**Search for:**

* springboot.product.views.count
* springboot.api.request.time

**Use filters:**

* product:abc
* env:dev

**DogStatsD is a high-performance, UDP-based metrics listener built into the Datadog Agent.**

**It's an extended version of** [**StatsD**](https://github.com/statsd/statsd)**, designed by Datadog, that lets you:**

**🔧 Push custom application metrics (counters, timers, gauges, etc.) from your app to the local Datadog Agent over UDP port 8125.**

**🧠 How DogStatsD Works (Architecture)**

**🎯 Your App (e.g., Spring Boot):**

* **Sends metrics like product.views.count:1|c|#env:dev,product:P123**
* **These are fire-and-forget UDP packets — fast and low overhead.**

**📡 Datadog Agent (on host):**

* **Listens on localhost:8125**
* **Receives, parses, batches, and forwards metrics to Datadog’s backend**
* **Adds host-level tags automatically (host:my-server, region:us-east)**

**🧾 Datadog Backend (Cloud):**

**What Types of Metrics DogStatsD Supports**

| **Metric Type** | **Use Case** | **Example** |
| --- | --- | --- |
| Counter | Increment events | e.g., orders.placed.count |
| Gauge | Current value | e.g., queue.depth, memory.used |
| Timer / Histogram | Duration tracking | e.g., api.response.time |
| Set | Unique elements | e.g., unique.users.daily |
| Event | Notify incidents | e.g., deploy.failed |
| Service Check | Health state | e.g., db.connection: OK / CRITICAL |

**🧰 When Should You Use DogStatsD?**

**✅ Use it when:**

* You want fast, real-time application-level metrics.
* You want to send dynamic tags (per user, product, etc.).
* Your app is short-lived (cron job, batch job) and cannot be scraped by Prometheus.
* You want minimal code — no need to manage HTTP requests manually.
* You’re already running the Datadog Agent on the host/containe

**✅ 1. Verify DogStatsD is Enabled in Datadog Agent Config**

Check the Datadog Agent config file:

**sudo grep dogstatsd\_port /etc/datadog-agent/datadog.yaml**

You should see:

**dogstatsd\_port: 8125**

**sudo systemctl restart datadog-agent**

**✅ 2. Verify UDP Port 8125 Is Listening**

Run this command to check that DogStatsD is listening on localhost:8125:

**sudo netstat -ulnp | grep 8125**

**sudo lsof -iUDP:8125**

✅ This confirms DogStatsD is active and listening.

**sudo datadog-agent check dogstatsd**

**sudo datadog-agent status**

**✅ 3. Send a Test Metric Using dog CLI Tool**

Use Datadog’s command-line utility dog to send a test DogStatsD metric:

**dog statsd metric send --metric test.dogstatsd.metric --type count --value 1 --tags env:test**

If dog is not installed, install the [Datadog CLI](https://github.com/DataDog/datadog-cli):

pip install datadog-cli

dog config --api-key <YOUR\_API\_KEY> --app-key <YOUR\_APP\_KEY>

**✅ 4. Check Agent Logs for Metric Receipt**

Run:

**✅ 4. Check Agent Logs for Metric Receipt**

Run:

**sudo journalctl -u datadog-agent -f**

Then hit your Spring Boot endpoint (or send a test metric). You should see lines like:

DogStatsD: received metric: springboot.product.views.count

✅ This confirms the Datadog Agent received metrics from your app.

**✅ 5. Verify in Datadog Web UI**

Go to Datadog Metrics Explorer:

* Search: springboot.product.views.count or test.dogstatsd.metric
* Filter by tag: env:test (if used)
* Set time range: Last 5 minutes