

Ideo-lab

November 2025



Datadog



**DATADOG**

## 1) Datadog en 2 lignes

Plateforme SaaS d'observabilité et de sécurité couvrant **infrastructure, APM/Traces, Logs, RUM & Synthetics, Network/DB monitoring, Cloud Security/CSM & SIEM, Cost monitoring**, etc. Tout est piloté par l'Agent **Datadog** (hosté ou conteneurisé) et un modèle de **tags** pour agréger/segmenter les métriques, logs et

## 2) Éditions & modules (vue rapide)

- **Infrastructure Monitoring** (Pro / Enterprise / variantes DevSecOps) – base hôte/conteneur, intégrations 750+, tableaux de bord, alerting. Datadog +1
- **APM + Continuous Profiler** – traces distribuées, flamegraphs, RUM et corrélation avec logs/metrics. Datadog
- **Log Management** – ingestion, pipelines, filtrage/sampling, rehydratation à la demande. Datadog
- **Synthetics & RUM** – tests browser/API, SLO. Datadog
- **Network/DB Monitoring** – flux, latence, requêtes, plans d'exécution. Datadog
- **Security (CSM, SIEM, ASM)** – posture cloud, détection d'événements, vulnérabilités appli. Datadog

## 3) Prix (ordre de grandeur 2025)

Les prix varient par produit et facturation (mensuelle/annuelle).

- **Infra Pro**: ~\$15/host/mo (annuel), \$18 on-demand. **Infra Enterprise**: ~\$23–27/host/mo. Variantes DevSecOps plus élevées. Datadog
  - Ordres de grandeur tiers pour **APM+Profiler** : à partir d'environ \$31/host/mo (références d'analyses externes — vérifier dans votre région & contrat). signoz.io +2
- 👉 **Recommandation**: utilisez la **page pricing officielle Datadog** et/ou le **simulateur commercial** pour votre périmètre exact (sites US/EU, remises volume). Datadog

## 4) Installation de l'Agent

### 4.1 Linux (Ubuntu/Debian)

1. Dans l'UI Datadog, allez sur **Agent Installation** > **Ubuntu** pour générer le script (inclut votre `DD_API_KEY` et `DD_SITE`). [docs.datadoghq... +1](#)
2. Exemple générique (sudo) :

```
bash
```

 Copier le code

```
DD_API_KEY="<votre_api_key>" DD_SITE="eu.datadoghq.com" \  
bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_script.sh)"  
# Vérifier le statut  
sudo datadog-agent status
```

Chemin de conf principal : `/etc/datadog-agent/datadog.yaml`. Intégrations : `/etc/datadog-agent/conf.d/`.

[docs.datadoghq...](#)

### 4.2 Windows

- Installeur MSI depuis l'UI; conf dans `%ProgramData%\Datadog\datadog.yaml`, service **DatadogAgent** (Services.msc). [docs.datadoghq...](#)

### 4.3 Docker (Agent en conteneur)

bash

 Copier le code

```
docker run -d --name dd-agent --restart unless-stopped \  
  -e DD_API_KEY=<key> -e DD_SITE=eu.datadoghq.com \  
  -e DD_APM_ENABLED=true -e DD_LOGS_ENABLED=true \  
  -v /var/run/docker.sock:/var/run/docker.sock:ro \  
  -v /proc:/host/proc:ro -v /sys/fs/cgroup:/host/sys/fs/cgroup:ro \  
  gcr.io/datadoghq/agent:latest
```

(Variables d'env à adapter; en K8s, préférer Helm/Operator ci-dessous.) [docs.datadoghq...](#)

### 4.4 Kubernetes (Helm/Operator)

- Helm chart officiel :

bash

 Copier le code

```
helm repo add datadog https://helm.datadoghq.com  
helm repo update  
helm upgrade --install datadog -n datadog --create-namespace datadog/datadog \  
  --set datadog.apiKey=<key> \  
  --set datadog.site=eu.datadoghq.com \  
  --set datadog.logs.enabled=true \  
  --set datadog.apm.enabled=true \  
  --set clusterAgent.enabled=true \  
  --set kubeStateMetricsCore.enabled=true
```

- Datadog Operator : gestion avancée via CRDs (↓ pour grands clusters). [GitHub +2](#)

## 5) Setup initial & paramètres clés

### 5.1 `datadog.yaml` (hôte ou Agent conteneur)

- `site`: `eu.datadoghq.com` (ou `datadoghq.com`, `us3`, `us5`, `ap1`, etc.)
- `tags`: normalisez (`env:prod`, `service:portal`, `team:platform`, `region:eu-west-1`).
- `apm_config`: `apm_enabled: true`
- `logs_config`: `logs_enabled: true` (+ sources via fichiers, sockets, ou Docker/K8s auto)
- `process_config`: `enabled: "true"` (process & container monitoring)
- `dogstatsd`: `non_local_traffic: true` si nécessaire.

Chemins & structure détaillés ici. [docs.datadoghq...](#)

### 5.2 Activer des intégrations (exemples)

- `Nginx`: `/etc/datadog-agent/conf.d/nginx.d/conf.yaml` → host/port + status URL
- `PostgreSQL/MariaDB`: `postgres.d/conf.yaml` / `mysql.d/conf.yaml` → creds en lecture
- `Redis, RabbitMQ, Elasticsearch, etc.` (modèles `conf.yaml.example` à copier/adapter). [docs.datadoghq...](#)

### 5.3 Dashboards, monitors, SLO

- Créez des `template variables` (tags) pour filtrer par `env`, `service`, `kube_namespace`.
- `Alerting`: seuils, anomalies, forecast; regroupez par `service` / `host`.
- `SLO`: objectifs par service (availability, latency p95). (Base produit Datadog). [Datadog](#)

## 6) Commandes & opérations courantes

- **Statut Agent** : `sudo datadog-agent status` (ou `datadog-agent status` dans conteneur). [docs.datadoghq...](#)
- **Vérif de conf** : `sudo datadog-agent configcheck`
- **Diagnostics (flare)** : `sudo datadog-agent flare` (pack support)
- **Service** : `sudo systemctl restart datadog-agent` / `start` / `stop`
- **Logs de l'Agent** : `/var/log/datadog/agent.log` (Linux), équivalents Windows. [docs.datadoghq...](#)

## 7) Champs d'application typiques (exemples concrets)

- **Stack web Django** (Ubuntu + Nginx + MariaDB/PostgreSQL) : metrics host, APM Python (uWSGI/Gunicorn), NGINX logs, DB monitoring (temps requêtes, locks), SLO et déploiements.
- **Kubernetes multi-tenant** : autodécouverte par tags, USM (Universal Service Monitoring), APM + Profiling, Network Monitoring (connexions, latence), Synthetics pour endpoints critiques.
- **Sécurité Cloud** : posture (CSM), menaces (SIEM), ASM (AppSec) avec corrélations.
- **FinOps observability** : cartes de coûts et **Optimisation de volumes de logs** (drop/sampling /rehydration). [Datadog](#)

## 8) Bonnes pratiques (rapides)

- **Tagging dès le départ** : imposez un schéma (`env`, `service`, `version`, `team`, `region`).
- **Monitors par SLO** : n'alertez que sur l'utilisateur final (p95/p99, availability, erreurs).
- **Cost control logs** : pipelines, filtres d'ingestion, durées de rétention adaptées, rehydratation ad hoc.
- **Dashboards modulaires** : 1 "Golden Signals" + 1 "Deep Dive" par service.
- **Infra as Code** : définissez dashboards/monitors via API/Terraform/Helm values. (Écosystème Datadog/Helm). [GitHub](#)

## 9) Références utiles (docs officielles)


- Getting Started Agent (différences host vs container, exemples) ; Linux platforms ; K8s install ; Operator ; Fichiers de configuration. [docs.datadoghq...](#) +4

## 10) Alternatives & positionnement (lecture tierce)

Bon pour contexte d'achat: Datadog est souvent plébiscité pour la richesse des intégrations et l'UX, mais l'onboarding et le coût peuvent grimper avec l'ampleur. Alternatives citées: LogicMonitor, Dynatrace, New Relic, Zabbix/Nagios, etc. [TechRadar](#) +2

## A) `datadog.yaml` (exemple minimal multi-signal)

yaml

 Copier le code

```
site: eu.datadoghq.com
api_key: "<VOTRE_API_KEY>"
tags:
  - env:prod
  - service:portal
  - team:platform
apm_config:
  enabled: true
logs_config:
  enabled: true
process_config:
  enabled: "true"
dogstatsd_non_local_traffic: true
```

## B) Intégration Nginx ( `/etc/datadog-agent/conf.d/nginx.d/conf.yaml` )

yaml

 Copier le code

```
init_config:
instances:
  - nginx_status_url: http://127.0.0.1:8080/nginx_status
  tags:
    - service:frontend
    - env:prod
```

## C) Helm values (K8s) – extrait

yaml

 Copier le code

```
datadog:
  apiKey: "<KEY>"
  site: "eu.datadoghq.com"
  logs:
    enabled: true
    containerCollectAll: true
  apm:
    enabled: true
  processAgent:
    enabled: true
  clusterAgent:
    enabled: true
  kubeStateMetricsCore:
    enabled: true
```









