**Datadog:**

What is Datadog?

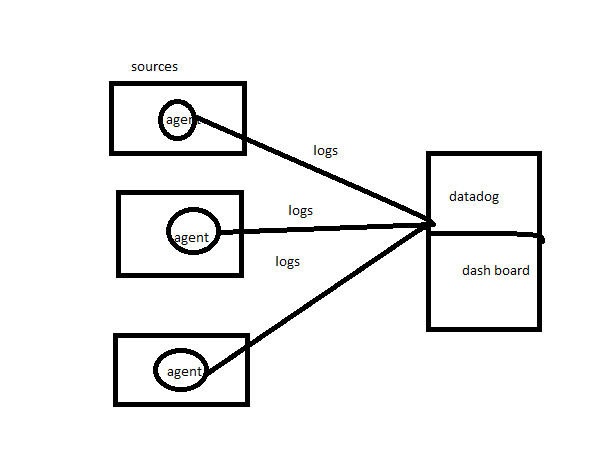
* Data dog is a company name.
* It is a monitoring software tool.

What kind of monitoring:

* Infra monitoring
* Cloud monitoring
* Log monitoring - Splunk
* APM (Application performance monitoring) - newrelic
* RUM (Real user monitoring)
* Container monitoring
* Security monitoring

Architecture of Datadog:

* Monitoring as a service-based cloud
* Agent is installed in your server



Datadog terminology:

Agent

DogstatsD

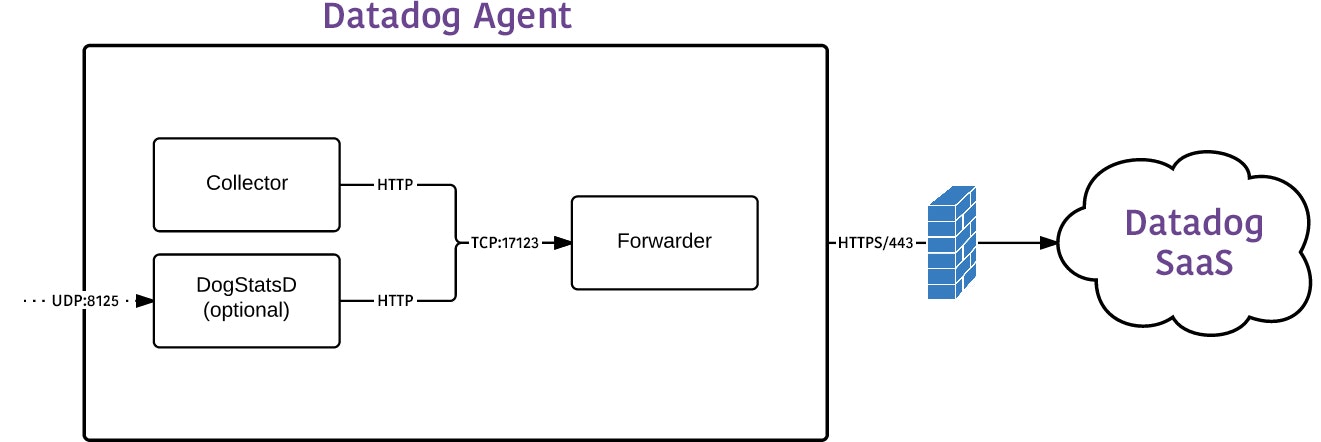
StatsD

Metrics

Agent:

* The Agent is lightweight software installed on your hosts.
* It reports metrics and events from your host to Datadog using [integrations](https://docs.datadoghq.com/integrations/), [DogStatsD](https://docs.datadoghq.com/metrics/dogstatsd_metrics_submission/), or the [API](https://docs.datadoghq.com/api/).
* With additional setup, the Agent can report [live processes](https://docs.datadoghq.com/infrastructure/process/), [logs](https://docs.datadoghq.com/logs/), and [traces](https://docs.datadoghq.com/tracing/).

How Datadog agent works?



[Collector](https://docs.datadoghq.com/agent/basic_agent_usage/?tab=agentv6v7#collector)

The collector gathers all standard metrics every 15 seconds. Agent v6 embeds a Python 2.7 interpreter to run integrations and [custom checks](https://docs.datadoghq.com/developers/custom_checks/write_agent_check/).

[Forwarder](https://docs.datadoghq.com/agent/basic_agent_usage/?tab=agentv6v7#forwarder)

The Agent forwarder send metrics over HTTPS to Datadog. Buffering prevents network splits from affecting metric reporting. Metrics are buffered in memory until a limit in size or number of outstanding send requests are reached. Afterwards, the oldest metrics are discarded to keep the forwarder’s memory footprint manageable. Logs are sent over an SSL-encrypted TCP connection to Datadog.

[DogStatsD](https://docs.datadoghq.com/agent/basic_agent_usage/?tab=agentv6v7#dogstatsd)

While StatsD accepts only metrics, DogStatsD accepts all three of the major Datadog data types: metrics, events, and service checks

Rough:

* Adding your API key to the Agent configuration: /etc/datadog-agent/datadog.yaml
* Setting SITE in the Agent configuration: /etc/datadog-agent/datadog.yam
* [Your Agent is running and functioning properly. It will continue to run in the](https://app.datadoghq.com/account/settings" \l "agent/mac)

[background and submit metrics to Datadog.](https://app.datadoghq.com/account/settings" \l "agent/mac)

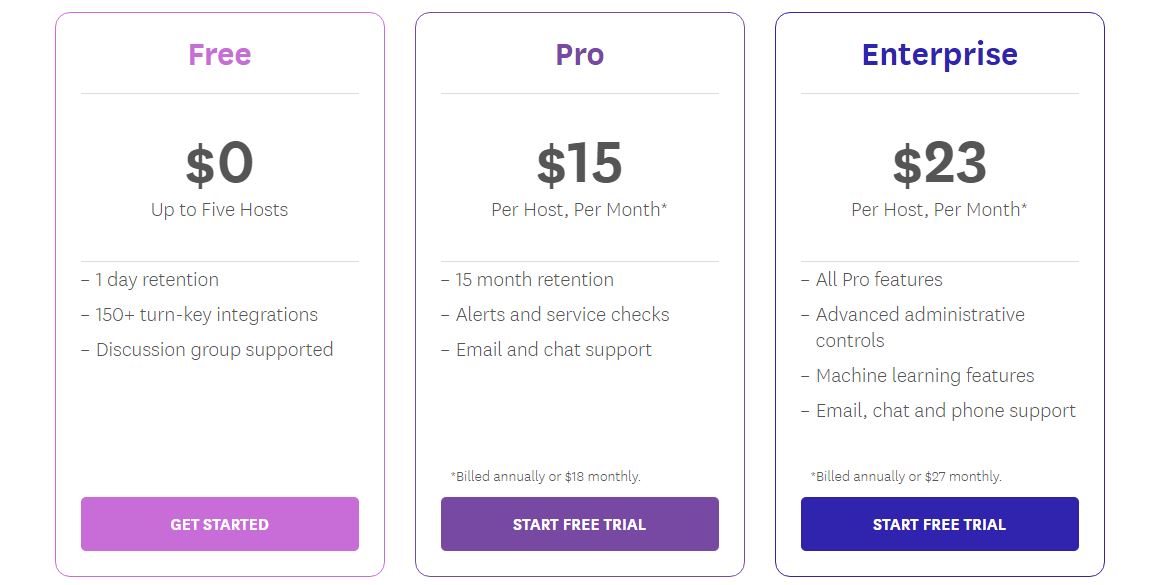
* [If you ever want to stop the Agent, run:](https://app.datadoghq.com/account/settings" \l "agent/mac)

[systemctl stop datadog-agent](https://app.datadoghq.com/account/settings" \l "agent/mac)

[And to run it again run:](https://app.datadoghq.com/account/settings" \l "agent/mac)

[systemctl start datadog-agent](https://app.datadoghq.com/account/settings" \l "agent/mac)

pricing



[Installing Agent on Amazon Linux](https://app.datadoghq.com/account/settings" \l "agent/windows)

* [one-step install](https://app.datadoghq.com/account/settings" \l "agent/windows)

DD\_AGENT\_MAJOR\_VERSION=7 DD\_API\_KEY=f9e3a5dd75a8f2652bc9eef785faa9f7 DD\_SITE="datadoghq.com" bash -c "$(curl -L <https://s3.amazonaws.com/dd-agent/scripts/install_script.sh>)"

| **DESCRIPTION** | **COMMAND** |
| --- | --- |
| Start Agent as a service | sudo systemctl start datadog-agent |
| Stop Agent running as a service | sudo systemctl stop datadog-agent |
| Restart Agent running as a service | sudo systemctl restart datadog-agent |
| Status of Agent service | sudo systemctl status datadog-agent |
| Status page of running Agent | sudo datadog-agent status |

[Integrate apache in datadog:](https://app.datadoghq.com/account/settings" \l "agent/debian)

Installation

The Apache check is packaged with the [Datadog Agent](https://app.datadoghq.com/account/settings#agent). To start gathering your Apache metrics and logs, you need to:

* [Install the Agent](https://docs.datadoghq.com/agent/) on your Apache servers.
* Install mod\_status on your Apache servers and enable ExtendedStatus. ()

Rough: 1. sudo vi /etc/httpd/conf/httpd.conf

2. <Location /server-status>

SetHandler server-status

Order deny,allow

Deny from all

Allow from 127.0.0.1

</Location>

ExtendedStatus On

3. httpd -t

4.sudo systemctl restart httpd

5.curl <http://localhost/server-status>

6. curl http://localhost/server-status/?refresh=5

**Configuration**

**Host**

To configure this check for an Agent running on a host:

**Metric collection**

1.Edit the apache.d/conf.yaml file in the conf.d/ folder at the root of your [Agent's configuration directory](https://docs.datadoghq.com/agent/guide/agent-configuration-files/#agent-configuration-directory) to start collecting your Apache metrics. See the [sample apache.d/conf.yaml](https://github.com/DataDog/integrations-core/blob/master/apache/datadog_checks/apache/data/conf.yaml.example) for all available configuration options.

init\_config:

instances:

## @param apache\_status\_url - string - required

## Status url of your Apache server.

#

- apache\_status\_url: http://localhost/server-status?auto

2.[Restart the Agent](https://docs.datadoghq.com/agent/guide/agent-commands/#start-stop-and-restart-the-agent).

**Log collection**

*Available for Agent versions >6.0*

1.Collecting logs is disabled by default in the Datadog Agent. Enable it in datadog.yaml:

logs\_enabled: true

2.Add this configuration block to your apache.d/conf.yaml file to start collecting your Apache logs, adjusting the path and service values to configure them for your environment:

logs:

- type: file

path: /path/to/your/apache/access.log

source: apache

service: apache

sourcecategory: http\_web\_access

- type: file

path: /path/to/your/apache/error.log

source: apache

service: apache

sourcecategory: http\_web\_error

See the [sample apache.d/conf.yaml](https://github.com/DataDog/integrations-core/blob/master/apache/datadog_checks/apache/data/conf.yaml.example) for all available configuration options.

3.[Restart the Agent](https://docs.datadoghq.com/agent/guide/agent-commands/#start-stop-and-restart-the-agent).