Tamarin Documentation

Release 1.0

DUO Interactive, LLC

Sep 27, 2017

Contents

Tamarin is a drop-in Django app for parsing/storing S3 access logs in a Django model. This allows to build queries against your historic data.

Note: Tamarin provides no analysis tools besides a model that you may query. Analysis is best left to other apps that use the models provided here.

CHAPTER 1

How it works

Tamarin uses celery or a Django management command to list the keys in log buckets that you specify. The contents of the keys are downloaded and parsed using pyparsing and stored in a Django model. From there, it is up to you to do what you'd like with the data.

No other functionality is included. It is up to you to analyze and use the data for your specific case. However, if you write anything cool that uses data from Tamarin, please open an issue on the issue tracker and we'll link you here.

CHAPTER 2

Usage cases

Tamarin may be useful to you if you are interested in monitoring...

- Which IP addresses consume the most bandwidth.
- Files that are responsible for the biggest chunk of your bill.
- 404s.
- Referring sites.

You might also like Tamarin if you want to...

- Perform simple or complex analysis of your S3 access logs using Django's ORM.
- Implement automatic banning of bandwidth hogs (not included in app).
- Create pretty charts and graphs (in another app).
- Set up bandwidth spike alarms.

CHAPTER 3

Learning more

Project Status: Stable

License: Tamarin is licensed under the BSD License.

These links may also be useful to you.

- Source repository: https://github.com/duointeractive/tamarin
- Issue tracker: https://github.com/duointeractive/tamarin/issues

CHAPTER 4

Documentation

Installation

Tamarin is installed much like most Django apps.

Requirements

- Python 2.5, 2.6, or 2.7
- Django >=1.2
- Boto >= 1.9b
- pyparsing >= 1.4

Obtaining the package

You may install Tamarin via the **`easy_install`** or **`pip`**:

```
easy_install tamarin
```

or:

```
pip install tamarin
```

Note: If you don't have access to **`pip`**, you may download a tarball/zip, from our GitHub project and install via the enclosed `setup.py`.

Integration

You'll then want to add *tamarin* to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'tamarin',
)
```

After this, if you are using South:

```
./manage.py migrate
```

If you are not using South, you'll want to:

```
./manage syncdb
```

Setting up the log puller

The module that actually does the pulling of your S3 access logs is called the log puller. There are currently two different ways to retrieve access logs automatically:

- A celery task that fires at configurable intervals.
- A management command, `tamarin_pull_logs`.

If you are already using celery, you should be all set. You can adjust the interval at which logs are pulled using the `TAMARIN_CELERY_PULL_PARSE_INTERVAL` setting in settings.py. This is an integer value (in minutes) for how often to check S3 for new logs.

If you don't have/want celery, you may set up a cron entry to run something like the following:

```
./manage.py tamarin_pull_logs
```

Set up bucket logging on S3

Before progressing any further, take a moment to set up bucket logging for one or more of your buckets. You may point more than one bucket at the same log bucket, but log buckets must only contain log files.

If you need details on how to do this, check out S3's bucket logging documentation.

Warning: If any files other than S3 access logs make their way into one of your log buckets, you will see errors, and the log puller will most likely not function.

Add buckets to monitor

At this point, you should have installed Tamarin and configured your choice of puller (celery or Django management command).

Log into your admin site, navigate to the Tamarin section. Add a 'S3 logged bucket'.

Tip: The `name` field is the bucket that the media resides in, not the name of its log bucket.

The `Monitor bucket` checkbox should default to being checked, but make sure it is if you want this bucket to be pulled/parsed.

Profit

Once a bucket is added, your puller should take care of the rest. Note that if you have a large backlog of logs to pull, this might take a good long while, and may take multiple calls to the puller.

For an overview of what models and fields are available for querying, see the *Model reference* page.

Settings

Tamarin only has a few settings to tweak with in your `settings.py`.

TAMARIN_CELERY_PULL_PARSE_INTERVAL

Default: `5`

This is an interval (in minutes) that determines how often to poll S3 for new access logs. You'll want to bump this up past the default of *5 minutes* if you have more than a handful of buckets to monitor.

Note: This setting is only used if you are running celery.

TAMARIN_PURGE_PARSED_KEYS

Default: `True`

When `True`, keys are removed from your S3 access log buckets after being successfully parsed. This will keep down the size of future requests, and avoid re-parsing logs that we have already seen.

Model reference

The following models should be the only thing you need interact with in your project code.

tamarin.models.S3LogRecord

The `S3LogRecord` class represents a single access log entry. The following fields are available for querying.

bucket_owner (CharField) The canonical id of the owner of the source bucket.

bucket (ForeignKey – `S3LoggedBucket`) The bucket that the request was processed against.

request_dtime (DateTimeField) The time at which the request was received.

remote_ip (IPAddressField) The apparent Internet address of the requester. Intermediate proxies and firewalls might obscure the actual address of the machine making the request.

requester (CharField, nullable) The canonical user id of the requester, or the string 'Anonymous' for unauthenticated requests. This identifier is the same one used for access control purposes.

request_id (CharField) A string generated by Amazon S3 to uniquely identify each request.

operation (CharField) Either SOAP.<operation> or REST.<HTTP_method.resource_type>.

key (TextField) The 'key' part of the request, URL encoded, or '-' if the operation does not take a key parameter.

request_method (CharField, nullable) The method used to retrieve the file. Typically either GET or POST.

request_uri (TextField) The Request-URI part of the HTTP request message.

http_version (CharField, nullable) HTTP version used in the request. Typically HTTP/1.0 "or HTTP/1.1."

http_status (PositiveIntegerField) The HTTP response code for the request.

error_code (CharField, nullable) The Amazon S3 Error Code, or '-' if no error occurred.")

bytes_sent (PositiveIntegerField, nullable) The number of response bytes sent, excluding HTTP protocol overhead, or '-' if zero.

object_size (PositiveIntegerField, nullable) The total size of the object in question.

total_time (PositiveIntegerField, nullable) The number of milliseconds the request was in flight from the server's perspective. This value is measured from the time your request is received to the time that the last byte of the response is sent. Measurements made from the client's perspective might be longer due to network latency.

turnaround_time (PositiveIntegerField, nullable) The number of milliseconds that Amazon S3 spent processing your request. This value is measured from the time the last byte of your request was received until the time the first byte of the response was sent.

referrer (TextField, nullable) The value of the HTTP Referrer header, if present. HTTP user-agents (e.g. browsers) typically set this header to the URL of the linking or embedding page when making a request.

user_agent (TextField, nullable) The value of the HTTP User-Agent header.

version_id (CharField, nullable) The version ID in the request, or '-' if the operation does not take a versionId parameter.

tamarin.models.S3LoggedBucket

The `S3LoggedBucket` class represents a bucket that is being logged.

name (CharField) The name of the bucket that is being logged. This is where the media that is being served resides.

log_bucket_name (CharField) The S3 bucket to monitor for log keys. This must be a separate bucket from the logged bucket. The contents of this bucket must only be log files.

monitor_bucket (BooleanField) When checked, pull logs from this bucket's log bucket.