ASTIOLAB

MONITORING ET SUPERVISION

PLAN DU COURS

- Cours :
  - Enjeux du monitoring en production
  - Distinction supervision / monitoring
  - Principes, métriques clés et principaux outils
  - Alertes et analyse d’incidents
- Pratique :
  - Mise en place d’une stack de monitoring et surveillance de nos premiers serveurs

ENJEUX DU MONITORING EN PRODUCTION

POURQUOI SURVEILLER SES DONNÉES ?

LE MONITORING, C'EST QUOI ?

- Le **monitoring** consiste à **observer, mesurer et analyser en temps réel ou en continu** le fonctionnement d'un système, d'une application, d'une infrastructure ou d'un service. L'objectif est de s'assurer que tout fonctionne correctement, de détecter les anomalies, et d'agir rapidement en cas de problème.

DES EXEMPLES DE CE QUE JE PEUX SURVEILLER

- **Infrastructure :**
  - **Serveurs** : Utilisation du CPU, de la mémoire (RAM), de l'espace disque, température des composants.
  - **Réseau** : Bande passante, latence, paquets perdus, disponibilité des services (DNS, VPN, etc.).
  - **Matériel** : État des disques durs, des routeurs, des switchs, etc.

DES EXEMPLES DE CE QUE JE PEUX SURVEILLER

- **Applications :**
  - **Performances** : Temps de réponse, taux d'erreur, nombre de requêtes par seconde.
  - **Logs** : Journaux d'activité pour tracer les erreurs ou les comportements suspects.
  - **Disponibilité** : Vérifier qu'une application ou un service est accessible (ex : "Est-ce que mon site web est en ligne ?").

DES EXEMPLES DE CE QUE JE PEUX SURVEILLER

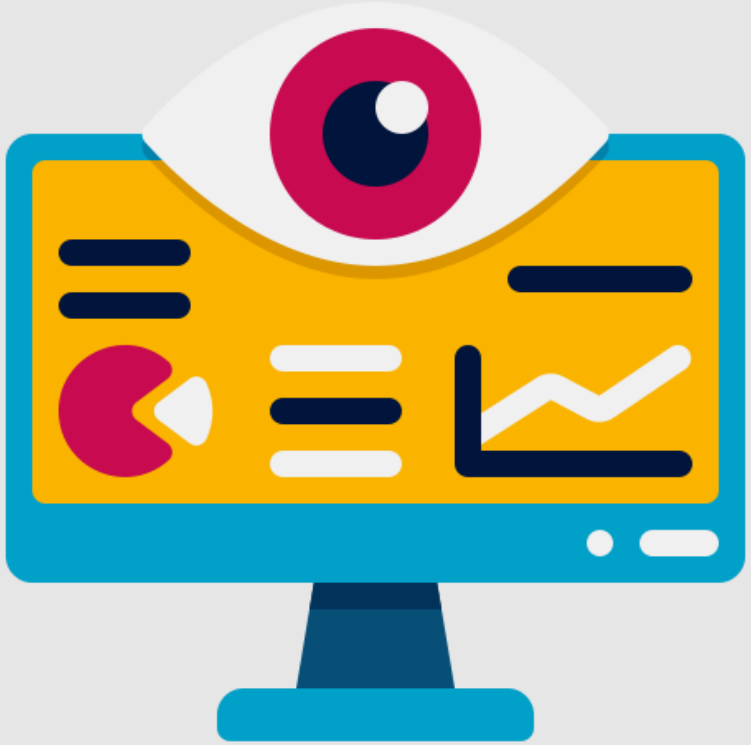
- **Expérience utilisateur :**
  - **Temps de chargement** des pages web ou des fonctionnalités.
  - **Parcours utilisateur** : Détection des étapes où les utilisateurs abandonnent (ex : panier d’achat non finalisé).

DES EXEMPLES DE CE QUE JE PEUX SURVEILLER

- **Sécurité**
  - **Activités suspectes** : Tentatives de connexion répétées, accès non autorisés.
  - **Vulnérabilités** : Détection de failles ou de comportements anormaux (ex : trafic inhabituel).

DISTINCTION SUPERVISION / MONITORING

QUELLE DIFFÉRENCE ?



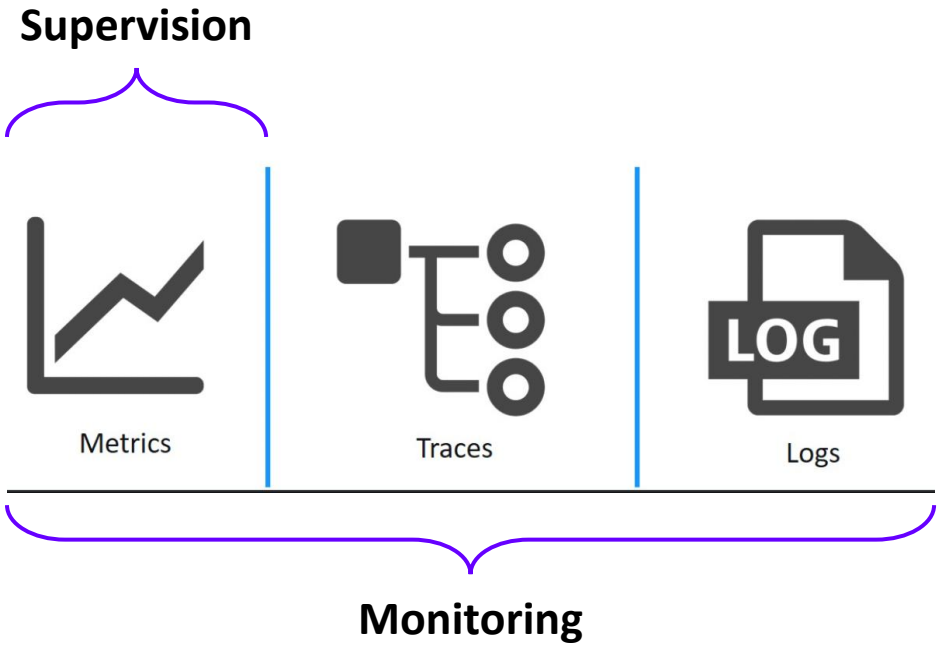
SUPERVISION

- Consiste à **surveiller en temps réel** l'état, les performances et la disponibilité d'un système, d'une infrastructure ou d'un service, **tout en mettant en œuvre des actions correctives ou préventives** pour en garantir le bon fonctionnement, l'optimisation et la résilience.

MONITORING

- **Processus de collecte, d’analyse et de visualisation de données** en temps réel ou quasi réel, visant à **évaluer l’état, les performances et la santé** d’un système, d’une application, d’une infrastructure ou d’un service.

DIFFÉRENCE



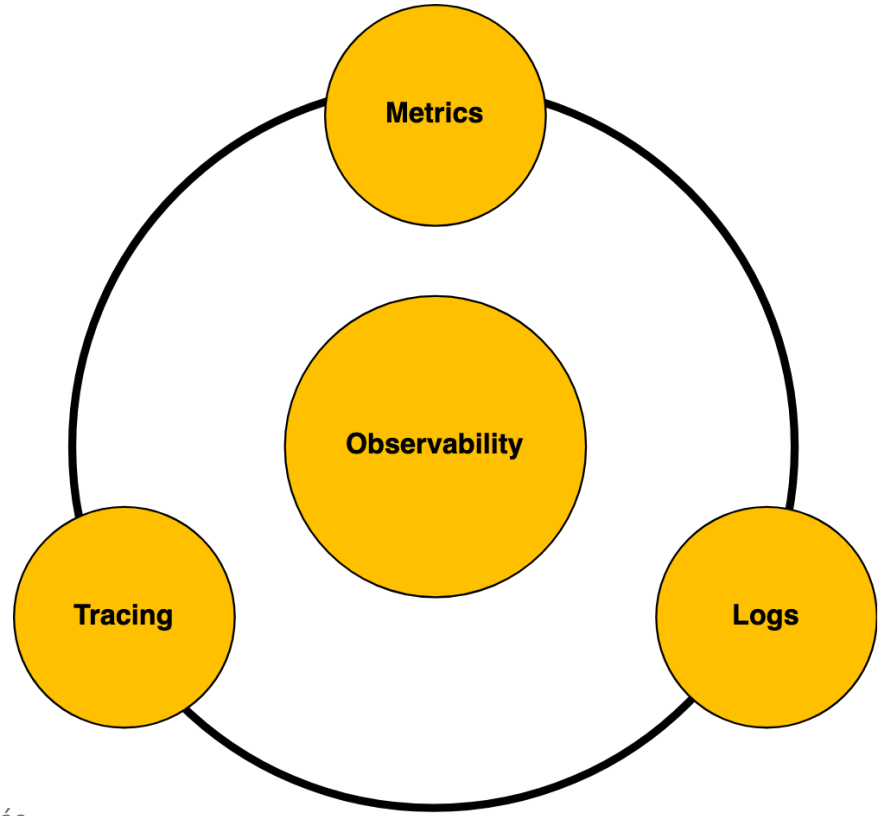
PRINCIPES, MÉTRIQUES CLÉS ET PRINCIPAUX OUTILS

GARDEZ UN ŒIL SUR VOTRE INFRA

PRINCIPES : OBSERVABILITÉ

- **Définition** : Capacité à comprendre l'état interne d'un système à partir de ses données externes (métriques, logs, traces).

- **Piliers** :
  - **Métriques** (ex. utilisation CPU).
  - **Logs** (ex. erreurs applicatives).
  - **Traces** (ex. latence par service).



PRINCIPES : PROACTIVITÉ

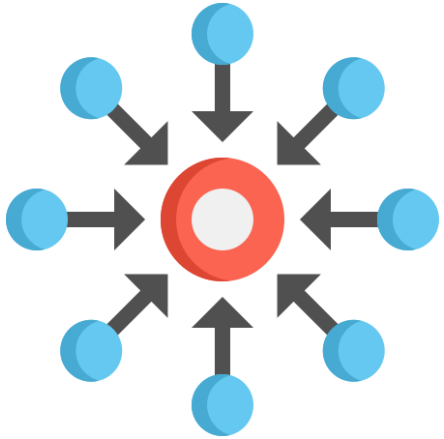
- **Détection précoce** des anomalies avant qu'elles n'impactent les utilisateurs.

- **Alertes intelligentes** : Réduction du bruit (ex. seuils dynamiques, regroupement d'alertes). Limiter le surplus d'alertes.

PRINCIPES : CENTRALISATION

- **Agrégation des données** issues de sources multiples (serveurs, applications, réseau) dans une plateforme unifiée.

- **Corrélation** des événements pour identifier les causes racines.



MÉTRIQUES CLÉS

- Infrastructure :

Métrique	Description	Seuil critique (exemple)
Utilisation CPU	% de CPU utilisé par un serveur ou un conteneur.	> 90% pendant 5 min.
Mémoire RAM	% de mémoire utilisée.	> 85%
Espace disque	Capacité restante sur les volumes de stockage.	< 10% libre.
Latence réseau	Temps de réponse entre deux points (ex. ping).	> 100 ms.
Débit réseau	Quantité de données transférées (ex. Mbps).	Saturation à 90% de la bande passante.

MÉTRIQUES CLÉS

- Applications :

Métrique	Description	Seuil critique (exemple)
Temps de réponse	Temps pour répondre à une requête (ex. API, page web).	> 2 secondes.
Taux d'erreur	% de requêtes aboutissant à une erreur (ex. HTTP 5xx).	> 1%.
Requêtes par seconde	Nombre de requêtes traitées.	Chute soudaine de 50%.
Latence applicative	Temps de traitement d'une requête (ex. temps SQL).	> 500 ms.

MÉTRIQUES CLÉS

- Expérience Utilisateur :

Métrique	Description	Seuil critique (exemple)
Temps de chargement	Temps pour afficher une page (ex. Largest Contentful Paint).	> 3 secondes.
Taux de rebond	% d'utilisateurs quittant la page sans interaction.	> 40%.
Erreurs frontend	Nombre d'erreurs JavaScript ou de plantages.	> 0.1% des sessions.

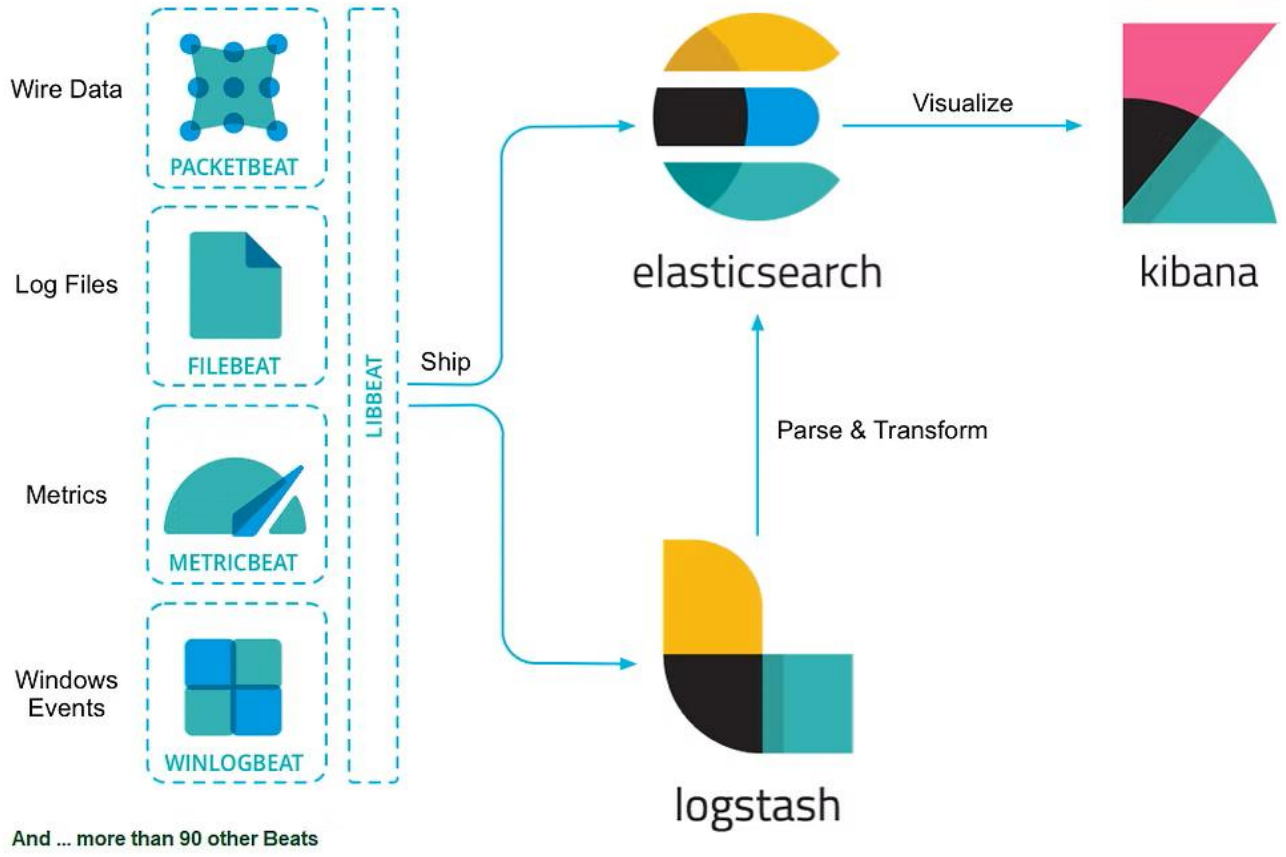
MÉTRIQUES CLÉS

- Sécurité :

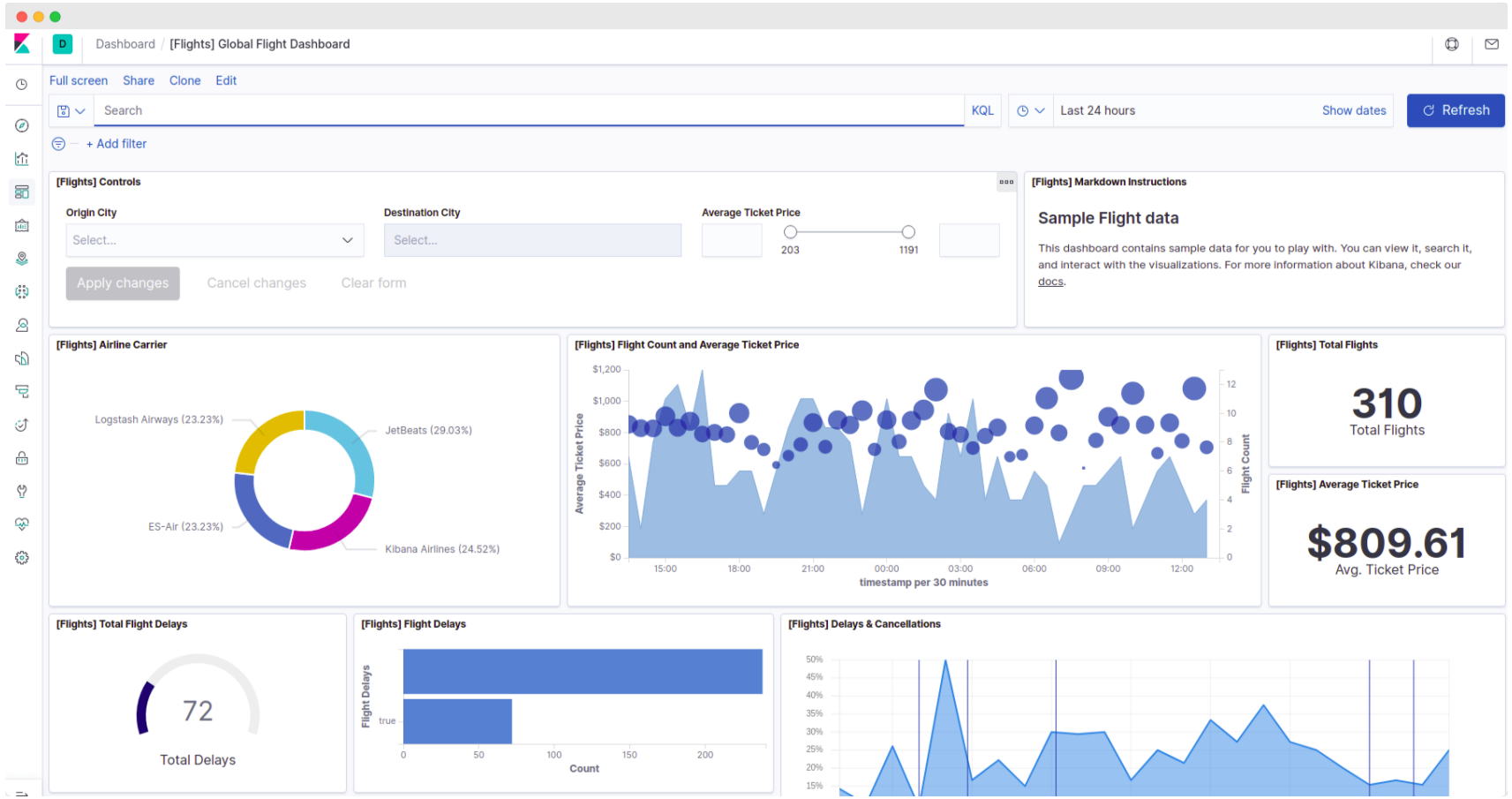
Métrique	Description	Seuil critique (exemple)
Tentatives de connexion	Nombre de tentatives infructueuses (ex. brute force).	> 10 tentatives/minute.
Traffic suspect	Requêtes provenant d'IPs blacklistées ou comportements anormaux.	Détection en temps réel.
Vulnérabilités	Nombre de failles non corrigées (ex. CVE).	Toute vulnérabilité critique (CVSS > 9).

PRINCIPAUX OUTILS POUR LE MONITORING

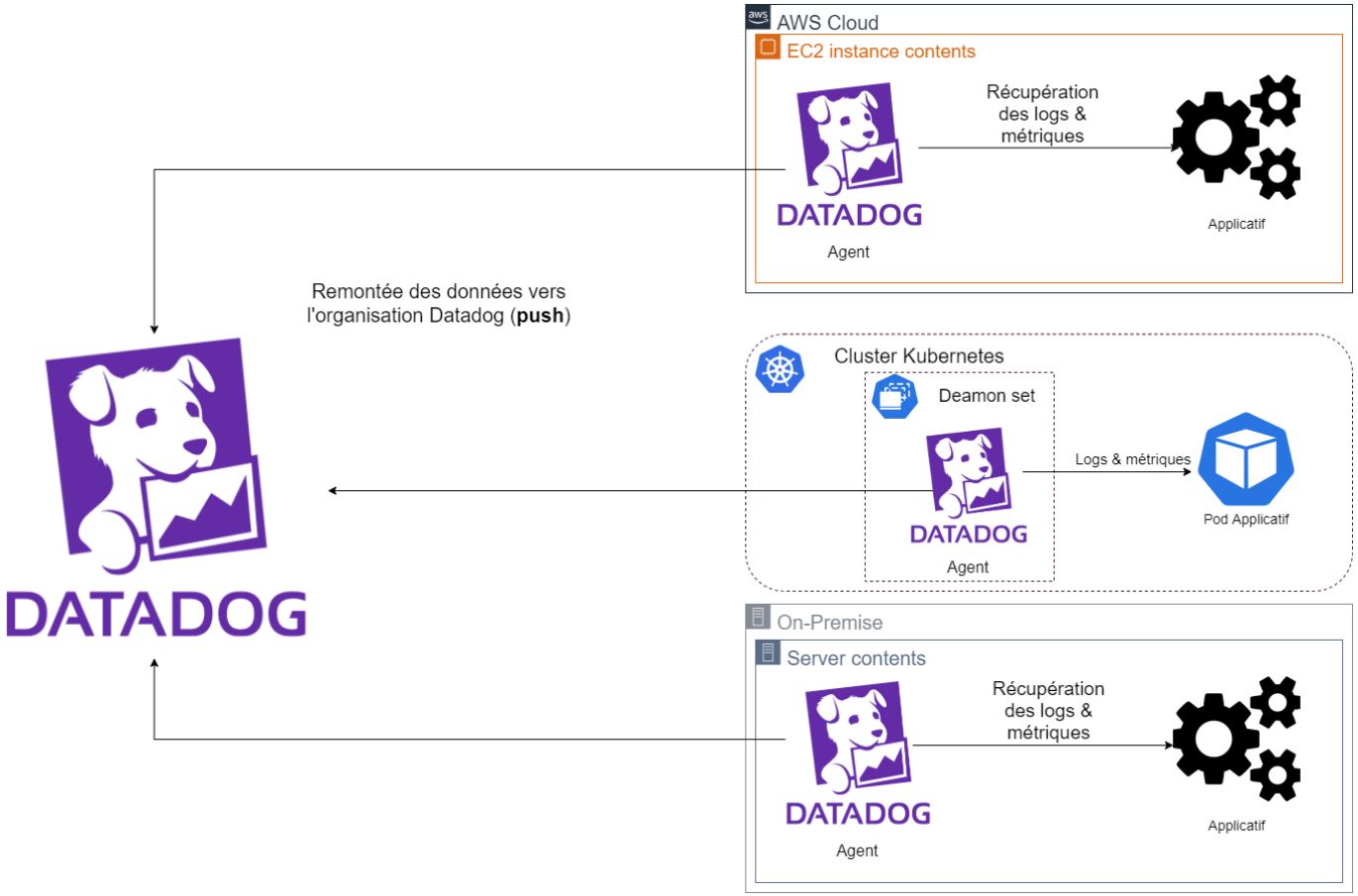
EXEMPLE DE STACK : ELK



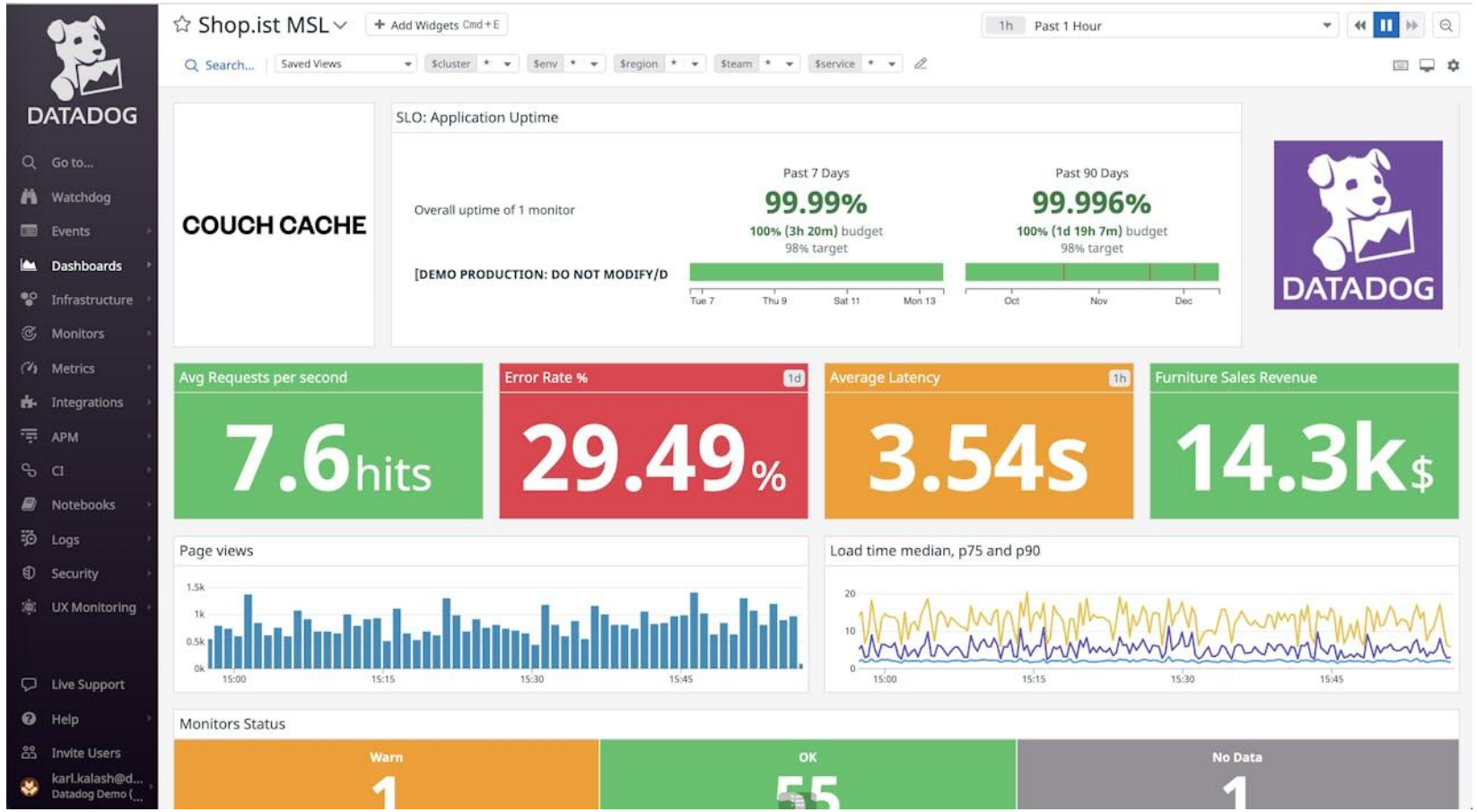
EXEMPLE DE STACK : ELK



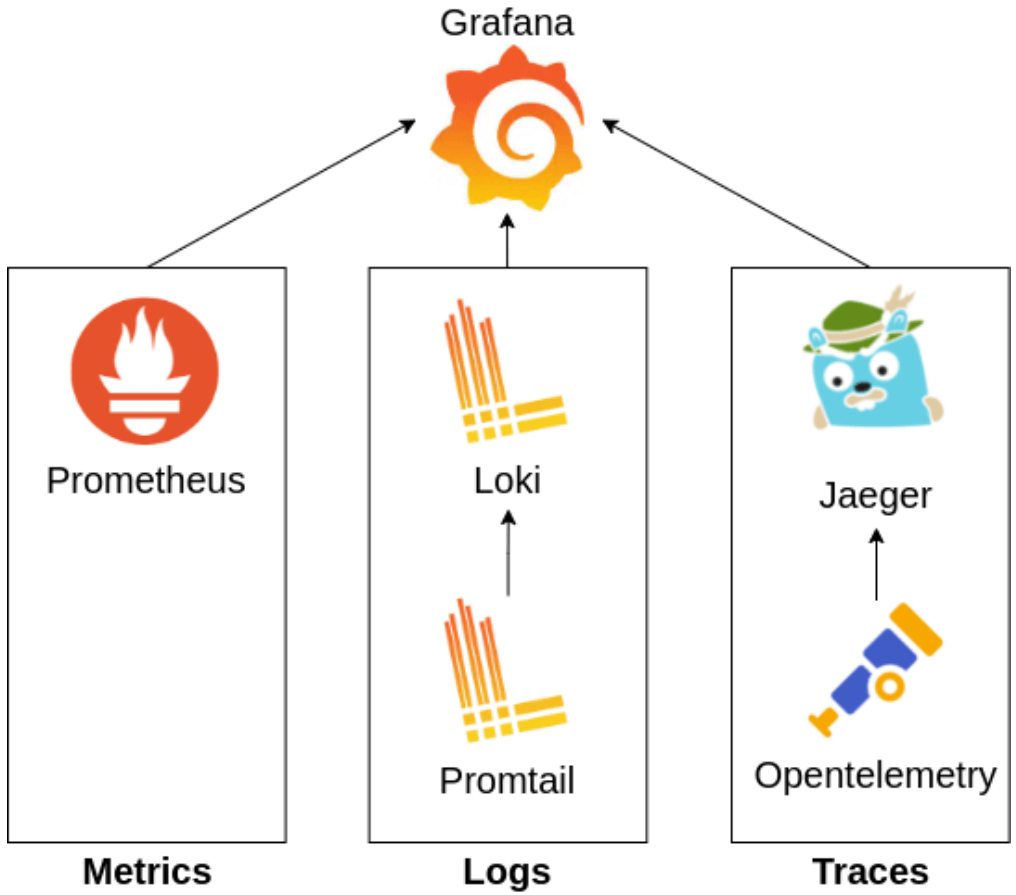
EXEMPLE DE STACK : DATADOG



EXEMPLE DE STACK : DATADOG



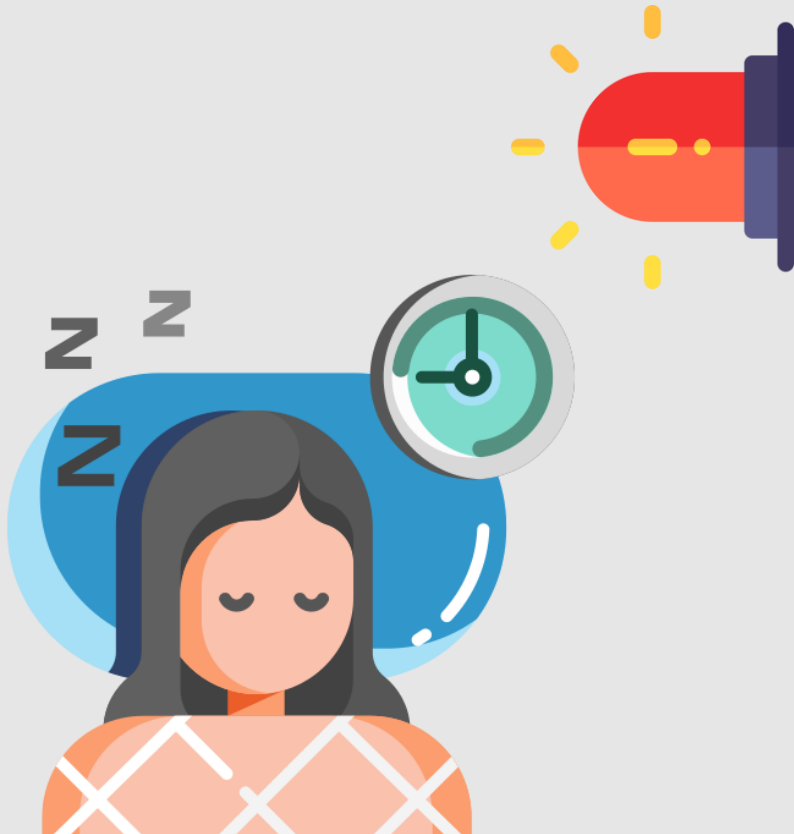
EXEMPLE DE STACK : GRAFANA/PROMETHEUS



EXEMPLE DE STACK : GRAFANA/PROMETHEUS

ALERTES ET ANALYSE D'INCIDENTS

PARCE QUE PERSONNE N'AIME LES SURPRISES (SURTOUT EN PRODUCTION)



C'EST QUOI L'INTÉRÊT ?

- **Alertes :**
  - Détecter les problèmes avant nos utilisateurs.
  - Plus un problème est détecté tôt, plus il est résolu rapidement.
  - Classer nos alertes en fonction de la gravité. (« warning », « critical », …).
  - Avoir des notifications (mails, slack, sms, …).

- **Analyse d'incident :**
  - Détection des anomalies automatiques.
  - Navigation entre métriques, logs et traces pour comprendre la situation.

COMMENT FAIRE DE BONNES ALERTES ?

- Une bonne alerte doit être :
  - **Pertinente** : L'alerte doit indiquer un vrai problème.
  - **Actionnable** : Elle doit dire *quoi faire* (ex : « Redémarrer le service X »).
  - **Contextuelle** : Inclure des liens vers les logs/dashboards.
  - **Priorisée** : Différencier « warning » et « critical ».

COMMENT ANALYSER UN INCIDENT ?

- **Détecter** : Via une alerte ou un signal utilisateur.
- **Trier** : Est-ce un faux positif ? Quel est l'impact ?
- **Diagnostiquer** :
  - Vérifier les métriques (dashboard).
  - Analyser les logs (filtres, corrélation).
  - Suivre les traces (si système distribué).
- **Résoudre** : Appliquer un fix ou un contournement.
- **Documenter** : Post-mortem et actions correctives.

BONNES PRATIQUES ET PIÈGES À ÉVITER

- Tester toutes ses alertes.
- Documenter les procédures de résolution.
- Former les équipes à la gestion d’incidents.
- Communiquer à ses collègues / utilisateurs.
- Enlever les alertes « inutiles »

C’EST L’HEURE DE LA PRATIQUE