



elastic

Search. Observe. Protect.

# Présentation de l'APM

Fonctionnement, avantages et mise en place.

[elastic.co/fr](https://elastic.co/fr)

# Table des matières

Pourquoi le monitoring des performances applicatives (APM) est-il nécessaire ?	3
Les applications, moteurs des entreprises	3
Une mauvaise performance provoque des pertes de chiffre d'affaires	4
Les applications cloud natives modernes sont complexes	4
Qu'est-ce que l'APM ?	5
Visibilité de bout en bout	5
Détection de la latence et des erreurs	6
L'APM au-delà de la production	6
Comment fonctionne l'APM	7
Agents APM	7
Instrumentation et configuration	8
Analyse	8
Termes clés	9
Traçage distribué	9
Intervalles	10
Transactions	10
Traces	11
Real User Monitoring (monitoring des utilisateurs réels)	11
Choisir un outil APM	12
Capacités techniques	12
Simplicité d'utilisation	13
Options de déploiement	13
Compatibilité avec les standards ouverts et les données ouvertes	13
Architecture et scalabilité	14
Sécurité	14
Capacités au-delà de l'APM	15
Tarifs	15
À propos d'Elastic	16



# Pourquoi le monitoring des performances applicatives (APM) est-il nécessaire ?

*Petite remarque : Tout au long de ce guide, nous allons nous intéresser au [monitoring des performances applicatives \(APM\)](#), qui est un pilier de l'[observabilité](#). Avec les logs et les indicateurs, l'APM joue un rôle essentiel dans la construction de systèmes observables.*

## Les applications, moteurs des entreprises

Les applications sont le visage public des entreprises modernes. C'est par le biais de ces applications que nous interagissons avec les produits et les services, que ce soit la vitrine d'une boutique en ligne, une application de covoiturage ou les différents outils de collaboration et de productivité que nous utilisons au quotidien. Lorsque nous ajoutons un casque audio à notre panier d'achats en ligne ou lorsque nous créons la playlist idéale pour un dîner, nous interagissons avec des applications et nous nous en faisons une opinion. La perception, la fidélisation et l'évaluation d'une marque reposent sur ces expériences.



# Une mauvaise performance provoque des pertes de chiffre d'affaires

Pensez aux applications que vous utilisez au travail ou sur vos appareils personnels. À quoi pensez-vous en premier ? Sont-elles faciles à utiliser ? Répondent-elles rapidement ? Rencontrent-elles souvent des erreurs ? Les réponses à ces questions influencent-elles votre perception des entreprises qui ont créé ces offres ?

Lorsque la performance des applications n'est pas à la hauteur des attentes, les entreprises subissent des conséquences directes. Le chiffre d'affaires et la confiance des clients sont en jeu.

Si un panier d'achats numérique ne fonctionne pas, les utilisateurs se rendront sur une autre boutique, ce qui entraîne une perte de ventes. Si les temps de réponse sont régulièrement longs ou si les erreurs sont fréquentes, même les utilisateurs fidèles peuvent décider de cesser complètement d'utiliser le produit ou le service. Chaque entreprise connaît les conséquences de l'indisponibilité sur ses résultats finaux. Le temps moyen de détection (MTTD, mean time to detect) et le temps moyen de réponse (MTTR, mean time to respond) ont un prix.



## Les applications cloud natives modernes sont complexes

L'évolution du développement et de la livraison des applications et la complexité qui découle de ce changement font du monitoring complet des performances un besoin encore plus urgent.

L'adoption des architectures en microservices a changé la manière dont on conçoit et maintient les applications, en offrant une livraison continue et une échelle qu'on ne pouvait imaginer au préalable. Mais la nature distribuée et polyglotte (développement avec des langages et frameworks multiples) de ces architectures ajoute un niveau de complexité. Les principes de la livraison continue et de l'automatisation ont permis des déploiements de code fréquents, mais le suivi de l'impact sur les performances est devenu encore plus important, afin que les problèmes puissent être corrigés rapidement ou que les modifications puissent être annulées si un déploiement ne se passait pas comme prévu.

Les principes de développement cloud natif permettent aux équipes d'offrir d'excellentes expériences numériques à leurs utilisateurs en continu, de réagir rapidement aux retours et de pivoter selon les besoins. Les utilisateurs ignorent tout des mécanismes qui fonctionnent en arrière-plan pour soutenir chacune de leurs requêtes. Alors, comment les équipes peuvent-elles garder un œil sur ces applications ?

Nous sommes humains, et nous ne pouvons pas traiter un tel volume de données de manière significative. Nous avons besoin d'une vue organisée avec une trajectoire claire vers les informations. C'est là que l'APM entre en jeu.

# Qu'est-ce que l'APM ?

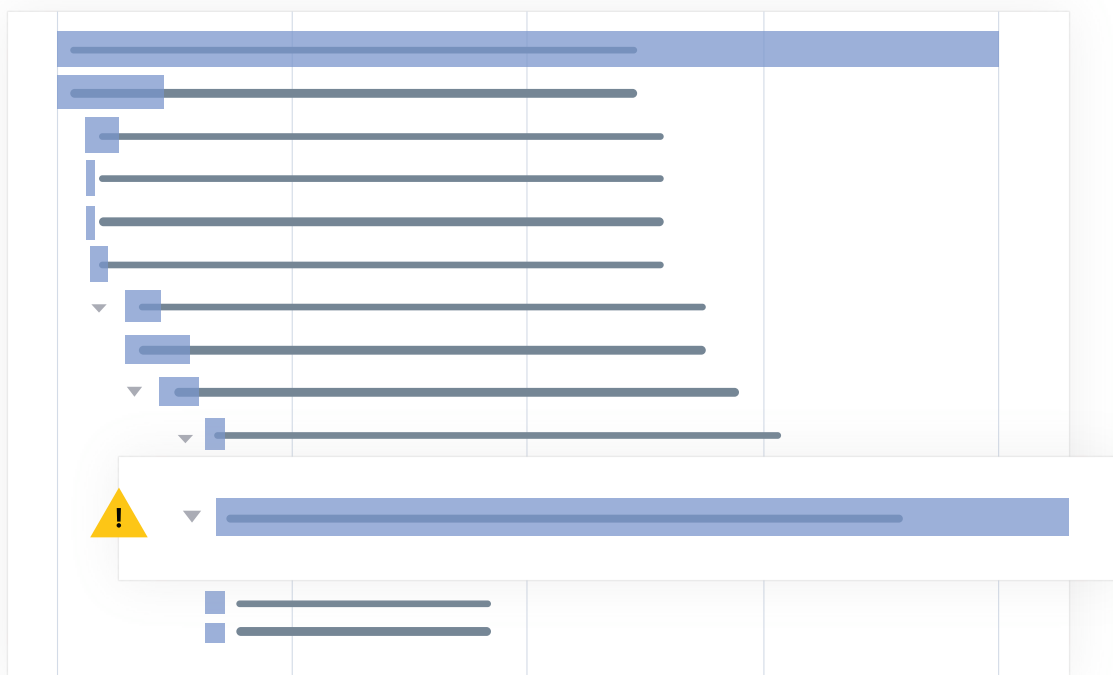
L'APM est la solution qui permet de collecter, de monitorer et d'analyser les performances de bout en bout et le comportement des transactions par le biais de vos applications (et des services dont elles dépendent). L'APM suit l'expérience de l'utilisateur, du moment où il commence à interagir avec l'application jusqu'à celui où il a atteint le résultat qu'il souhaite.

Grâce à ses liens directs avec l'expérience utilisateur, l'APM aide à relier l'informatique aux objectifs commerciaux. Le rapport n'est peut-être pas exactement de 1:1, mais il existe une corrélation positive. En fournissant des informations continues sur la performance de vos applications, l'APM vous permet d'être proactif, au lieu de réagir à un problème après une avalanche de plaintes de clients via votre système de tickets, ou même sur les réseaux sociaux.

## Visibilité de bout en bout

Une fois une application instrumentée, les données générées vous indiquent exactement ce qui se passe à l'intérieur. L'APM suit les transactions tout au long de leur parcours : chaque demande et chaque réponse sont enregistrées et mesurées, quelle que soit la complexité de votre architecture.

En reliant toutes ces traces, l'APM offre une vision complète des performances de votre application, d'une vue panoramique de vos services (et de leurs interactions) jusqu'aux informations au niveau du code.



*Mais... n'avons-nous pas déjà des logs qui nous indiquent ce qui se passe à l'intérieur des applications ? Il est vrai que les logs nous fournissent des données précieuses et des informations contextuelles sur les types d'erreurs ou d'autres événements qui se produisent. Toutefois, les logs ne montrent pas la totalité de l'expérience client, de bout en bout. Les traces APM (et le traçage distribué) permettent ainsi d'obtenir ces détails. En associant les traces aux logs et aux indicateurs, vous obtenez une visibilité unifiée de votre écosystème.*

## Détection de la latence et des erreurs

L'APM monitorise deux attributs qui interviennent dans l'expérience utilisateur : la durée d'une transaction et les erreurs. Combien de temps les requêtes HTTP prennent-elles habituellement ? Combien d'erreurs 500 rencontre-t-on habituellement en une heure ?

Les problèmes de latence surviennent lorsque la réponse à une requête prend plus de temps que d'habitude. Par exemple, un utilisateur clique pour afficher un produit, et il faut plus de quelques secondes à la réponse pour se charger.

Les erreurs, bien entendu, signifient qu'un résultat inattendu s'est produit. Cela se produit lorsque la requête n'est pas bien réalisée.

L'objectif final de toute analyse est de corriger les problèmes. Pour le faire efficacement, les équipes réalisent une analyse de la cause première afin de déterminer exactement le composant à l'origine du problème. Peut-on identifier rapidement le service qui constitue le goulet d'étranglement des performances ? Peut-on localiser précisément l'appel de fonction ou la méthode responsable du problème ? Avec l'APM, la réponse à ces deux questions est oui.

## L'APM au-delà de la production

On parle souvent de l'APM dans le contexte de l'expérience utilisateur des applications en phase de production, mais l'APM intervient plus tôt dans le processus, avec des impacts sur plusieurs équipes et environnements. Par exemple, l'APM fournit aux développeurs les retours dont ils ont besoin pour créer un code permettant d'atteindre une boucle développement-test-déploiement plus rapide.

Les entreprises utilisent l'APM à plusieurs étapes afin d'aider les équipes à concevoir, à vérifier la qualité, à déployer et à monitorer des applications de manière plus efficace. En utilisant l'APM dans les procédés de plusieurs équipes, les entreprises peuvent améliorer leurs processus et créer des workflows plus efficaces, afin de consacrer davantage de temps à l'innovation (et moins de temps à éteindre des incendies).

Nous avons parlé des avantages de l'APM ; voyons maintenant comment il fonctionne.

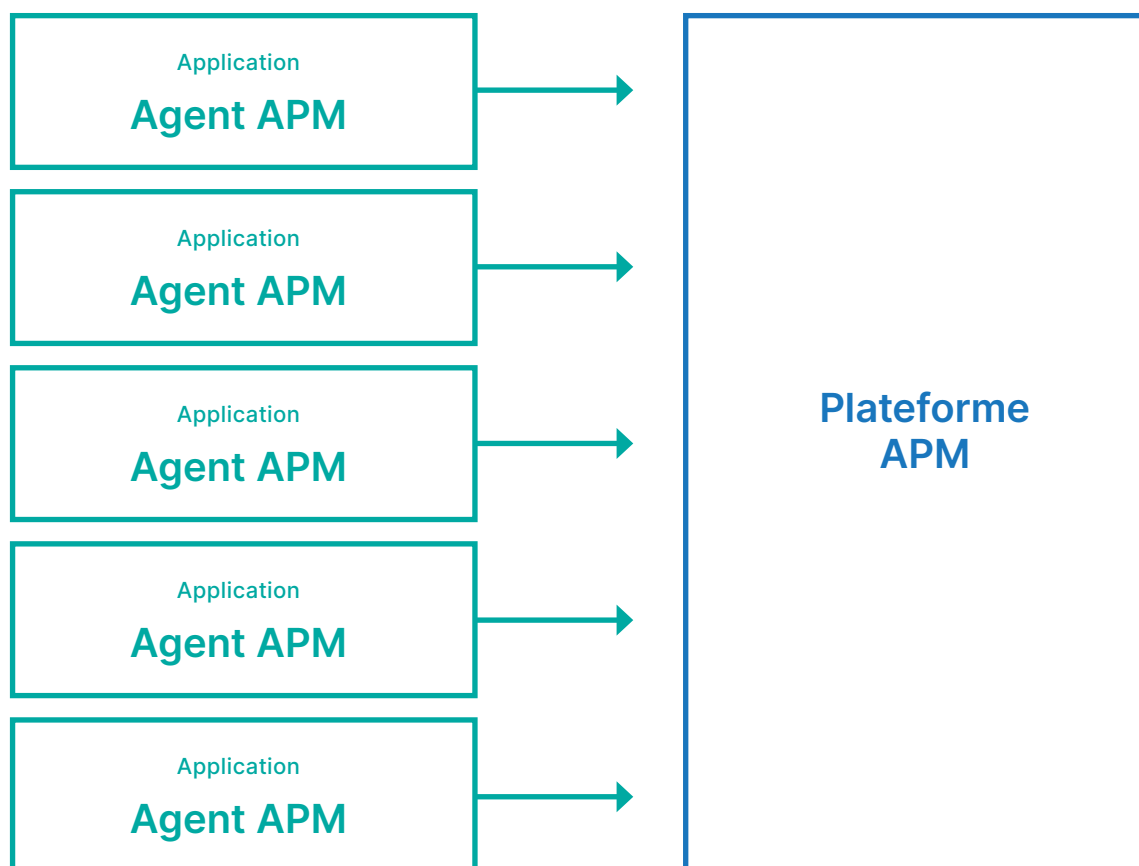
# Comment fonctionne l'APM

L'APM fonctionne en instrumentant chaque aspect de votre système que vous souhaitez monitorer (hôtes, conteneurs, applications, etc.), puis en envoyant les données de performances recueillies à l'emplacement de votre choix afin de les analyser.

## Agents APM

Un agent APM est une bibliothèque, un plug-in ou une extension qui monitorise les indicateurs de performances décrits plus haut. Selon ce que vous avez besoin de monitorer (et le langage dans lequel cet élément a été développé), vous aurez peut-être besoin de plusieurs agents. Après avoir identifié tous les éléments que vous souhaitez monitorer, vous déployez les agents APM sur chacun d'entre eux. Bien que les agents varient selon les fournisseurs, la plupart des agents instrumentent votre code, recueillent des données de performance, puis envoient les données à un serveur ou à un récupérateur.

### Machine en périphérie



# Instrumentation et configuration

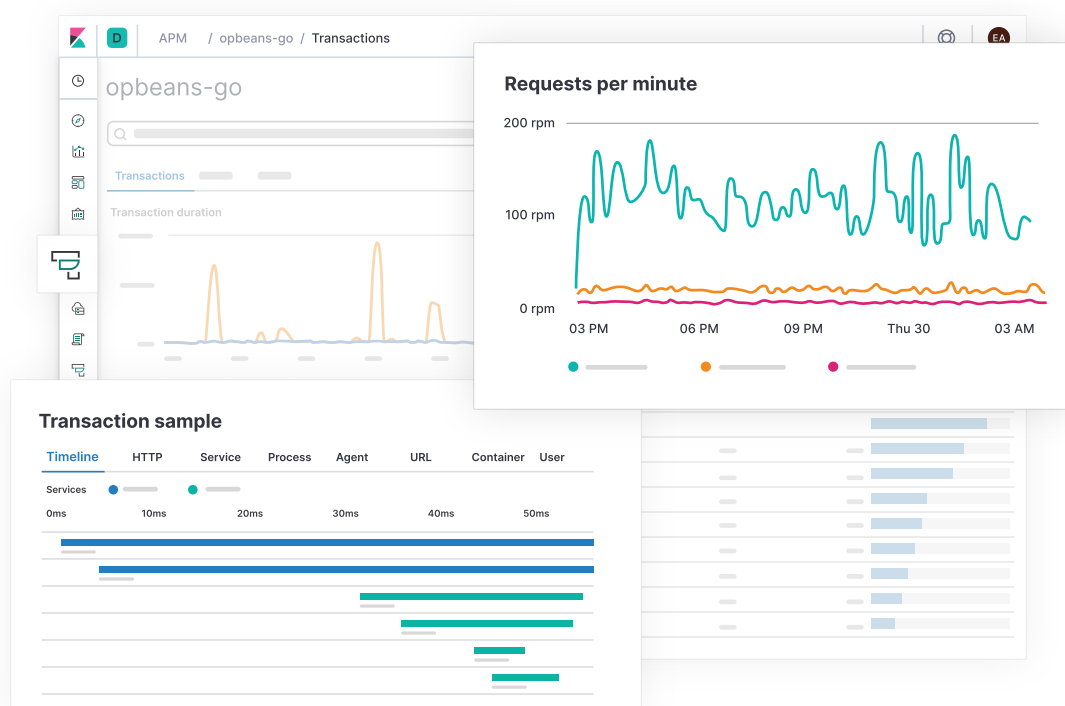
L'instrumentation est le processus qui consiste à étendre le code de votre application pour rapporter les données des traces. Après avoir installé le ou les agents nécessaires, vous pouvez instrumenter les éléments spécifiques de votre application que vous souhaitez analyser afin de recueillir et d'envoyer les données de performance à l'emplacement de votre choix (en général, un point de terminaison déterminé par votre outil APM).

Vous voudrez sans doute configurer des éléments tels que les noms d'environnements, les taux d'échantillonnage, les instrumentations et les indicateurs afin d'aider vos équipes à identifier et à analyser facilement les données diffusées dans votre outil APM. En général, cela se fait à l'aide de l'API et de l'interface utilisateur de l'outil, ou directement dans les variables de l'environnement.

## Analyse

Une fois les données de performance envoyées à l'emplacement de votre choix, vous pouvez passer à l'analyse des données. La plupart des outils comportent une interface utilisateur qui aide à identifier les erreurs, les problèmes de latence et autres anomalies qui touchent vos utilisateurs.

C'est un excellent point de départ pour analyser les problèmes rapportés ou, mieux encore, pour détecter les problèmes avant qu'ils n'impactent vos utilisateurs. Identifiez précisément les services (jusqu'au niveau du code) qui rencontrent des problèmes pour accélérer l'analyse de la cause principale et réduire le MTTR.







# Termes clés

Chaque outil APM peut avoir une définition un peu différente de ces termes, mais en général, ils enregistrent les mêmes données pour fournir à l'utilisateur des informations sur la performance de ses applications.

## Traçage distribué

En traçant toutes les demandes, de la requête web initiale au service front-end aux requêtes adressées aux services back-end, le traçage distribué vous permet d'analyser la performance de toute l'architecture de vos microservices en une seule vue.

Le traçage distribué permet de repérer facilement les goulots d'étranglement en affichant les événements complets par service, puis par requête dans ce service. Faire remonter les erreurs et autres problèmes de manière exploitable au niveau du code permet d'accélérer les analyses et de réduire le MTTR.



# Intervalles

Chaque unité ou élément du workflow constitue un intervalle. Les intervalles sont ce que l'on voit généralement dans la vue en cascade d'un outil d'analyse APM, généralement représentés sous la forme de barres horizontales. Ces segments sont au cœur du traçage distribué. Les intervalles mesurent du début à la fin d'une activité et contiennent des informations sur l'exécution d'un chemin de code spécifique.

Les attributs communs d'un intervalle comprennent :

- Heure de début
- Heure de fin
- Un nom
- Un type

## Transactions

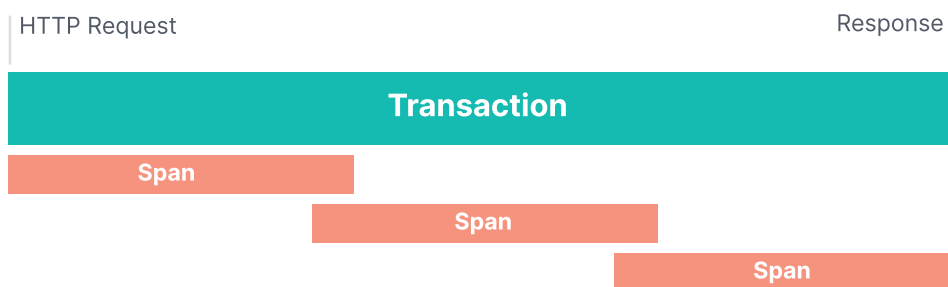
Les transactions sont un type d'intervalle décrivant un événement et pouvant également inclure plusieurs intervalles.

Voici quelques exemples de transactions :

- Requête adressée à votre serveur
- Tâche en lot
- Tâche en arrière-plan

Des attributs supplémentaires sont associés aux transactions, par exemple, des données sur l'environnement dans lequel l'événement est enregistré :

- Service : environnement, framework, langage, etc.
- Hôte : architecture, nom d'hôte, IP, etc.
- Processus : args, PID, PPID, etc.
- URL : complète, domaine, port, requête, etc.
- Utilisateur (s'il est fourni) : adresse e-mail, ID, nom d'utilisateur, etc.

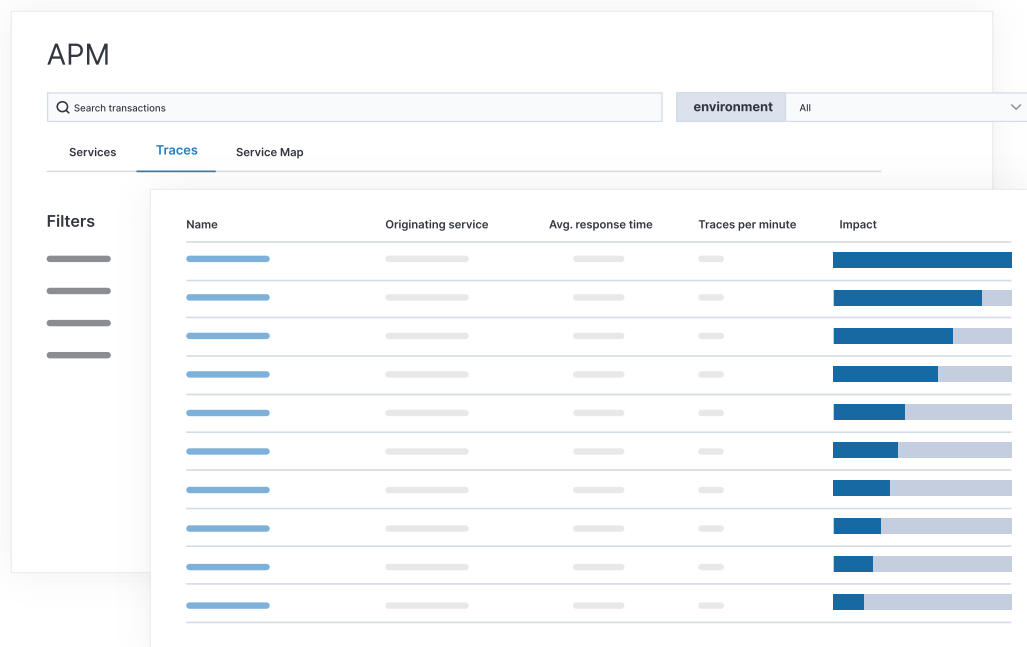


# Traces

Les traces mesurent l'état et la durée d'une requête réalisée au moyen de votre application. Ensemble, les intervalles et les transactions forment une trace, qui relie les points de la requête de l'utilisateur jusqu'à la réponse.

Vous pouvez observer des traces individuelles pour des actions terminées, comme le traitement d'un paiement, le traitement d'une commande terminée et la mise à jour du statut d'expédition.

Associées aux logs et aux indicateurs de l'application (et aux autres aspects de votre infrastructure), les traces offrent une visibilité totale sur l'ensemble de votre écosystème.



## Real User Monitoring (monitoring des utilisateurs réels)

Le monitoring des utilisateurs réels (RUM) enregistre les interactions directes des utilisateurs avec les clients dans les environnements de production. Par exemple, le RUM vous indique exactement comment un utilisateur interagit avec votre site web, en précisant les temps de réponse et les erreurs rencontrées. Pour cela, un agent sur le navigateur de l'utilisateur recueille les données de performance et les envoie à l'outil APM en vue de leur analyse.

# Choisir un outil APM

Comme lors de tout processus de sélection d'outil, commencez par définir clairement vos exigences. Lors de cette étape, tenez compte de votre croissance et de vos besoins futurs. Planifiez une preuve de faisabilité afin de vous assurer que l'outil répond réellement à vos besoins. Les capacités démontrées et commercialisées par le fournisseur fonctionnent-elles comme prévu dans votre environnement ? Une preuve de faisabilité aide à éviter les surprises et les coûts cachés en aval du processus.

## Capacités techniques

Créez une liste de contrôle des fonctionnalités requises en tenant compte de vos besoins techniques spécifiques. Soyez rigoureux et précis pour définir vos exigences. La liste suivante n'est pas exhaustive, mais elle contient quelques pistes de réflexion :



### Compatibilité des langages et des frameworks

- L'outil APM est-il compatible avec le(s) langage(s) et les frameworks largement déployés sur votre environnement ?



### Largeur et profondeur de la visibilité

- Offre-t-il une visibilité de la performance au niveau du code ?
- Permet-il une visibilité de bout en bout de votre suite technologique ?



### Capacités de recherche, d'analyse et d'exploration visuelle

- Comprend-il une interface utilisateur soignée et des workflows d'analyse réactifs ?
- Dispose-t-il d'outils visuels tels que des graphiques en cascade et des cartes de dépendances permettant d'étudier la performance des applications distribuées ?
- Offre-t-il un langage de requête flexible (et rapide) permettant des analyses ponctuelles ?
- Offre-t-il un framework de visualisation flexible au-delà des tableaux de bord standard prévus par le fournisseur ?



### Intégration dans votre écosystème au sens large

- S'intègre-t-il harmonieusement avec les autres outils standard de vos environnements ?  
Par exemple, PagerDuty pour la réponse aux incidents, ServiceNow pour la gestion des cas ou Slack pour les notifications. Vos frameworks d'intégration et de livraison continues vous aident-ils à suivre automatiquement l'impact de vos exécutions de code et de vos nouveaux déploiements sur les performances ?

## Simplicité d'utilisation

En étudiant vos exigences en matière de fonctionnalités techniques, évaluez la simplicité d'utilisation et l'accessibilité de ces fonctionnalités. Pensez aux différentes personnes qui vont interagir avec l'outil et tenez compte de leur expérience, de leurs attentes et de leurs workflows pour répondre aux questions sur la simplicité d'utilisation. Votre évaluation ne doit pas s'arrêter à l'expérience de démarrage, mais se prolonger dans la surcharge opérationnelle continue.

- Combien de temps faut-il pour instrumenter et obtenir des informations initiales ?
- L'interface utilisateur est-elle intuitive et facile à utiliser ?
- Quel est l'effort requis pour déployer la solution ?
- Est-il facile de mettre la plateforme à niveau ? Et les agents ?
- Les agents peuvent-ils être gérés centralement à partir d'un seul endroit ?
- Quel est l'effort requis pour scaler (vers le haut ou vers le bas) ?

## Options de déploiement

Il est important que l'outil APM choisi soit compatible avec vos préférences en matière de consommation logicielle.

- Préférez-vous une option SaaS pour réduire vos coûts opérationnels et administratifs ?
- Voulez-vous opter pour une solution pouvant vous aider à réduire les dépenses annuelles engagées pour le fournisseur cloud que vous avez choisi (AWS, Azure, Google Cloud, etc.) ?
- Avez-vous besoin d'une solution auto-gérée, car le cloud n'est pas possible pour des raisons de coûts ou de conformité ?
- Si vous avez une stratégie multi-cloud ou hybride, voulez-vous exécuter votre solution APM plus près de votre charge de travail pour réduire les coûts ou les latences du transfert des données ?

## Compatibilité avec les standards ouverts et les données ouvertes

L'espace de l'observabilité commence à se développer et à accepter les standards ouverts comme OpenTelemetry (formé par la fusion de deux autres standards ouverts, OpenTracing et OpenCensus) comme le framework d'instrumentation standard indépendant du fournisseur. L'objectif de ces initiatives est d'aider les développeurs à exporter facilement leur application sur une autre solution APM sans avoir à procéder de nouveau à l'instrumentation.

Si vous disposez déjà d'applications instrumentées au moyen d'un standard ouvert comme Jaeger, le passage à un outil compatible peut accélérer et simplifier la migration. Les standards ouverts aident également à assurer la pérennité de votre investissement.

## Architecture et scalabilité

Évaluez si l'outil APM que vous avez choisi repose sur une fondation et une architecture robustes afin de gérer votre taille actuelle et votre évolution. Votre capacité à analyser et résoudre rapidement et efficacement les problèmes de performances applicatives repose sur la performance de votre outil APM. Ne manquez pas de réaliser un test de charge de votre outil APM pendant le processus d'évaluation (ou d'examiner les données d'évaluation comparative si ce n'est pas possible) afin de vous assurer qu'il pourra gérer le volume de requêtes et d'ingestion prévu dans votre environnement sans ployer sous son poids.

- Repose-t-il sur une architecture simple ? Ou est-il basé sur un patchwork caché sous le capot et dont les composants finiront par lâcher ?
- Est-il compatible avec une disponibilité élevée ?
- Est-il facile de scaler pour gérer les pics de volumes de données de monitoring ?
- Existe-t-il des limites au volume (applications, indicateurs, requêtes, etc.) qu'il peut gérer ?

## Sécurité

Le processus d'évaluation de la sécurité doit faire partie intégrante de votre procédure d'évaluation d'outil. Pensez à étudier cette question sous les deux angles suivants :

1. L'engagement du fournisseur d'APM envers la sécurité relative à la conception et à la fourniture de l'outil
  - Les agents APM déployés sur vos applications nécessitent-ils des privilèges excessifs ?
  - Le trafic entre les composants internes est-il chiffré et sécurisé correctement ?
  - L'outil possède-t-il les certifications requises (surtout en SaaS) ?
  - L'outil utilise-t-il des extensions tierces ? Sont-elles sécurisées ?
2. La capacité à sécuriser et à contrôler l'accès à l'outil APM
  - L'outil s'intègre-t-il au système de gestion des identités et des accès de votre entreprise ?
  - L'outil est-il compatible avec un contrôle d'accès basé sur les rôles, avec un modèle de permissions granulaire ?

## Capacités au-delà de l'APM

Lors de l'évaluation d'outils, réfléchissez au-delà des limites typiques de l'APM. Le besoin de consolider les outils et d'obtenir une visibilité unifiée pousse de nombreuses entreprises à unifier l'APM avec les logs et les indicateurs (les deux autres "piliers de l'observabilité") pour simplifier les analyses et réduire le MTTR. Vérifiez si l'outil APM est compatible avec cette vision unifiée, grâce à une remontée des logs et des indicateurs dans le contexte des données de traces d'applications. Qu'en est-il de l'ingestion de données en dehors du paysage informatique typique, comme le sentiment social ou les demandes de support technique des clients ?

## Tarifs

Enfin, il est important d'examiner attentivement les options de tarification afin de vous assurer que l'outil que vous choisissez ne vous force pas à faire des compromis sur vos objectifs de visibilité ou de monitoring. Comme pour la plupart des autres critères, vous devrez pour cela tenir compte de votre utilisation et de votre architecture actuelles et futures. De nombreux facteurs interviennent dans la tarification des solutions APM : le nombre d'agents, le nombre d'hôtes, les ressources matérielles, etc. Certains fournisseurs imposent également des coûts supplémentaires lorsque vous dépassez certains seuils (par exemple, le nombre de conteneurs ou d'indicateurs). Voici quelques questions à poser sur les modèles de tarification :

- Le modèle de tarification correspond-il à vos besoins commerciaux et à vos choix en matière d'architecture ?
- Comment les coûts évolueront-ils avec la croissance que vous prévoyez et l'évolution de l'architecture (par exemple, passage d'une architecture monolithique aux microservices) ?
- Un niveau gratuit est-il disponible ? Que comprend le niveau gratuit ? Y a-t-il des limites d'utilisation ?
- Quel est le niveau de support technique compris ?



# À propos d'Elastic

Plateforme de logging open source la plus populaire, Elastic rend l'APM rapide, évolutif et pertinent.

Lorsque vous analysez un problème qui impacte les utilisateurs, chaque seconde est précieuse. Avec Elastic APM, vos données de performance sont enregistrées dans un index dans Elasticsearch, ce qui permet aux équipes de rechercher et d'identifier les goulots d'étranglement en temps réel. Elastic APM comporte également des cartes de service soutenues par le machine learning, des options d'alerting personnalisées, et bien plus encore, pour vous permettre d'offrir des expériences numériques optimales à vos utilisateurs. Rendez-vous sur [elastic.co/fr/apm](https://elastic.co/fr/apm) pour en savoir plus.

L'APM n'est qu'une pièce du puzzle. Unifiez vos logs, vos indicateurs et vos traces APM sur une seule et même plateforme pour parvenir à une véritable observabilité de tout votre écosystème. Rendez-vous sur [elastic.co/fr/observability](https://elastic.co/fr/observability) pour en savoir plus.

**Commencez votre essai gratuit aujourd'hui**