Gestion des services Internet

Cours 4 - Le monitoring et le logging de services Web

Un cours de Yann Fornier

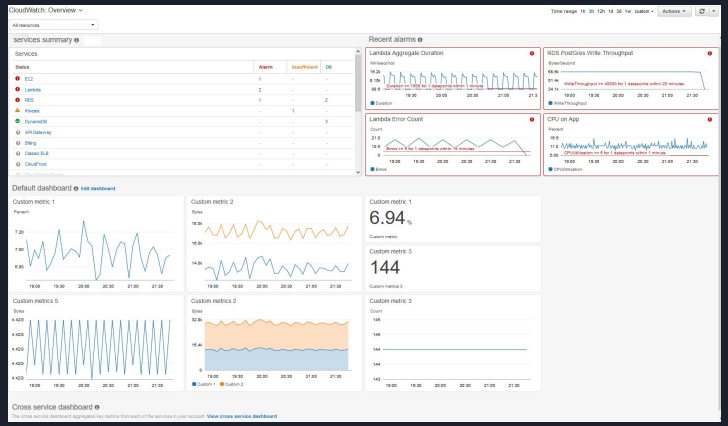
Thèmes abordés

Mise en place de monitoring et de logging pour suivre les performances et la disponibilité des services web

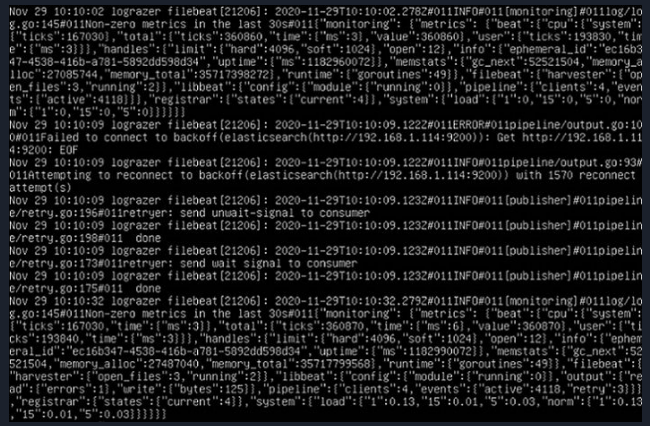
Les différents outils et techniques utilisées pour le monitoring et le logging des services web

Introduction

Le monitoring et le logging sont des pratiques clés pour assurer la disponibilité, la performance et la sécurité des services web.



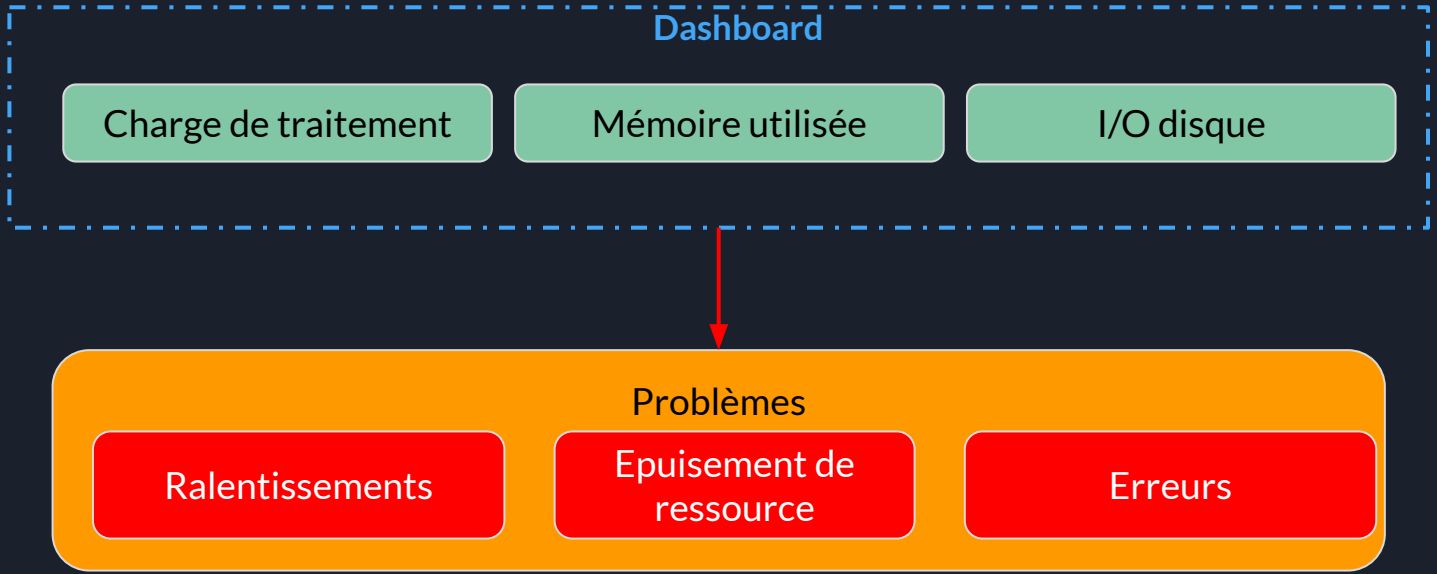
Monitoring



Logging

Définition du Monitoring

Le monitoring est la surveillance continue d'un système ou d'une application pour détecter les anomalies et les problèmes potentiels.

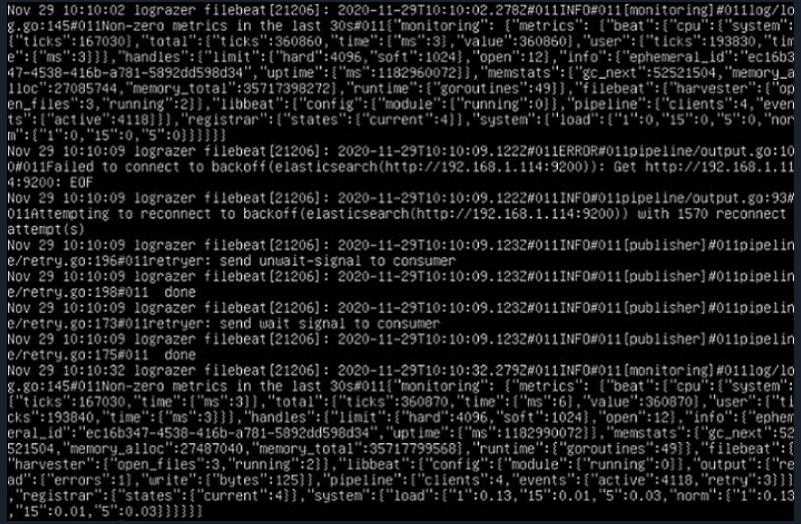


Définition du Logging

Le logging est l'enregistrement de données sur l'état et les activités d'un système.

Il permet de récupérer des informations sur les erreurs et les anomalies qui se produisent, et de les utiliser pour résoudre les problèmes, comprendre les tendances et identifier les tendances à venir.

Les données de journalisation peuvent également être utilisées pour répondre aux exigences réglementaires ou pour la conformité.

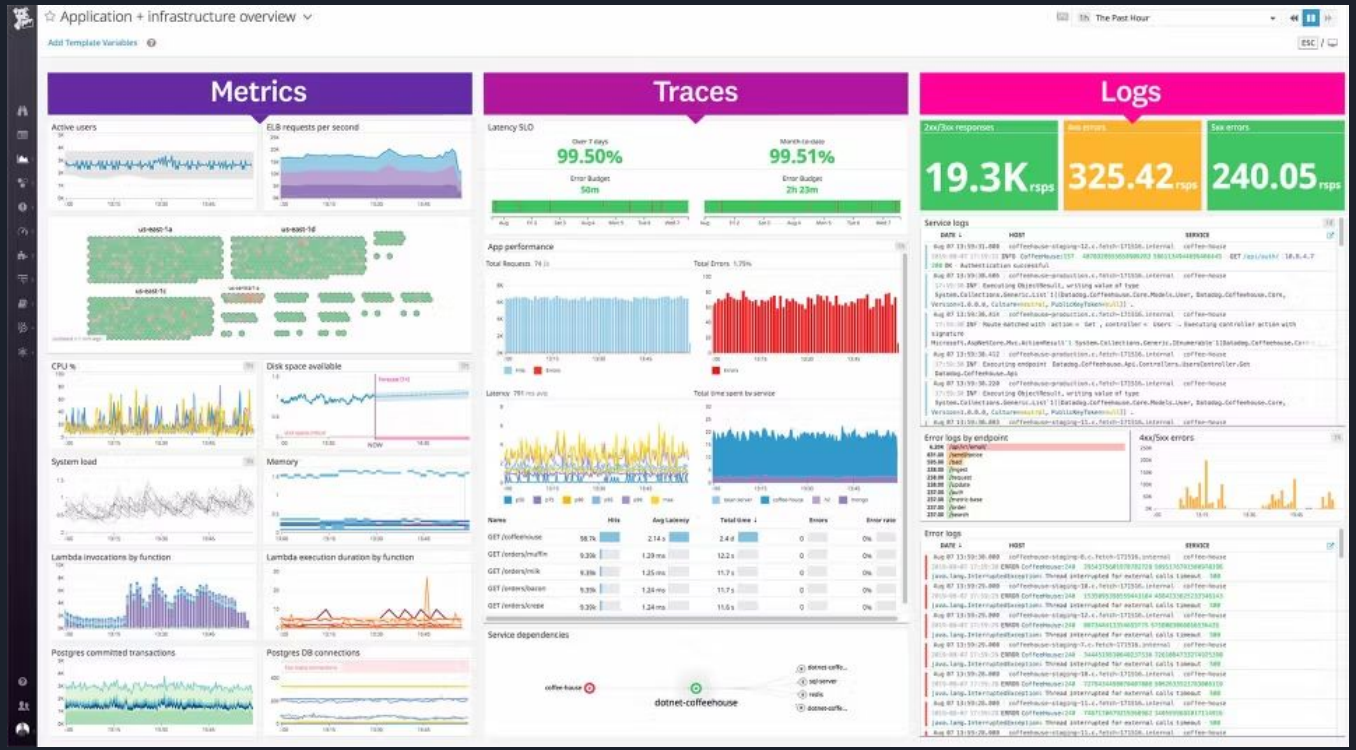


Logging

Le monitoring

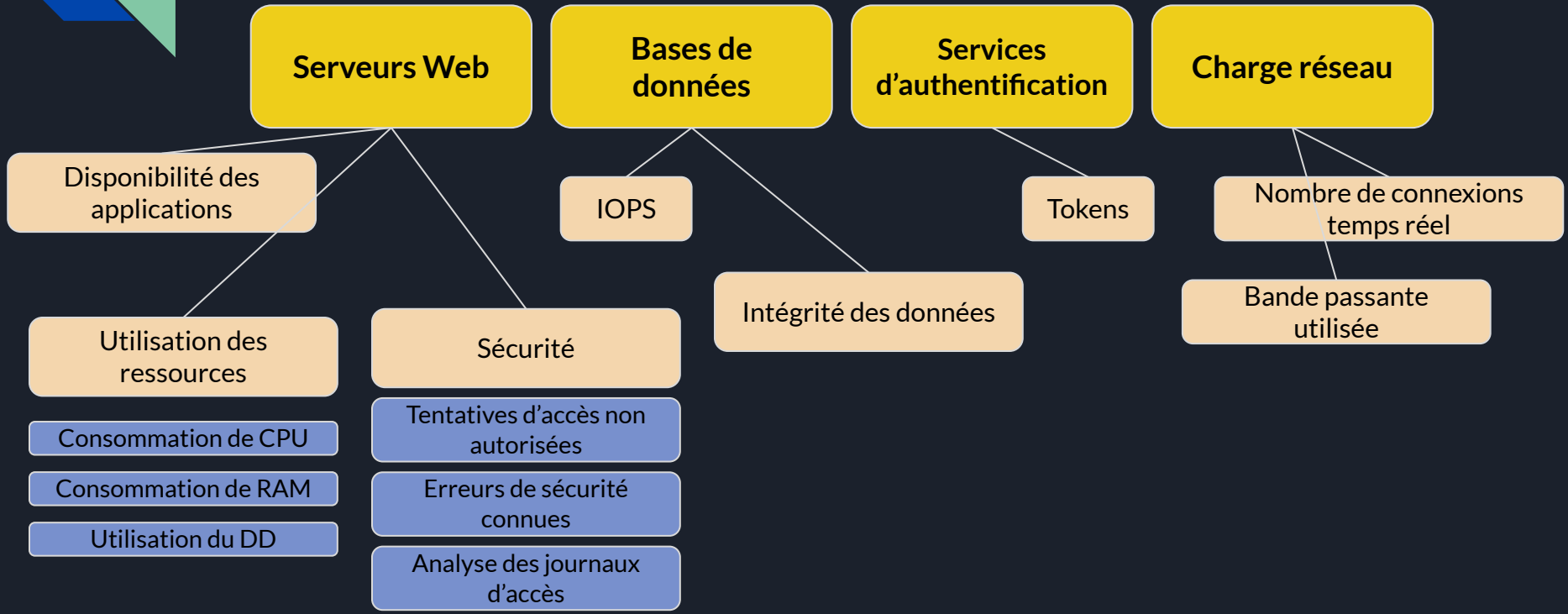
Les technologies de monitoring comme Datadog permettent aux administrateurs de systèmes de surveiller les performances de leurs services web en temps réel, collecter des métriques et des données de journalisation de leurs applications, et utiliser des alertes et des rapports pour identifier les problèmes potentiels avant qu'ils ne causent des perturbations de service.

Exemple d’outil de Monitoring

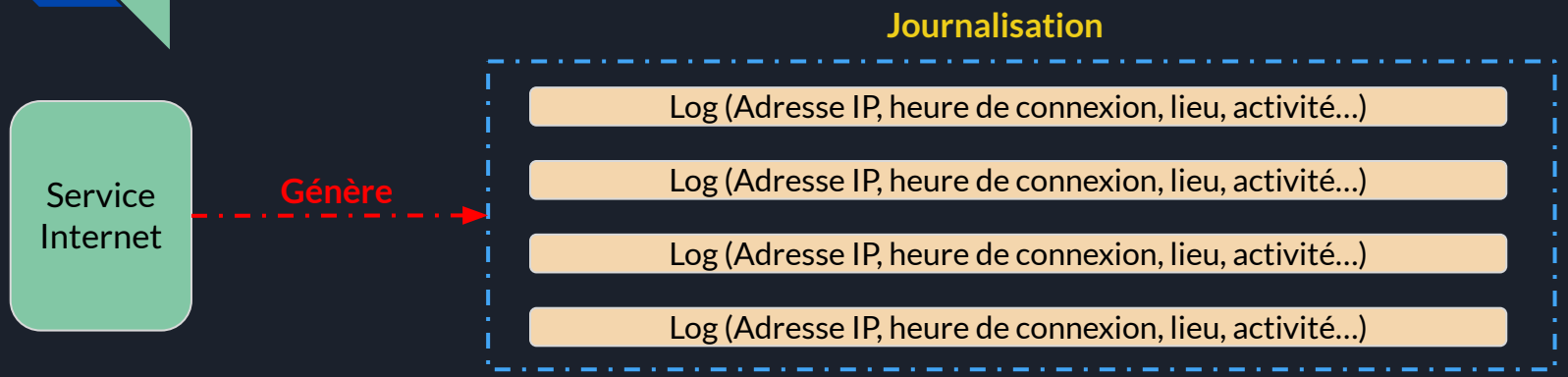


Datadog

Les objets à monitorer



Objectif du logging



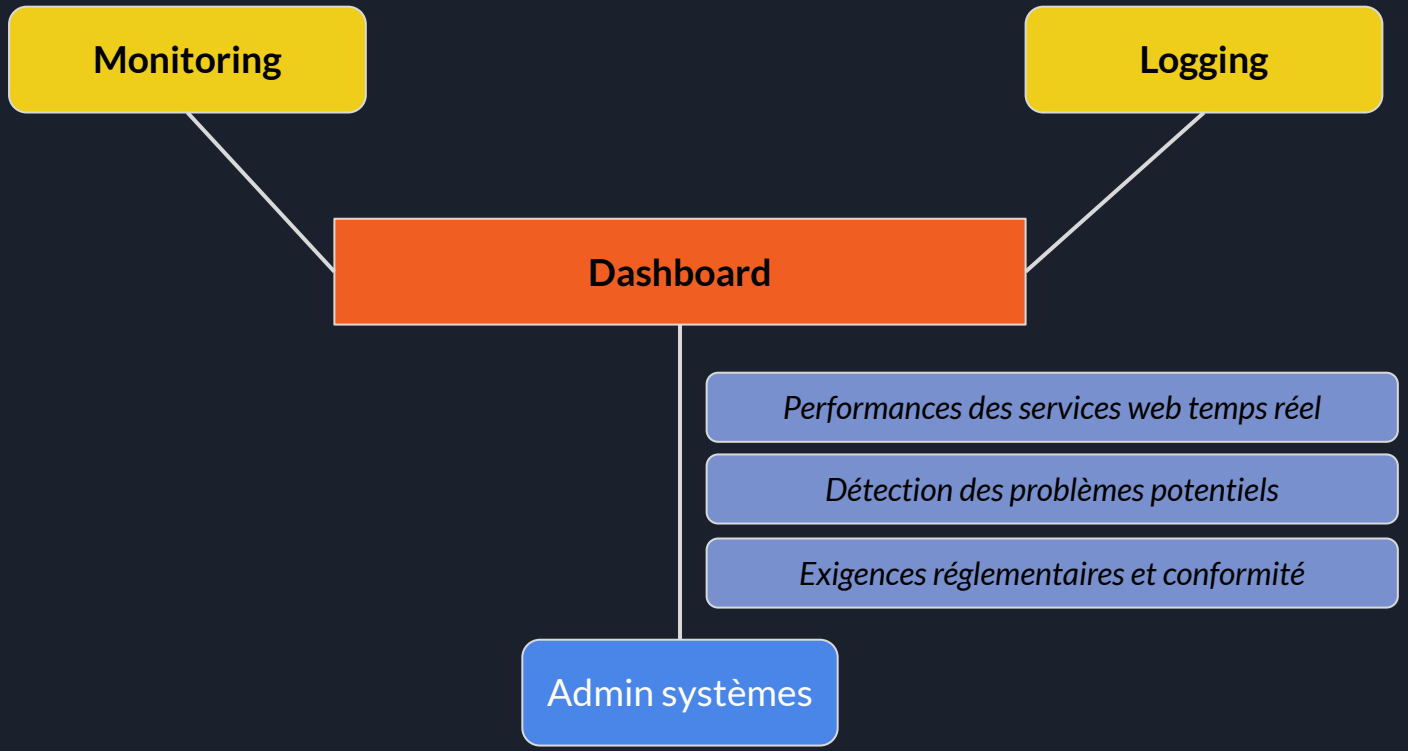
Analyse Préventive

Détecter un comportement inhabituel sur un SI et remonter le comportement pour voir si c'est un faux positif ou un évènement de cybersécurité

Forensic

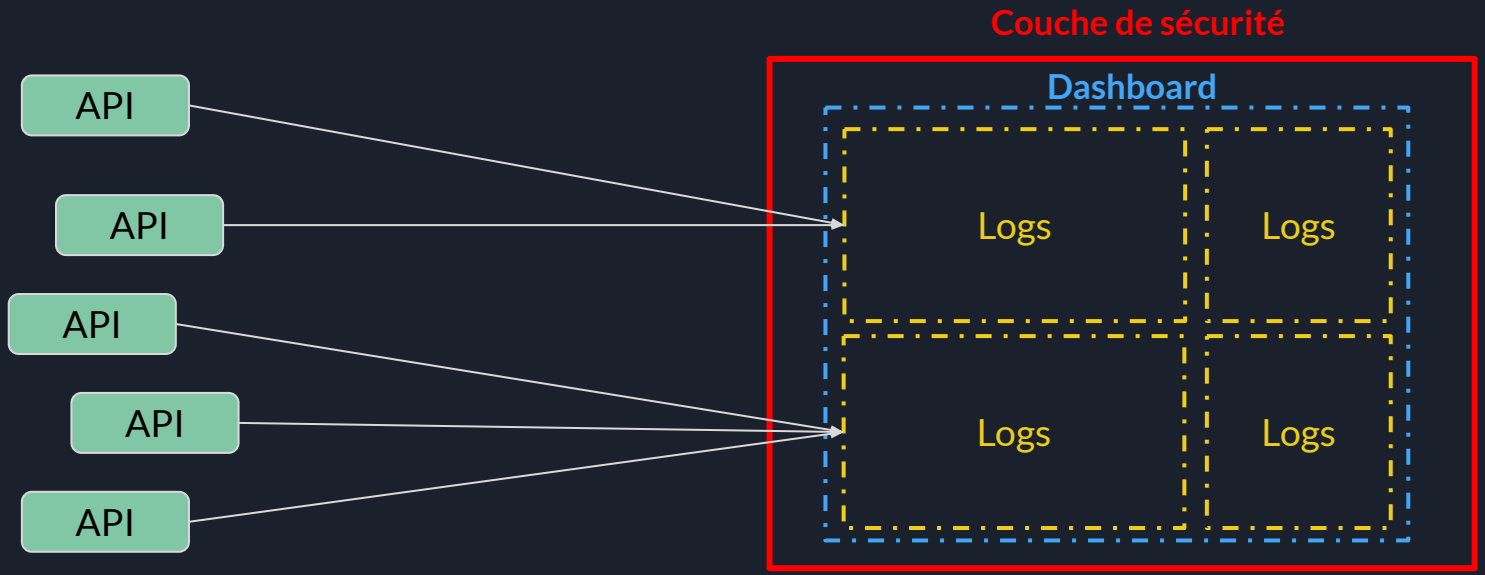
"Police scientifique" Étude des logs pour trouver des traces des pirates qui ont organisé une attaque

Dualité monitoring/logging



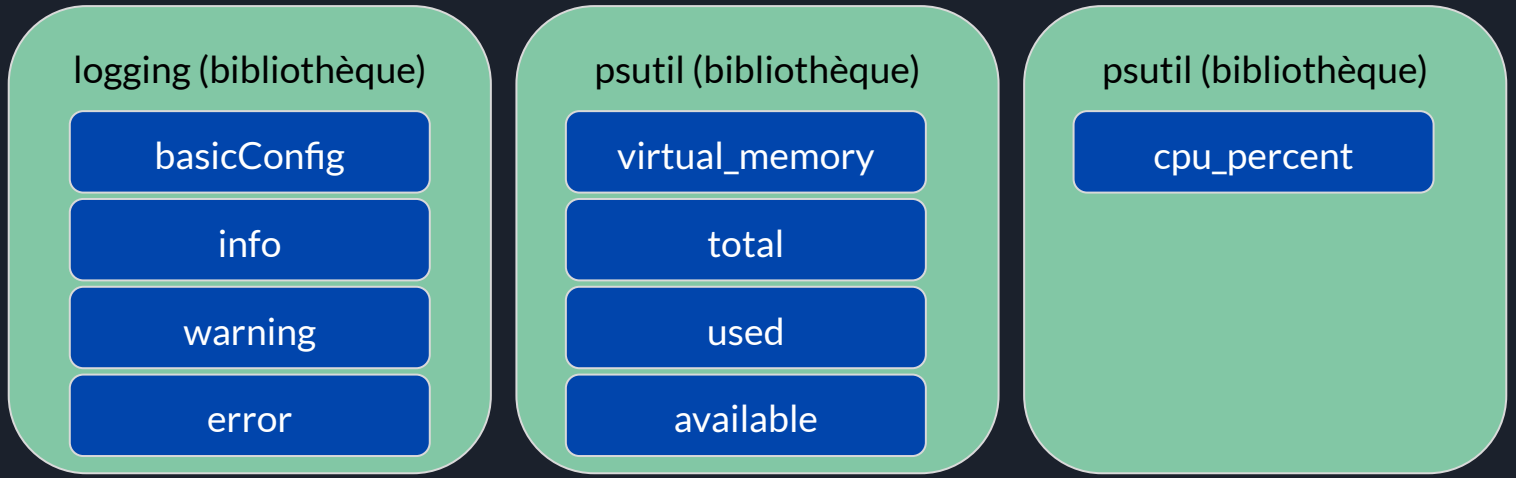
Sécurité des outils de monitoring

Il est important de noter que la sécurité de ces outils de monitoring et de logging doit être prise en considération pour éviter les fuites de données ou les accès non autorisés aux données collectées. Il est donc essentiel de mettre en place les protocoles de sécurité adéquats et de suivre régulièrement les politiques de sécurité.



Logging et monitoring en python

Python dispose d'une bibliothèque de journalisation intégrée appelée **logging**, qui permet de générer des messages de journalisation de manière simple et efficace. Il est également possible d'utiliser des bibliothèques de monitoring telles que psutil pour récupérer des informations sur les performances et les ressources système.



Journalisation de base

psutil (bibliothèque)

- basicConfig
- info
- warning
- error

```
import logging

logging.basicConfig(level=logging.INFO,
                    format = '%(asctime)s %(levelname)s %(message)s')

logging.info("Application started")
logging.warning("A warning message")
logging.error("An error occurred")
```

Monitoring de la mémoire utilisée

psutil (bibliothèque)

- basicConfig
- info
- warning
- error

```
import psutil

memory_info = psutil.virtual_memory()
print(“Total memory : ” , memory_info.total)
print(“Used memory : ” , memory_info.used)
print(“Available memory : ” , memory_info.available)
```

Monitoring de l’utilisation du processeur

psutil (bibliothèque)

cpu_percent

```
import psutil

cpu_info = psutil.cpu_percent(interval=1)
print(“CPU usage : ” , cpu_info)
```

Journalisation et monitoring combinés

```
import psutil
import logging

logging.basicConfig(level=logging.INFO,
                    format = '%(asctime)s %(levelname)s %(message)s')

logging.info("Application started")

#Perform calculations
result = 0
for i in range (1, 101):
        result += i
        cpu_info = psutil.cpu_percent(interval = 1)
        logging.info("CPU usage : %s", cpu_info)

logging.info("Application finished")
```

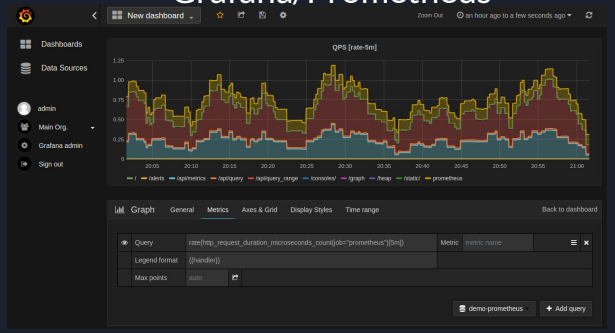
Journalisation et monitoring combinés

Il est important de noter que les exemples ci-dessus ne représentent qu'une petite partie des fonctionnalités offertes par les bibliothèques de journalisation et de monitoring disponibles en Python. Il est donc important de consulter la documentation de ces bibliothèques pour en savoir plus sur les fonctionnalités.

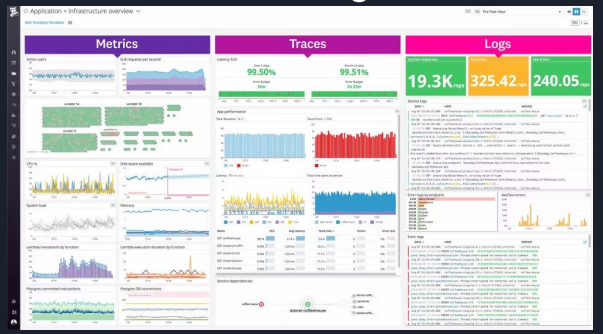
Etude de cas : Les outils pour du monitoring et log

Etude de cas (30 minutes)

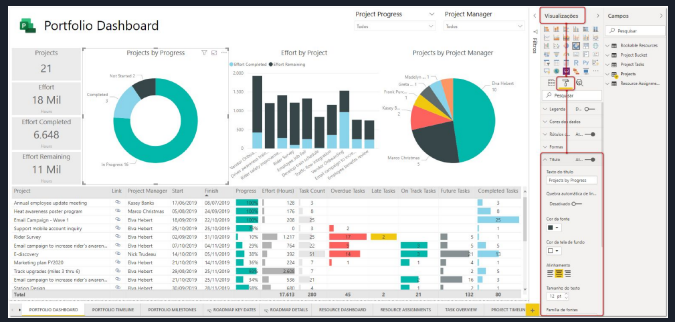
Groupe 1
Grafana/Prometheus



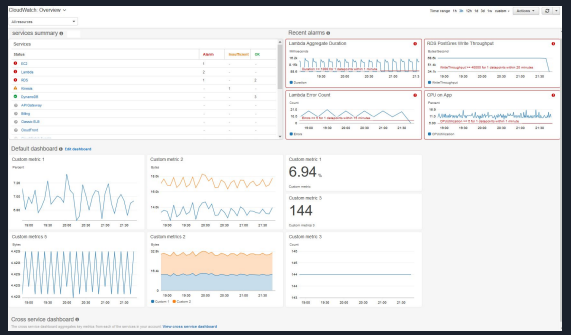
Groupe 2
DataDog



Groupe 3
PowerBI



Groupe 4
AWS CloudWatch



Grafana

Tableaux de bord interactifs

Dashboard personnalisé Grafana

Graphiques

- Prometheus
- InfluxDB
- Graphite
- Elasticsearch

Format de données

Ingestion temps réel

Données de performance

- Open source
- Visualisation de données
  - Analyse de tendances
  - Alertes temps réel
  - Collaboration temps réel
- Monitoring

Grafana

Dashboard personnalisé Grafana

Équipe DevOps

Surveillance de l'infrastructure temps réel

Équipe Cybersécurité

Identification des problèmes de sécurité temps réel

Équipe Gestion de performance

Gestion des performances temps réel

Grafana et Prometheus

Source 1

Source 2

Source 3

Data locale
agrégée

Prometheus

Collecte sous la forme de paire
clé-valeur
(Non relationnel)

PromQL
(Prometheus Query Language)

Dashboard personnalisé Grafana

Métriques
Indicateur de performance système
Ressources utilisées
Données d'application

Grafana et Prometheus

Ingestion de données (temps réel)

Source 1

Source 2

Source 3

Data locale agrégée

Prometheus

Collecte sous la forme de paire clé-valeur (Non relationnel)

Métriques

Indicateur de performance système
Ressources utilisées
Données d'application

PromQL
(Prometheus Query Language)

Data Visualization

Dashboard personnalisé Grafana

Travaux pratiques : Création d'un dashboard Grafana avec une source de donnée Prometheus

Objectif du TP : Réaliser le Lab sur Grafana

https://grafana.com/tutorials/grafana-fundamentals/

Ajouter des sources de données

Créer un Dashboard

Réaliser un fichier pdf avec captures d'écran expliquant vos résultats.

Deadline de rendu : **Mercredi 21 février 23h59**

Format attendu : "TP_Grafana_NOM_PRENOM"

Adresse de rendu : yann.fornier@gmail.com

Datadog

Datadog est une **plateforme de surveillance** complète

- Collecte de métriques
- Analyse de logs
- Tracing d'applications
- Alerting
- Intégration avec des services Cloud

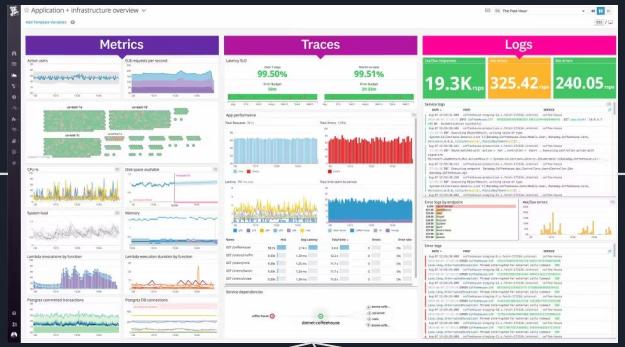
Grafana est davantage une plateforme de visualisation de données

Datadog

Métriques stockées dans un système distribué
Indicateur de performance système
Ressources utilisées
Données d'application

Services Web

- API
- API
- API
- API
- API



Cloud

- AWS
- Azure
- Google Cloud Platform

- Alertes pour des valeurs limites
- Utilisation de ressources anormales
- Erreurs
- Analyse de journaux
- Surveillance de l'infrastructure
- Performance des requêtes

Travaux pratiques : Création d'un APM (Application Performance Monitoring) Datadog

Objectif du TP : Réaliser le Lab sur Datadog

https://learn.datadoghq.com/courses/dd-101-dev

Réaliser un fichier pdf avec captures d'écran expliquant vos résultats.

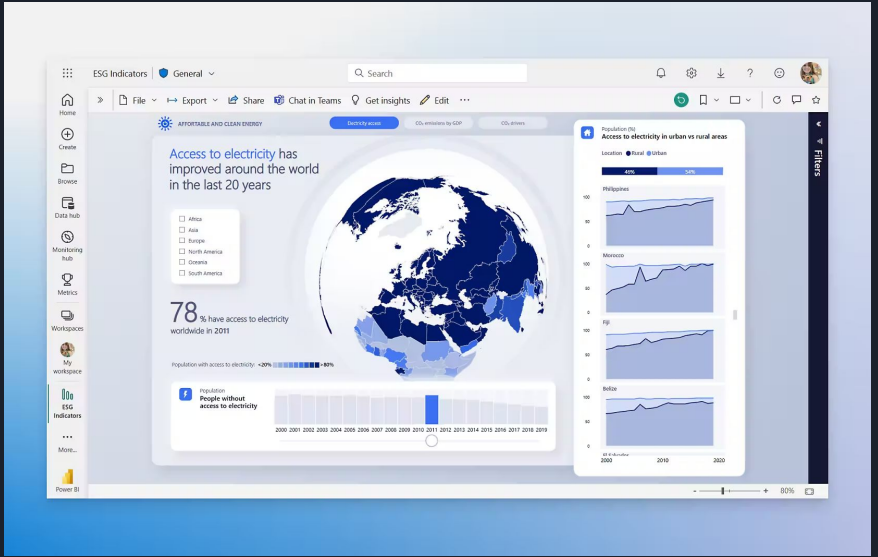
Deadline de rendu : **Mercredi 21 février 23h59**

Format attendu : "TP_DATADOG_NOM_PRENOM"

Adresse de rendu : yann.fornier@gmail.com

PowerBI

PowerBI est une suite d'outils d'analyse commerciale développée par Microsoft. Elle permet de faire de la visualisation et de l'analyse de donnée à partir de différentes sources.



Un dashboard PowerBI

PowerBI

Modélisation de données

PowerBI

Données diverses

Excel

Azure

Dynamics 365

Services Microsoft

Création de visualisations

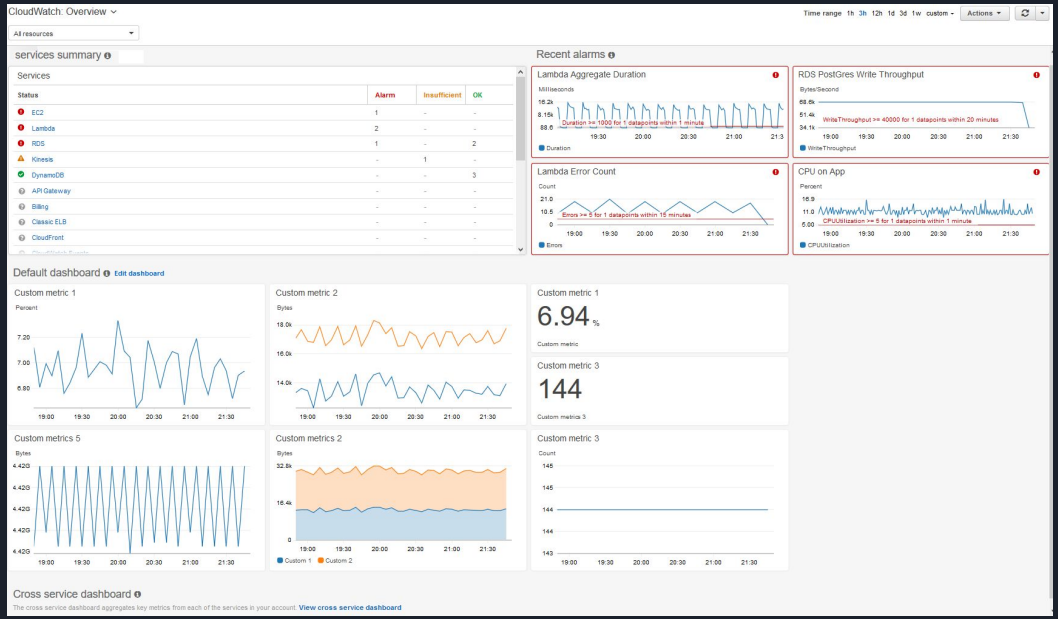
Analyse avancée

Génération de rapports

Modèles statistiques

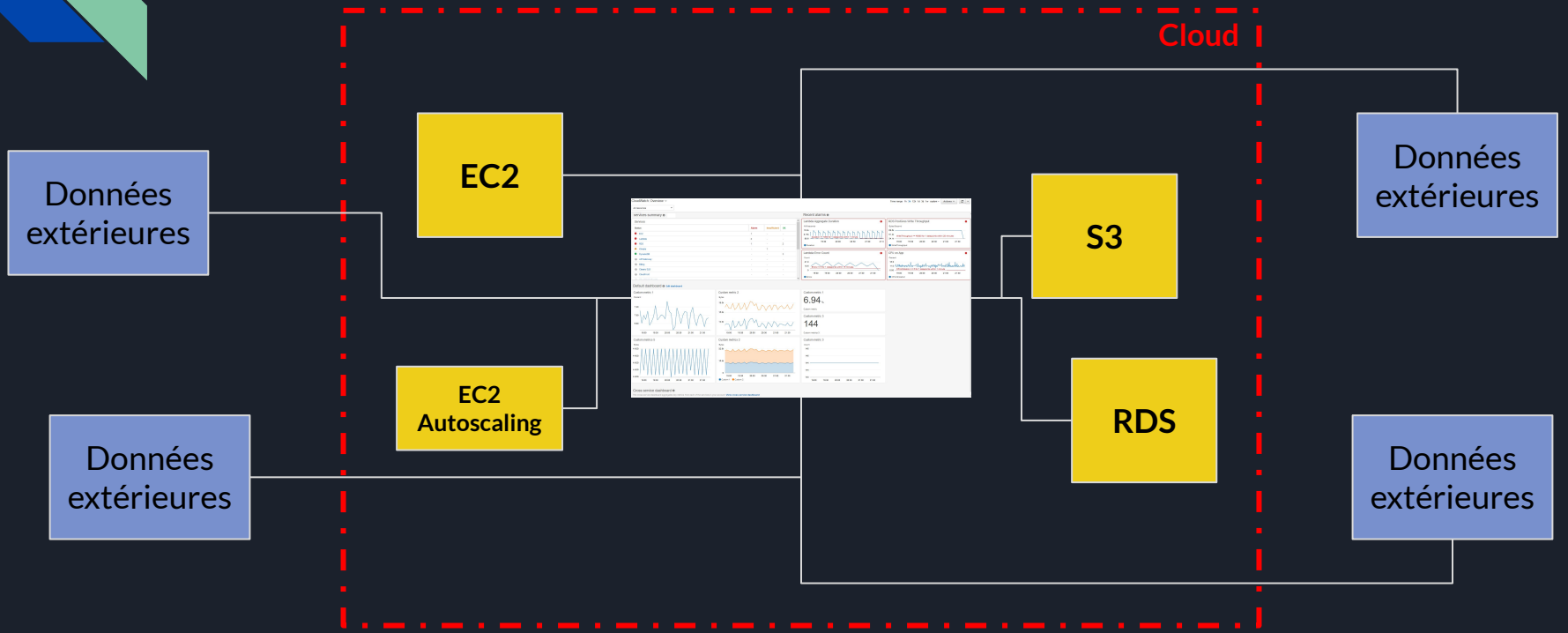
AWS Cloudwatch

AWS Cloudwatch est la plateforme de surveillance et de gestion des performances d'Amazon Web Services



Un dashboard CloudWatch

AWS CloudWatch



AWS CloudWatch