

# Importing Logs in JSON Format

You can import JSON formatted logs from Treasure Agent (td-agent), to continuously import the access logs into the cloud.

**td-agent handles log-rotation.** td-agent keeps a record of the last position of the log, ensuring that each line is read exactly once even if the td-agent process goes down. However, because the information is kept in a file, the "exactly once" guarantee breaks down if the file becomes corrupted.

- [Prerequisites](#)
- [Installing td-agent](#)
- [Modifying td-agent.conf](#)
  - [Confirming Data Import](#)
- [Next Steps](#)

## Prerequisites

td-agent : td-agent must have permission to read the logs.

## Installing td-agent

td-agent is under the [Fluentd project](#). td-agent extends Fluentd with custom plugins for Treasure Data.

'td-agent' must be installed on your application servers. td-agent is a daemon program dedicated to the streaming upload of any kind of the time-series data. td-agent is developed and maintained by Treasure Data.

To set up td-agent, refer to the following articles; we provide deb/rpm packages for Linux systems.

If you have...	Refer to...
MacOS X	<a href="#">Installing td-agent on MacOS X</a>
Ubuntu System	<a href="#">Installing td-agent for Debian and Ubuntu</a>
RHEL / CentOS System	<a href="#">Installing td-agent for Redhat and CentOS</a>
AWS Elastic Beanstalk	<a href="#">Installing td-agent on AWS Elastic Beanstalk</a>

## Modifying td-agent.conf

Specify your authentication key by setting the `apikey` option. You can view your API key from the TD Console.

Access `/etc/td-agent/td-agent.conf` to set the `apikey` option.

`YOUR_API_KEY` should be your API key string.

```
# Tailing the JSON formatted Logs
<source>
  type tail
  format json
  tag td.production.foo
  path /path/to/the/file/foo.json
  pos_file /var/log/td-agent/foo.pos
</source>

# Treasure Data Input and Output
<match td.*.*>
  type tdlog
  endpoint api.treasuredata.com
  apikey YOUR_API_KEY
  auto_create_table
  buffer_type file
  buffer_path /var/log/td-agent/buffer/td
  use_ssl true
</match>
```

Restart your agent when the following lines are in place.

```
$ sudo /etc/init.d/td-agent restart
```

td-agent tails the file, buffers the log (*var/log/td-agent/buffer/td*), and automatically uploads the log into the cloud.

## Confirming Data Import

The following example is a sample log file. Every time a new line is appended to the log file, td-agent parses the line and adds the record to the buffer. td-agent uploads the data into the cloud every 5 minutes. To upload the data immediately, send a SIGUSR1 signal.

```
$ tail -n 5 /path/to/the/file/foo.json
{"a"=>"b", "c"=>"d"}
{"a"=>"b", "c"=>"d", "e"=>1}
{"a"=>"b", "c"=>"d", "e"=>1, "f"=>2.0}
{"a"=>"b", "c"=>"d"}
{"a"=>"b", "c"=>"d", "e"=>1}
```

Issue the following commands to confirm that everything is configured correctly.

```
# append new entries
$ tail -n 3 /path/to/the/file/foo.json > sample.txt # take the last three lines of the log...
$ cat sample.txt >>/path/to/the/file/foo.json # and append them to the buffer file to trigger the tail plugin.

# flush the buffer
$ kill -USR1 `cat /var/run/td-agent/td-agent.pid`
```

To confirm that your data has been uploaded successfully, issue the `td tables` command as follows.

```
$ td tables
+-----+-----+-----+-----+
| Database | Table   | Type | Count |
+-----+-----+-----+-----+
| production | foo     | log  | 3     |
+-----+-----+-----+-----+
```

Check `/var/log/td-agent.log` if it's not working correctly. `td-agent:td-agent` must have permission to read the logs.

## Next Steps

We offer a schema mechanism that is more flexible than that of traditional RDBMSs. For queries, we leverage the Hive Query Language.

- [Schema Management](#)
- [Hive Query Language](#)
- [Programmatic Access with REST API and its Bindings](#)