RUBY AWS::SDK EXAMPLES (AWS-SDK GEM ~>2)

SETTINGS

You can setup the connection on global way:

```
Aws.config.update(
        endpoint: 'https://objects.dreamhost.com.',
        access_key_id: 'my-access-key',
        secret_access_key: 'my-secret-key',
        force_path_style: true,
        region: 'us-east-1'
)
```

and instantiate a client object:

```
s3_client = Aws::S3::Client.new
```

LISTING OWNED BUCKETS

This gets a list of buckets that you own. This also prints out the bucket name and creation date of each bucket.

```
s3_client.list_buckets.buckets.each do |bucket|
        puts "#{bucket.name}\t#{bucket.creation_date}"
end
```

The output will look something like this:

```
mahbuckat1   2011-04-21T18:05:39.000Z
mahbuckat2   2011-04-21T18:05:48.000Z
mahbuckat3   2011-04-21T18:07:18.000Z
```

CREATING A BUCKET

This creates a new bucket called `my-new-bucket`

```
s3_client.create_bucket(bucket: 'my-new-bucket')
```

If you want a private bucket:

acl option accepts: # private, public-read, public-read-write, authenticated-read

```
s3_client.create_bucket(bucket: 'my-new-bucket', acl: 'private')
```

LISTING A BUCKET'S CONTENT

This gets a list of hashes with the contents of each object This also prints out each object's name, the file size, and last modified date.

```
s3_client.get_objects(bucket: 'my-new-bucket').contents.each do |object|
        puts "#{object.key}\t#{object.size}\t#{object.last-modified}"
end
```

The output will look something like this if the bucket has some files:

```
myphoto1.jpg 251262  2011-08-08T21:35:48.000Z
myphoto2.jpg 262518  2011-08-08T21:38:01.000Z
```

DELETING A BUCKET

Note: The Bucket must be empty! Otherwise it won't work!

```
s3_client.delete_bucket(bucket: 'my-new-bucket')
```

FORCED DELETE FOR NON-EMPTY BUCKETS

First, you need to clear the bucket:

```
Aws::S3::Bucket.new('my-new-bucket', client: s3_client).clear!
```

after, you can destroy the bucket

```
s3_client.delete_bucket(bucket: 'my-new-bucket')
```

CREATING AN OBJECT

This creates a file `hello.txt` with the string `"Hello World!"`

```
s3_client.put_object(
        key: 'hello.txt',
        body: 'Hello World!',
        bucket: 'my-new-bucket',
        content_type: 'text/plain'
)
```

CHANGE AN OBJECT'S ACL

This makes the object `hello.txt` to be publicly readable, and `secret_plans.txt` to be private.

```
s3_client.put_object_acl(bucket: 'my-new-bucket', key: 'hello.txt', acl: 'public-read')

s3_client.put_object_acl(bucket: 'my-new-bucket', key: 'private.txt', acl: 'private')
```

DOWNLOAD AN OBJECT (TO A FILE)

This downloads the object `poetry.pdf` and saves it in `/home/larry/documents/`

```
s3_client.get_object(bucket: 'my-new-bucket', key: 'poetry.pdf', response_target: '/home/larr
```

DELETE AN OBJECT

This deletes the object `goodbye.txt`

```
s3_client.delete_object(key: 'goodbye.txt', bucket: 'my-new-bucket')
```

GENERATE OBJECT DOWNLOAD URLS (SIGNED AND UNSIGNED)

This generates an unsigned download URL for `hello.txt`. This works because we made `hello.txt` public by setting the ACL above. This then generates a signed download URL for `secret_plans.txt` that will work for 1 hour. Signed download URLs will work for the time period even if the object is private (when the time period is up, the URL will stop working).

```
puts Aws::S3::Object.new(
        key: 'hello.txt',
        bucket_name: 'my-new-bucket',
        client: s3_client
).public_url

puts Aws::S3::Object.new(
        key: 'secret_plans.txt',
        bucket_name: 'hermes_ceph_gem',
        client: s3_client
).presigned_url(:get, expires_in: 60 * 60)
```

The output of this will look something like:

```
http://objects.dreamhost.com/my-bucket-name/hello.txt
http://objects.dreamhost.com/my-bucket-name/secret_plans.txt?Signature=XXXXXXXXXXXXXXXXXXXXXXX
```

RUBY AWS::S3 EXAMPLES (AWS-S3 GEM)

CREATING A CONNECTION

This creates a connection so that you can interact with the server.

```
AWS::S3::Base.establish_connection!(
        :server            => 'objects.dreamhost.com',
        :use_ssl           => true,
        :access_key_id     => 'my-access-key',
        :secret_access_key => 'my-secret-key'
)
```

LISTING OWNED BUCKETS

This gets a list of AWS::S3::Bucket objects that you own. This also prints out the bucket name and creation date of each bucket.

```
AWS::S3::Service.buckets.each do |bucket|
        puts "#{bucket.name}\t#{bucket.creation_date}"
end
```

The output will look something like this:

```
mahbuckat1   2011-04-21T18:05:39.000Z
mahbuckat2   2011-04-21T18:05:48.000Z
mahbuckat3   2011-04-21T18:07:18.000Z
```

CREATING A BUCKET

This creates a new bucket called `my-new-bucket`

```
AWS::S3::Bucket.create('my-new-bucket')
```

LISTING A BUCKET'S CONTENT

This gets a list of hashes with the contents of each object This also prints out each object's name, the file size, and last modified date.

```
new_bucket = AWS::S3::Bucket.find('my-new-bucket')
new_bucket.each do |object|
        puts "#{object.key}\t#{object.about['content-length']}\t#{object.about['last-modified
end
```

The output will look something like this if the bucket has some files:

```
myphoto1.jpg 251262  2011-08-08T21:35:48.000Z
myphoto2.jpg 262518  2011-08-08T21:38:01.000Z
```

DELETING A BUCKET

Note: The Bucket must be empty! Otherwise it won't work!

```
AWS::S3::Bucket.delete('my-new-bucket')
```

FORCED DELETE FOR NON-EMPTY BUCKETS

```
AWS::S3::Bucket.delete('my-new-bucket', :force => true)
```

CREATING AN OBJECT

This creates a file `hello.txt` with the string `"Hello World!"`

```
AWS::S3::S3Object.store(
        'hello.txt',
        'Hello World!',
        'my-new-bucket',
        :content_type => 'text/plain'
)
```

CHANGE AN OBJECT'S ACL

This makes the object `hello.txt` to be publicly readable, and `secret_plans.txt` to be private.

```
policy = AWS::S3::S3Object.acl('hello.txt', 'my-new-bucket')
policy.grants = [ AWS::S3::ACL::Grant.grant(:public_read) ]
AWS::S3::S3Object.acl('hello.txt', 'my-new-bucket', policy)

policy = AWS::S3::S3Object.acl('secret_plans.txt', 'my-new-bucket')
policy.grants = []
AWS::S3::S3Object.acl('secret_plans.txt', 'my-new-bucket', policy)
```

DOWNLOAD AN OBJECT (TO A FILE)

This downloads the object `poetry.pdf` and saves it in `/home/larry/documents/`

```
open('/home/larry/documents/poetry.pdf', 'w') do |file|
        AWS::S3::S3Object.stream('poetry.pdf', 'my-new-bucket') do |chunk|
                file.write(chunk)
        end
end
```

DELETE AN OBJECT

This deletes the object `goodbye.txt`

```
AWS::S3::S3Object.delete('goodbye.txt', 'my-new-bucket')
```

GENERATE OBJECT DOWNLOAD URLS (SIGNED AND UNSIGNED)

This generates an unsigned download URL for `hello.txt`. This works because we made `hello.txt` public by setting the ACL above. This then generates a signed download URL for `secret_plans.txt` that will work for 1 hour. Signed download URLs will work for the time period even if the object is private (when the time period is up, the URL will stop working).

```
puts AWS::S3::S3Object.url_for(
        'hello.txt',
        'my-new-bucket',
        :authenticated => false
)

puts AWS::S3::S3Object.url_for(
        'secret_plans.txt',
        'my-new-bucket',
        :expires_in => 60 * 60
)
```

The output of this will look something like:

```
http://objects.dreamhost.com/my-bucket-name/hello.txt
http://objects.dreamhost.com/my-bucket-name/secret_plans.txt?Signature=XXXXXXXXXXXXXXXXXXXXXXX
```