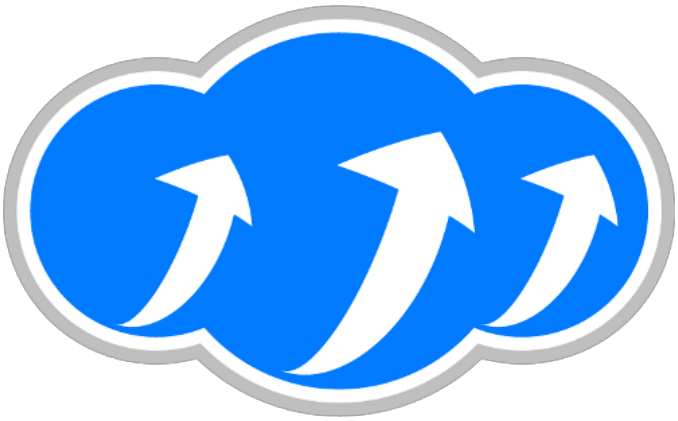


Avoiding Vendor Lock-In Using Apache Libcloud

Tomaz Muraus
tomaz@apache.org
Cloud Open 2012, San Diego, CA

Agenda

- Who am I?

- What is Libcloud?
- Why?
- History
- Project info
- Libcloud APIs
- What is currently going on & plans for the future
- Questions

Who am I?

- Tomaz Muraus, @KamiSLO, +Tomaz Muraus
- Github: github.com/Kami
- Libcloud Project chair & committer
- Engineer at Rackspace

- Dude who likes open standards and open source

What is Libcloud?

"Libcloud is a Python library which abstracts differences between cloud provider APIs and allows users to manage their cloud resources (servers, storage, load balancers, DNS) using a unified and easy to use interface."

What is Libcloud?

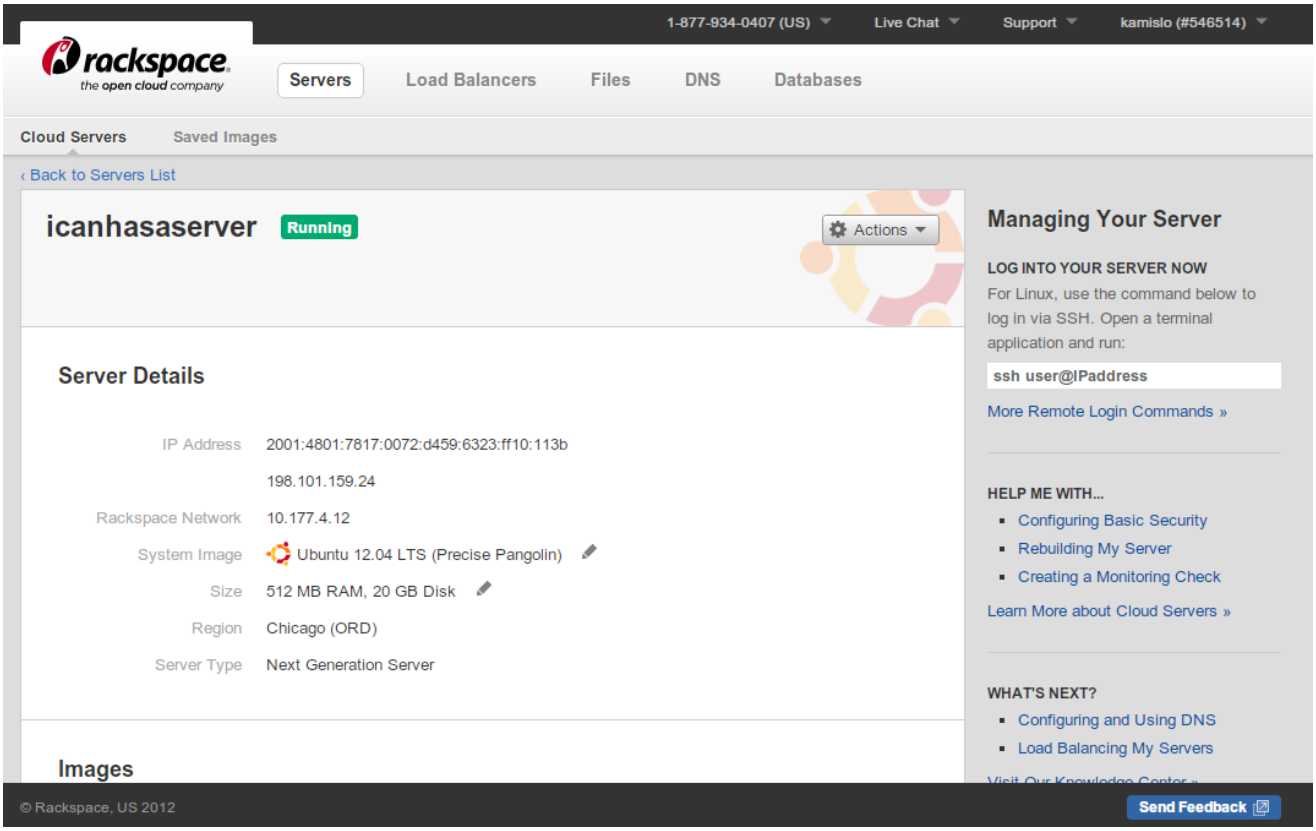
Turns this:

```
1  from libcloud.compute.types import Provider
2  from libcloud.compute.providers import get_driver
3
4  Cls = get_driver(Provider.RACKSPACE)
5  driver = Cls('username', 'api key')
6
7  size = driver.list_sizes()[0]
8  images = driver.list_images()[0]
9
10 node = driver.create_node(name='icanhasaserver', size=size,
11 image=image)
```

?

What is Libcloud?

Into this:



Why?

- Cloud interoperability and standards are (mostly) a lie
- Need for standardization
  - Different APIs
  - Different response formats (XML, JSON, text)
  - Different authentication methods
  - Different request signing mechanisms

Why - different response formats (XML)

```
<DescribeInstancesResponse
```

```
xmlns="http://ec2.amazonaws.com/doc/2010-08-31/">
  <requestId>56d0fffa-8819-4658-bdd7-548f143a86d2</requestId>
  <reservationSet>
    <item>
      <reservationId>r-07adf66e</reservationId>
      <instancesSet>
        <item>
          <instanceId>i-4382922a</instanceId>
          <imageId>ami-0d57b264</imageId>
          <instanceState>
            <code>0</code>
            <name>pending</name>
          </instanceState>
          <privateDnsName/>
```

Why - different response formats (text)

```
de:0:0:write:requests 466
rx 760681
vnc:password testpass
ide:0:0 f0202f1c-0b4f-4cfc-8ae3-e30951d09ef0
ide:0:0:read:requests 7467
ide:0:0:read:bytes 165395968
vnc:ip 178.22.66.28
boot ide:0:0
```

```
smp 1
started 1286568422
nic:0:model virtio
status active
user 93b34fd9-7986-4b25-8bfd-98a50383605d
ide:0:0:media disk
name cloudsigma node
```

Why - different response formats (JSON)

```
[
    {
        "ips": [
            {
                "address": "67.214.214.212"
            }
        ],
        "memory": 1073741824,
        "id": "99df878c-6e5c-4945-a635-d94da9fd3146",
        "storage": 21474836480,
        "hostname": "foo.apitest.blueboxgrid.com",
        "description": "1 GB RAM + 20 GB Disk",
        "cpu": 0.5,
        "status": "running"
```

```
    }
]
```

Why - different authentication methods

- Shared token / secret
- HMAC based
- HTTP basic / digest auth
- X509 certificate-based

Apache Libcloud

- Python library (pip install apache-libcloud)
- Originally developed at Cloudkick in 2009
- Later this year project joined Apache Incubator
- Graduated to Apache TLP in May 2011
- Current stable release is 0.11.1
- Similar libraries in other languages:
  - Fog (Ruby)
  - jclouds (Java)
  - deltacloud

Apache Libcloud

- 8 committers
- A decent amount of contributions from the community
- Mailing list: {users,dev}@libcloud.apache.org
- IRC channel: #libcloud on irc.freenode.net

Apache Libcloud - who is using it

- Rackspace

- Server Density
- CollabNet
- Salt Stack

Libcloud APIs

- Compute
- Storage
- DNS
- Load balancers

Libcloud APIs - Supported providers

Libcloud APIs - Compute

- Allows users to manage VMs / cloud servers across more than 25 different providers
- Includes limited support for block storage (EBS, etc.) management
- Supported services include: Amazon EC2, Rackspace Cloud Servers, OpenStack, CloudStack, ...

Libcloud APIs - Compute - Base API

- list_nodes
- list_images
- list_sizes
- list_locations
- create_node
- deploy_node
- reboot_node
- destroy_node

Libcloud APIs - Compute - Example

```
1  import os                                                              ?
2  from libcloud.compute.types import Provider
3  from libcloud.compute.providers import get_driver
4  from libcloud.compute.deployment import MultiStepDeployment, \
5  ScriptDeployment, SSHKeyDeployment
6
7  conn = get_driver(Provider.RACKSPACE)('username', 'key')
8
9  install_key = SSHKeyDeployment(open(os.path.expanduser("~/.ssh/id_rsa.p
10 install_puppet = ScriptDeployment("apt-get -y install puppet")
11
12 msd = MultiStepDeployment([install_key, install_puppet])
13
14 image = conn.list_images()[0]
15 size = conn.list_sizes()[0]
16
17 node = conn.deploy_node(name='test', image=image, size=size,
18 deploy=msd)
```

Libcloud APIs - Storage

- Allows users to manage cloud storage across 5 different

providers

- Supported services include: Amazon S3, Rackspace Cloud Files, Google Storage, OpenStack Swift

Libcloud APIs - Storage - Base API

- list_containers
- list_container_object
- get_container
- get_object

- create_container
- upload_object
- upload_object_via_stream
- download_object
- download_object_as_stream
- delete_container
- delete_object

Libcloud APIs - Storage - Example

```
1  import subprocess                                                    ?
2  from libcloud.storage.types import Provider
3  from libcloud.storage.providers import get_driver
4
5  driver = get_driver(Provider.CLOUDFILES_US)('username', 'key')
6
7  directory = '/home/some/path'
8  cmd = 'tar cvzpf - %s' % (directory)
9  container = driver.create_container('backups')
10
11 pipe = subprocess.Popen(cmd, bufsize=0, shell=True, stdout=subprocess.P
12 return_code = pipe.poll()
13
14 while return_code is None:
15     # Compress data in our directory and stream it directly to CF
16     container.upload_object_via_stream(iterator=pipe.stdout,
17                                        object_name='backup.tar.gz')
18     return_code = pipe.poll()
19 print 'Upload complete'
```

Libcloud APIs - Load balancers

- Allows users to manage cloud load balancers across 4 different providers
- Supported services: Rackspace LoadBalancers, CloudStack LoadBalancers, ...

Libcloud APIs - Load balancers - Base API

- list_protocols
- list_balancers
- balancer_list_members
- get_balancer
- create_balancer
- destroy_balancer
- balancer_attach_member
- balancer_attach_compute_node
- balancer_detach_member

Libcloud APIs - Load balancers - Example

```
1  Cls = get_driver(Provider.RACKSPACE_US)
2  driver = Cls('username', 'api key')
3
4  new_balancer = driver.create_balancer(name='test-lb',
5          algorithm=Algorithm.ROUND_ROBIN, port=80, protocol='http',
6  members=(Member(None, '192.168.86.1', 8080),
7                  Member(None, '192.168.86.2', 8080)))
8
9  # wait for the balancer to become ready
10 while True:
11     balancer = driver.get_balancer(balancer_id=new_balancer.id)
12     if balancer.state == State.RUNNING:
13         break
14     time.sleep(20)
15
16 # fetch list of members
17 members = balancer.list_members()
18 print members
```

Libcloud APIs - DNS

- Allows users to manage DNS across 3 different providers

- Supported services: Zerigo DNS, Rackspace DNS, Linode DNS

Libcloud APIs - DNS - Base API

- list_record_types
- list_zones
- list_records
- get_zone

- get_record
- create_zone
- update_zone
- create_record
- update_record
- delete_zone
- delete_record

Libcloud APIs - DNS - Example

```
1  from pprint import pprint
2
3  from libcloud.compute.providers import get_driver as get_compute_driv
4  from libcloud.compute.types import Provider as ComputeProvider
5  from libcloud.dns.providers import get_driver as get_dns_driver
6  from libcloud.dns.types import Provider as DNSProvider, RecordType
7
8  compute_driver = get_dns_driver(DNSProvider.ZERIGO)('username', 'api
9  dns_driver = get_dns_driver(DNSProvider.ZERIGO)('email', 'password')
10
11 nodes = compute_driver.list_nodes()
12
13 zone = dns_driver.create_zone(domain='mydomain2.com')
14
15 created = []
16 for node in nodes:
17     print 'Creating %s record (data=%s) for node %s' % ('A', ip, name
18     record = zone.create_record(name=node.name, type=RecordType.A,
19                                 data=node.public_ips[0])
```

```
20          created.append(record)
21
22  print 'Done, created %d records' % (len(created))
23  pprint(created)
```

What is currently going on & plans for the future

- Libcloud REST - talk to Libcloud over HTTP
- Improving pricing data distribution
- Revamping the whole “location” concept
- ...

Rackspace (SF) is hiring

- Python, Node.js, Lua, Ruby, Java, ...
- Chef, Cassandra, ZooKeeper, Scribe, ...
- Cloud, Big data, ...
- We <3 open source
- We <3 cloud
- http://rackertalent.com/sanfrancisco/
- http://github.com/racker/
- http://github.com/Rackspace/

Thank you & Questions

- Website: http://libcloud.apache.org
- Docs: http://libcloud.apache.org/docs/
- Mailing lists: {users,dev}@libcloud.apache.org
- IRC: #libcloud on irc.freenode.net