LibcloudStorageAPI

Storage API ideas

Base API should probably consist of three classes:

1. Driver

Methods:

- create_container(container_name) - create a new container
- delete_container(container_name) - delete an existing container
- list_containers() - return a list of all the containers - some providers allow you to filter the results by a prefix but I don't think we should expose this to the user. This method should also return a "lazy" list (more on this bellow).

2. Container - represents an object container (some libraries call it "bucket")

Properties:

- name / key - container name / key
- object_count - number of objects located in this container (not sure if this is supported by all the providers)
- extra - other attributes and meta data

Methods:

- list_objects() - returns a list of objects in this container. It makes sense for this method to return a "lazy" list (some APIs limit the maximum number of objects which can be returned in a single call, but this should be hidden away from the end user)
- delete() - remove a container and all the objects located in this container
- upload_object(file_path / file_like_object, object_name) - upload an object to this container.
  The actual blob should be split and uploaded in chunks (this also makes it possible to use the "parallel upload" Amazon S3 functionality).
- download_object(object_key, destination_path) - download an object to the specified path. We should also have another method which returns a File like object instead of downloading it. Returning a File like object has many advantages, one of them is that it can easily be used for file uploads (and other things) with frameworks like Django.
- delete_object(object_instance) - delete an object

3. Object - represents a binary object (blob)

Properties:

- name / key - object key
- size - object size
- container - name of the container
- object_hash - ideally, we would also have some kind of hash (md5 / sha1) so we can easily verify the file integrity (not sure if this is supported by all the providers).
- extra - other attributes and meta data

Methods:

- download(destination_path) - download an object to the destination path
- as_file() - return a File like object
- delete() - delete an object

Other things & notes:

- API should also support both "normal" and "object oriented" way of using the methods.

For example:

driver.delete_object(object_instance) and object_instance.delete()

- Some of the providers also offer CDN functionality, but I'm not sure if we should use extension methods for this (ex_) or move into the standard library