

Ruby SDK Training

Kyle Rames

About Me

- Ruby guy on DRG
- Fog MVP committer
- knife-rackspace maintainer
- Five years Ruby/RoR experience
- Failed Startup
- Java/.NET developer in a previous life

Agenda

- Introduction
- Cloud Servers
- Model Layer
- Cloud Files
- Request Layer
- Troubleshooting
- Contributing
- Questions?



37 Different Providers

Fog Service	Rackspace	Openstack
BlockStorage	✓	✓
CDN	✓	
Compute	✓	✓
Databases	✓	
DNS	✓	
Identity	✓	✓
Image		✓
LoadBalancers	✓	
Storage	✓	✓
Volume		✓

Gems using Fog

- knife-rackspace (Chef)
- Puppet Cloud Provisioner
- Paperclip
- Carrierwave

Cloud Servers

Bootstrap

1. Creates server
2. Waits for server to finish building
3. Create `ROOT_USER/.ssh/authorized_keys`
4. Lock password for root user
5. Create `ROOT_USER/attributes.json` file
6. Create `ROOT_USER/metadata.json` file

Documentation

- <http://developer.rackspace.com/#ruby>
 - SDK Quick Reference
 - API Reference
 - Examples
- <http://github.com/fog/fog>
- <http://fog.io>

Your turn!

- Create a server using `bootstrap`
- Execute `uname -a` on remote server using `ssh` method.

Remember <http://developer.rackspace.com/#ruby!>

Exceptions

NotFound	HTTP 404
ServiceError	Unknown HTTP Error
InternalServerError	HTTP 500
BadRequest	HTTP 400
Excon::Errors::Unauthorized	HTTP 401
Fog::Errors::TimeoutError	wait_for timeout has been exceeded

Note unless specifically noted is full namespace is Fog::Compute::RackspaceV2

Exceptions

StandardError

Fog::Errors::Error

Fog::Rackspace::Errors::ServiceError

Fog::Rackspace::Errors::BadRequest

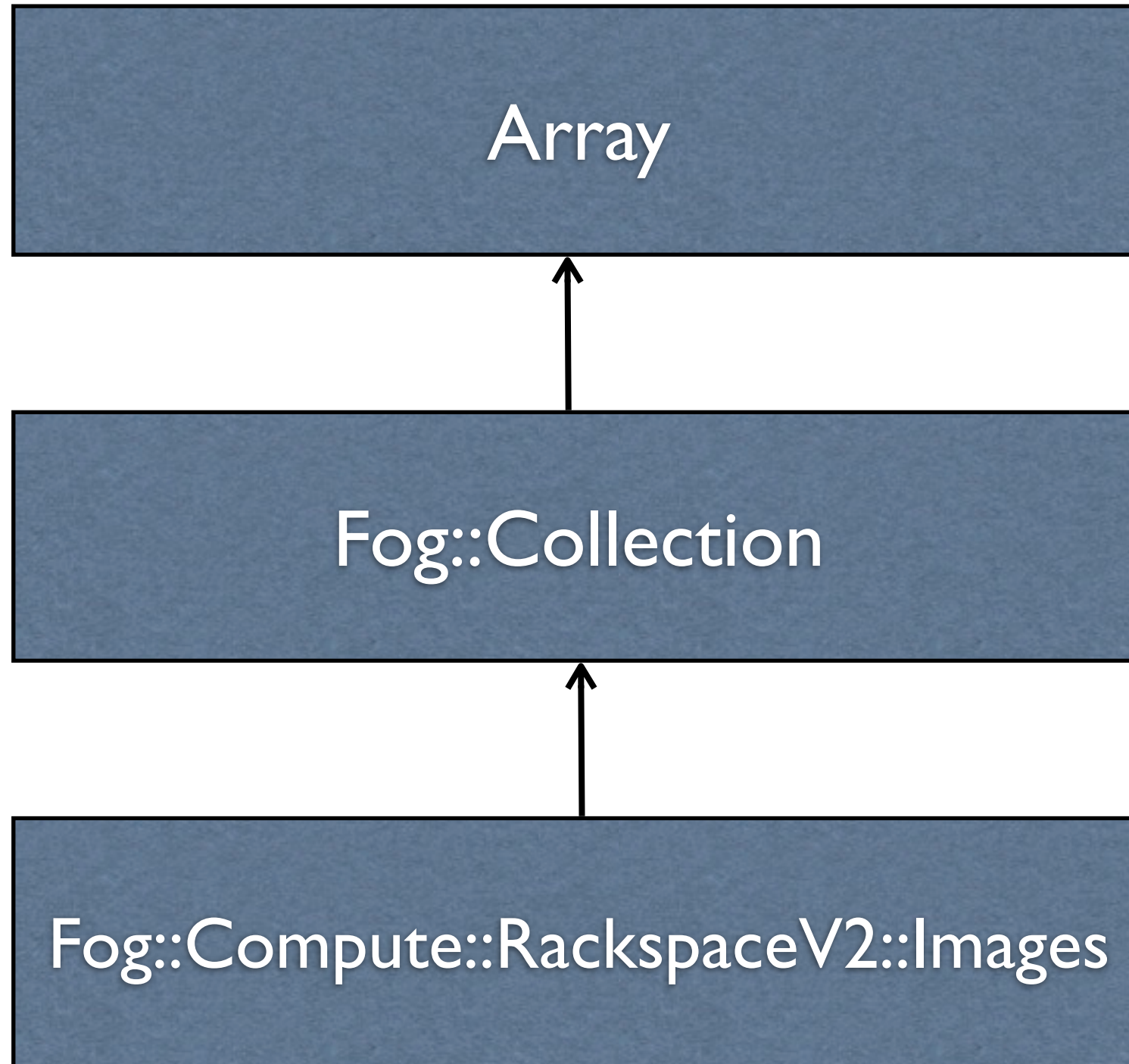
Fog::Compute::RackspaceV2::BadRequest

Model Layer

Remember?

`service.images`

Collection



Fog::Collection Methods

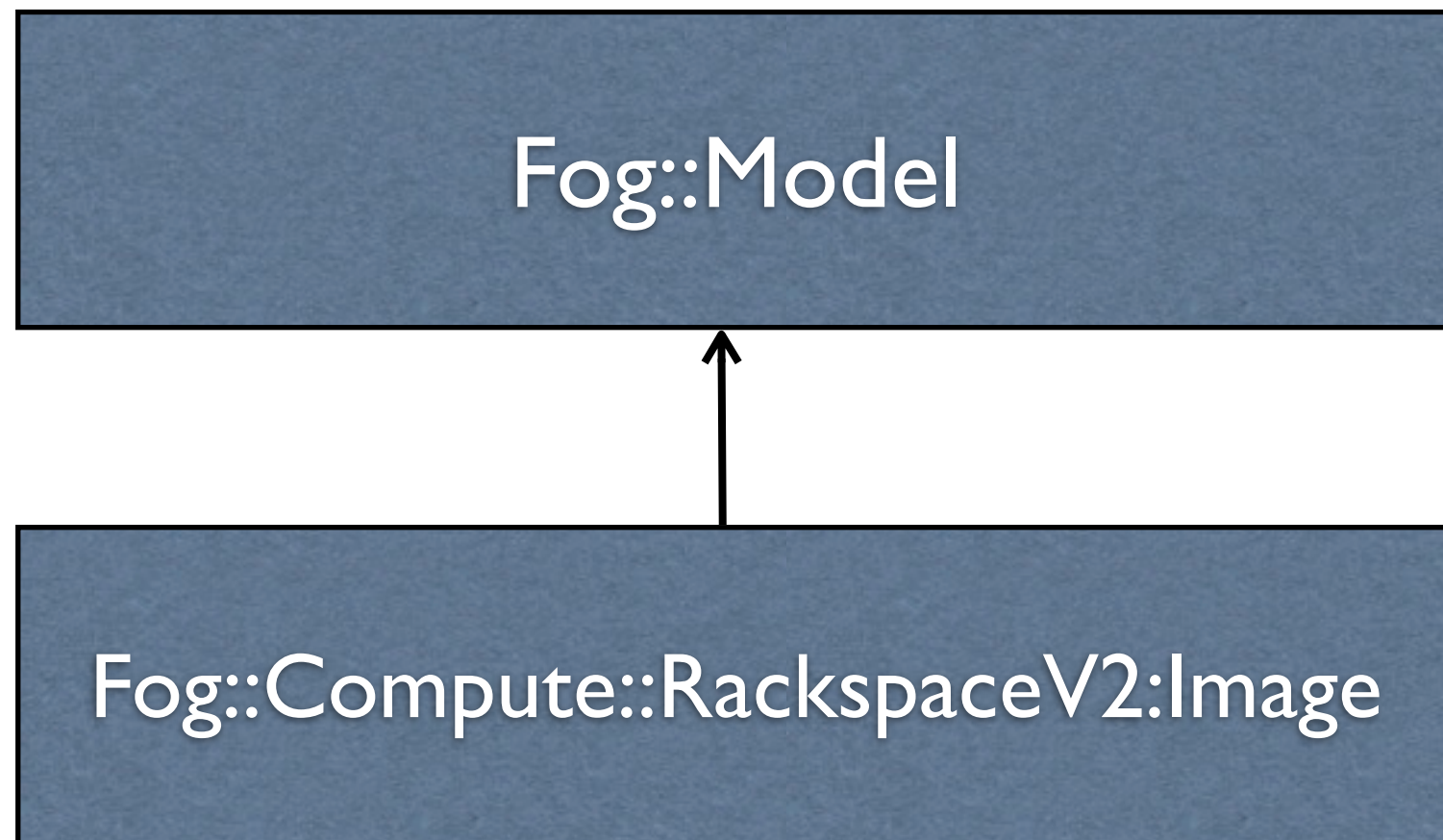
<code>all</code>	fetch every object of that type from the provider.
<code>get</code>	fetch a single object by it's identity from the provider.
<code>create</code>	initialize a new record locally and a remote resource with the provider.
<code>new</code>	initialize a new record locally, but do not create a remote resource with the provider.

Collections

```
service.collections
```

```
[ :servers, :flavors, :images,  
  :attachments, :networks ]
```

Model



Fog::Model Methods

<code>attributes</code>	Returns a hash containing the list of model attributes and values.
<code>save</code>	Saves object. (<i>not all object support update</i>)
<code>reload</code>	Updates object with latest state from service.
<code>ready?</code>	Returns true if object is in a ready state and able to perform actions.
<code>wait_for</code>	Periodically reloads model yielding to block.

Cloud Files

What's the best place
to find info about fog?

Your Turn!

- Create a public container
- Upload a file
- Print out the public URL, streaming URL and iOS URL

Request Layer

- Mirrors REST interface
- Most efficient way to access cloud
- Not portable between providers
- Preference model layer!

Requests

```
service.requests  
  
[:copy_object,  
:delete_container,  
:delete_object,  
:get_container,  
:get_containers, :get_object,  
...]
```

CDN

Fog deep dive

```
module Fog
  module Storage
    class Rackspace

      class Directories < Fog::Collection

        def all
          data = service.get_containers.body
          load(data)
        end
      end
    end
  end
end
```

Source

```
module Fog
  module Storage
    class Rackspace
      class Real
        def get_containers(options = {})
          options = options.reject {|key, value| value.nil?}
          request(
            :expects => [200, 204],
            :method   => 'GET',
            :path      => '',
            :query     => {'format' => 'json'}.merge!(options)
          )
        end
      end
    end
  end
end
```

Source

```
GET /<api version>/<account> HTTP/1.1  
Host: storage.clouddrive.com  
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

REST Docs

Troubleshooting

Check

- Ruby Version
- RubyGems Version
- Fog Version
- **Region**

Next Steps

- `EXCON_DEBUG=true`
- Use gist to share information
- Get code or configuration info if possible
- Try to duplicate locally
- Alternatively send them a sample to run
- Fail back to curl

Last Step

sdk-support@rackspace.com

Sample Issues

- <http://stackoverflow.com/questions/14158256/uploading-to-rackspace-cloud-files-with-paperclip-and-fog/14204155#14204155>
- <https://github.com/fog/fog/issues/1739>
- <https://gist.github.com/krames/65ea14d22d5504a70fdc>

Roadmap

- Refactor Rackspace provider share common base with OpenStack provider
- Fill out implementation of Fog Services

Contributing

- Any contributions are welcome!
- Write Tests!
- <https://github.com/fog/fog#contributing>
- Use personal Github account
- Fork from github.com/fog/fog
- Feel free to reach out to me if you need help getting started

~/.fog

default:

rackspace_username: *USERNAME*

rackspace_api_key: *API_KEY*

Additional Resources

- <http://developer.rackspace.com>
- <http://fog.io>
- #ruby-fog on freenode.net
- <https://groups.google.com/forum/?fromgroups=#!forum/ruby-fog>
- sdk-support@rackspace.com

Questions?