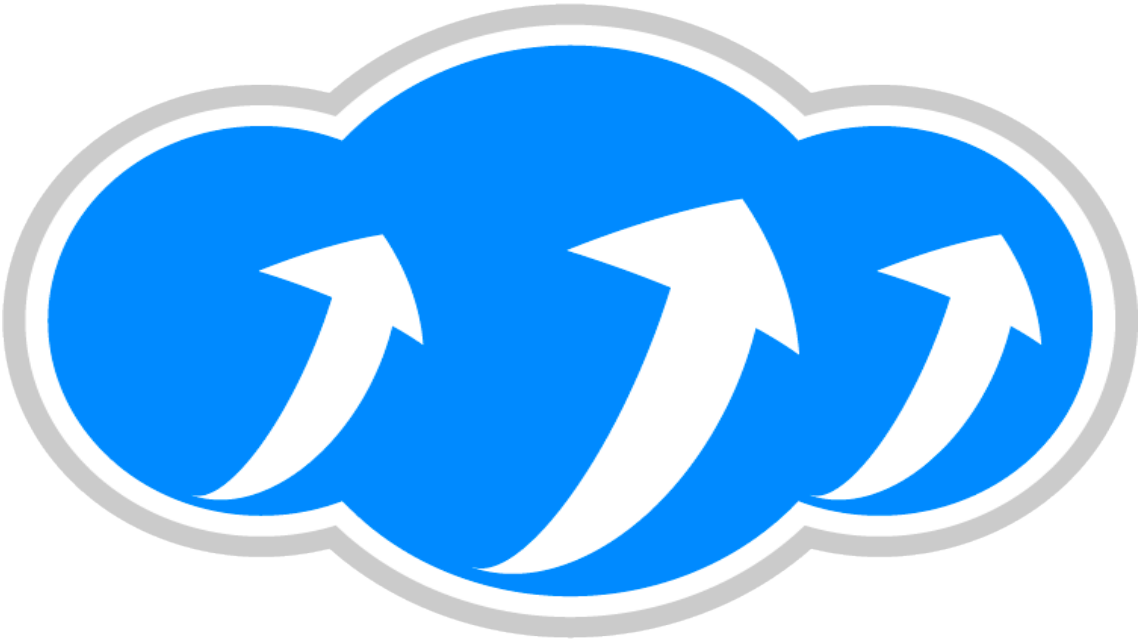


Managing the cloud with Libcloud

Tomaz Muraus
tomaz@apache.org
June 22, 2011

EUROPYTHON 2011
FLORENCE, JUNE 20–26

The Apache Software Foundation
http://www.apache.org/

Agenda

- Who am I
- What is Libcloud?
- Why?
- Libcloud history
- Libcloud APIs

Agenda

- Compute API
- Storage API
- Load Balancer API
- Plans for the future
- Questions

Who am I

- Tomaz Muraus, @KamiSLO
- Software developer at Cloudkick / Rackspace
- Author of multiple Python libraries and Django apps – http://github.com/Kami
- FOSS supporter & lover
- Apache Libcloud Committer

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) using a unif ed and easy to use interface.”

What is Libcloud?

What??

What is Libcloud?

Turns this

What is Libcloud?

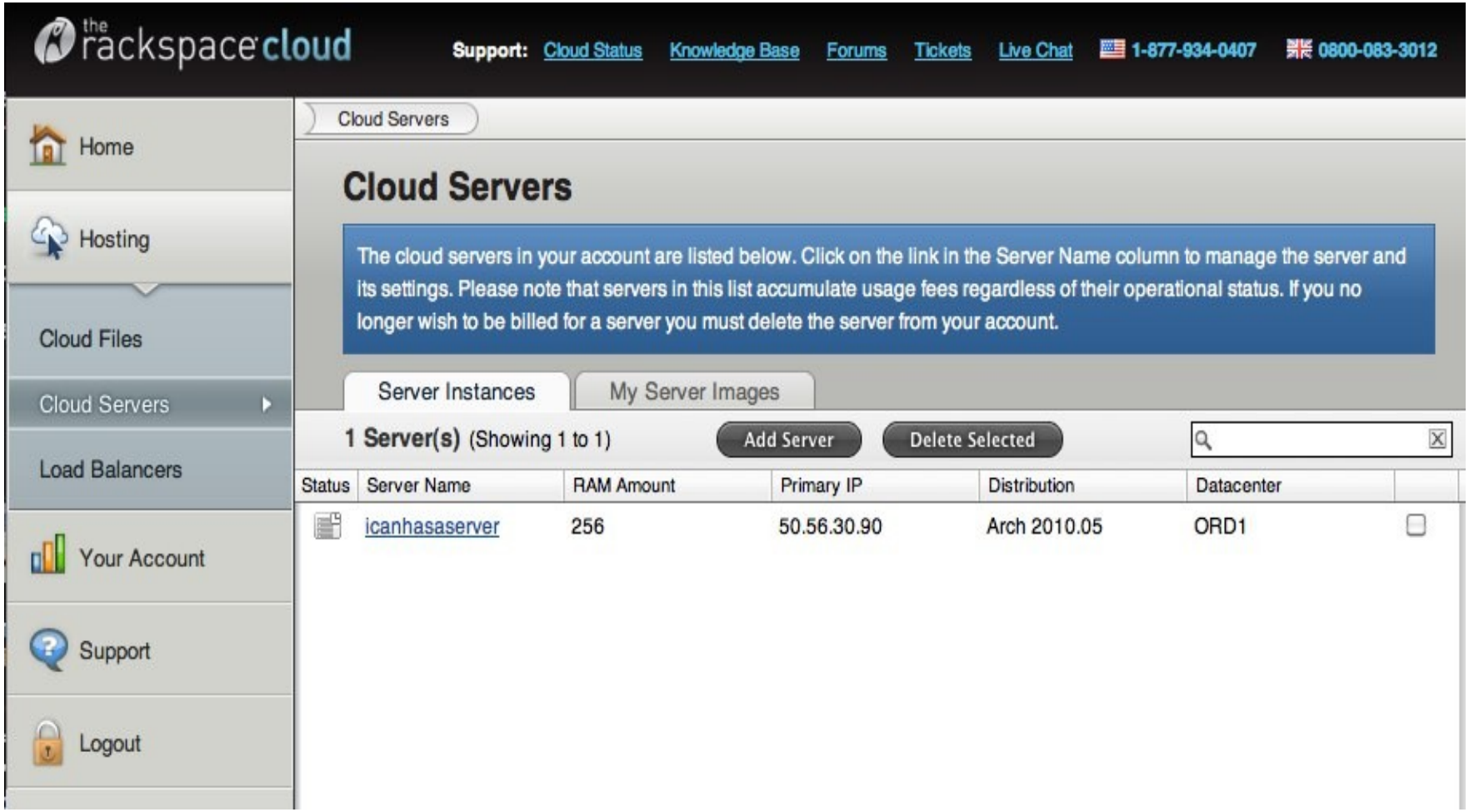
```
from libcloud.types import Provider
from libcloud.providers import get_driver

driver = get_driver(Provider.RACKSPACE)
conn = driver('username', 'api key')
node = conn.create_node(name='icanhasa
server',size=conn.list_sizes()[0],
image=conn.list_images()[0])
```

What is Libcloud?

Into this

What is Libcloud?



Why?

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

Why - different response formats

```
<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

```
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


```
[
    {"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

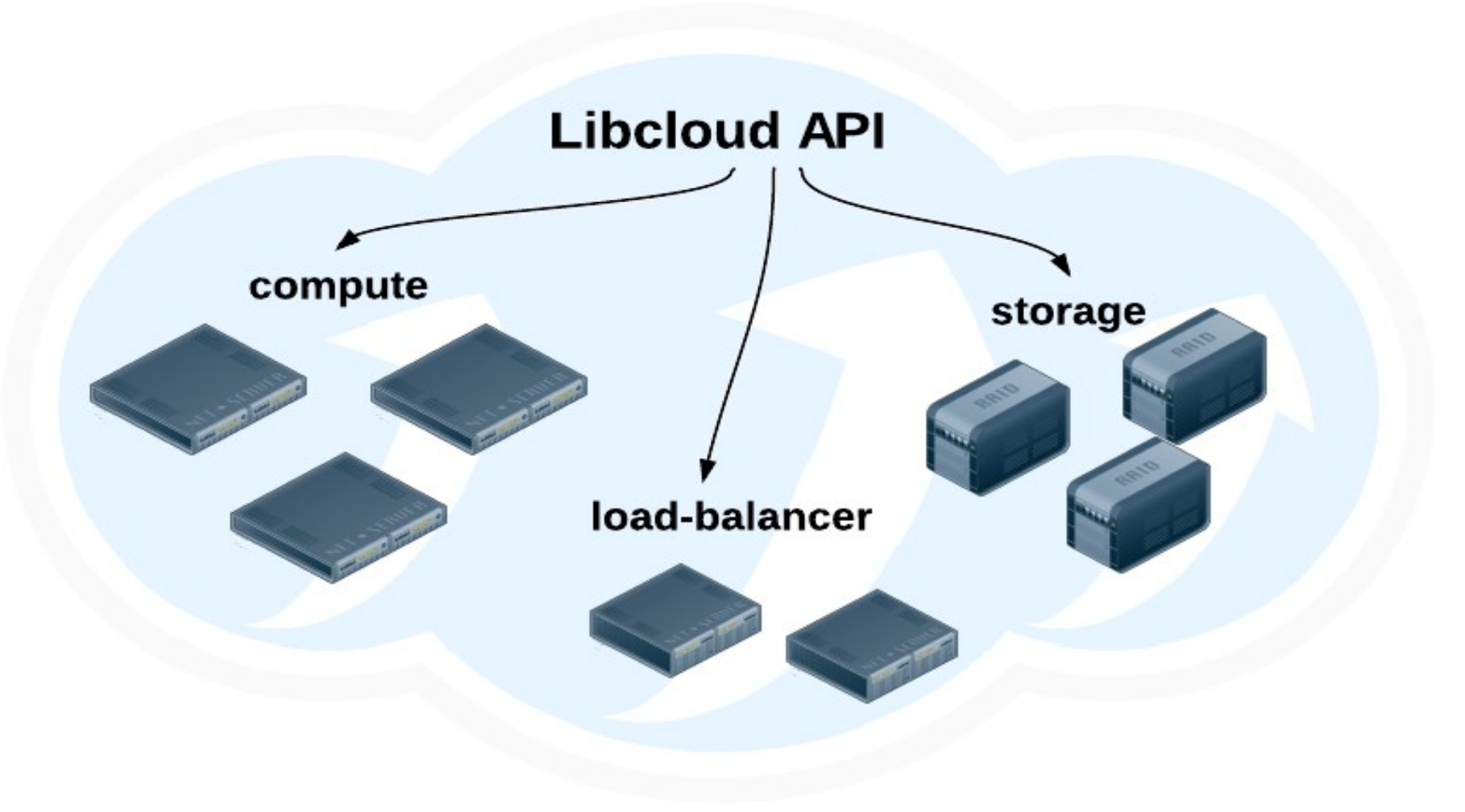
Apache Libcloud

- Originally developed at Cloudkick in 2009
- Later this year project joined Apache Incubator
- Recently graduated to an Apache TLP
- Current stable version is **0.5.0** (released in May)

Apache Libcloud

- Around ~8 active committers
- Relatively active mailing list and IRC channel
- Mailing list: {users, dev}@libcloud.apache.org
- IRC channel: #libcloud at freenode

Libcloud API



Compute API


brightbox
cloud beta
BLUE
BLUE BOX GROUP, LLC
ElasticHosts
Flexible servers in the cloud
slicehost
linode.com
OpenNebula
DreamHost
amazon
web services
enomaly
elastic computing
GOGRID
Eucalyptus
IBM
the rackspace cloud
serverlove
CLOUD SERVERS YOU'LL LOVE
SOFTLAYER
skalicloud
CREATE SERVERS, FORGET HARDWARE
RIMU
HOSTING
VPS NET
voxel
terremark
vmware
CloudSigma
openstack
OpSource

Compute API - terminology

- **image**

<NodeImage: id=aki-17765c63, name=ubuntu-kernels-milestone-eu/ubuntu-lucid-amd64-linux-image-2.6.32-305-ec2-v-2.6.32-305.9-kernel.img.manifest.xml, driver=Amazon EC2 (eu-west-1) ...>

- **size**

<NodeSize: id=m1.large, name=Large Instance, ram=7680 disk=850 bandwidth=None price=0.34 driver=Amazon EC2 (us-east-1) ...>

Compute API - terminology

- **location**

<EC2NodeLocation: id=0, name=Amazon Europe Ireland, country=IE, availability_zone=eu-west-1a driver=Amazon EC2 (eu-west-1)

- **node**

<Node: uuid=7e25ff022bfa952d15558d1c362e258d274b747e, name=i-db3fbcad, state=0, public_ip=['46.137.19.67'], provider=Amazon EC2 (eu-west-1) ...>

Compute API - methods

- list_images()
- list_sizes()
- list_locations()
- list_nodes()
- create_node()
- destroy_node()
- reboot_node()
- *deploy_node()*
- ex_* - provider specific functionality

Compute API - example

```
import os
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver
from libcloud.compute.deployment import MultiStepDeployment, ScriptDeployment, \
SSHKeyDeployment

conn = get_driver(Provider.RACKSPACE)('username', 'key')

install_key =
SSHKeyDeployment(open(os.path.expanduser("~/.ssh/id_rsa.pub")).read())
install_puppet = ScriptDeployment("apt-get -y install puppet")

msd = MultiStepDeployment([install_key, install_puppet])

image = conn.list_images()[0]
size = conn.list_sizes()[0]

node = conn.deploy_node(name='test', image=image, size=size, deploy=msd)
```

Storage API

Storage API - terminology

- **container**

<Container: name=test_container, provider=CloudFiles>

- **object**

<Object: name=test object, size=1500,
hash=sadj398998fhwqdasyhdsa, provider=CloudFiles ...>

Storage API - methods

- list_containers()
- list_container_objects()
- get_container()
- get_object()
- download_object()
- *download_objects_as_stream()*
- upload_object()
- *upload_object_via_stream()*
- delete_object()
- delete_container()

Storage API - methods

- create_container()
- ex_* - provider specific functionality

Storage API - example

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

driver = get_driver(Provider.CLOUDFILES_US)('username', 'key')

directory = '/home/some/path'
cmd = 'tar cvzpf - %s' % (directory)

# Create a container
container = driver.create_container('backups')

pipe = subprocess.Popen(cmd, bufsize=0, shell=True, stdout=subprocess.PIPE)
return_code = pipe.poll()

while return_code is None:
    # Compress data in our directory and stream it directly to CF
    container.upload_object_via_stream(iterator=pipe.stdout,
                                       object_name='backup.tar.gz')
    return_code = pipe.poll()
print 'Upload complete'
```

Load-balancer API

Load-balancer API - terminology

- **LoadBalancer**

<LoadBalancer: id=123, name=test loadbalancer, state=0>

- **Member**

<Member: id=326, address=10.0.0.10:12345>

- **Algorithm**

Algorithm.RANDOM, Algorithm.ROUND_ROBIN,
Algorithm.LEAST_CONNECTIONS

Load-balancer API - methods

- list_protocols()
- list_balancers()
- create_balancer()
- destroy_balancer()
- get_balancer()
- balancer_attach_compute_node()
- balancer_attach_member()
- balancer_detach_member()

Load-balancer API - methods

- balancer_list_members()
- ex_* - provider specific functionality

Load-balancer API - example

```
driver = get_driver(Provider.RACKSPACE_US) Rackspace('username', 'api key')
new_balancer_name = 'testlb'
new_balancer = driver.create_balancer(name=new_balancer_name,
        algorithm=Algorithm.ROUND_ROBIN, port=80, protocol='http',
        members=(Member(None, '192.168.86.1', 8080),
                 Member(None, '192.168.86.2', 8080)))
# wait for the balancer to become ready
while True:
    balancer = driver.get_balancer(balancer_id=new_balancer.id)
    if balancer.state == State.RUNNING:
        break
    time.sleep(20)
# fetch list of members
members = balancer.list_members()
print members
```

Plans for the future

- More storage and load-balancer provider drivers
- Support for even more services – DNS, CDN, Block storage, …
- Improving pricing data distribution
- Revamping the whole “location” concept
- ...

Sprint

- We are holding a sprint here after the conference – join us!
  - learn more about the library
  - contribute
  - have fun

Questions?

Questions?

- Thanks.

- http://libcloud.apache.org
- {users,dev}@libcloud.apache.org
- #libcloud at irc.freenode.net