.rackspace__1__visible_managed">0</meta>
        <meta key="com.rackspace__1__build_core">1</meta>
        <meta key="arch">x86-64</meta>
        <meta key="os_type">linux</meta>
        <meta key="org.openstack__1__architecture">x64</meta>
        <meta key="com.rackspace__1__build_managed">0</meta>
      </metadata>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/0cab6212-f231-4abd-9c70-608d0d0e04ba"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/0cab6212-f231-4abd-9c70-608d0d0e04ba"
        rel="bookmark"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/0cab6212-f231-4abd-9c70-608d0d0e04ba"
        type="application/vnd.openstack.image" rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2012-05-17T17:14:17Z"
      name="Red Hat Enterprise Linux 5.5"
      created="2012-04-06T22:11:47Z" minDisk="0" progress="100"
      minRam="256" id="644be485-411d-4bac-aba5-5f60641d92b5"
      OS-DCF:diskConfig="AUTO">
      <metadata>
        <meta key="os_distro">rhel</meta>
        <meta key="com.rackspace__1__visible_core">1</meta>
        <meta key="com.rackspace__1__build_rackconnect">1</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="com.rackspace__1__options">1</meta>
        <meta key="image_type">base</meta>
        <meta key="arch">x86-64</meta>
        <meta key="org.openstack__1__os_version">5.5</meta>
        <meta key="rax_managed">>false</meta>
        <meta key="os_version">5.5</meta>
        <meta key="rax_options">1</meta>
        <meta key="com.rackspace__1__visible_rackconnect">1</meta>
        <meta key="org.openstack__1__os_distro">com.redhat</meta>
        <meta key="com.rackspace__1__visible_managed">1</meta>
        <meta key="com.rackspace__1__build_core">1</meta>
        <meta key="rax_activation_profile">redhat</meta>
        <meta key="os_type">linux</meta>
        <meta key="org.openstack__1__architecture">x64</meta>
        <meta key="com.rackspace__1__build_managed">1</meta>
      </metadata>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/644be485-411d-4bac-aba5-5f60641d92b5"
        rel="self"/>
      <atom:link
        href="https://dfw.servers.api.rackspacecloud.com/658405/images/644be485-411d-4bac-aba5-5f60641d92b5"
        rel="bookmark"/>
      <atom:link
```



```
href="https://dfw.servers.api.rackspacecloud.com/658405/images/644be485-411d-4bac-
aba5-5f60641d92b5"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-05-17T17:14:18Z"
name="Ubuntu 11.04 (Natty Narwhal)"
created="2012-02-28T21:24:42Z" minDisk="10" progress="100"
minRam="256" id="8bf22129-8483-462b-a020-1754ec822770"
OS-DCF:diskConfig="AUTO">
<metadata>
<meta key="os_distro">ubuntu</meta>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">1</meta>
<meta key="auto_disk_config">True</meta>
<meta key="com.rackspace__1__options">0</meta>
<meta key="image_type">base</meta>
<meta key="org.openstack__1__os_version">11.10</meta>
<meta key="rax_managed">false</meta>
<meta key="os_version">11.10</meta>
<meta key="rax_options">0</meta>
<meta key="com.rackspace__1__visible_rackconnect">1</meta>
<meta key="org.openstack__1__os_distro">com.ubuntu</meta>
<meta key="com.rackspace__1__visible_managed">1</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="arch">x86-64</meta>
<meta key="os_type">linux</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">1</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/8bf22129-8483-462b-
a020-1754ec822770"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/8bf22129-8483-462b-
a020-1754ec822770"
rel="bookmark"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/8bf22129-8483-462b-
a020-1754ec822770"
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-05-17T17:12:59Z"
name="openSUSE 12.1" created="2012-02-28T19:42:04Z"
minDisk="10" progress="100" minRam="256"
id="096c55e5-39f3-48cf-a413-68d9377a3ab6"
OS-DCF:diskConfig="AUTO">
<metadata>
<meta key="os_distro">opensuse</meta>
<meta key="com.rackspace__1__visible_core">1</meta>
<meta key="com.rackspace__1__build_rackconnect">0</meta>
<meta key="auto_disk_config">True</meta>
<meta key="com.rackspace__1__options">0</meta>
<meta key="image_type">base</meta>
<meta key="org.openstack__1__os_version">12.0</meta>
<meta key="rax_managed">false</meta>
<meta key="os_version">12</meta>
<meta key="rax_options">0</meta>
<meta key="com.rackspace__1__visible_rackconnect">0</meta>
<meta key="org.openstack__1__os_distro"
>org.opensuse</meta>
<meta key="com.rackspace__1__visible_managed">0</meta>
<meta key="com.rackspace__1__build_core">1</meta>
<meta key="arch">x86-64</meta>
<meta key="os_type">linux</meta>
<meta key="org.openstack__1__architecture">x64</meta>
<meta key="com.rackspace__1__build_managed">0</meta>
</metadata>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/096c55e5-39f3-48cf-
a413-68d9377a3ab6"
rel="self"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/096c55e5-39f3-48cf-
a413-68d9377a3ab6"
rel="bookmark"/>
<atom:link
href="https://dfw.servers.api.rackspacecloud.com/658405/images/096c55e5-39f3-48cf-
a413-68d9377a3ab6"
```

```
type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-05-17T17:13:50Z"
  name="Debian 6 (Squeeze)" created="2012-02-28T19:41:44Z"
  minDisk="10" progress="100" minRam="256"
  id="a10eacf7-ac15-4225-b533-5744f1fe47c1"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">debian</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">6.0</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">6</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro">org.debian</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/a10eacf7-ac15-4225-
b533-5744f1fe47c1"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/a10eacf7-ac15-4225-
b533-5744f1fe47c1"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/a10eacf7-ac15-4225-
b533-5744f1fe47c1"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-05-17T17:12:26Z"
  name="Fedora 16 (Verne)" created="2012-02-28T19:41:07Z"
  minDisk="10" progress="100" minRam="256"
  id="bca91446-e60e-42e7-9e39-0582e7e20fb9"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">fedora</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">16.0</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">16</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro"
      >org.fedoraproject</meta>
    <meta key="com.rackspace__1__visible_managed">0</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">0</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/bca91446-
e60e-42e7-9e39-0582e7e20fb9"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/bca91446-
e60e-42e7-9e39-0582e7e20fb9"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/bca91446-
e60e-42e7-9e39-0582e7e20fb9"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
```

```
<image status="ACTIVE" updated="2012-05-17T17:14:17Z"
  name="CentOS 5.6" created="2012-02-28T19:40:46Z" minDisk="10"
  progress="100" minRam="256"
  id="03318d19-b6e6-4092-9b5c-4758ee0ada60"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">centos</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">5.6</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">5.6</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro">org.centos</meta>
    <meta key="com.rackspace__1__visible_managed">1</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/03318d19-
b6e6-4092-9b5c-4758ee0ada60"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/03318d19-
b6e6-4092-9b5c-4758ee0ada60"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/03318d19-
b6e6-4092-9b5c-4758ee0ada60"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2012-02-28T19:39:05Z"
  name="Ubuntu 11.10 (Oneiric Ocelot)"
  created="2012-02-28T19:38:57Z" minDisk="10" progress="100"
  minRam="256" id="3afe97b2-26dc-49c5-a2cc-a2fc8d80c001"
  OS-DCF:diskConfig="AUTO">
  <metadata>
    <meta key="os_distro">ubuntu</meta>
    <meta key="com.rackspace__1__visible_core">1</meta>
    <meta key="com.rackspace__1__build_rackconnect">1</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="com.rackspace__1__options">0</meta>
    <meta key="image_type">base</meta>
    <meta key="org.openstack__1__os_version">11.10</meta>
    <meta key="rax_managed">false</meta>
    <meta key="os_version">11.10</meta>
    <meta key="rax_options">0</meta>
    <meta key="com.rackspace__1__visible_rackconnect">1</meta>
    <meta key="org.openstack__1__os_distro">org.ubuntu</meta>
    <meta key="com.rackspace__1__visible_managed">1</meta>
    <meta key="com.rackspace__1__build_core">1</meta>
    <meta key="arch">x86-64</meta>
    <meta key="os_type">linux</meta>
    <meta key="org.openstack__1__architecture">x64</meta>
    <meta key="com.rackspace__1__build_managed">1</meta>
  </metadata>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/v2/658405/images/3afe97b2-26dc-49c5-a2cc-
a2fc8d80c001"
    rel="self"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/3afe97b2-26dc-49c5-a2cc-
a2fc8d80c001"
    rel="bookmark"/>
  <atom:link
    href="https://dfw.servers.api.rackspacecloud.com/658405/images/3afe97b2-26dc-49c5-a2cc-
a2fc8d80c001"
    type="application/vnd.openstack.image" rel="alternate"/>
</image>
</images>
```

2.6.3. Delete Image

Verb	URI	Description
DELETE	/images/ <i>id</i>	Deletes the specified image.

Normal Response Code: 204

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

Status Transition:	ACTIVE → DELETED
	ERROR → DELETED

This operation deletes the specified image from the system.

Specify the image ID as *id* in the URI.

Images are immediately removed.

Specify the image ID as *id* in the URI.

This operation does not require a request body.

This operation does not contain a response body. Successful deletion returns a 204 response code.

2.7. Metadata

You can list and set metadata for a server or an image after you create the server or image.

Verb	URI	Description
GET	/servers/ <i>id</i> /metadata	Lists metadata associated with a server or an image.
PUT	/servers/ <i>id</i> /metadata	Sets metadata for the specified server or image.
POST	/servers/ <i>id</i> /metadata	Updates metadata items for the specified server or image.
GET	/servers/ <i>id</i> /metadata/ <i>key</i>	Gets a metadata item associated with a server or an image.
PUT	/servers/ <i>id</i> /metadata/ <i>key</i>	Sets a metadata item for a specified server or image.
DELETE	/servers/ <i>id</i> /metadata/ <i>key</i>	Deletes a metadata item for a specified server or image.

2.7.1. List Metadata

Verb	URI	Description
GET	/servers/ <i>id</i> /metadata	Lists metadata associated with a server.
GET	/images/ <i>id</i> /metadata	Lists metadata associated with an image.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413), serviceUnavailable (503)

Lists all metadata associated with a server or an image.

Specify the server or image ID as *id* in the URI.

This operation does not require a request body.

This operation returns a response body.

Example 2.68. List Metadata: JSON Response

```
{
  "metadata": {
    "Label" : "Web",
    "Version" : "2.1"
  }
}
```

Example 2.69. List Metadata: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="Label">Web</meta>
  <meta key="Version">2.1</meta>
</metadata>
```

2.7.2. Set Metadata

Verb	URI	Description
PUT	/servers/ <i>id</i> /metadata	Sets metadata for the specified server.
PUT	/images/ <i>id</i> /metadata	Sets metadata for the specified image.

Normal Response Code: 200

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serviceUnavailable (503)

Sets metadata for the specified server or image.

Specify the server or image ID as *id* in the URI.

In the request body, specify the *metadata* element followed by attributes.

The following table describes the attributes that you can set in the request body:

Table 2.11. Set Metadata Request Attributes

Attribute	Description	Required
meta	Key and value pairs for metadata.	Yes

Existing metadata items are replaced with the ones provided in the request regardless of the names of the original metadata items.

If you exceed the maximum number of metadata items, the call throws an overLimit (413) fault.

You can query the maximum number of key-value pairs that can be supplied for each server through the `maxServerMeta` absolute limit. You can query the maximum number of key-value pairs for an image through the `maxImageMeta` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

Example 2.70. Set Metadata: JSON Request

```
{
  "metadata": {
    "Label" : "Web",
    "Version" : "2.1"
  }
}
```

Example 2.71. Set Metadata: JSON Response

```
{
  "metadata": {
    "Label" : "Web",
    "Version" : "2.1"
  }
}
```

Example 2.72. Set Metadata: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="Label">Web</meta>
  <meta key="Version">2.1</meta>
</metadata>
```

Example 2.73. Set Metadata: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">  
  <meta key="Label">Web</meta>  
  <meta key="Version">2.1</meta>  
</metadata>
```

2.7.3. Update Metadata

Verb	URI	Description
POST	/servers/ <i>id</i> /metadata	Updates metadata items for the specified server.
POST	/images/ <i>id</i> /metadata	Updates metadata items for the specified image.

Normal Response Code: 200

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415), serviceUnavailable (503)

Updates metadata items for a specified server or image.

Specify the server or image ID as *id* in the URI.

In the request body, specify the *metadata* element followed by attributes.

The following table describes the attributes that you can set in the request body:

Table 2.12. Update Metadata Request Attributes

Attribute	Description	Required
meta	Key and value pairs for metadata.	Yes

Updates replace existing metadata items with the same key. Items that are not explicitly mentioned in the request are not modified.

If you exceed the maximum number of metadata items in the request, the call throws an overLimit (413) fault. You can query the maximum number of key-value pairs that can be supplied for each server through the `maxServerMeta` absolute limit. You can query the maximum number of key-value pairs for an image through the `maxImageMeta` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

Example 2.74. Update Metadata: JSON Request

```
{
  "metadata": {
    "Label" : "Web2"
  }
}
```

Example 2.75. Update Metadata: JSON Response

```
{
  "metadata" : {
    "Label" : "Web2",
    "Version" : "2.1"
  }
}
```

Example 2.76. Update Metadata: XML Request

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

<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="Label">Web2</meta>
</metadata>
```


Example 2.77. Update Metadata: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="Label">Web2</meta>
  <meta key="Version">2.1</meta>
</metadata>
```

2.7.4. Get Metadata Item

Verb	URI	Description
GET	/servers/ <i>id</i> /metadata/ <i>key</i>	Gets a metadata item associated with a server.
GET	/images/ <i>id</i> /metadata/ <i>key</i>	Gets a metadata item associated with an image.

Normal Response Codes: 200 and 203

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), badRequest (400), forbidden (403), itemNotFound (404), Method Not Allowed (405), overLimit (413)

Retrieves a single metadata item by key.

Specify the server or image ID as *id* and the key as *key* in the URI.

This operation does not require a request body.

This operation returns a response body.

Example 2.78. Get Metadata Item: JSON Response

```
{
  "meta" : {
    "Label" : "Web"
  }
}
```

Example 2.79. Get Metadata Item: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<meta
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  key="Label">Web</meta>
```

2.7.5. Set Metadata Item

Verb	URI	Description
PUT	/servers/ <i>id</i> /metadata/ <i>key</i>	Sets a metadata item for a specified server.
PUT	/images/ <i>id</i> /metadata/ <i>key</i>	Sets a metadata item for a specified image.

Normal Response Code: 200

Error Response Codes: computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), badRequest (400), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), badMediaType (415)

Sets a metadata item by its key.

Specify the server or image ID as *id* and the metadata key as *key* in the URI.

You must also specify the `metadata` element followed by the key and value pair in the request body. The key specified in the request body must match the key specified in the URI request.

The following table describes the attributes that you can set in the request body:

Table 2.13. Set Metadata Item Request Attributes

Attribute	Description	Required
meta	Key and value pairs for metadata. The key specified must match the key specified in the URI request.	Yes

If you exceed the maximum number of metadata items in the request, the call throws an `overLimit (413)` fault. You can query the maximum number of key-value pairs that can be supplied for each server through the `maxServerMeta` absolute limit. You can query the maximum number of key-value pairs for an image through the `maxImageMeta` absolute limit. See [Section 1.8.3, "Get Limits" \[25\]](#).

Example 2.80. Set Metadata Item: JSON Request

```
{
  "meta" : {
    "Label" : "Web"
  }
}
```

Example 2.81. Set Metadata Item: JSON Response

```
{
  "meta" : {
    "Label" : "Web"
  }
}
```

Example 2.82. Set Metadata Item: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<meta
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  key="Label">Web</meta>
```

Example 2.83. Set Metadata Item: XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<meta
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  key="Label">Web</meta>
```

2.7.6. Delete Metadata Item

Verb	URI	Description
DELETE	/servers/ <i>id</i> /metadata/ <i>key</i>	Deletes a metadata item for the specified server.
DELETE	/images/ <i>id</i> /metadata/ <i>key</i>	Deletes a metadata item for the specified image.

Normal Response Code: 204

Error Response Codes: computeFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), Method Not Allowed (405), buildInProgress (409), overLimit (413), serviceUnavailable (503)

Deletes a metadata item.

Specify the server or image ID as *id* and the key as *key* in the URI.

The operation does not require a request body.

The operation does not return a response body.

Successful deletion returns a 204 response code.

3. Rackspace Extensions

Rackspace provides the following extensions to the OpenStack Compute API v2:

- **Bandwidth stats extension.** Shows bandwidth usage statistics for a given audit period.
- **Disk configuration extension.** Enables control of how the disk is partitioned when a server is created, rebuilt, or resized.
- **Extended status extension.** Shows extended statuses including the VM, task, and power statuses in the response bodies for the [list servers](#) and [get server details](#) calls.
- **Rescue mode extension.** Creates a new server with the file system for the specified version of Cloud Servers mounted to fix file system and configuration errors.
- **Used limits extension.** Returns the amount of absolute limit capacity that is currently used.
- **Volume attachment extension.** In conjunction with the Cloud Block Storage API, you can attach a volume to a server instance, list volume attachments for a server instance, get volume details for a volume attachment, and delete a volume attachment.

3.1. Bandwidth Stats Extension

The bandwidth stats extension adds a namespace-qualified bandwidth element within the server element in the response for a list servers or get server details API operation. The bandwidth element contains one or more interface elements with bandwidth usage statistics for each interface during the audit period.

See [Section 2.1.1, "List Servers" \[46\]](#) and [Section 2.1.3, "Get Server Details" \[63\]](#).

The namespace for this extended attribute is:

```
xmlns:rax-bandwidth="http://docs.rackspace.com/servers/api/ext/  
server_bandwidth/"
```

3.2. Disk Configuration Extension

The disk configuration extension adds a `OS-DCF:diskConfig` attribute on images and servers that controls how the disk is partitioned when servers are created, rebuilt, or resized. A server inherits the `OS-DCF:diskConfig` value from the image it was created with, and an image inherits the `OS-DCF:diskConfig` value of the server from which it was created. To override the inherited setting, you can include the `OS-DCF:diskConfig` attribute in the request body of a server create, rebuild, or resize request. Valid `OS-DCF:diskConfig` values are:

- **AUTO.** The server is built with a single partition the size of the target flavor disk. The file system is automatically adjusted to fit the entire partition. This keeps things simple and automated. **AUTO** is valid only for images and servers with a single partition that use the EXT3 file system. This is the default setting for applicable Rackspace base images.
- **MANUAL.** The server is built using whatever partition scheme and file system is in the source image. If the target flavor disk is larger, the remaining disk space is left

unpartitioned. This enables images to have non-EXT3 file systems, multiple partitions, and so on, and enables you to manage the disk configuration.



Note

Although Rackspace Windows images are configured with a `OS-DCF:diskConfig` value of `MANUAL`, the NTFS file system expands to the entire partition on only the first boot.

Resizing down requires the server to have a `OS-DCF:diskConfig` value of `AUTO`.

The namespace for this extended attribute is:

```
xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
```

3.2.1. Changes to Get Server/Image Details

A **GET** request against the `/servers/detail`, `/servers/id`, `/images/detail`, or `/images/id` resource returns the `OS-DCF:diskConfig` extended attribute. See the following examples:

- [Example 2.2, "List Servers: JSON Response \(detail\)" \[50\]](#)
- [Example 2.4, "List Servers: XML Response \(detail\)" \[53\]](#).
- [Example 2.14, "Get Server Details: JSON Response" \[66\]](#)
- [Example 2.15, "Get Server Details: XML Response" \[67\]](#).
- [Example 2.64, "List Images: JSON Response \(detail\)" \[108\]](#)
- [Example 2.65, "List Images: XML Response \(detail\)" \[110\]](#).
- [Example 2.66, "Get Image Details: JSON Response" \[110\]](#)
- [Example 2.67, "Get Image Details: XML Response" \[112\]](#).

3.2.2. Changes to Create Server

When you create a server from an image with the `OS-DCF:diskConfig` value set to `AUTO`, the server is built with a single partition that is expanded to the disk size of the flavor selected. When you set the `OS-DCF:diskConfig` attribute to `MANUAL`, the server is built by using the partition scheme and file system that is in the source image. If the target flavor disk is larger, remaining disk space is left unpartitioned. A server inherits the `OS-DCF:diskConfig` attribute from the image from which it is created. However, you can override the `OS-DCF:diskConfig` value when you create a server, as follows:

Example 3.1. Create Server with OS-DCF:diskConfig: JSON Request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "5f68715f-201f-4600-b5a1-0b97e2b1cb31",
    "flavorRef" : "2",
    "OS-DCF:diskConfig" : "MANUAL",
    "metadata" : {
```

```
    "My Server Name" : "Ubuntu 10.04 LTS manual"
  },
  "personality" : [
    {
      "path" : "/etc/banner.txt",
      "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lvbi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYmV5b3Vz
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlm dCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCg0KLWJpY2hhcmQgQmFjaA=="
    }
  ]
}
```

Example 3.2. Create Server with OS-DCF:diskConfig: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="5f68715f-201f-4600-b5a1-0b97e2b1cb31"
  flavorRef="2"
  OS-DCF:diskConfig="MANUAL"
  name="new-server-test"
>
  <metadata>
    <meta key="My Server Name">Ubuntu 10.04 LTS manual</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lvbi4uLnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYmV5b3Vz
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlm dCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCg0KLWJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>
```

In this example, the server is created with OS-DCF:diskConfig set to MANUAL, regardless of what value the image OS-DCF:diskConfig attribute is set to. Images also inherit the OS-DCF:diskConfig value from a server. So, if an image is created from the server, it also has a OS-DCF:diskConfig value of MANUAL.

3.2.3. Changes to Rebuild Server

You can set the OS-DCF:diskConfig attribute when you rebuild a server. In the following examples, the OS-DCF:diskConfig attribute is set to MANUAL, which allows unused disk space to be used for other partitions after the server is rebuilt.

If you do not set the OS-DCF:diskConfig attribute is not set during the rebuild, the original value of the attribute is retained.

Example 3.3. Rebuild Server with OS-DCF:diskConfig: JSON Request

```
{
  "rebuild" : {
```



```
"name" : "new-server-test",
"imageRef" : "d42f821e-c2d1-4796-9f07-af5ed7912d0e",
"flavorRef" : "2",
"diskConfig" : "manual",
"adminPass" : "diane123",
"metadata" : {
    "My Server Name" : "Apache1"
},
"personality" : [
    {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHNN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHNN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHhvIGdvIG5vdy4g
QnV0IHROZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRO
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYm5kIHlv
dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
    }
]
```

Example 3.4. Rebuild Server with OS-DCF:diskConfig: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<rebuild xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="rebuild-server"
  imageRef="d6dd6c70-a122-4391-91a8-decbla356549"
  OS-DCF:diskConfig="MANUAL"
  adminPass="diane123">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHNN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHNN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lubi4uLnRoXMGaXMGdGhlIHBSYWNlIHhvIGdvIG5vdy4g
QnV0IHROZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRO
ZSBwYXR0ZXJucyBiZWphbmQgYWxsIGNsb3VkcycwYm5kIHlv
dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA== </file>
    </personality>
  </rebuild>
```

3.2.4. Changes to Resize Server

You can set the `OS-DCF:diskConfig` attribute when you resize a server, which enables you to change the value of the attribute when you scale a server up or down.

If you do not set the `OS-DCF:diskConfig` attribute during the resize, the original value of the attribute is retained.

Example 3.5. Resize Server with OS-DCF:diskConfig: JSON Request

```
{
  "resize" : {
    "flavorRef" : "3",
    "diskConfig" : "manual"
  }
}
```

Example 3.6. Resize Server with OS-DCF:diskConfig: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<resize xmlns="http://docs.openstack.org/compute/api/v1.1"
  flavorRef="2"
  OS-DCF:diskConfig="MANUAL" />
```

3.3. Extended Status Extension

The extended status extension displays the VM, task, and power statuses for servers.

The extension displays these statuses in the following fields in the response bodies for the [list servers](#) and [get server details](#) calls:

Table 3.1. Extended Status Extension

Field	Description
OS-EXT-STS:vm_state	The virtual machine (VM) status. For possible values, see OS-EXT-STS:vm_state [] .
OS-EXT-STS:task_state	The task status. For possible values, see OS-EXT-STS:task_state [] .
OS-EXT-STS:power_state	The power status. For possible values, see OS-EXT-STS:power_state [] .



Note

The API does not regulate the VM and task status values so it is possible that these status values can be added, removed, or renamed.

Currently, the possible values for the VM, task, and power status fields are:

OS-EXT-STS:vm_state	The virtual machine (VM) status. Possible values are: <ul style="list-style-type: none">• active• build• deleted• error• paused• rescued• resized• soft_deleted• stopped• suspended
OS-EXT-STS:task_state	The task status. Possible values are: <ul style="list-style-type: none">• block_device_mapping• deleting• image_snapshot• image_backup• migrating• networking• pausing• powering_off• powering_on• rebooting

- rebooting_hard
- rebuilding
- rebuild_block_device_mapping
- rebuild_spawning
- rescuing
- resize_confirming
- resize_finish
- resize_migrated
- resize_migrating
- resize_prep
- resize_reverting
- resuming
- scheduling
- spawning
- starting
- stopping
- suspending
- unpausing
- unrescuing
- updating_password

OS-EXT-STS:power_state The power status. Possible values are:

- 0. The instance is powered down.
- 1. The instance is powered up.
- 4. The instance is shut off.

The following table shows the server statuses that correspond with the VM and tasks statuses:

Table 3.2. Server Statuses and Corresponding VM and Task Statuses

Server status	OS-EXT-STS:vm_state	OS-EXT-STS:task_state
ACTIVE	active	^a
HARD_REBOOT	active	rebooting_hard
MIGRATING	active	migrating
PASSWORD	active	updating_password
REBOOT	active	rebooting
REBUILD	active	rebuilding
REBUILD	active	rebuild_block_device_mapping
REBUILD	active	rebuild_spawning
RESIZE	active	resize_prep
RESIZE	active	resize_migrating
RESIZE	active	resize_migrated
RESIZE	active	resize_finish
BUILD	building	^a
DELETED	deleted	^a
ERROR	error	^a
PAUSED	paused	^a
RESCUE	rescued	^a

Server status	OS-EXT-STS:vm_state	OS-EXT-STS:task_state
VERIFY_RESIZE	resized	^a
REVERT_RESIZE	resized	resize_reverting
DELETED	soft_deleted	^a
SHUTOFF	stopped	^a
SUSPENDED	suspended	^a

^aPossible task statuses include the following:

- block_device_mapping
- deleting
- image_snapshot. Indicates that the image creation is a result of a create image action. This state is set for the duration of the backup. image_backup is the results of a scheduled backup.
- image_backup. Indicates that the image creation is a result of a scheduled backup action. This state is set for the duration of the backup.
- migrating
- networking
- pausing
- powering_off
- powering_on
- rebooting
- rebooting_hard
- rebuilding
- rebuild_block_device_mapping
- rebuild_spawning
- rescuing
- resize_confirming
- resize_finish
- resize_migrated
- resize_migrating
- resize_prep
- resize_reverting
- resuming
- scheduling
- spawning
- starting
- stopping
- suspending
- unpausing
- unrescuing
- updating_password

The namespace for this extended attribute is:

```
xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"
```

For information about server statuses, see [Server Status Values \[48\]](#).

3.4. Rescue Mode Extension

Rescue mode creates a new Cloud Server with the file system for the specified Cloud Server system mounted to fix file system and configuration errors. See also [Section 2.3.7, "Rescue Server" \[90\]](#) and [Section 2.3.8, "Unrescue Server" \[92\]](#).

When you place a server in rescue mode, the following events occur:

1. The server is shut down.
2. A new server is created, as follows:
 - The new server is based on the image from which the original server was created, with a random password. This password is returned to you in a response to issuing the rescue mode API call.
 - The new server has a secondary disk that is the file system of the original server. Use the clean rescue server to fix problems on the original server.

To place a server in rescue mode, issue the request body in a **POST** request to `/servers/id/action`. When you put a server into rescue mode, you cannot use it until its status goes from `ACTIVE` to `RESCUE`. This does not happen immediately.

After you resolve any problems and reboot the rescued server, you can unrescue the server, which restores the repaired image to running state with its original password. The unrescue operation does not return a response body. The HTTP status code is 202 (Accepted) for a successful unrescue.

The following JSON request and response examples show how to place a server in rescue mode:

Example 3.7. Place a Server in Rescue Mode: JSON Request

```
{  
  "rescue" : null  
}
```

After you place a server in rescue mode, the following response is returned:

Example 3.8. Place a Server in Rescue Mode: JSON Response

```
{  
  "adminPass" : "Qy7gCeHeYaT7"  
}
```

The following XML request and response examples show how to place a server in rescue mode:

Example 3.9. Place a Server in Rescue Mode: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<rescue
  xmlns="http://docs.openstack.org/compute/ext/rescue/api/v1.1"/>
```

After you place a server in rescue mode, the following response is returned:

Example 3.10. Place a Server in Rescue Mode: XML Response

```
<adminPass>eBHcCgGBVj6Z</adminPass>
```

The following XML example shows how to unrescue a server that is in rescue mode:

Example 3.11. Unrescue a Server in Rescue Mode: XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<unrescue
  xmlns="http://docs.rackspacecloud.com/servers/api/v1.1"/>
```

3.5. Used Limits Extension

Verb	URI	Description
GET	v2/tenant_id/limits	Extends limits to include information about the absolute limits that are currently used.

Returns absolute and rate limit information, including information about the currently used absolute limits.

Absolute and rate limits are part of the core API. See [Section 1.8, "Limits" \[23\]](#). The used limits extension adds attributes to the response body that show how much capacity is currently being used.

The following table describes the URI parameters:

Parameter	Description
tenant_id	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not require a request body.

This operation returns a response body.

In the following response example, the `totalRAMUsed` value is an extended attribute.

The following example response shows a JSON response:

Example 3.12. Used Limits: JSON Response

```
{
  "limits": {
    "rate": [
      {
        "uri": "*",
        "regex": ".*",
        "limit": [
          {
```

```
        "value": 10,
        "verb": "POST",
        "remaining": 2,
        "unit": "MINUTE",
        "next-available": "2011-12-15T22:42:45Z"
      },
      {
        "value": 10,
        "verb": "PUT",
        "remaining": 2,
        "unit": "MINUTE",
        "next-available": "2011-12-15T22:42:45Z"
      },
      {
        "value": 100,
        "verb": "DELETE",
        "remaining": 100,
        "unit": "MINUTE",
        "next-available": "2011-12-15T22:42:45Z"
      }
    ]
  },
  {
    "uri": "*changes-since*",
    "regex": "changes-since",
    "limit": [
      {
        "value": 3,
        "verb": "GET",
        "remaining": 3,
        "unit": "MINUTE",
        "next-available": "2011-12-15T22:42:45Z"
      }
    ]
  },
  {
    "uri": "*/servers",
    "regex": "^/servers",
    "limit": [
      {
        "verb": "POST",
        "value": 25,
        "remaining": 24,
        "unit": "DAY",
        "next-available": "2011-12-15T22:42:45Z"
      }
    ]
  }
],
"absolute": {
  "maxTotalRAMSize": 51200,
  "totalRAMUsed": 1024,
  "maxServerMeta": 5,
  "maxImageMeta": 5,
  "maxPersonality": 5,
  "maxPersonalitySize": 10240
}
```

4. Document Change History

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

Revision Date	Summary of Changes
Feb 4, 2013	<ul style="list-style-type: none">Updated the change administrator password API operation to define administrator password as the root password for the server.
Jan 31, 2013	<ul style="list-style-type: none">Minor updates.
Jan 23, 2013	<ul style="list-style-type: none">Updated the links and references topic to show the personality attribute in the JSON examples.
Jan 21, 2013	<ul style="list-style-type: none">Updated the list images API operation to correct the <code>SERVER</code> filter to <code>SNAPSHOT</code>.Abbreviated the list images example responses.
Jan 18, 2013	<ul style="list-style-type: none">Updated the <code>diskConfig</code> extended attribute to <code>OS-DCF:diskConfig</code> throughout the guide.Removed normal response code 300 from the get server details API operation.
Jan 16, 2013	<ul style="list-style-type: none">Updated the Section 2.4, "Volume Attachment Actions" [96] to correct the JSON and XML requests and responses.
Jan 8, 2013	<ul style="list-style-type: none">Updated Section 2.1.3, "Get Server Details" [63] to add explanations to the <code>accessIPv4</code> and <code>accessIPv6</code> fields.
Dec 4, 2012	<ul style="list-style-type: none">Corrected the <code>NOVA_RACK_AUTH</code> environment variable to <code>NOVA_RAX_AUTH</code>.Corrected <code>volumeId</code> attribute in the attributes table in the attach volume to server API call.
Nov 7, 2012	<ul style="list-style-type: none">Added <code>power_state</code> value of 4 to the description of OS-EXT-STS extended attribute description.
Nov 2, 2012	<ul style="list-style-type: none">Corrected formatting errors in code samples.
Oct 30, 2012	<ul style="list-style-type: none">Updated the limits section to describe the <code>maxTotalPrivateNetworks</code> absolute limit.Phased-release launch of Cloud Networks.Added new Cloud Networks books to the resources topic.
Oct 16, 2012	<ul style="list-style-type: none">Updated the list images examples.
Oct 16, 2012	<ul style="list-style-type: none">Updated the list servers API to include example cURL requests.Updated Section 2.3.7, "Rescue Server" [90] to correct the normal response code to 200.
Oct 5, 2012	<ul style="list-style-type: none">Updated Section 2.1.2, "Create Server" [55] to add new <code>networks</code> element with <code>uuid</code> attribute for creating private isolated networks.
Sep 11, 2012	<ul style="list-style-type: none">Updated Section 1.8, "Limits" [23] with correct absolute and rate limits and JSON and XML examples.
Sep 10, 2012	<ul style="list-style-type: none">Updated Section 1.8, "Limits" [23] with correct absolute and rate limits and JSON and XML examples.Clarified that you must specify a flavor ID and not a full flavor reference in the Section 2.1.1, "List Servers" [46], Section 2.1.2, "Create Server" [55], Section 2.1.3, "Get Server Details" [63], and Section 2.3.3, "Rebuild Server" [81] requests.Clarified that you must specify an image ID and not a full image reference in the Section 2.1.1, "List Servers" [46], Section 2.1.2, "Create Server" [55], Section 2.1.3, "Get Server Details" [63], and Section 2.3.3, "Rebuild Server" [81] requests.Corrected the <code>OS-DCF:diskConfig</code> request attribute to <code>OS-DCF:diskConfig</code> in Section 2.1.2, "Create Server" [55], Section 2.3.3, "Rebuild Server" [81], and Section 2.3.4, "Resize Server" [86].
Sep 7, 2012	<ul style="list-style-type: none">Updated the Section 1.7, "Efficient Polling with the <code>Changes-Since</code> Parameter" [22] examples to show correct <code>retryAt</code> syntax.Updated the list images JSON and XML examples to reflect the latest images.Updated the XML request example in Section 2.3.7, "Rescue Server" [90] to add missing closing backslash.Updated the request examples in Section 2.3.8, "Unrescue Server" [92] to show correct <code>unrescue</code> action.

Revision Date	Summary of Changes
Aug 28, 2012	<ul style="list-style-type: none"> Updated the Section 1.7, "Efficient Polling with the <i>Changes-Since</i> Parameter" [22] example to include the seconds value.
Aug 20, 2012	<ul style="list-style-type: none"> Updated examples for the list servers and get server details calls with the <code>OS-EXT-STS:vm_state</code>, <code>OS-EXT-STS:task_state</code>, and <code>OS-EXT-STS:power_state</code> extended attributes. Added the <code>OS-EXT-STS</code> extension description. See Section 3.3, "Extended Status Extension" [144].
Aug 15, 2012	<ul style="list-style-type: none"> Next generation Cloud Servers UK launch date. Added authentication endpoint for UK accounts to the documentation and removed the preview release notes for authentication updates.
Aug 1, 2012	<ul style="list-style-type: none"> Next generation Cloud Servers US launch date. Updated Rackspace contact information in the preface. Updated authentication topic to remove redundancy and add regional support. Updated Chapter 1, "General API Information" [1] to update examples for API v2. Updated Section 1.1, "Cloud Servers Concepts" [1] to revise definitions. Updated examples in Chapter 2, "API Operations" [45] to reflect API v2 changes. Updated Chapter 3, "Rackspace Extensions" [140] to update extensions descriptions.
Jul 16, 2012	<ul style="list-style-type: none"> Updated request and response examples in the create server API operation.
Jul 11, 2012	<ul style="list-style-type: none"> Removed links for error codes in list servers API operations.
Jul 9, 2012	<ul style="list-style-type: none"> Updated authentication section with pointer to preview release notes for UK authentication. Updated list servers API operation to include namespaces for the <code>OS-DCF:diskConfig</code> and <code>bandwidth</code> extended attributes.
Jun 27, 2012	<ul style="list-style-type: none"> Added a note about changing the name or host name of a server to the create server and update server API operations. Removed backup schedules extension. Added status <code>MIGRATING</code> to the list servers operation.
Jun 21, 2012	<ul style="list-style-type: none"> Updated the rate limits.
Jun 13, 2012	<ul style="list-style-type: none"> Added preliminary HTTP status codes topic, with explanations of status codes.
Jun 5, 2012	<ul style="list-style-type: none"> Updated references to legacy Cloud Servers to first generation Cloud Servers.
May 9, 2012	<ul style="list-style-type: none"> Updated Section 2.1.2.3, "Server Personality" [60] with guidelines about file injection. Updated Section 2.1.1, "List Servers" [46], Section 2.1.3, "Get Server Details" [63], and Section 3.1, "Bandwidth Stats Extension" [140] with bandwidth extension information.
May 1, 2012	<ul style="list-style-type: none"> Limited availability release date. Added volume attachment API operations.
Feb 29, 2012	<ul style="list-style-type: none"> Private beta release date. First edition of this document.

5. Resources

Next generation Cloud Servers v2	Cloud Networks v2	First generation Cloud Servers v1.0
<ul style="list-style-type: none">• Next Generation Cloud Servers Release Notes• Next Generation Cloud Servers Getting Started• Next Generation Cloud Servers Developer Guide• Cloud Identity Client Developer Guide v2.0	<ul style="list-style-type: none">• Cloud Networks Release Notes• Cloud Networks Getting Started• Cloud Networks Developer Guide	<ul style="list-style-type: none">• First Generation Cloud Servers Developer Guide• Cloud Servers™ API Schema Types• Cloud Servers - Frequently Asked Questions• Cloud Authentication Client Developer Guide v1.1

For Cloud Servers and Cloud Networks service resources, see the [Rackspace Cloud](#) site, which provides related documents and links to Rackspace support channels including [knowledge base articles](#), phone, chat, and tickets.

For product updates and announcements through Twitter, see <http://twitter.com/rackspace>.

Glossary

API

Application Programming Interface. Enables programmers to create application services by using an open application.

authentication

To make API calls, you authenticate by issuing a request to the Rackspace Cloud Identity Service. In response to valid credentials, the request returns an authentication token and the Cloud services catalog with the endpoints to request Cloud services. Each time that you make an API call, you include the endpoint for the Cloud service that you want to access and the authentication token.

bandwidth

The amount of available data used by communication resources such as the Internet. Bandwidth refers to the amount of data that is used to download things or the amount of data available to download.

flavor

A resource configuration for a server. Each flavor is a unique combination of disk, memory, vCPUs, and network bandwidth.

HTTP

HyperText Transfer Protocol. The protocol that tells browsers where to find information.

image

A collection of files for a specific operating system (OS) that you use to create or rebuild a server. Rackspace provides pre-built images. You can also create custom images from servers that you have launched. Custom images can be used for data backups or as "gold" images for additional servers.

REST

REpresentational State Transfer. A style of architecture for hypermedia systems that is used for the World Wide web.

RESTful

A kind of web service API that uses REST.

server

A computer that provides explicit services to the client software running on that system. A server is a virtual machine (VM) instance in the Cloud Servers environment. To create a server, you must specify a name, flavor reference, and image reference.