Last updated Saturday, Nov 16, 2019

Configuring and working with Cloudfront Logs

Example of how to setup Cloudfront to log to S3, enable log rotation and how to download and work with combined Cloudfront log files.

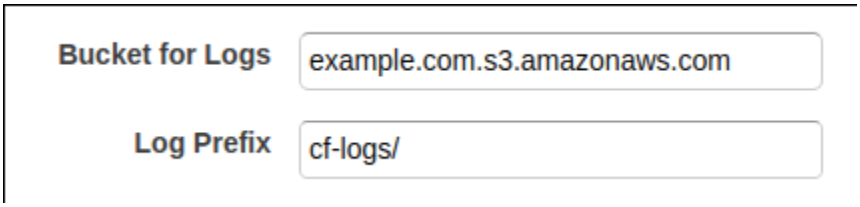
Estimated reading time: 4 minutes

Table of contents

- Setting up logging on Cloudfront
- Setting up log rotation on S3
- Fetching log files
- Working with the log files
- Further reading

Setting up logging on Cloudfront

Cloudfront supports logging to an Amazon S3 bucket. Create the bucket first and then edit the Cloudfront distribution. Under the general tab specify a Bucket for Logs and also a log prefix. The log prefix to is set to `cf-logs/` so it can be targeted with lifecycle rules in the S3 bucket.



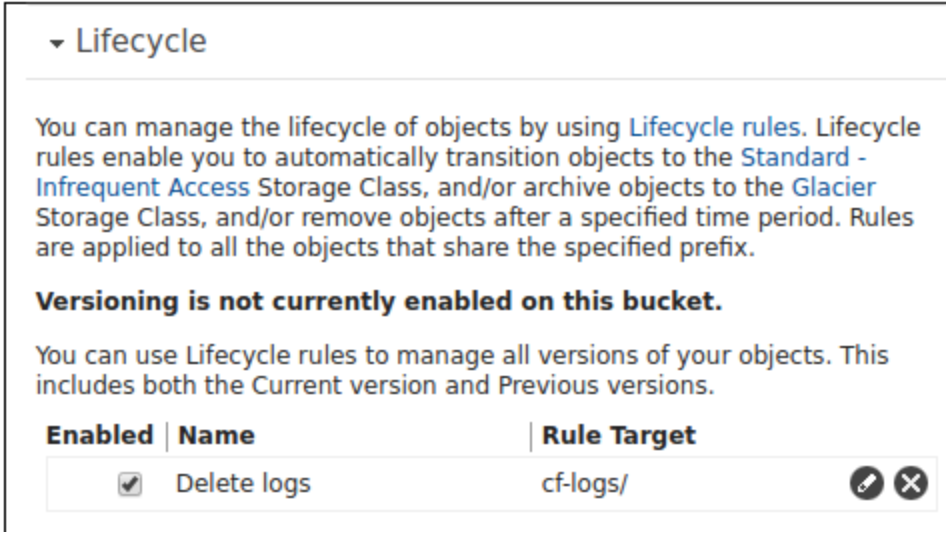
Once configured log files will be written to the S3 bucket as traffic flows through the Cloudfront distribution. Files are written as gzipped text files in the W3C extended log file format. This is good as they can be used with a variety of tools to analyse them.

More details on how AWS logs Cloudfront requests is available on the Cloudfront Developer Guide (http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html)

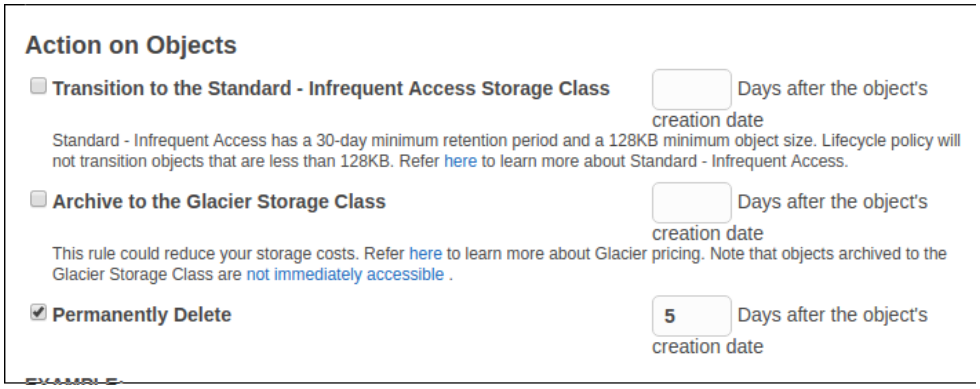
Setting up log rotation on S3

Cloudfront writes logs to an S3 bucket which means that any of the features available on S3 can be used. Cloudfront logs each request so it is unlikely that there is a need to store this information forever and it also incurs a cost to store it. S3's lifecycle feature can be used to remove files after a certain period.

To enable lifecycle management open the S3 bucket and click on properties. Then click on lifecycle. Then add a rule that targets the folder where the Cloudfront logs are stored. In this example this is the `cf-logs/` prefix.



It is possible to choose to permanently delete files or to transition them to Amazon Glacier. In this example files are deleted after 5 days.



More information on lifecycle management is available on the S3 Developer Guide (http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html).

Fetching log files

Now the log files are being written and being rotated they can be analysed. Amazon provide some services for interrogating logs but with some UNIX skill most requirements can be achieved by downloading the files.

To download the files here is a simple bash script to download the files, combine them into a single file and removing any comments. The script depends on the `aws-cli` tool that is readily available on all platforms.

More information on installing the `aws-cli` tool is available on the AWS CLI User Guide (http://docs.aws.amazon.com/cli/latest/userguide/installing.html)

```
#!/usr/bin/env bash

BUCKET=$1
CWD=$(pwd)

if [[ -n $1 ]]; then
  aws s3 sync s3://$BUCKET/cf-logs .
  cat *.gz > combined.log.gz
  find $CWD ! -name 'combined.log.gz' -name '*.gz' -type f -exec rm -
  gzip -d combined.log.gz
  sed -i '/^#/ d' combined.log
  exit 0
else
  echo "Error: no bucket name provided"
```

```
  exit 1
fi
```

The script does the following:

- Reads the bucket name as the first argument
- Synchronises the current working directory with the specified S3 bucket
- Combines the gzipped log files into a single file
- Removes all files other than the combined file
- Decompresses the file
- Removes comments

The script is saved as `aws-cf-logs`. To fetch a combined log file is then as simple as

```
aws-cf-logs example-bucket
ls
combined.log
```

The script is available as this gist (https://gist.github.com/shapeshed/e25bdf3b1116899fa8c47c16db9aa8e0). Feel free to fork, extend or improve it as you wish.

Working with the log files

The log file is in a UNIX friendly standard format so it is easy to extract information from it using standard UNIX tools.

In the following example a file is generated with a list of 404 URLs ordered by frequency. This can be useful for finding broken links.

```
grep '404' combined.log | cut -f 8 | sort | uniq -c | sort -n  -r
242 /apple-touch-icon.png
238 /apple-touch-icon-precomposed.png
 54 /example-url/
 40 /another-example-url/
...
```

In the following example a list of IP addresses is generated and sorted by frequency of occurrence. This can be useful for finding out bad bots.

```
cut -f 5 combined.log | sort | uniq -c | sort -n -r
1298 51.254.130.62
1846 216.244.66.240
1383 157.55.39.84
1325 68.180.228.227
...
```

In the following example the number of cache hits and cache misses is shown.

```
grep -c 'Hit' combined.log
19325
grep -c 'Miss' combined.log
8345
```

Further reading

- AWS Cloudfront Developer Guide - AccessLogs (http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html)
- AWS S3 Developer Guide - Lifecycle Management (http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html)
- AWS CLI Developer Guide - Installing (http://docs.aws.amazon.com/cli/latest/userguide/installing.html)

Have an update or suggestion for this article? You can edit it here and send me a pull request. (https://github.com/shapeshed/shapeshed.com/edit/master/content/posts/aws-cloudfront-logs.md)

Tags

- UNIX (/tags/unix)
- Linux (/tags/linux)
- AWS (/tags/aws)

Recent Posts

About the author

George Ornbo is a UK based human.

He is interested in people, music, food and writing. In a previous version of himself he wrote books (https://www.amazon.com/Sams-Teach-Yourself-Hours-Programming/dp/0672338033) on technology (https://www.amazon.com/Sams-Teach-Yourself-Node-js-Hours/dp/0672335956) .

← http://shapeshed.com (/)