d3ploy Documentation

Release 1.3.2

dryan

November 06, 2013

Contents

To install, run `pip install d3ploy`. To use, run `d3ploy`. Additional arguments may be specified. Run `d3ploy --help` for more information.

CHAPTER 1

Authentication

Your AWS credentials can be set in a number of ways:

1. In a ".boto" file in your home folder. See Boto's documentation for how to create this file.
2. In a ".aws" file in the folder you're running `d3ploy` in. Follows the same format as ".boto".
3. In the environment variables "AWS_ACCESS_KEY_ID" and "AWS_SECRET_ACCESS_KEY".
4. Passed in as arguments. `-a` or `--access-key` for the Access Key ID and `-s` or `--access-secret` for the Secret Access Key.
5. In the per-enviroment configuration outlined below.

CHAPTER 2

Configuration options

When you run `d3ploy`, it will look in the current directory for a "deploy.json" file that defines the different deploy enviroments and their options. At a minimum, a "default" environment is required and is the environment used if you pass no arguments to `d3ploy`. Additionally, you may pass in a different path for you config file with the `-c` or `--config` options.

You can add as many environments as needed. Deploy to an environment by passing in its key like `d3ploy staging`. Environments besides "default" will inherit any settings not explicitly set from the default configuration.

The only required option for any environment is "bucket" for the S3 bucket to upload to. Additionally, you may define:

- "local_path" to upload only the contents of a directory under the current one; defaults to "." (current directory)
- "bucket_path" to upload to a subfolder in the bucket; defaults to "/" (root)
- "aws_key" to specify the AWS Access Key ID to use for uploading
- "aws_secret" to specify the AWS Secret Access Key to use for uploading
- "exclude" to specify patterns to not upload
- "gzip" to automatically gzip files before uploading to S3
- "delete" to remove files on S3 that are not present in the local directory
- "charset" to set the charset flag on 'Content-Type' headers of text files
- "cache" to set the Cache-Control header for various mimetypes. See below for more.
- "gitignore" to add all entries in a .gitignore file to the exclude patterns

CHAPTER 3

Cache-Control Headers

If you want to set Cache-Control headers on various files, add a `cache` object to your config file like:

```
"cache": {
  "text/css": 2592000,
  "application/javascript": 2592000,
  "image/png": 22896000,
  "image/jpeg": 22896000,
  "image/webp": 22896000,
  "image/gif": 22896000
}
```

Each key is the mimetype of the kind of file you want to have cached, with a value that is the seconds the `max-age` flag set to. In the above example, CSS and JavaScript files will be cached for 30 days while images will be cached for 1 year. For more about Cache-Control, read Leverage Browser Caching.

CHAPTER 4

OS X Notification Center

d3ploy will attempt to alert you via Notification Center when it is completed. To enable this feature, you need pyobjc; run `pip install pyobjc` to install.

Caution About Using the gzip Option

Almost all modern browsers will support files that are served with gzip compression. The notable exception is non-smartphone mobile browsers. If you have significant traffic over those browsers, it is advisable to avoid the gzip option.