## 1 Documentation

1.1 Main ................................................................. 3  
1.1.1 Getting Started .............................................. 3  
1.1.2 Changelog ..................................................... 4  
1.1.3 Supported Providers ........................................ 39  
1.1.4 Compute ....................................................... 47  
1.1.5 Object Storage ............................................... 48  
1.1.6 Load Balancer ............................................... 49  
1.1.7 DNS ........................................................... 49  
1.1.8 Upgrade Notes ............................................... 50  
1.1.9 Troubleshooting ............................................. 60  
1.1.10 Frequently Asked Questions (FAQ) ........................ 61  
1.1.11 Registering a third party driver ......................... 62  
1.1.12 SSL Certificate Validation ............................... 63  
1.1.13 Using Libcloud in multi-threaded and async environments ............................................... 63  
1.1.14 Working with the object oriented APIs ................. 64  
1.2 Developer Information .......................................... 67  
1.2.1 Developer Information ....................................... 67  
1.2.2 Development .................................................. 68  
1.3 Committer Guide ................................................ 71  
1.3.1 Committer Guide ............................................ 71  
1.4 Other ............................................................ 77  
1.4.1 Security ....................................................... 77
Apache Libcloud is a Python library which hides differences between different cloud provider APIs and allows you to manage different cloud resources through a unified and easy to use API.

Resource you can manage with Libcloud are divided in the following categories:

- **Cloud Servers and Block Storage** - services such as Amazon EC2 and RackSpace CloudServers
- **Cloud Object Storage and CDN** - services such as Amazon S3 and Rackspace CloudFiles
- **Load Balancers as a Service** - services such as Amazon Elastic Load Balancer and GoGrid LoadBalancers
- **DNS as a Service** - services such as Amazon Route 53 and Zerigo

Fig. 1: A subset of supported providers in Libcloud.
1.1 Main

1.1.1 Getting Started

Installation (stable version)

Libcloud is available on PyPi. You can install latest stable version using pip:

```shell
pip install apache-libcloud
```

Installation (development version)

You can install latest development version from our Git repository:

```shell
pip install -e https://git-wip-us.apache.org/repos/asf/libcloud.git@trunk#egg=apache-libcloud
```

Upgrading

If you used pip to install the library you can also use it to upgrade it:

```shell
pip install --upgrade apache-libcloud
```

Using it

This section describes a standard work-flow which you follow when working with any of the Libcloud drivers.

1. Obtain reference to the provider driver

   ```python
   from pprint import pprint
   from libcloud.compute.types import Provider
   from libcloud.compute.providers import get_driver
   
   cls = get_driver(Provider.RACKSPACE)
   ```

2. Instantiate the driver with your provider credentials
3. Start using the driver

```python
pprint(driver.list_sizes())
pprint(driver.list_nodes())
```

4. Putting it all together

```python
from pprint import pprint
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver
cls = get_driver(Provider.RACKSPACE)
driver = cls('my username', 'my api key')
pprint(driver.list_sizes())
pprint(driver.list_nodes())
```

You can find more examples with common patterns which can help you get started on the [Compute Examples page](http://s.apache.org/lc0140un).

**Where to go from here?**

### 1.1.2 Changelog

```---
code: utf-8
---
```

Changes with Apache Libcloud 0.14.0-beta3

***) General

- If the file exists, read pricing data from `~/.libcloud/pricing.json` by default. If the file doesn't exist, fall back to the old behavior and use pricing data which is bundled with the release.
  [Tomaz Muraus]

- Add `libcloud.pricing.download_pricing_file` function for downloading and updating the pricing file.
  [Tomaz Muraus]

- Fix `libcloud.utils.py3.urlquote` so it works with unicode strings under Python 2. (LIBCLOUD-429)
  [Michael Farrell]

***) Compute

- Refactor Rackspace driver classes and make them easier to use. Now there are two Rackspace provider constants - `Provider.RACKSPACE` which represents new next-gen OpenStack servers and `Provider.RACKSPACE_FIRST_GEN` which represents old first-gen cloud servers.

  Note: This change is backward incompatible. For more information on those changes and how to update your code, please visit "Upgrade Notes" documentation page - [http://s.apache.org/lc0140un](http://s.apache.org/lc0140un)
- Deprecate the following EC2 provider constants: EC2_US_EAST, EC2_EU, EC2_EU_WEST, EC2_AP_SOUTHEAST, EC2_AP_NORTHEAST, EC2_US_WEST_OREGON, EC2_SA_EAST, EC2_SA_EAST and replace it with a new EC2 constant. Driver referenced by this new constant now takes a "region" argument which dictates to which region to connect.

Note: Deprecated constants will continue to work until the next major release. For more information on those changes and how to update your code, please visit "Upgrade Notes" documentation page - http://s.apache.org/lc0140un

- Add support for volume related functions to OpenNebula driver. (LIBCLOUD-354) [Emanuele Rocca]

- Add methods for managing storage volumes to the OpenStack driver. (LIBCLOUD-353) [Bernard Kerckenaere]

- Add new driver for Google Compute Engine (LIBCLOUD-266, LIBCLOUD-386) [Rick Wright]

- Fix create_node "features" metadata and update affected drivers. (LIBCLOUD-367) [John Carr]

- Update EC2 driver to accept the auth kwarg (it will accept NodeAuthSSH objects and automatically import a public key that is not already uploaded to the EC2 keyring). (Follow on from LIBCLOUD-367). [John Carr]

- Unify extension argument names for assigning a node to security groups in EC2 and OpenStack driver. Argument in the EC2 driver has been renamed from ex_securitygroup to ex_security_groups. For backward compatibility reasons, old argument will continue to work until the next major release. (LIBCLOUD-375) [Tomaz Muraus]

- Add ex_import_keypair_from_string and ex_import_keypair method to the CloudStack driver. (LIBCLOUD-380) [Sebastien Goasguen]

- Add support for managing floating IP addresses to the OpenStack driver. (LIBCLOUD-382) [Ivan Kusalic]

- Add extension methods for handling port forwarding to the CloudStack driver, rename CloudStackForwardingRule class to CloudStackIPForwardingRule. (LIBCLOUD-383) [sebastien goasguen]

- Hook up deploy_node functionality in the CloudStack driver and unify extension arguments for handling security groups. (LIBCLOUD-388) [sebastien goasguen]
- Allow user to pass "args" argument to the ScriptDeployment and ScriptFileDeployment class. This argument tells which command line arguments get passed to the ScriptDeployment script. (LIBCLOUD-394)

Note: This change is backward incompatible. For more information on how this affects your code and how to update it, visit "Upgrade Notes" documentation page - http://s.apache.org/lc0140un
[Tomaz Muraus]

- Allow user to specify IAM profile to use when creating an EC2 node. (LIBCLOUD-403)
[Xavier Barbosa]

- Add support for keypair management to the OpenStack driver. (LIBCLOUD-392)
[L. Schaub]

- Allow user to specify disk partitioning mode using ex_disk_config argument in the OpenStack based drivers. (LIBCLOUD-402)
[Brian Curtin]

- Add new driver for NephoScale provider (http://nephoscale.com/). (LIBCLOUD-404)
[Markos Gogoulos]

- Update network related extension methods so they work correctly with both, OpenStack and Rackspace driver. (LIBCLOUD-368)
[Tomaz Muraus]

- Add tests for networking functionality in the OpenStack and Rackspace driver.
[Tomaz Muraus]

- Allow user to pass all supported extension arguments to ex_rebuild_server method in the OpenStack driver. (LIBCLOUD-408)
[Dave King]

- Add pricing information for Rackspace Cloud Sydney region.
[Tomaz Muraus]

- Update EC2 instance type map and pricing data. High Storage instances are now also available in Sydney and Singapore region.
[Tomaz Muraus]

- Add new methods for managing storage volumes and snapshots to the EC2 driver (list_volumes, list_snapshots, destroy_volume_snapshot, create_volume_snapshot) (LIBCLOUD-409)
[Oleg Suharev]

- Add the following new extension methods to EC2 driver: ex_destroy_image, ex_modify_instance_attributes, ex_delete_keypair. (LIBCLOUD-409)
[Oleg Suharev]

- Allow user to specify a port range when creating a port forwarding rule. (LIBCLOUD-409)
[Oleg Suharev]

- Align Joyent driver with other drivers and deprecate "location" argument
in the driver constructor in favor of "region" argument.

Note: Deprecated argument will continue to work until the next major release.
[Tomaz Muraus]

- Deprecate the following ElasticHosts provider constants: ELASTICHOSTS_UK1, ELASTICHOSTS_UK2, ELASTICHOSTS_US1, ELASTICHOSTS_US2, ELASTICHOSTS_US3, ELASTICHOSTS_CA1, ELASTICHOSTS_AU1, ELASTICHOSTS_CN1 and replace it with a new ELASTICHOSTS constant. Driver referenced by this new constant now takes a "region" argument which dictates to which region to connect.

Note: Deprecated constants will continue to work until the next major release. For more information on those changes and how to update your code, please visit "Upgrade Notes" documentation page - http://s.apache.org/lc0140un (LIBCLOUD-383) [Michael Bennett, Tomaz Muraus]

- Add log statements to our ParamikoSSHClient wrapper. This should make debugging deployment issues easier. (LIBCLOUD-414) [Tomaz Muraus]

- Add new "NodeState.STOPPED" node state. Update HostVirual and EC2 driver to also recognize this new state. (LIBCLOUD-296) [Jayy Vis]

- Add new Hong Kong endpoint to Rackspace driver. [Brian Curtin]

- Fix ex_delete_keypair method in the EC2 driver. (LIBCLOUD-415) [Oleg Suharev]

- Add the following new extension methods for elastic IP management to the EC2 driver: ex_allocate_address, ex_disassociate_address, ex_release_address. (LIBCLOUD-417) [Patrick Armstrong]

- For consistency and accuracy, rename "ex_associate_addresses" method in the EC2 driver to "ex_associate_address_with_node".

Note: Old method will continue to work until the next major release. [Tomaz Muraus]

- Add new driver for CloudFrames (http://www.cloudfounders.com/CloudFrames) provider. (LIBCLOUD-358) [Bernard Kerckenaere]

- Update default kernel versions which are used when creating a Linode server.

Old default kernel versions:
- x86 - 2.6.18.8-x86_64-linode1
- x86_64 - 2.6.39.1-linode34

New default kernel versions:
- x86 - 3.9.3-x86-1inode52
  - x86_64 - 3.9.3-x86_64-1inode33

(LIBCLOUD-424)
[Tomaz Muraus, Jon Chen]

- Disable cache busting functionality in the OpenStack and Rackspace next-gen driver and enable it only for Rackspace first-gen driver.
  [Tomaz Muraus]

- Update Google Compute Engine driver to v1beta16.
  [Rick Wright]

- Modify auth_url variable in the OpenStack drivers so it works more like users would expect it to.
  Previously path specified in the auth_url was ignored and only protocol, hostname and port were used. Now user can provide a full url for the auth_url variable and the path provided in the url is also used.  
  [DaeMyung Kang, Tomaz Muraus]

- Allow user to associate arbitrary key/value pairs with a node by passing "ex_metadata" argument (dictionary) to create_node method in the EC2 driver.
  Those values are associated with a node using tags functionality.  
  (LIBCLOUD-395)  
  [Ivan Kusalic]

- Add "ex_get_metadata" method to EC2 and OpenStack driver. This method reads metadata dictionary from the Node object.  
  (LIBCLOUD-395)  
  [Ivan Kusalic]

- Multiple improvements in the Softlayer driver:
  - Map "INITIATING" node state to NodeState.PENDING
  - If node is launching remap "halted" state to "pending"
  - Add more node sizes
  - Add ex_stop_node and ex_start_node method
  - Update tests response fixtures
  (LIBCLOUD-416)  
  [Markos Gogoulos]

- Modify list_sizes method in the KT UCoud driver to work, even if the item doesn't have 'diskofferingid' attribute.  
  (LIBCLOUD-435)  
  [DaeMyung Kang]

- Add new c3 instance types to the EC2 driver.
  [Tomaz Muraus]

- Fix an issue with the ex_list_keypairs and ex_list_security_groups method in the CloudStack driver which caused an exception to be thrown if the API returned no keypairs / security groups.  
  (LIBCLOUD-438)  
  [Carlos Reategui, Tomaz Muraus]

- Fix a bug in the OpenStack based drivers with not correctly checking if the auth token has expired before re-using it.  
  (LIBCLOUD-428)  
  
  Reported by Michael Farrell.
*) Storage

- Deprecate CLOUDFILES_US and CLOUDFILES_UK provider constant and replace it with a new CLOUDFILES constant. Driver referenced by this new constant takes a "region" keyword argument which can be one of 'ord', 'dfw', 'iad', 'syd', 'lon'.

  Note: Deprecated constants will continue to work until the next major release. For more information on this change, please visit "Upgrade Notes" documentation section - http://s.apache.org/lc0140un

  [Tomaz Muraus]

- Allow users to filter objects starting with a prefix by passing ex_prefix argument to the list_container_objects method in the S3, Google Storage and CloudFiles driver. (LIBCLOUD-369)

  [Stefan Friesel]

- Fix an issue with mutating connectionCls.host attribute in the Azure driver. This bug prevented user from having multiple Azure drivers with different keys instantiated at the same time. (LIBCLOUD-399)

  [Olivier Grisel]

- Add a new driver for KT UCloud based on the OpenStack Swift driver. (LIBCLOUD-431).

  [DaeMyung Kang]

*) Load Balancer

- Deprecate RACKSPACE_US and RACKSPACE_UK provider constant and replace it with a new RACKSPACE constant. Driver referenced by this new constant takes a "region" keyword argument which can be one of the following: 'ord', 'dfw', 'iad', 'syd', 'lon'.

  Note: Deprecated constants will continue to work until the next major release. For more information on this change, please visit "Upgrade Notes" documentation section - http://s.apache.org/lc0140un

  [Tomaz Muraus]

- Add new driver for Google Compute Engine (LIBCLOUD-386)

  [Rick Wright]

- Add new Hong Kong endpoint to Rackspace driver.

  [Brian Curtin]

*) DNS

- Deprecate RACKSPACE_US and RACKSPACE_UK provider constant and replace it with a new RACKSPACE constant. Driver referenced by this new constant takes a "region" keyword argument which can be one of the following: 'us', 'uk'.

  Note: Deprecated constants will continue to work until the next major release. For more information on this change, please visit "Upgrade Notes"
Use string instead of integer for RecordType ENUM value.

Note: If you directly use an integer instead of RecordType ENUM class you need to update your code to use the RecordType ENUM otherwise the code won't work. For more information on how to do that, see "Upgrade Notes" documentation section - http://s.apache.org/lc0140un

Add "export_zone_to_bind_format" and export_zone_to_bind_zone_file method which allows users to export Libcloud Zone to BIND zone format. (LIBCLOUD-398)

Update issue with inexistent zone / record handling in the get_zone and get_record method in the Linode driver. Those issues were related to changes in the Linode API. (LIBCLOUD-425)

Changes with Apache Libcloud 0.13.2

*) General

- Don't sent Content-Length: 0 header with POST and PUT request if "raw" mode is used. This fixes a regression which could cause broken behavior in some storage driver when uploading a file from disk. (LIBCLOUD-396)

*) Compute

- Added Ubuntu Linux 12.04 image to ElasticHost driver image list. (LIBCLOUD-364)

- Update ElasticHosts driver to store drive UUID in the node 'extra' field. (LIBCLOUD-357)

*) Storage

- Store last_modified timestamp in the Object extra dictionary in the S3 driver. (LIBCLOUD-373)

*) Load Balancer

- Expose CloudStack driver directly through the Provider.CLOUDSTACK constant.

*) DNS

- Modify Zerigo driver to include record TTL in the record 'extra' attribute if a record has a TTL set.
- Modify values in the Record 'extra' dictionary attribute in the Zerigo DNS driver to be set to None instead of an empty string ('') if a value for the provided key is not set. [Tomaz Muraus]

Changes with Apache Libcloud 0.13.1

*) General

- Fix a regression introduced in 0.13.0 and make sure to include Content-Length 0 with PUT and POST requests. (LIBCLOUD-362, LIBCLOUD-390) [Tomaz Muraus]

*) Compute

- Fix a bug in the ElasticHosts driver and check for right HTTP status code when determining drive imaging success. (LIBCLOUD-363) [Bob Thompson]

- Update Opsource driver to include node public ip address (if available). (LIBCLOUD-384) [Michael Bennett]

*) Storage

- Fix a regression with calling encode_container_name instead of encode_object_name on object name in get_object method. Reported by Ben Meng (LIBCLOUD-366) [Tomaz Muraus]

- Ensure that AWS S3 multipart upload works for small iterators. (LIBCLOUD-378) [Mahendra M]

Changes with Apache Libcloud 0.13.0:

*) General

- Add homebrew curl-ca-bundle path to CA_CERTS_PATH. This will make Libcloud use homebrew curl ca bundle file (if available) for server certificate validation. (LIBCLOUD-324) [Robert Chiniquy]

- Modify OpenStackAuthConnection and change auth_token_expires attribute to be a datetime object instead of a string. [Tomaz Muraus]

- Modify OpenStackAuthConnection to support re-using of the existing auth token if it's still valid instead of re-authenticating on every authenticate() call. [Tomaz Muraus]

- Modify base Connection class to not send Content-Length header if body is not provided. [Tomaz Muraus]

- Add the new error class ProviderError and modify InvalidCredsError to inherit from it. (LIBCLOUD-331)
*) Misc

- Add unittest2 library dependency for tests and update some tests to use it.  
  [Tomaz Muraus]

*) Compute

- Fix destroy_node method in the experimental libvirt driver.  
  [Aymen Fitati]

- Add ex_start_node method to the Joyent driver. (LIBCLOUD-319)  
  [rszabo50]

- Fix Python 3 compatibility issue in the ScriptFileDeployment class.  
  (LIBCLOUD-321)  
  [Arfrever Frehtes Taifersar Araheis]

- Add ex_set_metadata_entry and ex_get_metadata method to the VCloud driver.  
  (LIBCLOUD-318)  
  [Michel Samia]

- Various improvements and bug-fixes in the VCloud driver. (LIBCLOUD-323)  
  [Michel Samia]

- Various bug fixes and improvements in the HostVirtual driver.  
  (LIBCLOUD-249)  
  [Dinesh Bhoopathy]

- Modify list_sizes method in the OpenStack driver to include  
  OpenStackNodeSize object which includes 'vcpus' attribute which holds a number of virtual CPUs for this size. (LIBCLOUD-325)  
  [Carlo]

- For consistency rename "ex_describe_keypairs" method in the EC2 driver to "ex_describe_keypair".  
  [Tomaz Muraus]

- Modify "ex_describe_keypair" method to return key fingerprint in the return value. (LIBCLOUD-326)  
  [Andre Merzky, Tomaz Muraus]

- Populate private_ips attribute in the CloudStack drive when returning a Node object from the create_node method. (LIBCLOUD-329)  
  [Sebastien Goasguen, Tomaz Muraus]

- Allow user to pass extra arguments via "extra_args" argument which are then passed to the "deployVirtualMachine" call in the CloudStack driver create_node method. (LIBCLOUD-330)  
  [Sebastien Goasguen, Tomaz Muraus]

- Update Gandi driver to handle new billing model. (LIBCLOUD-317)  
  [Aymeric Barantal]

- Fix a bug in the Linode driver and remove extra newline which is added when generating a random root password in create_node. (LIBCLOUD-334)
- Add extension methods for managing keypairs to the CloudStack driver. (LIBCLOUD-333)
  [sebastien goasguen]
- Add extension methods for managing security groups to the CloudStack driver. (LIBCLOUD-332)
  [sebastien goasguen]
- Add extension methods for starting and stopping the node to the CloudStack driver. (LIBCLOUD-338)
  [sebastien goasguen]
- Fix old _wait_until_running method. (LIBCLOUD-339)
  [Bob Thompson]
- Allow user to override default async task completion timeout by specifying ex_clone_timeout argument. (LIBCLOUD-340)
  [Michal Galet]
- Fix a bug in the GoGrid driver get_uuid method. (LIBCLOUD-341)
  [Bob Thompson]
- Fix a bug with deploy_node not respecting 'timeout' kwarg.
  [Kevin Carter]
- Modify create_node method in CloudStack driver to return an instance of CloudStackNode and add a new "expunging" node state. (LIBCLOUD-345)
  [sebastien goasguen]
- Update API endpoint hostnames in the ElasticHost driver and use hostnames which return a valid SSL certificate. (LIBCLOUD-346)
  [Bob Thompson]
- Add ex_list_networks method and missing tests for list_templates to the CloudStack driver. (LIBCLOUD-349)
  [Philipp Strube]
- Correctly throw InvalidCredsError if user passes invalid credentials to the DigitalOcean driver.
  [Tomaz muraus]

*) Storage

- Fix an issue with double encoding the container name in the CloudFiles driver upload_object method.
  Also properly encode container and object name used in the HTTP request in the get_container and get_object method. (LIBCLOUD-328)
  [Tomaz Muraus]

*) Load Balancer

- Add ex_list_current_usage method to the Rackspace driver.

Changes with Apache Libcloud 0.12.4:
- Fix a regression in Softlayer driver caused by the xmlrpclib changes. (LIBCLOUD-310)
  [Jason Johnson]

- Allow user to pass alternate ssh usernames to deploy_node
  (ssh_alternate_usernames kwarg) which are used for authentication if the
default one doesn't work. (LIBCLOUD-309)
  [Chris Psaltis, Tomaz Muraus]

- Fix a bug in EC2 list_locations method - 'name' attribute didn't contain a
  the right value.
  [Tomaz Muraus]

- Add new ScriptFileDeployment deployment class which reads deploy script
  from a file.
  [Rudolf J Streif]

- Add support for API version 5.1 to the vCloud driver and accept any value
  which is a multiple of four for ex_vm_memory kwarg in create_node method.
  (LIBCLOUD-314)
  [Trevor Powell]

*) Storage

- Fix a regression with removed ex_force_service_region constructor kwarg in
  the CloudFiles driver. (LIBCLOUD-260)

Changes with Apache Libcloud 0.12.3:

*) General

- Fix Python 3.x related regressions. (LIBCLOUD-245)
  Reported by Arfrever Frehtes Taiferaar Arahesis.
  [Tomaz Muraus]

- Fix a regression introduced with recent xmlrpclib changes which broke all
  the Gandi.net drivers. (LIBCLOUD-288)
  Reported by Hutson Betts.
  [Tomaz Muraus]

- Improve deploy code to work correctly if the ssh user doesn't have access
to the /root directory.

  Previously the ScriptDeployment script was stored in /root folder by
default. Now it's stored in users home directory under filename
  ~/.libcloud_deployment_<random>.sh. (LIBCLOUD-302)
  Reported by rotem on #libcloud.
  [Tomaz Muraus]

*) Compute

- Improve public and private IP address handling in OpenStack 1.1 driver.
  Assume every IP address which doesn't have a label "public" or "internet"
is private. (LIBCLOUD-297)
  [Grischa Meyer, Tomaz Muraus]
  [LIBCLOUD-304]
  [Tomaz Muraus]

- Fix a regression in ParamikoSSHClient.run method which caused this method
to only work as expected if you passed an absolute or a relative path to
the script to it. (LIBCLOUD-278)
  [Tomaz Muraus]

*) DNS

- Allow user to specify 'priority' extra argument when creating a MX or SRV record.
  [Brian Jinwright, Tomaz Muraus]

Changes with Apache Libcloud 0.12.1:

*) General

- Deprecate LazyList method of iteration over large paginated collections
  and use a new, more efficient generator based approach which doesn't
  require the iterator to be pre-exhausted and buffering all of the values
  in memory.

  Existing list_* methods which previously used LazyList class are
  preserving the old behavior and new iterate_* methods which use a new
  generator based approach have been added. (LIBCLOUD-254)
  [Mahendra M]

- Replace old ENUM style provider constants and replace them with a string
  version.
  This change allows users to dynamically register new drivers using a new
  set_driver method. (LIBCLOUD-255)
  [Mahendra M]

- Allow user to explicitly specify which CA file is used for verifying
  the server certificate by setting 'SSL_CERT_FILE' environment variable.

  Note: When this variable is specified, the specified path is the only
  CA file which is used to verifying the server certificate. (LIBCLOUD-283)
  [Tomaz Muraus, Erinn Looney-Triggs]

- Add a common module (libcloud.common.xmlrpc) for handling XML-RPC
  requests using Libcloud http layer.

  Also refactor existing drivers which use xmlrpclib directly (VCL, Gandi,
  Softlayer) to use this module.

  This change allows drivers to support LIBCLOUD_DEBUG and SSL certificate
  validation functionality. Previously they have bypassed Libcloud http
  layer so this functionality was not available. (LIBCLOUD-288)
  [John Carr]

*) Compute

- Fix string interpolation bug in __repr__ methods in the IBM SCE driver.
  (LIBCLOUD-242)
  [Tomaz Muraus]
- Fix test failures which happened in Python 3.3 due to:
  - hash randomization
  - changes in xml.etree module
  - changes in xmlrpc module
  (LIBCLOUD-245)
  [Tomaz Muraus]

- Improvements and additions in vCloud driver:
  - Expose generic query method (ex_query)
  - Provide functionality to get and set control access for vApps. This way
    created vApps can be shared between users/groups or everyone.
  (LIBCLOUD-251)
  [Michal Galet]

- Update EC2 pricing data to reflect new, lower prices —
  [Tomaz Muraus]

- Update EC2 instance size to reflect new m3 instance types. Also refactor
  the code to make it easier to maintain.
  [Tomaz Muraus]

- Add a new driver for HostVirtual (http://www.vr.org) provider.
  (LIBCLOUD-249)
  [Dinesh Bhoopathy]

- Fix a bug where a numeric instead of a string value was used for the
  content-length header in VCloud driver. (LIBCLOUD-256)
  [Brian DeGeeter, Tomaz Muraus]

- Add a new driver for new Asia Pacific (Sydney) EC2 region.
  [Tomaz Muraus]

- Add support for managing security groups to the OpenStack driver. This
  patch adds the following extension methods:
  - ex_list_security_groups, ex_get_node_security_groups methods
  - ex_create_security_group, ex_delete_security_group
  - ex_create_security_group_rule, ex_delete_security_group_rule
  (LIBCLOUD-253)
  [L. Schaub]

- Modify ElasticStack driver class to pass 'vnc auto' instead of
  'vnc:ip auto' argument to the API when creating a server.
  It looks like 'vnc:ip' has been replaced with 'vnc'.
  [Rick Copeland, Tomaz Muraus]

- Add new EC2 instance type — High Storage Eight Extra Large Instance
  (hs1.8xlarge).
  [Tomaz Muraus]

- Map 'shutting-down' node state in EC2 driver to UNKNOWN. Previously
  it was mapped to TERMINATED. (LIBCLOUD-280)

  Note: This change is backward incompatible which means you need to update
  your code if you rely on the old behavior.
  [Tomaz Muraus, Marcin Kuzminski]

- Change _wait_until_running method so it supports waiting on multiple nodes
and make it public (wait_until_running). (LIBCLOUD-274)
[Nick Bailey]

- Add new EC2 instance type - High Memory Cluster Eight Extra Large.
  (cr1.8xlarge).
  [Tomaz Muraus]

  [Jaume Devesa]

- Allow user to pass 'ex_blockdevicemappings' kwarg to the EC2 driver
  'create_node' method. (LIBCLOUD-282)
  [Joe Miller, Tomaz Muraus]

- Improve error handling in the Brightbox driver.
  [Tomaz Muraus]

- Fix the ScriptDeployment step to work correctly if user provides a
  relative path for the script argument. (LIBCLOUD-278)
  [Jaume Devesa]

- Fix Softlayer driver and make sure all the code is up to date and works
  with the latest version of the actual Softlayer deployment (v3).
  (LIBCLOUD-287)
  [Kevin McDonald]

- Update EC2 driver, m3 instance types are now available in all the regions
  except Brazil.
  Also update pricing to reflect new (decreased) prices.
  [Tomaz Muraus]

- Minor improvements in the HostVirtual driver and add new ex_get_node and
  ex_build_node extension method. (LIBCLOUD-249)
  [Dinesh Bhoopathy]

- Add ex_destroy_image method to IBM SCE driver. (LIBCLOUD-291)
  [Perry Zou]

- Add the following new regions to the ElasticHosts driver: sjc-c, syd-v,
  hkg-e. (LIBCLOUD-293)
  [Tomaz Muraus]

- Fix create_node in OpenStack driver to work correctly if 'adminPass'
  attribute is not present in the response.
  [Gavin McCance, Tomaz Muraus]

- Allow users to filter images returned by the list_images method in the EC2
  driver by providing ex_image_ids argument. (LIBCLOUD-294)
  [Chris Psaltis, Joseph Hall]

- Add support for OpenNebula 3.8. (LIBCLOUD-295)
  [Guillaume ZITTA]

- Add missing 'deletd' -> terminated mapping to OpenStack driver.
  (LIBCLOUD-276)
  [Jayyy V]
- Fix create_node in OpenStack driver to work correctly if 'adminPass' attribute is not present in the response. (LIBCLOUD-292) [Gavin McCance, Tomaz Muraus]

*) Storage

- Add a new local storage driver. (LIBCLOUD-252, LIBCLOUD-258, LIBCLOUD-265, LIBCLOUD-273) [Mahendra M]

- Fix a bug which caused the connection to not be closed when using Python 2.6 and calling get_object on an object which doesn't exist in the S3 driver. (LIBCLOUD-257) [John Carr]

- Add a new generator based method for listing / iterating over the containers (iterate_containers). (LIBCLOUD-261) [Mahendra M]

- Add ex_purge_object_from_cdn method to the CloudFiles driver. (LIBCLOUD-267) [Tomaz Muraus]

- Support for multipart uploads and other improvements in the S3 driver so it can more easily be re-used with other implementations (e.g. Google Storage, etc.).

Also default to a multipart upload when using upload_object_via_stream. This methods is more efficient compared to old approach because it only requires buffering a single multipart chunk (5 MB) in memory. (LIBCLOUD-269) [Mahendra M]

- Add new driver for Windows Azure Storage with support for block and page blobs. (LIBCLOUD-80) [Mahendra M]

*) DNS

- Update 'if type' checks in the update_record methods to behave correctly if users passes in RecordType.A with a value of 0 - if type is not None. (LIBCLOUD-247) [Tomaz Muraus]

- New driver for HostVirtual provider (www.vr.org). (LIBCLOUD-249) [Dinesh Bhoopathy]

- Finish Amazon Route53 driver. (LIBCLOUD-132) [John Carr]

- Add new driver for Gandi provider (https://www.gandi.net). (LIBCLOUD-281) [John Carr]

*) Load-Balancer

- Add new driver for AWS Elastic Load Balancing service. (LIBCLOUD-169) [John Carr]
Changes with Apache Libcloud 0.11.4:

*) General

- Fix some of tests failures which happened in Python 3.3 due to randomized dictionary ordering. (LIBCLOUD-245)
  [Tomaz Muraus]

*) Compute

- Fix a bug where a numeric instead of a string value was used for the content-length header in VCloud driver. (LIBCLOUD-256)
  [Brian DeGeeter, Tomaz Muraus]

*) Storage

- Don't ignore ex_force_service_region argument in the CloudFiles driver. (LIBCLOUD-260)
  [Dan Di Spaltro]

- Fix a bug which caused the connection to not be closed when using Python 2.6 and calling get_object on an object which doesn't exist in the S3 driver. (LIBCLOUD-257)
  [John Carr]

*) DNS

- Update 'if type' checks in the update_record methods to behave correctly if users passes in RecordType.A with a value of 0 - if type is not None. (LIBCLOUD-247)
  [Tomaz Muraus]

Changes with Apache Libcloud 0.11.3:

*) Storage

- Include 'last_modified' and 'content_type' attribute in the Object 'extra' dictionary when retrieving object using get_object in the S3 driver. Also modify 'meta_data' dictionary to include all the headers prefixed with 'x-amz-meta-'.
  [Tomaz Muraus]

Changes with Apache Libcloud 0.11.2:

*) General

Fix a bug with the Libcloud SSL verification code. Code was too strict and didn't allow "-" character in the sub-domain when using a wildcard certificate.

Note: This is NOT a security vulnerability. (LIBCLOUD-244)
  [Tomaz Muraus]

*) Compute

- Add new Rackspace Nova driver for Chicago (ORD) location (LIBCLOUD-234)
  [Brian McDaniel]
- Add capacity information to Vdc objects and implement power operations.
  (LIBCLOUD-239)
  [Michal Galet]

- Allow user to pass 'timeout' argument to the 'deploy_node' method.
  [Tomaz Muraus]

- Add ex_list_security_groups, ex_authorize_security_group and
  ex_describe_all_keypairs methods to the EC2 driver. (LIBCLOUD-241,
  LIBCLOUD-243)
  [Nick Bailey]

- Add new methods for managing storage volumes and other extension methods
  to the IBM SCE driver. (LIBCLOUD-242)
  [Sengor Kusturica]

*) Storage

- Add the following new methods to the CloudFiles driver:
  ex_set_account_metadata_temp_url_key, ex_get_object_temp_url. (GITHUB-72)
  [Shawn Smith]

*) Load-balancer

- Add 'balancer' attribute to the Member instance. This attribute refers to
  the LoadBalancer instance this member belongs to.
  [Tomaz Muraus]

Changes with Apache Libcloud 0.11.1:

*) General

- Fix hostname validation in the SSL verification code (CVE-2012-3446).
  Reported by researchers from the University of Texas at Austin (Martin
  Georgiev, Suman Jana and Vitaly Shmatikov).

Changes with Apache Libcloud 0.11.0:

*) Compute

- Add a timeout of 10 seconds to OpenStackAuthConnection class.
  (LIBCLOUD-199)
  [Chris Gilmer]

- Add time.sleep(wait_period) to _ssh_client_connect to prevent busy loops
  when we immediately can't connect to a server. (LIBCLOUD-197)
  [Jay Doane]

- Fix a bug with Python 3 support in the following drivers
  - IBM SCE,
  - CloudStack
  - CloudSigma
  - OpenNebula
  - VpsNet
  - EC2
  - ElasticStack
  - vCloud
- Make CloudStack driver more robust and make it work if list_images() call returns no images. (LIBCLOUD-202) [Gabriel Reid]

- Add force_ipv4 argument to _wait_until_running and default it to True. This will make Libcloud ignore IPv6 addresses when using deploy_node. [Jay Doane, Tomaz Muraus]

- Include error text if a CloudStack async job returns an error code. (LIBCLOUD-207) [Gabriel Reid]

- Add extension methods for block storage volume management to the CloudStack driver. (LIBCLOUD-208) [Gabriel Reid]

- New driver for KT UCloud (http://home.ucloud.olleh.com/main.kt) based on the CloudStack driver. [DaeMyung Kang]

- Add a standard API and methods for managing storage volumes to the EC2 and CloudStack drivers. Base API consistent of the following methods: create_volume, destroy_volume, attach_volume, detach_volume. (LIBCLOUD-213) [Gabriel Reid]

- Change ex_describe_tags, ex_create_tags and ex_delete_tags methods signature in the EC2 driver. Argument is now called resource (previously it was called node). This methods work with both Node and StorageVolume objects. (LIBCLOUD-213) [Gabriel Reid, Tomaz Muraus]

- Add Rackspace Nova London driver. [Chris Gilmer]

- Fix a bug - If user doesn't pass in 'network_id' argument to the create_node method in the CloudStack driver, don't explicitly define it. (LIBCLOUD-219) [Bruno Mahé, Tomaz Muraus]

- Modify EC2 driver to also return cc2.8xlarge cluster compute instance in the eu-west-1 region. [Tomaz Muraus]

- Add 'auth_user_variable' to the OpenStackAuthConnection class. [Mark Everett]

- Fix a bug with repeated URLs in some requests the vCloud driver. (LIBCLOUD-222) [Michal Galet]

- New Gridspot driver with basic list and destroy functionality.
- Add methods for managing storage volumes to the Gandi driver. (LIBCLOUD-225) [Aymeric Barantal]

*) DNS

- Add support for GEO RecordType to Zerigo driver. (LIBCLOUD-203) [Gary Wilson]
- Fix a bug with Python 3 support in the following drivers (LIBCLOUD-204)
  - Zerigo [Tomaz Muraus]
- Add support for URL RecordType to Zerigo driver. (LIBCLOUD-209) [Bojan Mihelac]
- Properly handle record creation when user doesn't provider a record name and wants to create a record for the actual domain. Reported by Matt Perry (LIBCLOUD-224) [Tomaz Muraus]

*) Storage

- Fix a bug with Python 3 support in the following drivers
  - Atmos
  - Google Storage
  - Amazon S3 (LIBCLOUD-204) [Tomaz Muraus]
- Fix a bug in the CloudFiles driver which prevented it to work with accounts which use a non ORD endpoint. (LIBCLOUD-205) [Geoff Greer]
- Fix a bug in the enable_container_cdn method. (LIBCLOUD-206) [Geoff Greer]
- Allow user to specify container CDN TTL when calling container.enable_cdn() using ex_ttl keyword argument in the CloudFiles driver. [Tomaz Muraus]
- Add ex_enable_static_website and ex_set_error_page method to the CloudFiles driver. [Tomaz Muraus]
- Propagate kwargs passed to container.download_object() to driver.download_object(). (LIBCLOUD-227) [Benno Rice]
- Fix a bug with not escaping container and object name in the Atmos driver. [Russell Keith-Magee, Benno Rice]
- Fix upload_object_via_stream method in the Atmos driver. (LIBCLOUD-228) [Benno Rice]
- Fix a bug with uploading zero-sized files in the OpenStack Swift / CloudFiles driver.
  [Tomaz Muraus]

- Fix a bug with content_type and encoding of object and path names in the Atmos driver.
  [Russell Keith-Magee]

*) Other:
- Unify docstrings formatting in the compute drivers. (LIBCLOUD-229)
  [Ilgiz Islamgulov]

Changes with Apache Libcloud 0.10.1:

*) General:
- Add timeout attribute to base 'Connection' class and pass it to the connection class constructor if Python version is not 2.5.
  [Chris Gilmer]

*) Compute:
- Update IBM SBC driver so it works with IBM Smart Cloud Enterprise. (LIBCLOUD-195)
  [Sengor Kusturica]

- Add `ex_register_iso` method to the CloudStack driver. (LIBCLOUD-196)
  [Daemian Mack]

- Allow user to specify which IP to use when calling deploy_node.
  (defaults to 'public_ips'). Previously it only worked with public IP, now user can pass 'private_ips' as an argument and SSH client will try to connect to the node first private IP address.
  [Jay Doane]

- Fix CloudSigmaLvsNodeDriver connectionCls bug.
  [Jerry Chen]

- Add 'ex_keyname' argument to the create_node method in the OpenStack driver. (LIBCLOUD-177)
  [Jay Doane]

- Fix a problem in deploy_node - make it work with providers which don't instantly return created node in the list_node response. Also add __str__ and __repr__ method to DeploymentError so the error message is more useful. (LIBCLOUD-176)
  [Jouke Waleson, Tomaz Muraus]

- Add 'ssh_key' feature to Brightbox driver. This way it works with deploy_node. (LIBCLOUD-179)
  [Neil Wilson]

- Add Joyent compute driver.
  [Tomaz Muraus]

- Store auth token expire times on the connection class in the attribute called 'auth_tokenExpires'. (LIBCLOUD-178)
- Add new driver for VCL cloud
  (http://www.educause.edu/blog/hes8/CloudComputingandtheVirtualCloud/167931)
  (LIBCLOUD-180)
  [Chris Gilmer, Brad Morgan]

- Add ex_userdata argument to the OpenStack 1.1 driver. (LIBCLOUD-185)
  [Jay Doane]

- Modify VMware vCloud driver and implement new features
  for the vCloud version 1.5. (LIBCLOUD-183)
  [Michal Galet, Sengor Kusturica]

- Add ex_rescue and ex_unrescue method to OpenStack 1.1 driver. (LIBCLOUD-193)
  [Shawn Smith]

- Allow user to pass mode argument to SSHClient.put method and default it to
  'w'. (LIBCLOUD-188)
  [Jay Doane]

- Modify SSHKeyDeployment step to use append mode so it doesn't overwrite
  existing entries in .ssh/authorized_keys. (LIBCLOUD-187)
  [Jay Doane]

- Modify ParamikoSSHClient to connect to the SSH agent and automatically
  look for private keys in ~/.ssh if the 'auth' and 'ssh_key' argument
  is not specified when calling deploy_node. (LIBCLOUD-182)
  [Tomaz Muraus]

- Include 'password' in the node extra dictionary when calling deploy_node
  if the password auth is used.
  [Juan Carlos Moreno]

- Add FileDeployment class to libcloud.compute.deployment module. This can
  be used as a replacement for ex_files argument if the provider supports
  deployment functionality. (LIBCLOUD-190)
  [Jay Doane]

- Large object upload support for CloudFiles driver
- Add CLOUDFILES_SWIFT driver to connect to OpenStack Swift
  [Dmitry Russkikh, Roman Bogorodskiy]
*) Load-balancer

- Don't include 'body_regex' attribute in the Rackspace driver body if body_regex is None or empty string. (LIBCLOUD-186) [Bill Woodward]

- Don't split Load balancer IP addresses into public and private list. Include all the addresses in the 'virtualIps' variable in the extra dictionary (Rackspace driver). (LIBCLOUD-191) [Adam Pickeral]

Changes with Apache Libcloud 0.9.1:

*) General:

- Make parsing of the Auth API responses in the OpenStack drivers more flexible and extensible.

  Now, every connection class that inherits from the openstack base connection must implement get_endpoint(), who's job is to return the correct endpoint out of the service catalog.

  Note: The openstack.py base driver no longer works by default with Rackspace nova. The default endpoint parsed from the service catalog is the default compute endpoint for devstack. (LIBCLOUD-151) [Brad Morgan]

- Allow user to pass ex_tenant_name keyword argument to the OpenStack node driver class. This scopes all the endpoints returned by the Auth API endpoint to the provided tenant. (LIBCLOUD-172) [James E. Blair]

- Allow user to specify OpenStack service catalog parameters (service type, name and region). This way base OpenStack driver can be used with different providers without needing to subclass. (LIBCLOUD-173) [James E. Blair]

- Fix a bug with handling compressed responses in the Linode driver. (LIBCLOUD-158) [Ben Agricola]

*) Compute:

- Add new RackspaceNovaBeta and RackspaceNovaDfw driver based on the OpenStack. (LIBCLOUD-151) [Brad Morgan]

- Include 'created' and 'updated' attribute in the OpenStack 1.1 driver. (LIBCLOUD-155) [Chris Gilmer]

- Include 'minRam' and 'minDisk' attribute in the OpenStack 1.1 driver Node extra dictionary. (LIBCLOUD-163) [Chris Gilmer]

- Allow users to use a list of tuples for the query string parameters inside the OpenStack connection classes. This way same key can be specified multiple times (LIBCLOUD-153)
- Allow user to pass 'max_tries' keyword argument to deploy_node method.
  [Tomaz Muraus]

- Include original exception error message when re-throwing an exception
  inside _run_deployment_script method.
  [Tomaz Muraus]

- Add support for ElasticHosts new United States (Los Angeles) and Canada
  (Toronto) locations. (GITHUB-53)
  [Jaime Irurzun]

- Add serverId attribute to the NodeImage object extra dictionary in the
  OpenStack driver.
  [Mark Everett]

- Add new EC2 instance type - ml.medium.
  [Tomaz Muraus]

- Allow user to re-use auth tokens and pass 'ex_force_auth_token' keyword
  argument to the OpenStack driver constructor. (LIBCLOUD-164)
  [Dave King]

- Add new experimental libvirt driver.
  [Tomaz Muraus]

- Properly handle OpenStack providers which return public IP addresses under
  the 'internet' key in the addresses dictionary.
  [Tomaz Muraus]

- Update create_node in Linode driver and make it return a Node object
  instead of a list. Reported by Jouke Waleson. (LIBCLOUD-175)
  [Tomaz Muraus]

*) Storage:

- Don't lowercase special header names in the Amazon S3 storage driver.
  (LIBCLOUD-149)
  [Tomaz Muraus]

*) Load-balancer:

- Allow user to specify a condition and weight when adding a member in
  the Rackspace driver.
  [Adam Pickeral]

- Add an extension method (ex_balancer_attach_members) for attaching
  multiple members to a load balancer in the Rackspace driver.
  (LIBCLOUD-152)
  [Adam Pickeral]

- Add ex_create_balancer method to the Rackspace driver and allow user to
  pass 'vip' argument to it. (LIBCLOUD-166)
  [Adam Pickeral]

- Update Rackspace driver to support Auth 2.0. (LIBCLOUD-165)
  [Dave King]
- Add new `ex_create_balancer_access_rule` and `ex_create_balancer_access_rule_no_poll` method to the Rackspace driver. (LIBCLOUD-170)
  [Dave King]

*) DNS:

- Update Rackspace driver to support Auth 2.0. (LIBCLOUD-165)
  [Dave King]

Changes with Apache Libcloud 0.8.0:

*) General:

- Add 'request_kwargs' argument to the `get_poll_request_kwargs` method. This argument contains kwargs which were previously used to initiate the poll request.
  [Mark Everett]

- Add support for handling compressed responses (deflate, gzip). Also send "Accept-Encoding" "gzip, deflate" header with all the requests.
  [Tomaz Muraus]

- Fix debug module (LIBCLOUD_DEBUG env variable) so it works with Python 3
  [Tomaz Muraus]

*) Compute:

- Added support for retrieving OpenNebula v3.2 instance types, OpenNebula v3.0 network Public attribute support, and additional code coverage tests.
  [Hutson Betts]

- Add implementation for `ex_save_image` method to the OpenStack 1.1 driver.
  [Shawn Smith]

- Add support for Amazon new South America (Sao Paulo) location.
  [Tomaz Muraus]

- Fix a bug in OpenStack driver when 2.0_apikey or 2.0_password 'auth_version' is used.
  [Tomaz Muraus]

- Current OpenNebula OCCI implementation does not support a proper restart method. Rather it suspends and resumes. Therefore, `restart_node` has been removed from the OpenNebula driver.
  [Hutson Betts]

- Enable `ex_delete_image` method in the OpenStack 1.1 driver.
  [Shawn Smith]

- Return `NodeImage` instance in OpenStack 1.1 driver `ex_save_image` method (LIBCLOUD-138)
  [Shawn Smith]

- Enable `reboot_node` method in the OpenNebula 3.2 driver.
  [Hutson Betts]
- Fix a public_ips Node variable assignment in the Gandi.net driver.  
  [Aymeric Barantal]

- Updated the list of node states for OpenNebula drivers. (LIBCLOUD-148)  
  [Hutson Betts]

*) Storage:

- Propagate extra keyword arguments passed to the Rackspace driver connection class.  
  [Dave King]

*) Load-balancer:

- Add 'extra' attribute to the LoadBalancer object and retrieve all the virtual IP addresses in the Rackspace driver.  
  [Dave King]

- Add list_supported_algorithms() method to the base LoadBalancer class. This method returns a list of supported algorithms by the provider.  
  [Dave King]

- Update Rackspace driver:
  - Add two new supported algorithms: WEIGHTED_ROUND_ROBIN, WEIGHTED_LEAST_CONNECTIONS
  - Add ex_list_algorithm_names method
  - Add ex_get_balancer_error_page method
  - Add ex_balancer_access_list method
  - Populate LoadBalancer extra dictionary with more attributes
  - Add support for health monitors and connection throttling
  - Add more balancer states
  - ex_list_protocols_with_default_ports  
  [Dave King]

- Propagate extra keyword arguments passed to the Rackspace driver connection class.  
  [Dave King]

- Add 'extra' attribute to the Member object and populate it in the Rackspace driver.  
  [Mark Everett]

- Adds status to the Member object and conditions an 'enum' (Rackspace driver).  
  [Mark Everett]

- Add update_balancer method to the base LoadBalancer class.  
  [Mark Everett]

- Add update_balancer method to the Rackspace driver.  
  [Mark Everett]

- Add created and updated attribute to the LoadBalancer extra dictionary in the Rackspace driver.  
  [Mark Everett]

- Fix protocol name mapping in the Rackspace driver.  
  [Bill Woodward]
Changes with Apache Libcloud 0.7.1:

*) General:
- Fix a minor bug in debug mode (LIBCLOUD_DEBUG=/dev/stderr) which has been introduced when adding Python 3 compatibility layer.  
  [Paul Querna]
- Update OpenStack Auth API endpoint paths.  
  [Paul Querna]

Changes with Apache Libcloud 0.7.0:

*) General:
- Add support for Python 3.x.  
  [Tomaz Muraus]
- Remove old deprecated paths.  
  [Tomaz Muraus]

*) Compute:
- Update CloudSigma Zurich API endpoint address.  
  [Tomaz Muraus]
- Add new US Las Vegas endpoint to CloudSigma driver (types.CLOUDSIGMA_US)  
  [Tomaz Muraus]
- Allow user to specify drive type (hdd, ssd) when creating a CloudSigma server.  
  Note 'ssd' drive_type doesn't work with the API yet.  
  [Tomaz Muraus]
- Update OpenStack 1.1 driver to comply with the API specs. Need to make another call to retrieve node name and ip addresses when creating a node, because the first call only returns an id an the password. (GITHUB-40)  
  [Dave King]
- Add ex_node_ids argument to the EC2 driver list_nodes method. (GITHUB-39)  
  [Suvish Vt]
- If OpenStack Auth 2.0 API is used, also parse out tenant id and name and save it on the connection class (conn.tenant['id'], conn.tenant['name']).  
  [Tomaz Muraus]
- Add new "Cluster Compute Eight Extra Large" size to the Amazon EC2 driver.  
  [Tomaz Muraus]
- Add the following extension methods to the EC2 compute driver: ex_describe_allAddresses, ex_associate_addresses, ex_start_node, ex_stop_node.  
  [Suvish Vt]
- Change public_ip and private_ip attribute on the Node object to the public_ips and private_ips since both of the objects are always a list.

  Note: For backward compatibility you can still access public_ip and private_ip attributes, but this will be removed in the next release.
  [Tomaz Muraus]

- Fix an inconsistency in IBM SBC driver and make sure public_ips and private_ips attributes are a list.
  [Tomaz Muraus]

- Fix an inconsistency in OpSource driver and make sure public_ips is an empty list ([]), not 'unknown'
  [Tomaz Muraus]

- Updated support for OpenNebula.org v1.4, v2.x, and v3.x APIs and included additional compute tests validating functionality. (LIBCLOUD-121)
  [Hutson Betts]

*) Load-balancer:

- Add ex_member_address argument to the Rackspace driver list_balancers method. If this argument is provided, only loadbalancers which have a member with the provided IP address attached are returned.
  [Tomaz Muraus]

Changes with Apache Libcloud 0.6.2:

*) General

- Fix a bug in PollingConnection class - actually use and don't ignore the poll_interval
  [Tomaz Muraus]

*) Compute:

- Add support for Auth 2.0 API (keystone) to the OpenStack Auth connection class.
  [Brad Morgan]

- Add list_locations method to the OpenStack driver and fix some inconsistencies in the OpenStack driver extension method signatures.
  [Brad Morgan]

- Update Amazon EC2 driver and pricing data to support a new region - US West 2 (Oregon)
  [Tomaz Muraus]

- Expose 'CLOUDSTACK' provider. This driver can be used with an arbitrary CloudStack installation.
  [Tomaz Muraus]

*) Storage:

- Update Amazon S3 driver to support a new region - US West 2 (Oregon)
  [Tomaz Muraus]

*) DNS:
- Increase the default poll interval in the Rackspace driver to 2.5 seconds.
  [Tomaz Muraus]

- Fix a bug in Rackspace Cloud DNS driver and make sure to throw an exception if an unexpected status code is returned. Reported by "jeblair".
  [Tomaz Muraus]

Changes with Apache Libcloud 0.6.1:

*) General:

- Modify ParamikoSSHClient.connect so it supports authentication using a key file, (LIBCLOUD-116)
  [Jay Doane]

- User must now explicitly specify a path when using LIBCLOUD_DEBUG environment variable. (LIBCLOUD-95)
  [daveb, Tomaz Muraus]

- Add new XmlResponse and JsonResponse base class and modify all the driver-specific response classes to inherit from one of those two classes where applicable.
  [Caio Romão]

- Add new 'PollingConnection' class. This class can work with 'async' APIs. It sends an initial request and then periodically poll the API until the job has completed or a timeout has been reached.
  [Tomaz Muraus]

*) Compute:

- Add 24GB size to the GoGrid driver
  [Roman Bogorodskiy]

- Fix API endpoint URL in the Softlayer driver
  [Tomaz Muraus]

- Add support for OpenNebula 3.0 API (LIBCLOUD-120)
  [Hutson Betts]

- Add more attributes to the extra dictionary in the EC2 driver. (GITHUB-31)
  [Juan Carlos Moreno]

- Fix IP address assignment in the EC2 driver. Don't include "None" in the public_ip and private_ip Node list attribute.
  [Tomaz Muraus]

- Make deploy_node functionality more robust and don't start deployment if node public_ip attribute is an empty list.
  [Tomaz Muraus]

- Support SSH key authentication when using deploy_node.
  [Russell Haering, Tomaz Muraus]

- Enable deploy_node functionality in the EC2 driver using SSH key
- Enable paramiko library debug log level if LIBCLOUD_DEBUG is used and paramiko is installed. [Tomaz Muraus]

- Fix the request signature generation in the base EC2 compute driver. If the endpoint is using a non-standard port (Eucalyptus based installations), append it to the hostname used to generate the signature. [Simon Delamare]

- Add new "unavailable" state to the BrightboxNodeDriver class. [Tim Fletcher]

- Increase a PollingConnection timeout in the CloudStack connection and fix the context dictionary creation in the _async_request method. [gigimon]

- Fix networks retrieval in the CloudStack driver create_node method. Also only pass 'networkids' field to the API if there are any networks available. [gigimon, Tomaz Muraus]

- Fix list_nodes in the CloudStack driver. Private IPs aren't always available. [Tomaz Muraus]

*) Load-balancer:

- Add a missing argument to the method call inside LoadBalancer.attach_compute_node and Driver.balancer_attach_compute_node. [Tim Fletcher, Tomaz Muraus]

- Add missing destroy() method to the LoadBalancer class. [Tomaz Muraus]

*) DNS:

- New drivers for Rackspace Cloud DNS (US and UK region) [Tomaz Muraus]

- Add list_record_types() method. This method returns a list of record types supported by the provider. [Tomaz Muraus]

Changes with Apache Libcloud 0.6.0-beta1

*) General:

- All the driver classes now inherit from the BaseDriver class [Tomaz Muraus]

- Prefer simplejson (if available) over json module. (LIBCLOUD-112) [Geoff Greer]

- Update compute demo and change the syntax of test credentials stored in
- Enable SSL certificate verification by default and throw an exception if CA certificate files cannot be found. This can be overridden by setting `libcloud.security.VERIFY_SSL_CERT_STRICT` to False.
  [Tomaz Muraus]

*) Compute:

- Support for 1.1 API and many other improvements in the OpenStack driver; LIBCLOUD-83
  [Mike Nerone, Paul Querna, Brad Morgan, Tomaz Muraus]

- Add some extra methods to the Gandi.net driver (LIBCLOUD-115)
  [Aymeric Barantal]

- Add `ex_delete_image` method to the Rackspace driver. (GITHUB-27)
  [David Busby]

- Linode driver now supports new 'Japan' location
  [Jed Smith]

- Rackspace driver now inherits from the OpenStack one instead of doing it vice versa. (LIBCLOUD-110)
  [Mike Nerone]

- Properly populate NodeImage "details" dictionary in the Rackspace compute driver. (LIBCLOUD-107)
  [Lucy Mendel]

- Fix a bug in Eucalyptus driver `ex_describe_addresses` method.
  [Tomaz Muraus]

- Add the following new extension methods to the Rackspace driver:
  `ex_resize`, `ex_confirm_resize`, `ex_revert_resize`.
  [Tomaz Muraus]

- Also allow user to pass in Node object to some `ex_` methods in the Rackspace compute driver.
  [Tomaz Muraus]

- Throw an exception in `deploy_node` if `paramiko` library is not available
  [Tomaz Muraus]

- Fix `chmod` argument value which is passed to the `sftpclient.put` method; GITHUB-17
  [John Carr]

- New driver for Ninefold.com. (LIBCLOUD-98)
  [Benno Rice]

*) Storage:

- New driver for Google Storage based on the v1.0 / legacy API
  [Tomaz Muraus]
- New driver for Ninefold.com. (GITHUB-19)  
  [Benno Rice]

- Fix a bug in uploading an object with some versions of Python 2.7  
  where httplib library doesn't automatically call str() on the  
  header values.  
  [Tomaz Muraus]

- Allow users to upload (create) 0-bytes large (empty) objects  
  [Tomaz Muraus]

*) Load-balancer:

- New driver for Rackspace UK location  
  [Tomaz Muraus]

- New driver for Ninefold.com. (LIBCLOUD-98)  
  [Benno Rice]

*) DNS:

- Drivers for Linode DNS and Zerigo DNS  
  [Tomaz Muraus]

- Brand new DNS API!  
  [Tomaz Muraus]

Changes with Apache Libcloud 0.5.2

*) Compute:

- New driver for serverlove.com and skalicloud.com  
  [Tomaz Muraus]

- Fix node name and tag handling in the Amazon EC2 driver  
  [Wiktor Kolodziej]

- Fix pricing and response handling in the OpenStack driver  
  [Andrey Zhuchkov]

- Fix deploy_node() method and make it more robust  
  [Tomaz Muraus]

- Users can now pass file like objects to ScriptDeployment and  
  SSHKeyDeployment constructor.  
  [Tomaz Muraus]

- Include node tags when calling list_nodes() in the Amazon EC2  
  driver  
  [Trevor Pounds]

- Properly handle response errors in the Rackspace driver and  
  only throw InvalidCredsError if the returned status code is 401  
  [Brad Morgan]

- Fix the create_node method in the Nimbus driver and make the  
  "ex_create_tag" method a no-op, because Nimbus doesn't support creating  
  tags.  
  [Tomaz Muraus]
Storage:
- Fix handling of the containers with a lot of objects. Now a LazyList object is returned when user calls list_container_objects() method and this object transparently handles pagination.

Changes with Apache Libcloud 0.5.0

- Existing APIs directly on the libcloud.* module have been deprecated and will be removed in version 0.6.0. Most methods were moved to the libcloud.compute.* module.
- Add new libcloud.loadbalancers API, with initial support for:
  - GoGrid Load Balancers
  - Rackspace Load Balancers

- Add new libcloud.storage API, with initial support for:
  - Amazon S3
  - Rackspace CloudFiles

- Add new libcloud.compute drivers for:
  - Bluebox [Christian Paredes]
  - Gandi.net [Aymeric Barantal]
  - Nimbus [David LaBissoniere]
  - OpenStack [Roman Bogorodskiy]
  - Opsource.net [Joe Miller]

- Added "pricing" module and improved pricing handling.

- Updates to the GoGrid compute driver:
  - Use API version 1.0.
  - Remove sandbox flag.
  - Add ex_list_ips() to list IP addresses assigned to the account.
  - Implement ex_edit_image method which allows changing image attributes like name, description and make image public or private.

- Updates to the Amazon EC2 compute driver:
  - When creating a Node, use the name argument to set a Tag with the value. [Tomaz Muraus]
  - Add extension method for modifying node attributes and changing the node size. [Tomaz Muraus]
  - Add support for the new Amazon Region (Tokyo). [Tomaz Muraus]
  - Added ex_create_tags and ex_delete_tags. [Brandon Rhodes]
  - Include node Elastic IP addresses in the node public_ip attribute for the EC2 nodes. [Tomaz Muraus]
  - Use ipAddress and privateIpAddress attribute for the EC2 node public and private ip. [Tomaz Muraus]
  - Add ex_describe_addresses method to the EC2 driver. [Tomaz Muraus]

- Updates to the Rackspace CloudServers compute driver:
  - Add ex_rebuild() and ex_get_node_details() [Andrew Klochkov]
  - Expose URI of a Rackspace node to the node meta data. [Paul Querna]
*) Minor fixes to get the library and tests working on Python 2.7 and PyPy.
[Tomaz Muraus]

Changes with Apache Libcloud 0.4.2 (Released January 18, 2011)

*) Fix EC2 create_node to become backward compatible for NodeLocation.
[Tomaž Muraus]

*) Update code for compatibility with CPython 2.5
[Jerry Chen]

*) Implement ex_edit_node method for GoGrid driver which allows changing node attributes like amount of RAM or description.
[Roman Bogorodskiy]

*) Add ex_set_password and ex_set_server_name to Rackspace driver.
[Peter Herndon, Paul Querna]

*) Add Hard and Soft reboot methods to Rackspace driver.
[Peter Herndon]

*) EC2 Driver availability zones, via ex_list_availability_zones; list_locations rewrite to include availability zones
[Tomaž Muraus]

*) EC2 Driver Idempotency capability in create_node; LIBCLOUD-69
[David Labissoniere]

*) SSL Certificate Name Verification:
- libcloud.security module
- LibcloudHTTPSConnection, LibcloudHTTPConnection (alias)
- Emits warning when not verifying, or CA certs not found

*) Append ORD1 to available Rackspace location, but keep in the same node as DFW1, because it's not readable or writeable from the API.
[Per suggestion of Grig Gheorghiu]

*) ex_create_ip_group, ex_list_ip_groups, ex_delete_ip_group, ex_share_ip, ex_unshare_ip, ex_list_ip_addresses additions to Rackspace driver
[Andrew Klochkov]

*) New driver for CloudSigma.
[Tomaž Muraus]

*) New driver for Brightbox Cloud. (LIBCLOUD-63)
[Tim Fletcher]

*) Deployment capability to ElasticHosts
[Tomaž Muraus]

*) Allow deploy_node to use non-standard SSH username and port
[Tomaž Muraus]

*) Added Rackspace UK (London) support
[Chmouel Boudjnah]
*) GoGrid driver: add support for locations, i.e. listing of locations and creation of a node in specified location
   [Roman Bogorodskiy]

*) GoGrid and Rackspace drivers: add ex_save_image() extra call to convert running node to an image
   [Roman Bogorodskiy]

*) GoGrid driver: add support for creating 'sandbox' server and populate isSandbox flag in node's extra information.
   [Roman Bogorodskiy]

*) Add ImportKeyPair and DescribeKeyPair to EC2. (LIBCLOUD-62)
   [Philip Schwartz]

*) Update EC2 driver and test fixtures for new API.
   [Philip Schwartz]

Changes with Apache Libcloud 0.4.0 [Released October 6, 2010]

*) Add create keypair functionality to EC2 Drivers. (LIBCLOUD-57)
   [Grig Gheorghiu]

*) Improve handling of GoGrid accounts with limited access API keys. [Paul Querna]

*) New Driver for ElasticHosts. (LIBCLOUD-45)
   [Tomaz Muraus]

*) Use more consistent name for GoGrid driver and use http POST method for 'unsafe' operations
   [Russell Haering]

*) Implement password handling and add deployment support for GoGrid nodes.
   [Roman Bogorodskiy]

*) Fix behavior of GoGrid's create_node to wait for a Node ID.
   [Roman Bogorodskiy]

*) Add ex_create_node_nowait to GoGrid driver if you don't need to wait for a Node ID when creating a node.
   [Roman Bogorodskiy]

*) Removed libcloud.interfaces module.
   [Paul Querna]

*) Removed dependency on zope.interfaces.
   [Paul Querna]

*) RimuHosting moved API endpoint address.
   [Paul Querna]

*) Fix regression and error in GoGrid driver for parsing node objects.
   [Roman Bogorodskiy]

*) Added more test cases for GoGrid driver. (LIBCLOUD-34)
*) Fix parsing of Slicehost nodes with multiple Public IP addresses.  
[Paul Querna]

*) Add exit_status to ScriptDeployment. (LIBCLOUD-36)  
[Paul Querna]

*) Update prices for several drivers.  
[Brad Morgan, Paul Querna]

*) Update Linode driver to reflect new plan sizes.  
[Jed Smith]

*) Change default of 'location' in Linode create_node. (LIBCLOUD-41)  
[Jed Smith, Steve Steiner]

*) Document the Linode driver.  
[Jed Smith]

*) Request a private, LAN IP address at Linode creation.  
[Jed Smith]

Changes with Apache Libcloud 0.3.1 [Released May 11, 2010]

*) Updates to Apache License blocks to correctly reflect status as an  
Apache Project.

*) Fix NOTICE file to use 2010 copyright date.

*) Improve error messages for when running the test cases without  
first setting up a secrets.py

Changes with Apache Libcloud 0.3.0 [Tagged May 6, 2010, not released]

*) New Drivers for:  
- Dreamhost  
- Eucalyptus  
- Enomaly ECP  
- IBM Developer Cloud  
- OpenNebula  
- SoftLayer

*) Added new deployment and bootstrap API.

*) Improved Voxel driver.

*) Added support for Amazon EC2 Asia Pacific (Singapore) Region.

*) Improved test coverage for all drivers.

*) Add support for multiple security groups in EC2.

*) Fixed bug in Rackspace and RimuHosting when using multiple threads.

*) Improved debugging and logging of HTTP requests.

*) Improved documentation for all classes and methods.
### 1.1.3 Supported Providers

This pages lists supported providers and methods for all the APIs.

#### Compute

**Provider Matrix**

<table>
<thead>
<tr>
<th>Provider</th>
<th>Documentation</th>
<th>Provider constant</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abiquo</td>
<td>ABIQUO</td>
<td>libcloud.compute.drivers.abiquo</td>
<td>libcloud.compute.drivers.abiquo</td>
</tr>
<tr>
<td>Bluebox Blocks</td>
<td>BLUEBOX</td>
<td>libcloud.compute.drivers.bl</td>
<td>libcloud.compute.drivers.bl</td>
</tr>
<tr>
<td>Brightbox</td>
<td>BRIGHTBOX</td>
<td>libcloud.compute.drivers.bl</td>
<td>libcloud.compute.drivers.bl</td>
</tr>
<tr>
<td>CloudFrames</td>
<td>CLOUDFRAMES</td>
<td>libcloud.compute.drivers.cl</td>
<td>libcloud.compute.drivers.cl</td>
</tr>
<tr>
<td>CloudSigma</td>
<td>CLOUDSIGMA</td>
<td>libcloud.compute.drivers.cl</td>
<td>libcloud.compute.drivers.cl</td>
</tr>
<tr>
<td>CloudStack</td>
<td>CLOUDSTACK</td>
<td>libcloud.compute.drivers.cl</td>
<td>libcloud.compute.drivers.cl</td>
</tr>
<tr>
<td>Digital Ocean</td>
<td>DIGITAL_OCEAN</td>
<td>libcloud.compute.drivers.di</td>
<td>libcloud.compute.drivers.di</td>
</tr>
<tr>
<td>Dreamhost</td>
<td>DREAMHOST</td>
<td>libcloud.compute.drivers.dr</td>
<td>libcloud.compute.drivers.dr</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>EC2</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (ap-northeast-1)</td>
<td>EC2_AP_NORTHEAST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-1)</td>
<td>EC2_AP_SOUTHEAST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-2)</td>
<td>EC2_AP_SOUTHEAST2</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>EC2_EU</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>EC2_EU_WEST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (sa-east-1)</td>
<td>EC2_SA_EAST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>EC2_US_EAST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-1)</td>
<td>EC2_US_WEST</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-2)</td>
<td>EC2_US_WEST_OREGON</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Xenomaly</td>
<td>ECP</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>ElasticHosts (syd-y)</td>
<td>ELASTICHOSTS</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (tor-p)</td>
<td>ELASTICHOSTS_CA1</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (cn-1)</td>
<td>ELASTICHOSTS_CN1</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (lon-p)</td>
<td>ELASTICHOSTS_UK1</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (lon-b)</td>
<td>ELASTICHOSTS_UK2</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (sat-p)</td>
<td>ELASTICHOSTS_US1</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (lax-p)</td>
<td>ELASTICHOSTS_US2</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>ElasticHosts (sjc-c)</td>
<td>ELASTICHOSTS_US3</td>
<td>libcloud.compute.drivers.el</td>
<td>libcloud.compute.drivers.el</td>
</tr>
<tr>
<td>Eucalyptus</td>
<td>EUCALYPTUS</td>
<td>libcloud.compute.drivers.ec</td>
<td>libcloud.compute.drivers.ec</td>
</tr>
<tr>
<td>Gandi</td>
<td>GANDI</td>
<td>libcloud.compute.drivers.ga</td>
<td>libcloud.compute.drivers.ga</td>
</tr>
<tr>
<td>Google Compute Engine</td>
<td>GCE</td>
<td>libcloud.compute.drivers.gc</td>
<td>libcloud.compute.drivers.gc</td>
</tr>
<tr>
<td>GoGrid</td>
<td>GOGRID</td>
<td>libcloud.compute.drivers.go</td>
<td>libcloud.compute.drivers.go</td>
</tr>
<tr>
<td>HostVirtual</td>
<td>HOSTVIRTUAL</td>
<td>libcloud.compute.drivers.ho</td>
<td>libcloud.compute.drivers.ho</td>
</tr>
<tr>
<td>IBM SmartCloud Enterprise</td>
<td>IBM</td>
<td>libcloud.compute.drivers.ib</td>
<td>libcloud.compute.drivers.ib</td>
</tr>
</tbody>
</table>
### Table 1.1 – continued from previous page

<table>
<thead>
<tr>
<th>Provider</th>
<th>Documentation</th>
<th>Provider constant</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joyent</td>
<td>JOYENT</td>
<td>libcloud.compute.drivers.joyent</td>
<td></td>
</tr>
<tr>
<td>KTUCloud</td>
<td>KTUCLOUD</td>
<td>libcloud.compute.drivers.ktucloud</td>
<td></td>
</tr>
<tr>
<td>Libvirt</td>
<td>LIBVIRT</td>
<td>libcloud.compute.drivers.libvirt</td>
<td></td>
</tr>
<tr>
<td>Linode</td>
<td>LINODE</td>
<td>libcloud.compute.drivers.libcloud</td>
<td></td>
</tr>
<tr>
<td>NephoScale</td>
<td>NEPHOSCALE</td>
<td>libcloud.compute.drivers.nephoscale</td>
<td></td>
</tr>
<tr>
<td>&quot;Nimbus&quot;_</td>
<td>Click</td>
<td>NIMBUS</td>
<td>libcloud.compute.drivers.ec2</td>
</tr>
<tr>
<td>Ninefold</td>
<td>NINEFOLD</td>
<td>libcloud.compute.drivers.ninefold</td>
<td></td>
</tr>
<tr>
<td>OpenNebula</td>
<td>OPENNEBULA</td>
<td>libcloud.compute.drivers.opennebula</td>
<td></td>
</tr>
<tr>
<td>OpenStack</td>
<td>OPENSTACK</td>
<td>libcloud.compute.drivers.openstack</td>
<td></td>
</tr>
<tr>
<td>Opsource</td>
<td>OPSOURCE</td>
<td>libcloud.compute.drivers.opsource</td>
<td></td>
</tr>
<tr>
<td>Rackspace Cloud (Next Gen)</td>
<td>RACKSPACE</td>
<td>libcloud.compute.drivers.rackspace</td>
<td></td>
</tr>
<tr>
<td>Rackspace Cloud (First Gen)</td>
<td>RACKSPACE_FIRST_GEN</td>
<td>libcloud.compute.drivers.rackspace</td>
<td></td>
</tr>
<tr>
<td>RimuHosting</td>
<td>RIMUHOSTING</td>
<td>libcloud.compute.drivers.rimuhosting</td>
<td></td>
</tr>
<tr>
<td>ServerLove</td>
<td>SERVERLOVE</td>
<td>libcloud.compute.drivers.serverlove</td>
<td></td>
</tr>
<tr>
<td>skalicloud</td>
<td>SKALICLOUD</td>
<td>libcloud.compute.drivers.skalicloud</td>
<td></td>
</tr>
<tr>
<td>Slicehost</td>
<td>SLICEHOST</td>
<td>libcloud.compute.drivers.slicehost</td>
<td></td>
</tr>
<tr>
<td>SoftLayer</td>
<td>SOFTLAYER</td>
<td>libcloud.compute.drivers.softlayer</td>
<td></td>
</tr>
<tr>
<td>vCloud</td>
<td>TERREMARK</td>
<td>libcloud.compute.drivers.vcloud</td>
<td></td>
</tr>
<tr>
<td>VCL</td>
<td>VCL</td>
<td>libcloud.compute.drivers.vcl</td>
<td></td>
</tr>
<tr>
<td>vCloud</td>
<td>VCLUSION</td>
<td>libcloud.compute.drivers.vcl</td>
<td></td>
</tr>
<tr>
<td>vps.net</td>
<td>VPSNET</td>
<td>libcloud.compute.drivers.vpsnet</td>
<td></td>
</tr>
</tbody>
</table>

**Supported Methods (Main)**

<table>
<thead>
<tr>
<th>Provider</th>
<th>list nodes</th>
<th>create node</th>
<th>reboot node</th>
<th>destroy node</th>
<th>list images</th>
<th>list sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abiquo</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Bluebox Blocks</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Brightbox</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>CloudFrames</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudSigma</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudSigma</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudStack</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Digital Ocean</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Dreamhost</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-northeast-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-2)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (sa-east-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-2)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-2)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Enomalys Elastic Computing Platform</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (syd-y)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Continued
### Table 1.2 – continued from previous page

<table>
<thead>
<tr>
<th>Provider</th>
<th>list nodes</th>
<th>create node</th>
<th>reboot node</th>
<th>destroy node</th>
<th>list images</th>
<th>list sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElasticHosts (tor-p)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (cn-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (lon-p)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (lon-b)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (lax-p)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ElasticHosts (sjc-c)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Eucalyptus</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Gandi</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Google Compute Engine</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GoGrid</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HostVirtual</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>IBM SmartCloud Enterprise</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Joyent</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>KTUCloud</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Libvirt</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Linode</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>NephoScale</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Nimbus</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Ninefold</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>OpenNebula</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>OpenStack</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Opsource</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace Cloud (Next Gen)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace Cloud (First Gen)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>RimuHosting</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ServerLove</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>skalicloud</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Slicehost</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SoftLayer</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>vCloud</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>VCL</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>vCloud</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Voxel VoxCLOUD</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>vps.net</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

#### Supported Methods (Block Storage)

<table>
<thead>
<tr>
<th>Provider</th>
<th>list volumes</th>
<th>create volume</th>
<th>destroy volume</th>
<th>attach volume</th>
<th>detach volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abiquo</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Bluebox Blocks</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Brightbox</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CloudFrames</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CloudSigma</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CloudSigma</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CloudStack</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Digital Ocean</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Dreamhost</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Provider</td>
<td>list volumes</td>
<td>create volume</td>
<td>destroy volume</td>
<td>attach volume</td>
<td>detach volume</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-northeast-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (ap-southeast-2)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (eu-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (sa-east-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-1)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon EC2 (us-west-2)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Enomaly Elastic Computing Platform</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (syd-y)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (tor-p)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (cn-1)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (lon-p)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (lon-b)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (sat-p)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (lax-p)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ElasticHosts (lax-p)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Eucalyptus</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Gandi</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Google Compute Engine</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GoGrid</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>HostVirtual</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>IBM SmartCloud Enterprise</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Joyent</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>KTUCloud</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Libvirt</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Linode</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>NephoScale</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Nimbus</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Ninefold</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>OpenNebula</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>OpenStack</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Opsource</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Rackspace Cloud (Next Gen)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace Cloud (First Gen)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>RimuHosting</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ServerLove</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>skalcloud</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Slicehost</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>SoftLayer</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>vCloud</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>VCL</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>vCloud</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Voxel VoxCLOUD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>vps.net</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
## Load Balancer

### Provider Matrix

<table>
<thead>
<tr>
<th>Provider</th>
<th>Documentation</th>
<th>Provider constant</th>
<th>Module</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightbox</td>
<td></td>
<td>BRIGHT-BOX</td>
<td>libcloud.loadbalancer.drivers.BrightboxLBDriver</td>
<td>BrightboxLBDriver</td>
</tr>
<tr>
<td>CloudStack</td>
<td></td>
<td>CLOUD-STACK</td>
<td>libcloud.loadbalancer.drivers.CloudStackLBDriver</td>
<td>CloudStackLBDriver</td>
</tr>
<tr>
<td>ELB</td>
<td></td>
<td>ELB</td>
<td>libcloud.loadbalancer.drivers.ElasticLBDriver</td>
<td>ElasticLBDriver</td>
</tr>
<tr>
<td>Google Compute Engine</td>
<td></td>
<td>GCE</td>
<td>libcloud.loadbalancer.drivers.GCELBDriver</td>
<td>GCELBDriver</td>
</tr>
<tr>
<td>GoGrid LB</td>
<td></td>
<td>GOGRID</td>
<td>libcloud.loadbalancer.drivers.GoGridLBDriver</td>
<td>GoGridLBDriver</td>
</tr>
<tr>
<td>Ninefold LB</td>
<td></td>
<td>NINEFOLD</td>
<td>libcloud.loadbalancer.drivers.NinefoldLBDriver</td>
<td>NinefoldLBDriver</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td></td>
<td>RACKSPACE</td>
<td>libcloud.loadbalancer.drivers.RackspaceLBDriver</td>
<td>RackspaceLBDriver</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td></td>
<td>RACKSPACE</td>
<td>libcloud.loadbalancer.drivers.RackspaceLBDriver</td>
<td>RackspaceLBDriver</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td></td>
<td>RACKSPACE</td>
<td>libcloud.loadbalancer.drivers.RackspaceLBDriver</td>
<td>RackspaceLBDriver</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td></td>
<td>RACKSPACE</td>
<td>libcloud.loadbalancer.drivers.RackspaceLBDriver</td>
<td>RackspaceLBDriver</td>
</tr>
</tbody>
</table>

### Supported Methods

<table>
<thead>
<tr>
<th>Provider</th>
<th>create balancer</th>
<th>list balancers</th>
<th>list members</th>
<th>attach member</th>
<th>detach member</th>
<th>attach compute node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightbox</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudStack</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ELB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Google Compute Engine</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GoGrid LB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Ninefold LB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace LB</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
## Object Storage

### Provider Matrix

<table>
<thead>
<tr>
<th>Provider</th>
<th>Documentation</th>
<th>Provider constant</th>
<th>Module</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Azure (blobs)</td>
<td></td>
<td>AZURE_BLOBS</td>
<td>libcloud.storage.drivers.azure_blobs</td>
<td>AzureBlobsStorageDriver</td>
</tr>
<tr>
<td>CloudFiles (SWIFT)</td>
<td></td>
<td>CLOUDFILES_SWIFT</td>
<td>libcloud.storage.drivers.cloudfiles</td>
<td>CloudFilesSwiftStorageDriver</td>
</tr>
<tr>
<td>CloudFiles (UK)</td>
<td></td>
<td>CLOUDFILES_UK</td>
<td>libcloud.storage.drivers.cloudfiles</td>
<td>CloudFilesUKStorageDriver</td>
</tr>
<tr>
<td>CloudFiles (US)</td>
<td></td>
<td>CLOUDFILES_US</td>
<td>libcloud.storage.drivers.cloudfiles</td>
<td>CloudFilesUSStorageDriver</td>
</tr>
<tr>
<td>Google Storage</td>
<td>Click</td>
<td>GOOGLE_STORAGE</td>
<td>libcloud.storage.drivers.google_storage</td>
<td>GoogleStorageDriver</td>
</tr>
<tr>
<td>KTUCloud Storage</td>
<td></td>
<td>KTUCLOUD</td>
<td>libcloud.storage.drivers.ktucloud</td>
<td>KTUCloudStorageDriver</td>
</tr>
<tr>
<td>Local Storage</td>
<td></td>
<td>LOCAL</td>
<td>libcloud.storage.drivers.local</td>
<td>LocalStorageDriver</td>
</tr>
<tr>
<td>‘Nimbus’</td>
<td></td>
<td>NIMBUS</td>
<td>libcloud.storage.drivers.nimbus</td>
<td>NimbusStorageDriver</td>
</tr>
<tr>
<td>Ninefold</td>
<td></td>
<td>NINEFOLD</td>
<td>libcloud.storage.drivers.ninefold</td>
<td>NinefoldStorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (standard)</td>
<td>S3</td>
<td>S3</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3StorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (ap-northeast-1)</td>
<td></td>
<td>S3_AP_NORTHEAST</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3APNEStorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (ap-southeast-1)</td>
<td></td>
<td>S3_AP_SOUTHEAST</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3APSEStorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (eu-west-1)</td>
<td>S3_EU_WEST</td>
<td>S3</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3EUWestStorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (us-west-1)</td>
<td>S3_US_WEST</td>
<td>S3</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3USWestStorageDriver</td>
</tr>
<tr>
<td>Amazon S3 (us-west-2)</td>
<td>S3_US_WEST_OREGON</td>
<td>S3_US_WEST</td>
<td>libcloud.storage.drivers.s3</td>
<td>S3USWestOregonStorageDriver</td>
</tr>
</tbody>
</table>
## Supported Methods (Main)

<table>
<thead>
<tr>
<th>Provider</th>
<th>list containers</th>
<th>list objects</th>
<th>create container</th>
<th>delete container</th>
<th>upload object</th>
<th>streaming object upload</th>
<th>download object</th>
<th>streaming object download</th>
<th>delete object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Azure (blobs)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>CloudFiles (SWIFT)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudFiles (UK)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudFiles (US)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Google Storage</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>KTUCloud Storage</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Local Storage</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>‘Nimbus’</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Ninefold</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (standard)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon S3 (ap-northeast-1)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon S3 (ap-southeast-1)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon S3 (eu-west-1)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon S3 (us-west-1)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Amazon S3 (us-west-2)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Supported Methods (CDN)

<table>
<thead>
<tr>
<th>Provider</th>
<th>enable container cdn</th>
<th>enable object cdn</th>
<th>get container cdn URL</th>
<th>get object cdn URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Azure (blobs)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CloudFiles</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudFiles (SWIFT)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudFiles (UK)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>CloudFiles (US)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Google Storage</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>KTUCloud Storage</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Local Storage</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>‘Nimbus’</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Ninefold</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (standard)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (ap-northeast-1)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (ap-southeast-1)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (eu-west-1)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (us-west-1)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amazon S3 (us-west-2)</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

DNS

Provider Matrix

<table>
<thead>
<tr>
<th>Provider</th>
<th>Documentation</th>
<th>Provider constant</th>
<th>Module</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gandi DNS</td>
<td>Gandi</td>
<td>libcloud.dns.drivers.gandi</td>
<td>GandiDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Host Virtual DNS</td>
<td>HostVirtual</td>
<td>libcloud.dns.drivers.hostvirtual</td>
<td>HostVirtualDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Linode DNS</td>
<td>Linode</td>
<td>libcloud.dns.drivers.linode</td>
<td>LinodeDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Rackspace DNS</td>
<td>RACKSPACE</td>
<td>libcloud.dns.drivers.rackspace</td>
<td>RackspaceDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Rackspace DNS (UK)</td>
<td>RACKSPACE_UK</td>
<td>libcloud.dns.drivers.rackspace</td>
<td>RackspaceUKDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Rackspace DNS (US)</td>
<td>RACKSPACE_US</td>
<td>libcloud.dns.drivers.rackspace</td>
<td>RackspaceUSDNSDriver</td>
<td></td>
</tr>
<tr>
<td>Route53 DNS</td>
<td>ROUTE53</td>
<td>libcloud.dns.drivers.route53</td>
<td>Route53DNSDriver</td>
<td></td>
</tr>
<tr>
<td>Zerigo DNS</td>
<td>ZERIGO</td>
<td>libcloud.dns.drivers.zerigo</td>
<td>ZerigoDNSDriver</td>
<td></td>
</tr>
</tbody>
</table>
Supported Methods

<table>
<thead>
<tr>
<th>Provider</th>
<th>list zones</th>
<th>list records</th>
<th>create zone</th>
<th>update zone</th>
<th>create record</th>
<th>update record</th>
<th>delete zone</th>
<th>delete record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gandi DNS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Host Virtual DNS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Linode DNS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace DNS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace DNS (UK)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace DNS (US)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Route53 DNS</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Zerigo DNS</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

1.1.4 Compute

The compute component of libcloud allows you to manage cloud and virtual servers offered by different providers, more than 20 in total.

In addition to managing the servers this component also allows you to run deployment scripts on newly created servers. Deployment or “bootstrap” scripts allow you to execute arbitrary shell commands. This functionality is usually used to prepare your freshly created server, install your SSH key, and run a configuration management tool (such as Puppet, Chef, or cfengine) on it.

Besides managing cloud and virtual servers, compute component also allows you to manage cloud block storage (not to be confused with cloud object storage) for providers which support it. Block storage management is lives under compute API, because it is in most cases tightly coupled with compute resources.

Terminology

Compute

- **Node** - represents a cloud or virtual server.
- **NodeSize** - represents node hardware configuration. Usually this is amount of the available RAM, bandwidth, CPU speed and disk size. Most of the drivers also expose hourly price (in dollars) for the Node of this size.
- **NodeImage** - represents an operating system image.
- **NodeLocation** - represents a physical location where a server can be.
- **NodeState** - represents a node state. Standard states are: running, rebooting, terminated, pending, and unknown.

Block Storage

- **StorageVolume** - represents a block storage volume
- **VolumeSnapshot** - represents a point in time snapshot of a StorageVolume
Supported Providers

For a list of supported providers see supported providers page.

Pricing

For majority of the compute providers Libcloud provides estimated pricing information which tells users how much it costs per hour to run a Node with a specific NodeSize.

For more information, please see the pricing page.

Deployment

Libcloud provides deployment functionality which makes bootstrapping a server easier. It allows you to create a server and run shell commands on it once the server has been created.

For more information and examples, please see the deployment page.

Examples

We have examples of several common patterns.

API Reference

For a full reference of all the classes and methods exposed by the compute API, see this page.

1.1.5 Object Storage

Note: Object Storage API is available in Libcloud 0.5.0 and higher.

Storage API allows you to manage cloud object storage (not to be confused with cloud block storage) and services such as Amazon S3, Rackspace CloudFiles, Google Storage and others.

Besides managing cloud object storage, storage component also exposes simple CDN management functionality.

Terminology

• **Object** - represents an object or so called BLOB.

• **Container** - represents a container which can contain multiple objects. You can think of it as a folder on a file system. Difference between container and a folder on file system is that containers cannot be nested. Some APIs and providers (e.g. AWS) refer to it as a Bucket.

Supported Providers

For a list of supported providers see supported providers page.
Examples

We have examples of several common patterns.

API Reference

There is a reference to all the methods on the base storage driver.

1.1.6 Load Balancer

Note: Load Balancer API is available in Libcloud 0.5.0 and higher.

Load Balancer API allows you to manage Load Balancers as a service and services such as Rackspace Cloud Load Balancers, GoGrid Load Balancers and Ninefold Load Balancers.

Terminology

- **LoadBalancer** - represents a load balancer instance.
- **Member** - represents a load balancer member.
- **Algorithm** - represents a load balancing algorithm (round-robin, random, least connections, etc.).

Supported Providers

For a list of supported providers see supported providers page.

Examples

We have examples of several common patterns.

API Reference

There is a reference to all the methods on the base loadbalancer driver.

1.1.7 DNS

Note: DNS API is available in Libcloud 0.6.0 and higher.

DNS API allows you to manage DNS as A Service and services such as Zerigo DNS, Rackspace Cloud DNS and others.
Terminology

- **Zone** - Represents a DNS zone or so called domain.
- **Record** - Represents a DNS record. Each record belongs to a Zone and has a type and data attribute. Value of the data attribute depends on the record type. Some record types also require user to associate additional attributes with them. Those additional attributes are stored in the extra attribute (dictionary) on the record object. An example include MX and SRV record type which also contains a priority.
- **RecordType** - Represents a DNS record type (A, AAAA, MX, TXT, SRV, PTR, NS, etc.)
- **Zone Type** - Each zone has a type attribute. This attribute represents a zone type. Type can either be master (also called primary) or slave (also called secondary).

Supported Providers

For a list of supported providers see supported providers page.

Examples

We have examples of several common patterns.

API Reference

There is a reference to all the methods on the base dns driver.

1.1.8 Upgrade Notes

This page describes how to upgrade from a previous version to a new version which contains backward incompatible or semi-incompatible changes and how to preserve the old behavior when this is possible.

Libcloud 0.14.0

To make drivers with multiple regions easier to use, one of the big changes in this version is move away from the old “one class per region” model to a new single class plus region argument model.

More information on how this affects existing drivers and your code can be found bellow.

Cache busting functionality is now only enabled in Rackspace first-gen driver

Cache busting functionality has been disabled in the Rackspace next-gen driver and all of the OpenStack drivers. It’s now only enabled in the Rackspace first-gen driver.

Cache busting functionality works by appending a random query parameter to every GET HTTP request. It was originally added to the Rackspace first-gen driver a long time ago to avoid excessive HTTP caching on the provider side. This excessive caching some times caused list_nodes and other calls to return stale data.

This approach should not be needed with Rackspace next-gen and OpenStack drivers so it has been disabled.

No action is required on the user’s side.
New default kernel versions used when creating Linode servers

Kernel versions which are used by default when creating Linode servers have been updated.

Old default kernel versions:

- x86 (no paravirt-ops) - 2.6.18.8-x86_64-linode1 (#60)
- x86 (paravirt-ops) - 2.6.18.8-x86_64-linode1 (#110)
- x86_64 (no paravirt-ops) - 2.6.39.1-linode34 (#107)
- x86 (paravirt-ops)64 - 2.6.18.8-x86_64-linode1 (#111)

New default kernel versions:

- x86 - 3.9.3-x86-linode52 (#137)
- x86_64 - 3.9.3-x86_64-linode33 (#138)

Those new kernel versions now come with paravirt-ops by default.

If you want to preserve the old behavior, you can pass `ex_kernel` argument to the `create_node` method.

Keep in mind that using old kernels is strongly discouraged since they contain known security holes.

For example:

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.LINODE)

driver = cls('username', 'api_key')
driver.create_node(..., ex_kernel=110)
```

Addition of new “STOPPED” node state

This version includes a new state called `libcloud.compute.types.NodeState.STOPPED`. This state represents a node which has been stopped and can be started later on (unlike `TERMINATED` state which represents a node which has been terminated and can’t be started later on).

As such, `EC2` and `HostVirtual` drivers have also been updated to recognize this new state.

Before addition of this state, nodes in this state were mapped to `NodeState.UNKNOWN`.

Amazon EC2 compute driver changes

Amazon EC2 compute driver has moved to single class plus `region` argument model. As such, the following provider constants have been deprecated:

- `EC2_US_EAST`
- `EC2_US_WEST_OREGON`
- `EC2_EU`
- `EC2_EU_WEST`
- `EC2_AP_SOUTHEAST`
- `EC2_AP_SOUTHEAST2`
• EC2_AP_NORTHEAST
• EC2_SA_EAST

And replaced with a single constant:

• EC2 - Supported values for the region argument are: us-east-1, us-west-1, us-west-2, eu-west-1, ap-southeast-1, ap-northeast-1, sa-east-1, ap-southeast-2. Default value is us-east-1.

List which shows how old classes map to a new region argument value:

• EC2_US_EAST -> us-east-1
• EC2_US_WEST -> us-west-1
• EC2_US_WEST_OREGON -> us-west-2
• EC2_EU -> eu-west-1
• EC2_EU_WEST -> eu-west-1
• EC2_AP_SOUTHEAST -> ap-southeast-1
• EC2_AP_SOUTHEAST2 -> ap-southeast-2
• EC2_AP_NORTHEAST -> ap-northeast-1
• EC2_SA_EAST -> sa-east-1

Old code:

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls1 = get_driver(Provider.EC2)
cls2 = get_driver(Provider.EC2_EU_WEST)

driver1 = cls('username', 'api_key')
driver2 = cls('username', 'api_key')
```

New code:

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.EC2)

driver1 = cls('username', 'api_key', region='us-east-1')
driver2 = cls('username', 'api_key', region='eu-west-1')
```

**Rackspace compute driver changes**

Rackspace compute driver has moved to single class plus region argument model. As such, the following provider constants have been removed:

• RACKSPACE
• RACKSPACE_UK
• RACKSPACE_AU
• RACKSPACE_NOVA_ORD
And replaced with two new constants:

- **RACKSPACE** - Supported values for the `region` argument are: us, uk. Default value is us.
- **RACKSPACE_FIRST_GEN** - Supported values for the `region` argument are: dfw, ord, iad, lon, syd, hkg. Default value is dfw.

Besides that, the `RACKSPACE` provider constant now defaults to next-generation OpenStack based servers. Previously it defaulted to first generation cloud servers.

If you want to preserve old behavior and use first-gen drivers you need to use `RACKSPACE_FIRST_GEN` provider constant.

First generation cloud servers now also use auth 2.0 by default. Previously they used auth 1.0.

Because of the nature of this first-gen to next-gen change, old constants have been fully removed and unlike region changes in other driver, this change is not backward compatible.

List which shows how old, first-gen classes map to a new `region` argument value:

- **RACKSPACE** -> us
- **RACKSPACE_UK** -> uk

List which shows how old, next-gen classes map to a new `region` argument value:

- **RACKSPACE_NOVA_ORD** -> ord
- **RACKSPACE_NOVA_DFW** -> dfw
- **RACKSPACE_NOVA_LON** -> lon
- **RACKSPACE_AU** -> syd

More examples which show how to update your code to work with a new version can be found below.

Old code (connecting to a first-gen provider):

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

c1 = get_driver(Provider.RACKSPACE)  # US region

c2 = get_driver(Provider.RACKSPACE_UK)  # UK region

driver1 = c1('username', 'api_key')
driver2 = c2('username', 'api_key')
```

New code (connecting to a first-gen provider):

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

c = get_driver(Provider.RACKSPACE_FIRST_GEN)

driver1 = c('username', 'api_key', region='us')
driver2 = c('username', 'api_key', region='uk')
```

Old code (connecting to a next-gen provider)
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls1 = get_driver(Provider.RACKSPACE_NOVA_ORD)
cls2 = get_driver(Provider.RACKSPACE_NOVA_DFW)
cls3 = get_driver(Provider.RACKSPACE_NOVA_LON)

driver1 = cls('username', 'api_key')
driver2 = cls('username', 'api_key')
driver3 = cls('username', 'api_key')

New code (connecting to a next-gen provider)

from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.RACKSPACE)

driver1 = cls('username', 'api_key', region='ord')
driver2 = cls('username', 'api_key', region='dfw')
driver3 = cls('username', 'api_key', region='lon')

CloudStack compute driver changes

CloudStack driver received a lot of changes and additions which will make it more pleasant to use. Backward incompatible changes are listed below:

- CloudStackForwardingRule class has been renamed to CloudStackIPForwardingRule
- create_node method arguments are now more consistent with other drivers. Security groups are now passed as ex_security_groups, SSH keypairs are now passed as ex_keyname and userdata is now passed as ex_userdata.
- For advanced networking zones, multiple networks can now be passed to the create_node method instead of a single network id. These networks need to be instances of the CloudStackNetwork class.
- The extra_args argument of the create_node method has been removed. The only arguments accepted are now the defaults name, size, image, location plus ex_keyname, ex_userdata, ex_security_groups and networks.

Joyent compute driver changes

Joyent driver has been aligned with other drivers and now the constructor takes region instead of location argument.

For backward compatibility reasons, old argument will continue to work until the next major release.

Old code:

from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.JOYENT)

driver = cls('username', 'api_key', location='us-east-1')

Old code:
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.JOYENT)

driver = cls('username', 'api_key', region='us-east-1')

ElasticHosts compute driver changes

ElasticHosts compute driver has moved to single class plus region argument model. As such, the following provider constants have been deprecated:

- ELASTICHOSTS_UK1
- ELASTICHOSTS_UK1
- ELASTICHOSTS_US1
- ELASTICHOSTS_US2
- ELASTICHOSTS_US3
- ELASTICHOSTS_CA1
- ELASTICHOSTS_AU1
- ELASTICHOSTS_CN1

And replaced with a single constant:

- ELASTICHOSTS - Supported values for the region argument are: lon-p, lon-b, sat-p, lax-p, sjc-c, tor-p, syd-y, cn-1 Default value is sat-p.

List which shows how old classes map to a new region argument value:

- ELASTICHOSTS_UK1 -> lon-p
- ELASTICHOSTS_UK1 -> lon-b
- ELASTICHOSTS_US1 -> sat-p
- ELASTICHOSTS_US2 -> lax-p
- ELASTICHOSTS_US3 -> sjc-c
- ELASTICHOSTS_CA1 -> tor-p
- ELASTICHOSTS_AU1 -> syd-y
- ELASTICHOSTS_CN1 -> cn-1

Because of this change main driver class has also been renamed from libcloud.compute.drivers.elastichosts.ElasticHostsBaseNodeDriver to libcloud.compute.drivers.elastichosts.ElasticHostsNodeDriver.

Only users who directly instantiate a driver and don’t use recommended get_driver method are affected by this change.

Old code:

from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls1 = get_driver(Provider.ELASTICHOSTS_UK1)
Unification of extension arguments for security group handling in the EC2 driver

To unify extension arguments for handling security groups between drivers, `ex_securitygroup` argument in the EC2 `create_node` method has been renamed to `ex_security_groups`.

For backward compatibility reasons, old argument will continue to work for until a next major release.

CloudFiles Storage driver changes

`CLOUDFILES_US` and `CLOUDFILES_UK` provider constants have been deprecated and a new `CLOUDFILES` constant has been added.

User can now use this single constant and specify which region to use by passing `region` argument to the driver constructor.

Old code:

```python
from libcloud.storage.types import Provider
from libcloud.storage.providers import get_driver

cls1 = get_driver(Provider.CLOUDFILES_US)
cls2 = get_driver(Provider.CLOUDFILES_UK)
driver1 = cls1('username', 'api_key')
driver2 = cls1('username', 'api_key')
```

New code:

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

cls = get_driver(Provider.CLOUDFILES)
driver1 = cls('username', 'api_key', region='dfw')
driver2 = cls('username', 'api_key', region='lon')
```

Rackspace DNS driver changes

Rackspace DNS driver has moved to one class plus `region` argument model. As such, the following provider constants have been deprecated:
• RACKSPACE_US
• RACKSPACE_UK

And replaced with a single constant:

• RACKSPACE - Supported values for region arguments are us, uk. Default value is us.

Old code:

```python
from libcloud.dns.types import Provider
from libcloud.dns.providers import get_driver
cls1 = get_driver(Provider.RACKSPACE_US)
cls2 = get_driver(Provider.RACKSPACE_UK)
driver1 = cls1('username', 'api_key')
driver2 = cls1('username', 'api_key')
```

New code:

```python
from libcloud.dns.types import Provider
from libcloud.dns.providers import get_driver
cls = get_driver(Provider.RACKSPACE)
driver1 = cls1('username', 'api_key', region='us')
driver2 = cls1('username', 'api_key', region='uk')
```

**Rackspace load balancer driver changes**

Rackspace loadbalancer driver has moved to one class plus region argument model. As such, the following provider constants have been deprecated:

• RACKSPACE_US
• RACKSPACE_UK

And replaced with a single constant:

• RACKSPACE - Supported values for region arguments are dfw, ord, iad, lon, syd, hkg. Default value is dfw.

Old code:

```python
from libcloud.loadbalancer.types import Provider
from libcloud.loadbalancer.providers import get_driver
cls1 = get_driver(Provider.RACKSPACE_US)
cls2 = get_driver(Provider.RACKSPACE_UK)
driver1 = cls1('username', 'api_key')
driver2 = cls1('username', 'api_key')
```

New code:

```python
from libcloud.loadbalancer.types import Provider
from libcloud.loadbalancer.providers import get_driver
cls = get_driver(Provider.RACKSPACE)
```
driver1 = cls1('username', 'api_key', region='ord')
driver2 = cls1('username', 'api_key', region='lon')

ScriptDeployment and ScriptFileDeployment constructor now takes args argument

libcloud.compute.deployment.ScriptDeployment and libcloud.compute.deployment.ScriptFileDeployment class constructor now take args as a second argument.

Previously this argument was not present and the second argument was name.

If you have a code which instantiate those classes directly and passes two or more arguments (not keyword arguments) to the constructor you need to update it to preserve the old behavior.

Old code:

sd = ScriptDeployment('#!/usr/bin/env bash echo "ponies!"', 'ponies.sh')

New code:

sd = ScriptDeployment('#!/usr/bin/env bash echo "ponies!"', None, 'ponies.sh')

Even better (using keyword arguments):

sd = ScriptDeployment(script='#!/usr/bin/env bash echo "ponies!"', name='ponies.sh')

Pricing data changes

By default this version of Libcloud tries to read pricing data from the ~/.libcloud/pricing.json file. If this file doesn’t exist, Libcloud falls back to the old behavior and the pricing data is read from the pricing file which is shipped with each release.

For more information, please see using-custom-pricing-file page.

RecordType ENUM value is now a string

libcloud.dns.types.RecordType ENUM value used be an integer, but from this version on, it’s now a string. This was done to make it simpler and remove unnecessary indirection.

If you use RecordType class in your code as recommended, no changes are required, but if you use integer values directly, you need to update your code to use RecordType class otherwise it will break.

OK:

# ...
record = driver.create_record(name=www, zone=zone, type=RecordType.A, data='127.0.0.1')

Not OK:

# ...
record = driver.create_record(name=www, zone=zone, type=0, data='127.0.0.1')
Libcloud 0.8

- `restart_node` method has been removed from the OpenNebula compute driver, because OpenNebula OCCI implementation does not support a proper restart method.
- `ex_save_image` method in the OpenStack driver now returns a NodeImage instance.

For a full list of changes, please see the CHANGES file.

Libcloud 0.7

- For consistency, `public_ip` and `private_ip` attribute on the Node object have been renamed to `public_ips` and `private_ips` respectively.

In 0.7 you can still access those attributes using the old way, but this option will be removed in the next major release.

Note: If you have places in your code where you directly instantiate a “Node“ class, you need to update it.

Old code:

```python
node = Node(id='1', name='test node', state=NodeState.PENDING,
private_ip=['10.0.0.1'], public_ip=['88.77.66.77'],
driver=driver)
```

Updated code:

```python
node = Node(id='1', name='test node', state=NodeState.PENDING,
private_ips=['10.0.0.1'], public_ips=['88.77.66.77'],
driver=driver)
```

- Old deprecated paths have been removed. If you still haven’t updated your code you need to do it now, otherwise it won’t work with 0.7 and future releases.

Bellow is a list of old paths and their new locations:

- `libcloud.base` -> `libcloud.compute.base`
- `libcloud.deployment` -> `libcloud.compute.deployment`
- `libcloud.drivers.*` -> `libcloud.compute.drivers.*`
- `libcloud.ssh` -> `libcloud.compute.ssh`
- `libcloud.types` -> `libcloud.compute.types`
- `libcloud.providers` -> `libcloud.compute.providers`

In the contrib/ directory you can also find a simple bash script which can perform a search and replace for you - `migrate_paths.py`.

For a full list of changes, please see the CHANGES file.

Libcloud 0.6

- SSL certificate verification is now enabled by default and an exception is thrown if CA certificate files cannot be found.

To revert to the old behavior, set `libcloud.security.VERIFY_SSL_CERT STRICT` variable to `False`:

```python
libcloud.security.VERIFY_SSL_CERT STRICT = False
```
Note: You are strongly discouraged from disabling SSL certificate validation. If you disable it and no CA certificates files are found on the system you are vulnerable to a man-in-the-middle attack.

More information on how to acquire and install CA certificate files on different operating systems can be found on the SSL Certificate Validation page.

- OpenStack driver now defaults to using OpenStack 1.1 API.

To preserve the old behavior and use OpenStack 1.0 API, pass `api_version='1.0'` keyword argument to the driver constructor.

For example:

```python
cls = get_provider(Provider.OPENSTACK)
driver = cls('user_name', 'api_key', False, 'host', 8774, api_version='1.0')
```

- OpenNebula driver now defaults to using OpenNebula 3.0 API.

To preserve the old behavior and use OpenNebula 1.4 API, pass `api_version='1.4'` keyword argument to the driver constructor.

For example:

```python
cls = get_provider(Provider.OPENNEBULA)
driver = cls('key', 'secret', api_version='1.4')
```

For a full list of changes, please see the CHANGES file.

### 1.1.9 Troubleshooting

This page contains various tips which can help you troubleshoot and debug code with interfaces with libcloud.

#### Debugging

**Note:** If you are sharing debug output on our IRC channel or issue tracker using Pastebin, Github Gists or a similar service, make sure to remove your credentials and any other data you consider private from the output.

Libcloud has a special debug mode which when enabled outputs all the outgoing HTTP requests and all the incoming HTTP responses. Output also includes cURL commands which can be used to re-produce the requests.

When this mode is enabled and if the `paramiko` library is installed (used for deployment), paramiko library log level is also set to `DEBUG` which helps with debugging the deployment issues.

To enable it, set `LIBCLOUD_DEBUG` environment variable and make it point to a file where the output should be saved.

For example if you want the output to be logged to the standard error (on Unix) you can set it to `/dev/stderr`:

```
LIBCLOUD_DEBUG=/dev/stderr python my_script.py
```

**Example output:**

```bash
# -------- begin 4431824872 request ---------
# -------- begin 4431824872:4431825232 response ---------
HTTP/1.1 200 OK
X-Amz-Id-2: 1234
Server: AmazonS3
```
1.1.10 Frequently Asked Questions (FAQ)

Why are the block storage related management methods located in the compute API?

Block storage related management methods are located in the compute API because in most cases block storage API is tightly coupled with the compute API meaning that you can’t manage block storage independent of the compute API.

This also makes sense because in most cases you are only interested in attaching or detaching volumes from and to the compute nodes.

What are the extension methods and arguments?

Libcloud acts as a lowest common denominator and exposes a unified base API which allows you to work with many different cloud providers through a single code base.

Being a lowest common denominator by definition means that not all of the functionality offered by different cloud service providers is available through a base API.

Libcloud solves this problem and allows user to access provider specific functionality through a so called extension methods and arguments. Extension methods and arguments are all the methods and arguments which are prefixed with **ex_**.

Extension methods are there for your convenience, but you should be careful when you use them because they make switching or working with multiple providers harder.

How do I test if provider credentials are valid?

Libcloud makes the whole authentication process transparent to the user. As such, the easiest way to check if provider credentials are valid is by instantiating a driver and calling a method which results in an HTTP call.

If the credentials are valid, method will return a result, otherwise **libcloud.common.types.InvalidCredsError** exception will be thrown.

An example of such method is `libcloud.compute.base.NodeDriver.list_nodes()`. Keep in mind that depending on the account state, list_nodes method might return a lot of data.

If you want to avoid unnecessarily fetching a lot of data, you should find a method specific to your provider which issues a request which results in small amount of data being retrieved.
I want to add a new provider driver what should I do?

For now the best thing to do is to look at an existing driver and test cases for examples.
Libcloud currently supports more than 25 different providers. This means we have a broad range of examples of different APIs and authentication methods. APIs range from simple JSON based REST APIs to SOAP APIs. Authentication methods range from simple shared token and digest auth to HMAC signed requests.

I want to add / propose a new API, what should I do?

We are always open to accepting a now top level API as long as it matches the following criteria:

1. API must be intended to manage an online infrastructure oriented Cloud service
2. Similar service is offered by multiple providers
3. It’s possible to build a common API on top of services provided by different services

Last point is the most important one, because Libcloud acts as a lowest common denominator and can be used with many different providers. Sometimes it doesn’t make sense to build a common Libcloud API even if multiple providers offer a similar service. Usually the case is that the APIs are vastly different and there aren’t enough common points which would allow us to build a cross-provider API which would still provide enough benefit to the end user.

If the API matches the criteria mentioned above you should send a proposal to our mailing list where we can discuss it further. Ideally proposal should also contain a prototype of a driver for at least two different providers. This help with making sure that API you designed is not biased against a single provider.

1.1.11 Registering a third party driver

Driver is considered third party if it’s not bundled with a Libcloud release.

To register a third party driver you should use `provider.set_driver()` function from the corresponding component.

`set_driver()` takes the following arguments:

```
set_driver('provider_name', 'path.to.the.module', 'DriverClass')
```

Keep in mind that the `provider_name` needs to be unique and this function needs to be called before using a third party driver.

For example:

```python
from libcloud.compute.providers import get_driver
from libcloud.compute.providers import set_driver

set_driver('stratuslab',
        'stratuslab.libcloud.stratuslab_driver',
        'StratusLabNodeDriver')

# Your code which uses the driver.
# For example:
driver = get_driver('stratuslab')
```

An example of an existing third party driver can be found at https://github.com/StratusLab/libcloud-drivers
1.1.12 SSL Certificate Validation

When establishing a secure connection to a cloud provider endpoint, Libcloud verifies server SSL certificate. By default, Libcloud searches paths listed in `libcloud.security.CA_CERTS_PATH` for CA certificate files.

`CA_CERTS_PATH` contains common paths to CA bundle installations on the following platforms:

- openssl on CentOS / Fedora
- ca-certificates on Debian / Ubuntu / Arch / Gentoo
- ca_root_nss on FreeBSD
- curl-ca-bundle on Mac OS X

If no valid CA certificate files are found, you will see an error message similar to the one below:

No CA Certificates were found in CA_CERTS_PATH.

Acquiring CA Certificates

If the above packages are unavailable to you, and you don’t wish to roll your own, the makers of cURL provides an excellent resource, generated from Mozilla: [http://curl.haxx.se/docs/caextract.html](http://curl.haxx.se/docs/caextract.html).

1.1.13 Using Libcloud in multi-threaded and async environments

Libcloud’s primary task is to communicate with different provider APIs using HTTP. This means most of the work is not CPU intensive, but performing all those HTTP requests includes a lot of waiting which makes the library I/O bound.

Most of the time you want to perform more operations in parallel or just want your code to finish faster (for example starting a lot of servers or periodically polling for node status).

Problems like this are usually solved using threads or async libraries such as Twisted, Tornado or gevent.

This page contains some information and tips about how to use Libcloud in such environments.

Libcloud and thread-safety

Important thing to keep in mind when dealing with threads is thread-safety. Libcloud driver instance is not thread safe. This means if you don’t want to deal with complex (and usually inefficient) locking the easiest solution is to create a new driver instance inside each thread.

Using Libcloud with gevent

gevent has an ability to monkey patch and replace functions in the Python `socket, urllib2, httplib` and `time` module with its own functions which don’t block.

You need to do two things when you want to use Libcloud with gevent:

- Enable monkey patching

```python
from gevent import monkey
monkey.patch_all()
```

- Create a separate driver instance for each Greenlet. This is necessary because a driver instance reuses the same Connection class.
For an example see Efficiently download multiple files using gevent.

Using Libcloud with Twisted

Libcloud has no Twisted support included in the core which means you need to be careful when you use it with Twisted and some other async frameworks.

If you don’t use it properly it can block the whole reactor (similar as any other blocking library or a long CPU-intensive task) which means the execution of other pending tasks in the event queue will be blocked.

A simple solution to prevent blocking the reactor is to run Libcloud calls inside a thread. In Twisted this can be achieved using `threads.deferToThread` which runs a provided method inside the Twisted thread pool.

The example below demonstrates how to create a new node inside a thread without blocking the whole reactor.

```python
from __future__ import absolute_import
from pprint import pprint
from twisted.internet import defer, threads, reactor
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

@defer.inlineCallbacks
def create_node(name):
    node = yield threads.deferToThread(_thread_create_node, name=name)
    pprint(node)
    reactor.stop()

def _thread_create_node(name):
    Driver = get_driver(Provider.RACKSPACE)
    conn = Driver('username', 'api key')
    image = conn.list_images()[0]
    size = conn.list_sizes()[0]
    node = conn.create_node(name=name, image=image, size=size)
    return node

def stop(*args, **kwargs):
    reactor.stop()

d = create_node(name='my-lc-node')
d.addCallback(stop)
d.addErrback(stop)
reactor.run()
```

1.1.14 Working with the object oriented APIs

To make it easier for the end user, Libcloud components expose a fully object-oriented API.

This means that besides the driver object you also work with `NodeImage`, and `NodeSize` object in the compute API, `Container` and `Object` object in the Storage API, `Zone` and `Record` object in the DNS API and so on.
Methods which operate on those resources usually require you to pass in an instance of the resource you want to manipulate or work with and not just an id.

To obtain a reference to this resource, Libcloud providers corresponding get and / or list methods.

A couple of examples are shown bellow.

**Example 1 - listing records for a zone with a known id**

```python
from libcloud.dns.providers import get_driver
from libcloud.dns.types import Provider

CREDENTIALS_ZERIGO = ('email', 'api key')
ZONE_ID = 'example.myzone.com'

Cls = get_driver(Provider.ZERIGO)
driver = Cls(*CREDENTIALS_ZERIGO)

zone = driver.get_zone(zone_id=ZONE_ID)
records = driver.list_records(zone=zone)
```

In this example, `driver.get_zone()` method call results in an HTTP call.

**Example 2 - creating an EC2 instance with a known NodeSize and NodeImage id**

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

ACCESS_ID = 'your access id'
SECRET_KEY = 'your secret key'

IMAGE_ID = 'ami-c8052d8d'
SIZE_ID = 't1.micro'

cls = get_driver(Provider.EC2_US_WEST)
driver = cls(ACCESS_ID, SECRET_KEY)

# Here we select size and image
sizes = driver.list_sizes()
images = driver.list_images()

size = [s for s in sizes if s.id == SIZE_ID][0]
image = [i for i in images if i.id == IMAGE_ID][0]

node = driver.create_node(name='test-node', image=image, size=size)
```

In this example, both `driver.list_sizes()` and `driver.list_images()` method calls result in an HTTP call.

As you can see above, most of those getter methods retrieve extra information about the resource from the provider API and result in an HTTP request.

There are some cases when you might not want this:

- You don’t care if a resource doesn’t exist
- You don’t care about the extra attributes
- You want to avoid an extra HTTP request

1.1. Main
• You want to avoid holding a reference to the resource object

If that is true for you, you can directly instantiate a resource with a known id. You can see how to do this in the examples bellow.

Example 1 - listing records for a zone with a known id

```python
from libcloud.dns.base import Zone
from libcloud.dns.providers import get_driver
from libcloud.dns.types import Provider

CREDENTIALS_ZERIGO = ('email', 'api key')
ZONE_ID = 'example.myzone.com'

Cls = get_driver(Provider.ZERIGO)
driver = Cls(*CREDENTIALS_ZERIGO)

zone = Zone(ZONE_ID, domain=None, type=None, ttl=None, driver=driver)
records = driver.list_records(zone=zone)
```

Example 2 - creating an EC2 instance with a known NodeSize and NodeImage id

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver
from libcloud.compute.base import NodeSize, NodeImage

ACCESS_ID = 'your access id'
SECRET_KEY = 'your secret key'

IMAGE_ID = 'ami-c8052d8d'
SIZE_ID = 't1.micro'

cls = get_driver(Provider.EC2_US_WEST)
driver = cls(ACCESS_ID, SECRET_KEY)

size = NodeSize(id=SIZE_ID, name=None, ram=None, disk=None, bandwidth=None, price=None, driver=driver)
image = NodeImage(id=IMAGE_ID, name=None, driver=driver)

node = driver.create_node(name='test-node', image=image, size=size)
```

Example 3 - creating an EC2 instance with an IAM profile

```python
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

ACCESS_ID = 'your access id'
SECRET_KEY = 'your secret key'
IAM_PROFILE = 'your IAM profile arn or IAM profile name'

IMAGE_ID = 'ami-c8052d8d'
SIZE_ID = 't1.micro'

cls = get_driver(Provider.EC2_US_WEST)
```
driver = cls(ACCESS_ID, SECRET_KEY)

# Here we select size and image
sizes = driver.list_sizes()
images = driver.list_images()

size = [s for s in sizes if s.id == SIZE_ID][0]
image = [i for i in images if i.id == IMAGE_ID][0]

node = driver.create_node(name='test-node', image=image, size=size,
                          ex_iamprofile=IAM_PROFILE)

1.2 Developer Information

1.2.1 Developer Information

Mailing Lists

All of the communication about Libcloud development happens on our mailing lists.

- users@libcloud.apache.org - Mailing list for general talk about Libcloud and other off-topic things (users-archive)
- dev@libcloud.apache.org - General mailing list for developers (dev-archive)
- commits@libcloud.apache.org - Commits messages go to this mailing list. Keep in mind that unlike the others, this mailing list is fairly noisy. (commits-archive)

Archive of old incubator mailing lists:

- incubator-libcloud
- incubator-libcloud-commits

Issue Tracker

For bug and issue tracking we use JIRA located at https://issues.apache.org/jira/browse/LIBCLOUD.

Testing

For information how to run the tests and how to generate the test coverage report, please see the Testing page.

Continuous Integration

For continuous integration we use Apache buildbot instance and Travis-CI. You can find build reports on the following two links:

- https://travis-ci.org/apache/libcloud

Travis-CI builder is also integrated with Github which means that if you open a pull request there, Travis-CI will automatically build it.
Test Coverage

Test coverage report is automatically generated after every push and can be found at http://ci.apache.org/projects/libcloud/coverage.

1.2.2 Development

Contributing

We welcome contributions of any kind (ideas, code, tests, documentation, examples, ...). This page explains how you can contribute to the Libcloud project. If you get stuck at any point during this process, stop by on our IRC channel (#libcloud on freenode) and we will do our best to assist you.

Style guide

• We follow PEP8 Python Style Guide
• Use 4 spaces for a tab
• Use 79 characters in a line
• Make sure edited file doesn’t contain any trailing whitespace
• You can verify that your modifications don’t break any rules by running the flake8 script - e.g. flake8 libcloud/edited_file.py. or tox -e lint. Second command fill run flake8 on all the files in the repository.

Git pre-commit hook

To make complying with our style guide easier, we provide a git pre-commit hook which automatically checks modified Python files for violations of our style guide.

You can install it by running following command in the root of the repository checkout:

```
ln -s contrib/pre-commit.sh .git/hooks/pre-commit
```

After you have installed this hook it will automatically check modified Python files for violations before a commit. If a violation is found, commit will be aborted.

General guidelines

• Any non-trivial change must contain tests
• All the functions and methods must contain Sphinx docstrings which are used to generate API documentation. You can find a lot of examples of docstrings in the existing code e.g. libcloud/compute/base.py
• If you are adding a new feature, make sure to add corresponding documentation

Contribution workflow

1. Start a discussion on the mailing list

If you are implementing a big feature or a change, start a discussion on the mailing list first.
2. Open a new issue on our issue tracker

Go to our issue tracker and open a new issue for your changes there. This issue will be used as an umbrella place for your changes. As such, it will be used to track progress and discuss implementation details.

3. Fork our Github repository

Fork our Github git repository. Your fork will be used to hold your changes.

4. Create a new branch for your changes

For example:

```
git checkout -b <jira_issue_id>_<change_name>
```

5. Make your changes

6. Write tests for your changes and make sure all the tests pass

Make sure that all the code you have added or modified has appropriate test coverage. Also make sure all the tests including the existing ones still pass.

For more information on how to write and run tests, please see Testing page.

7. Commit your changes

Make a single commit for your changes. If a corresponding JIRA ticket exists, make sure the commit message contains the ticket number.

For example:

```
git commit -m "<change description>"
```

8. Open a pull request with your changes


Make sure the pull request name is prefixed with a JIRA ticket number, e.g. [LIBCLOUD-436] Improvements to DigitalOcean compute driver and that the pull request description contains link to the JIRA ticket.

9. Wait for the review

Wait for your changes to be reviewed and address any outstanding comments.

10. Attach a final patch with your changes to the corresponding JIRA ticket

Once the changes has been reviewed, all the outstanding issues have been addressed and the pull request has been +1’ed, close the pull request, generate a patch and attach it to the JIRA issue you have created earlier.
git format-patch --stdout trunk > patch_name.patch

Note about Github

Github repository is a read-only mirror of the official Apache git repository (https://git-wip-us.apache.org/repos/asf/libcloud.git). This mirror script runs only a couple of times per day which means this mirror can be slightly out of date.

You are advised to add a separate remote for the official upstream repository:

git remote add upstream https://git-wip-us.apache.org/repos/asf/libcloud.git

Github read-only mirror is used only for pull requests and code review. Once a pull request has been reviewed, all the comments have been addresses and it’s ready to be merged, user who submitted the pull request must close the pull request, create a patch and attach it to the original JIRA ticket.

Syncing your git(hub) repository with an official upstream git repository

This section describes how to synchronize your git clone / Github fork with an official upstream repository.

It’s important that your repository is in-sync with the upstream one when you start working on a new branch and before you generate a final patch. If the repository is not in-sync, generated patch will be out of sync and we won’t be able to cleanly merge it into trunk.

To synchronize it, follow the steps bellow in your git clone:

1. Add upstream remote if you haven’t added it yet

git remote add upstream https://git-wip-us.apache.org/repos/asf/libcloud.git

2. Synchronize your trunk branch with an upstream one

git checkout trunk

git pull upstream trunk

3. Create a branch for your changes and start working on it

git checkout -b my_new_branch

4. Before generating a final patch which is to be attached to the JIRA ticket, make sure your repository and branch is still in-sync

git pull upstream trunk

5. Generate a patch which can be attached to the JIRA ticket

git format-patch --stdout remotes/upstream/trunk > patch_name.patch

Contributing Bigger Changes

If you are contributing a bigger change (e.g. large new feature or a new provider driver) you need to have have signed Apache Individual Contributor License Agreement (ICLA) in order to have your patch accepted.

You can find more information on how to sign and file an ICLA on the Apache website.

When filling the form, leave field preferred Apache id(s) empty and in the notify project field, enter Libcloud.
Supporting Multiple Python Versions

Libcloud supports a variety of Python versions so your code also needs to work with all the supported versions. This means that in some cases you will need to include extra code to make sure it works in all the supported versions.

Some examples which show how to handle those cases are described below.

Context Managers

Context managers aren’t available in Python 2.5 by default. If you want to use them make sure to put from __future__ import with_statement on top of the file where you use them.

Exception Handling

There is no unified way to handle exceptions and extract the exception object in Python 2.5 and Python 3.x. This means you need to use a sys.exc_info()[1] approach to extract the raised exception object.

For example:

```python
try:
    some code
except Exception:
    e = sys.exc_info()[1]
    print e
```

Utility functions for cross-version compatibility

You can find a lot of utility functions which make code easier to work with Python 2.x and 3.x in libcloud.utils.py3 module.

You can find some more information on changes which are involved in making the code work with multiple versions on the following link - Lessons learned while porting Libcloud to Python 3

1.3 Committer Guide

1.3.1 Committer Guide

New committer guidelines

Note: This section outlines steps which need to be completed by new team members and is intended for people who have been voted to join Libcloud project as a committer and / or PMC member. It also assumes you have already filled your ICLA and your Apache account has been created.

First congratulations and welcome to the team!
1. Subscribe to the public mailing lists

If you haven’t yet, subscribe to {dev,users,commits}@apache.libcloud.org mailing lists. Commits mailing list is especially important because all of the JIRA notification, Github Pull Request notifications and build notifications are sent there.

2. Subscribe to the private mailing list

Subscribe to private@libcloud.apache.org by sending an email to private-subscribe@libcloud.apache.org. Keep in mind that this list is private and your subscription needs to be approved by a moderator.

3. Create PyPi account

Creating a PyPi account by going to https://pypi.python.org/pypi?%3Aaction=register_form and send an email to private@libcloud.apache.org requesting to be added as a maintainer to apache-libcloud package. Make sure to select a strong and unique password (pwgen -s 48 1 is your friend).

After you have registered go to “Your details” page and populate PGP Key ID field with your PGP key ID.

Applying a patch

When applying a third-party patch created using git format-patch command, use the following command:

```
git am --signoff < patch_name.patch
```

--signoff argument signs the patch and lets others know that you have reviewed and merged a patch.

After the patch has been applied, make sure to update CHANGES file.

Making a release (for release managers)

1. Pre-release check list
   - Make sure tests pass on all the supported Python versions (tox)
   - Make sure CHANGES file is up to date
   - Make sure __version__ string in libcloud/__init__.py is up to date

2. Creating release artifacts

We have a script that runs the required setup.py commands and then hashes and signs the files. To run it:

```
cd dist
./release.sh -u <yourusername>@apache.org
```

-u argument will be used to find a key with a matching email address in your local GPG database.

This should result in a set of apache-libcloud-${VERSION}.{tar.bz2,tar.gz,zip}{,asc,md5,sha1} files that are suitable to be uploaded for a release.

3. Uploading release artifacts to Apache servers

   - Add release artifacts to the dist SVN repository at https://dist.apache.org/repos/dist/release/libcloud/

   It may take up to a day for the artifacts to be available on all the Apache mirrors, but they should be instantly available at http://www.apache.org/dist/libcloud/.
• If there is more than one older release in the repository, delete the oldest one. Old releases are automatically archived and available at https://dist.apache.org/repos/dist/release/libcloud/.

4. Tagging a release

```
git tag <tag> <commit hash>
```

For example:

```
git tag v0.13.0 105b9610835f99704996d861613c5a9a8b3f8b1
```

5. Publishing package to PyPi

For consistency and security reasons packages are always uploaded to PyPi manually using the web interface and not using the `setup.py upload` command.

• Run `python setup.py register upload` command. This will register a new version on PyPi, but it won’t upload the actual release artifacts.

• Go to the PyPi release management page, find a new release and click on “files” link.

• Once you are there, upload all the release artifacts (.tar.bz2, .tar.gz and .zip). For File Type select Source and for Python Version select Any (ie. pure Python). Make sure to also select and upload a PGP signature for each file (PGP signature (.asc) field).

Once all the files have been uploaded, the page should look similar to the screenshot below.

7. Updating doap_libcloud.rdf file

Add information about the new release to the `doap_libcloud.rdf` file in the root of the main code repository.
8. Updating website
   • Update “News” page (content/news.mdtext file)
   • Update “Downloads” page (content/downloads.mdtext file)
   • Update “Get it” section in the sidebar (templates/blocks/other.html file)

9. Sending announcements
   • Send a release announcement to {dev,users}@libcloud.apache.org. If it’s a major release also send it to announce@apache.org.
   • Send a release announcement to Twitter and Google+

If needed, use Apache URL shortening service - http://s.apache.org/

Mailing list email templates

This section includes email templates which can be used when sending out official communication to the mailing lists.

Release voting thread template

This template should be used when starting a voting thread for a new release.

Subject:

[VOTE] Release Apache Libcloud <version>

Body:

This is a voting thread for Libcloud <version>.

<short description and highlights of this release>

Full changelog can be found at <link to the CHANGES file for this release>.

Release artifacts can be found at <link to your Apache space where a release artifacts can be found>.

Please test the release and post your votes.

+/- 1
[ ] Release Apache Libcloud <version>

Vote will be opened until <date, it should be at minimum today + 3 days> (or longer, if needed).

Thanks,

<name>

For example:

Subject:

[VOTE] Release Apache Libcloud 0.13.2

Body:
This is a voting thread for Libcloud 0.13.2.

This is another primarily a bug-fix release. Previous release included a fix for the Content-Length bug which didn't fully fix the original issue. It missed out "raw" requests which are fixed in this release (LIBCLOUD-396).

Besides this bug fix, it includes a couple of other smaller bug fixes and changes. Full change log can be found at https://git-wip-us.apache.org/repos/asf?p=libcloud.git;a=blob;f=CHANGES;h=b7747f777a257f82fe52a3f2efe83a00d5137276;hb=c4b3daae946049652a500a8515929b4cbf14a6b4

Release artifacts can be found at http://people.apache.org/~tomaz/libcloud/.

Please test the release and post your votes.

+/- 1

[ ] Release Apache Libcloud 0.13.2

Vote will be opened until September 18th, 2013 (or longer, if needed).

Thanks,
Tomaz

Release announcement

This template should be used when sending out a release announcement.

Subject:

[ANNOUNCE] Apache Libcloud 0.13.1 release

Body:

Libcloud is a Python library that abstracts away the differences among multiple cloud provider APIs. It allows users to manage cloud services (servers, storage, loadbalancers, DNS) offered by many different providers through a single, unified and easy to use API.

We are pleased to announce the release of Libcloud <version>!

<short description of the release which should include release highlights>

Full change log can be found at <link to CHANGES file for this release>

Download

Libcloud <version> can be downloaded from http://libcloud.apache.org/downloads.html or installed using pip:

```
pip install apache-libcloud
```

It is possible that the file hasn't been synced to all the mirrors yet. If this is the case, please use the main Apache mirror - http://www.apache.org/dist/libcloud.

Upgrading

If you have installed Libcloud using pip you can also use it to upgrade it:

```
pip install --upgrade apache-libcloud
```
Upgrade notes

A page which describes backward incompatible or semi-incompatible changes and how to preserve the old behavior when this is possible can be found at http://libcloud.apache.org/upgrade-notes.html.

Documentation

API documentation can be found at http://libcloud.apache.org/apidocs/<version>/.

We also have a new Sphinx documentation which can be found at https://libcloud.apache.org/docs/.

Bugs / Issues

If you find any bug or issue, please report it on our issue tracker <https://issues.apache.org/jira/browse/LIBCLOUD>. Don't forget to attach an example and / or test which reproduces your problem.

Thanks

Thanks to everyone who contributed and made this release possible! Full list of people who contributed to this release can be found in the CHANGES file <link to the changes file for this release>.

For example:

Subject:

[ANNOUNCE] Apache Libcloud 0.13.1 release

Body:

Libcloud is a Python library that abstracts away the differences among multiple cloud provider APIs. It allows users to manage cloud services (servers, storage, loadbalancers, DNS) offered by many different providers through a single, unified and easy to use API.

We are pleased to announce the release of Libcloud 0.13.1!

This is a bug-fix only release. Among some smaller bugs it also fixes Content-Length regression which broke create and update operations in the Bluebox Compute and Azure Storage driver (LIBCLOUD-362, LIBCLOUD-3901).

Full change log can be found at <https://git-wip-us.apache.org/repos/asf?p=libcloud.git;a=blob;f=CHANGES;h=ca90c84e296ca82e2206eb86ed7364c588aad503;hb=602b6a7a27dca6990a38eb887e1d6615826387d5>

Download

Libcloud 0.13.1 can be downloaded from http://libcloud.apache.org/downloads.html or installed using pip:

    pip install apache-libcloud

It is possible that the file hasn't been synced to all the mirrors yet. If this is the case, please use the main Apache mirror - http://www.apache.org/dist/libcloud.

Upgrading

If you have installed Libcloud using pip you can also use it to upgrade it:
pip install --upgrade apache-libcloud

Upgrade notes

A page which describes backward incompatible or semi-incompatible changes and how to preserve the old behavior when this is possible can be found at http://libcloud.apache.org/upgrade-notes.html.

Documentation

API documentation can be found at http://libcloud.apache.org/apidocs/0.13.1/.

We also have a new Sphinx documentation which can be found at https://libcloud.apache.org/docs/. Keep in mind though, that this documentation reflects state in trunk which includes some backward incompatible changes which aren't present in 0.13.1. All the examples in the documentation which only work with trunk are clearly marked with a note.

Bugs / Issues

If you find any bug or issue, please report it on our issue tracker <https://issues.apache.org/jira/browse/LIBCLOUD>. Don't forget to attach an example and / or test which reproduces your problem.

Thanks

Thanks to everyone who contributed and made this release possible! Full list of people who contributed to this release can be found in the CHANGES file <https://git-wip-us.apache.org/repos/asf?p=libcloud.git;a=blob;f=CHANGES;h=ca90c84e296ca82e2206eb86ec8f9f5082e154bc;hb=602b6a7a27dca6990a38eb887e1d6615826387d5>.

1.4 Other

1.4.1 Security

Reporting a vulnerability

Note: Please do not report security issues using our public JIRA instance. Use the private mailing list described bellow.

If you believe you found a security issue or a vulnerability, please send a description of it to our private mailing list at security@libcloud.apache.org You are also encouraged to encrypt this email using PGP. Keys of our developers can be found at https://www.apache.org/dist/libcloud/KEYS.

Once you’ve submitted an issue, you should receive an acknowledgment from one of our team members in 48 hours or less. If further action is necessary, you may receive additional follow-up emails.

How are vulnerabilities handled?

We follow a standard Apache Software Foundation vulnerability handling process which is described at http://www.apache.org/security/committers.html#vulnerability-handling

1.4. Other
Note: Unless noted otherwise, all of the examples and code snippets in the documentation are licensed under the Apache 2.0 license.