

# Python Apps Import Integration

Treasure Data provides provides [td-agent](#) to collect server-side logs and events and to seamlessly import the data from Python applications.

Continue to the following topics:

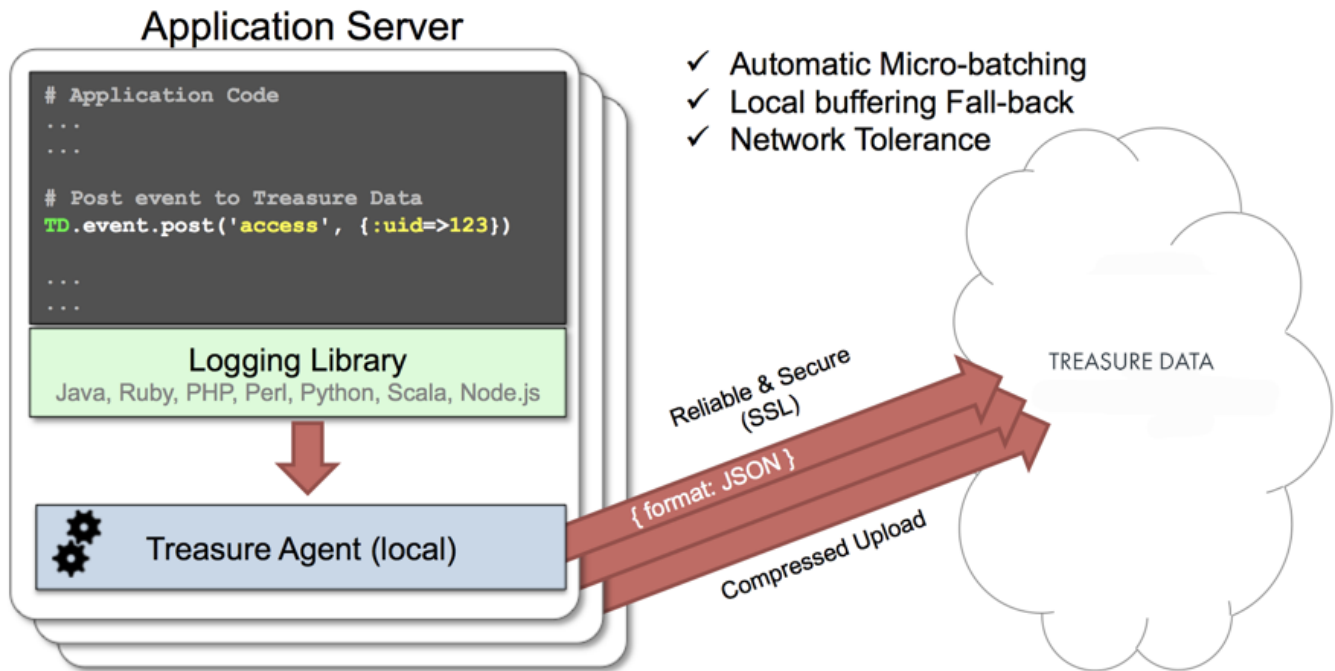
- [Prerequisites](#)
- [Installing td-agent](#)
  - [td-agent Install Options](#)
  - [Modifying /etc/td-agent/td-agent.conf](#)
  - [Using fluent-logger-python](#)
  - [Confirm the Import](#)
- [Production Deployments](#)
  - [High Availability Configurations of td-agent](#)
  - [Monitoring td-agent](#)

## Prerequisites

- Basic knowledge of Python.
- Basic knowledge of Treasure Data, including the [TD Toolbelt](#).
- Python 2.6 or higher (for local testing).

## Installing td-agent

Install `td-agent` on your application servers. `td-agent` sits within your application servers, focusing on uploading application logs to the cloud.



The [fluent-logger-python](#) library enables Python applications to post records to their local `td-agent`. `td-agent`, in turn, uploads the data to the cloud every 5 minutes. Because the daemon runs on a local node, the logging latency is negligible.

## td-agent Install Options

To install `td-agent`, run one of the following commands based on your environment. The agent program is installed automatically by using the package management software for each platform like `rpm/deb/dmg`.

### RHEL/CentOS 5,6,7

```
$ curl -L https://toolbelt.treasuredata.com/sh/install-redhat-td-agent3.sh | sh
```

## Ubuntu and Debian

```
# 18.04 Bionic
$ curl -L https://toolbelt.treasuredata.com/sh/install-ubuntu-bionic-td-agent3.sh | sh
# 16.04 Xenial (64bit only)
$ curl -L https://toolbelt.treasuredata.com/sh/install-ubuntu-xenial-td-agent3.sh | sh
```

Legacy support for EOL versions is still available

```
# 14.04 Trusty
$ curl -L https://toolbelt.treasuredata.com/sh/install-ubuntu-trusty-td-agent3.sh | sh
# 12.04 Precise
$ curl -L https://toolbelt.treasuredata.com/sh/install-ubuntu-precise-td-agent3.sh | sh
# Debian Stretch (64-bit only)
$ curl -L https://toolbelt.treasuredata.com/sh/install-debian-stretch-td-agent3.sh | sh
# Debian Jessie (64-bit only)
$ curl -L https://toolbelt.treasuredata.com/sh/install-debian-jessie-td-agent3.sh | sh
# Debian Squeeze (64-bit only)
$ curl -L https://toolbelt.treasuredata.com/sh/install-debian-squeeze-td-agent2.sh | sh
```

## Amazon Linux

You can choose Amazon Linux 1 or Amazon Linux 2. Refer to [Installing td-agent on AWS Linux](#).

## MacOS X 10.11+

```
$ open 'https://td-agent-package-browser.herokuapp.com/3/macosx'
```

MacOS X 10.11.1 (El Capitan) introduces some security changes. After the td-agent is installed, edit the `/Library/LaunchDaemons/td-agent.plist` file to change `/usr/sbin/td-agent` to `/opt/td-agent/usr/sbin/td-agent`.

## Windows Server 2012+

The Windows installation requires the steps detailed in:

- [Installing Treasure Agent using .msi Installer \(Windows\)](#)

## Opscode Chef Repository

```
$ echo 'cookbook "td-agent"' >> Berksfile
$ berks install
```

[AWS Elastic Beanstalk](#) is also supported. Windows is NOT supported.

## Modifying /etc/td-agent/td-agent.conf

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

```
# Treasure Data Input and Output
<source>
  type forward
  port 24224
</source>

<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
</match>
```

`YOUR_API_KEY` should be your actual apikey string. Using a [write-only API key](access-control#rest-apis-access) is recommended.

Restart your agent after the following lines are added.

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

# MacOS X
$ sudo launchctl unload /Library/LaunchDaemons/td-agent.plist
$ sudo launchctl load /Library/LaunchDaemons/td-agent.plist
```

td-agent now accepts data via port 24224, buffers the (`var/log/td-agent/buffer/td`), and automatically uploads the data into the cloud.

## Using fluent-logger-python

Install the fluent-logger library via pip.

```
$ pip install fluent-logger
```

Initialize and post the records as follows.

```
# Initialize
from fluent import sender
from fluent import event
sender.setup('td.test_db', host='localhost', port=24224)
event.Event('follow', {
  'from': 'userA',
  'to': 'userB'
})
```

## Confirm the Import

Execute the program.

```
$ python test.py
```

Sending a SIGUSR1 signal flushes td-agent's buffer. Upload starts immediately.

```
# Linux
$ kill -USR1 `cat /var/run/td-agent/td-agent.pid`

# MacOS X
$ sudo kill -USR1 `sudo launchctl list | grep td-agent | cut -f 1`
```

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

```
$ td tables
+-----+-----+-----+-----+
| Database | Table   | Type | Count |
+-----+-----+-----+-----+
| test_db  | follow  | log  | 1     |
+-----+-----+-----+-----+
```

## Production Deployments

### Use gunicorn, tornado or modwsgi

We recommend that you use [gunicorn](#), [tornado](#) or [modwsgi](#). Other setups have not been fully validated.

### High Availability Configurations of td-agent

For high-traffic websites (more than 5 application nodes), use a high availability configuration of td-agent to improve data transfer reliability and query performance.

- [High-Availability Configurations of td-agent](#)

### Monitoring td-agent

Monitoring td-agent itself is also important. For general monitoring methods for td-agent, see:

- [Monitoring td-agent](#)

td-agent is fully open-sourced under the [Fluentd project](#).