Cours 5
Monitoring et logging de services Web

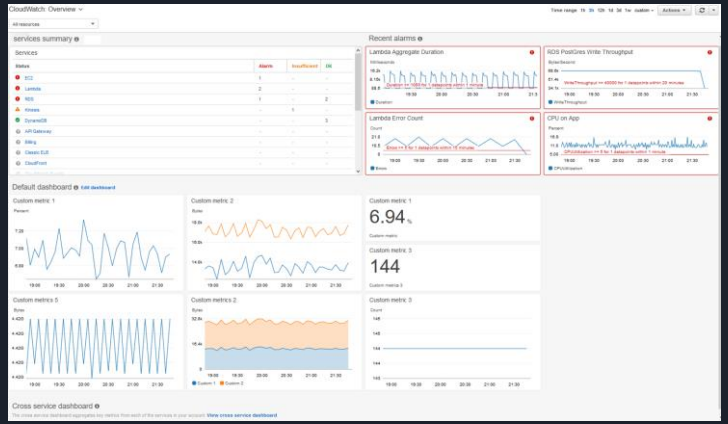
Un cours de Yann Fornier

Session 5 : Monitoring et logging des services web

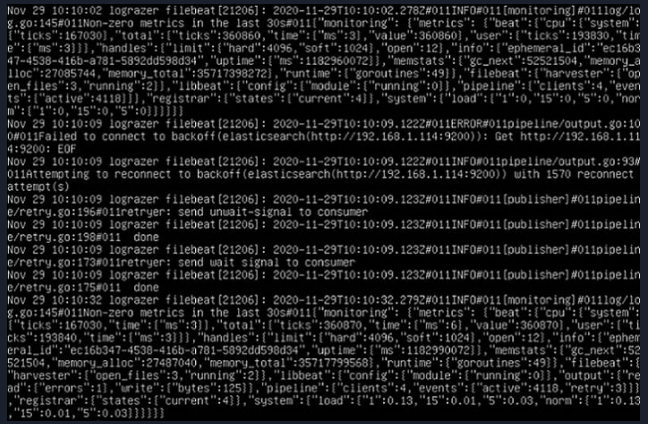
- La mise en place de monitoring et de logging pour suivre les performances et la disponibilité des services web
- Les différents outils et techniques utilisés pour le monitoring et le logging des services web
- **QCM** : connaissances sur le monitoring et le logging des services web

Introduction

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



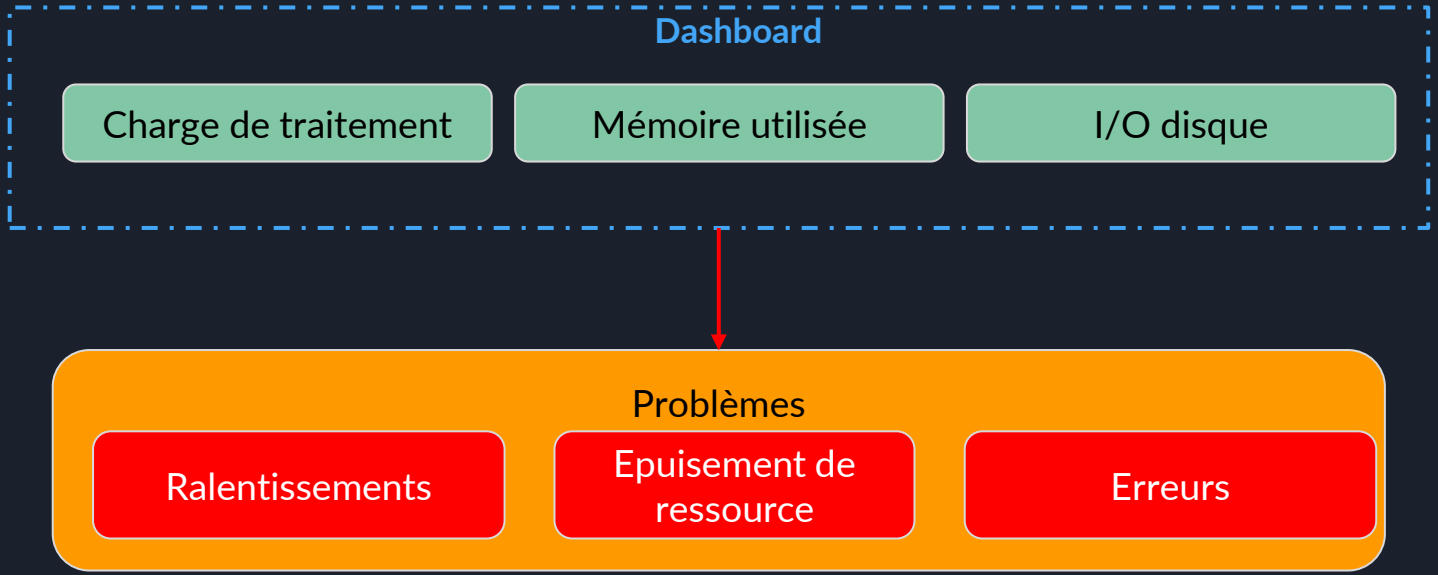
Monitoring



Logging

Monitoring

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

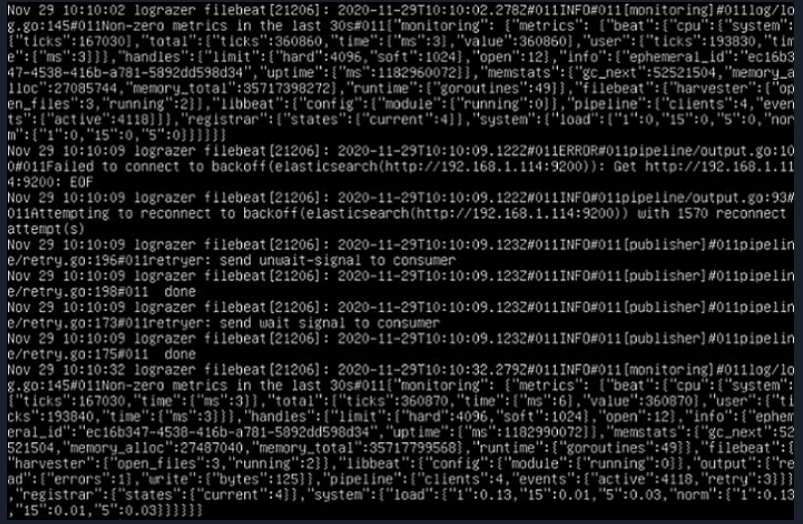


Logging

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

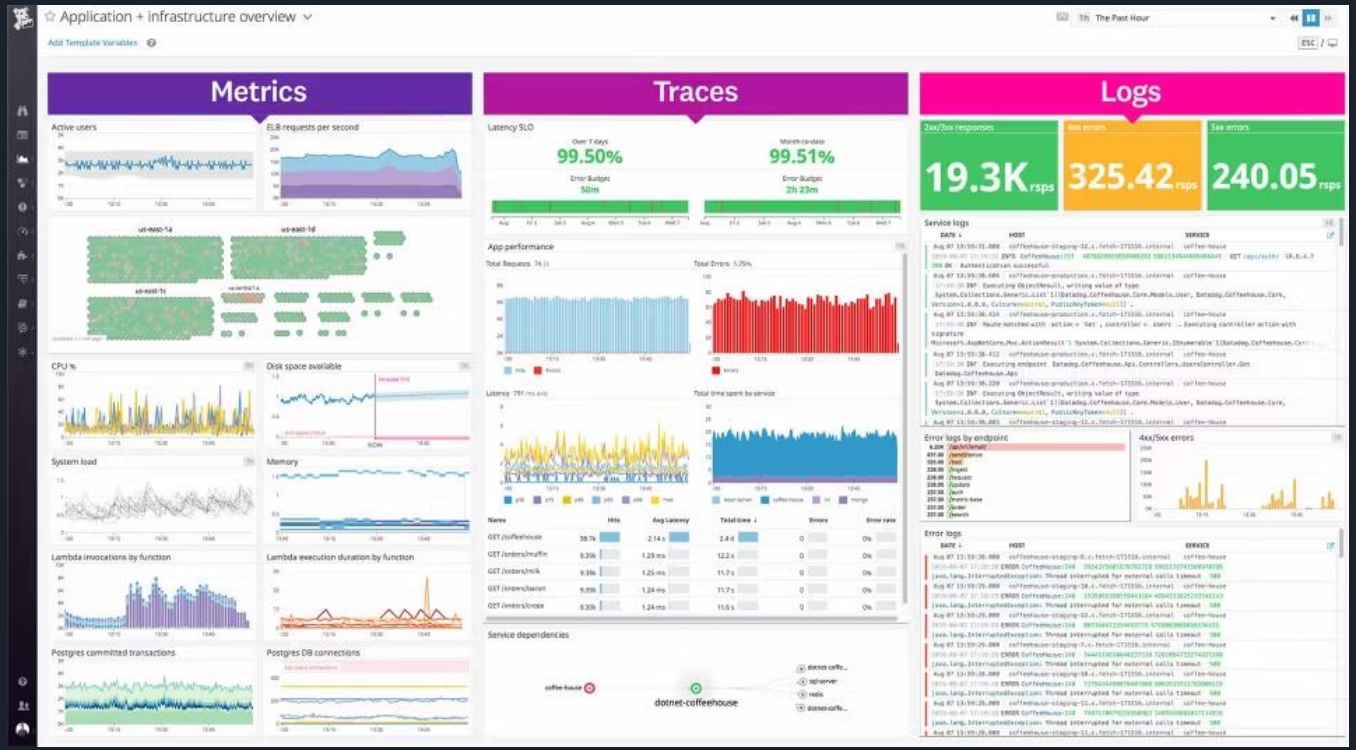
Il permet de récupérer des informations sur les erreurs et les anomalies qui se produisent, et de les utiliser pour résoudre les problèmes, comprendre les tendances et identifier les tendances à venir.

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



Logging

Monitoring



Datadog

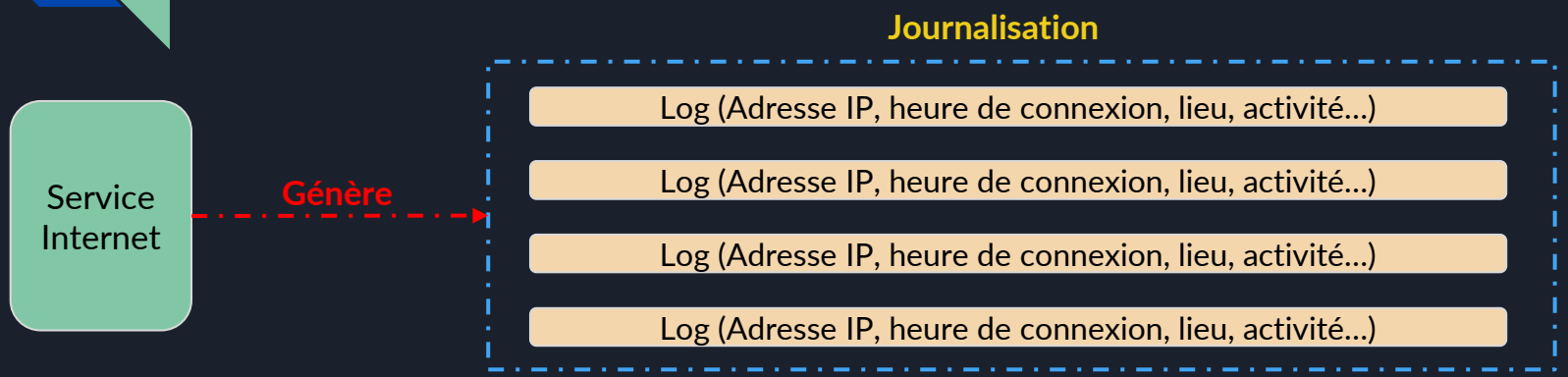
Le monitoring

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.

Les objets à monitorer

- Serveurs Web
- Bases de données
- Services d’authentification
- Charge réseau

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

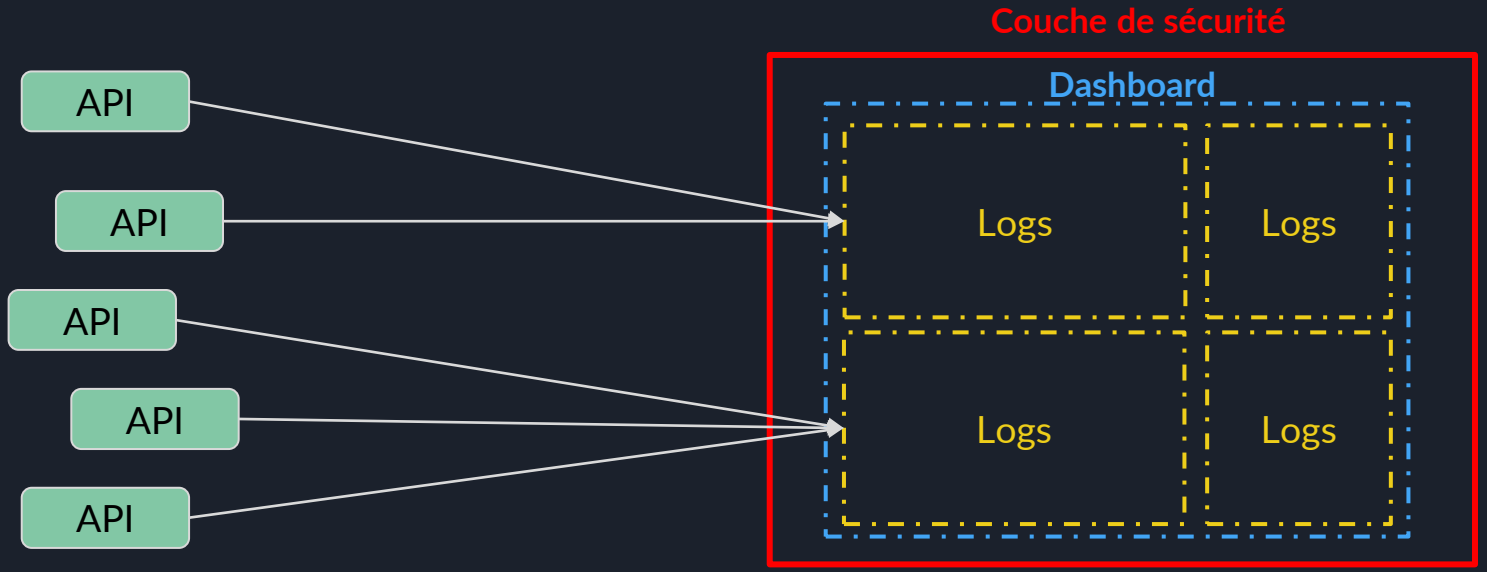
"Police scientifique" Étude des logs pour trouver des traces des pirates qui ont organisé une attaque

Objectif du logging

En utilisant des outils de monitoring et de logging, les administrateurs de systèmes peuvent surveiller les performances des services web en temps réel, détecter les problèmes potentiels avant qu'ils ne causent des perturbations de service, et répondre aux exigences réglementaires et à la conformité.

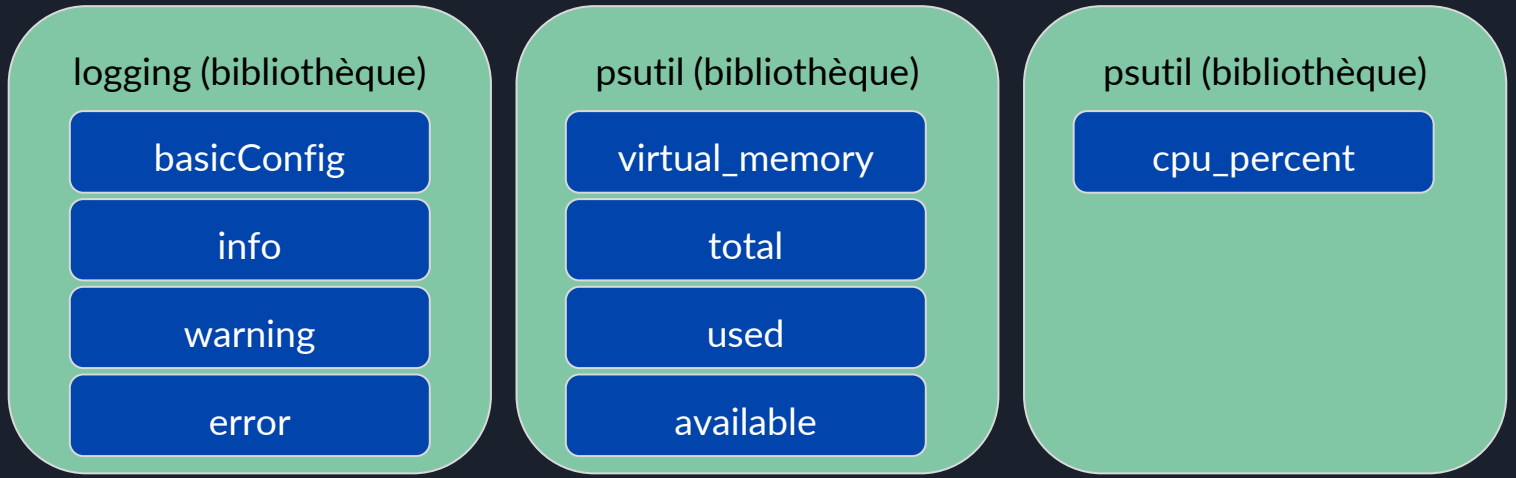
Sécurité des outils de monitoring

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 logging
import psutil

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()
    logging.info("CPU usage: %s", cpu_info)

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

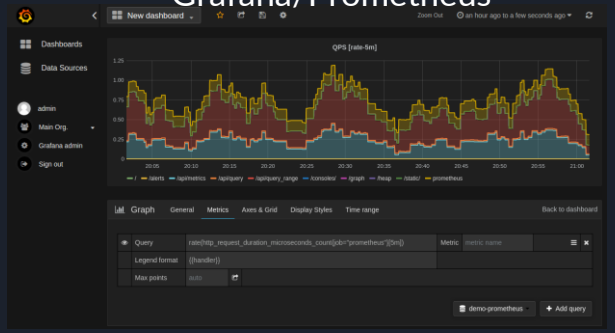
Journalisation et monitoring combinés

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

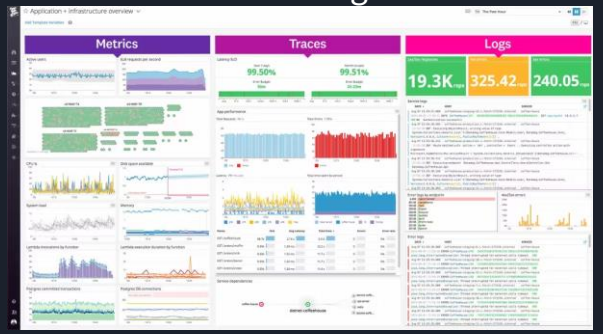
Les outils pour du monitoring et log

Etude de cas (30 minutes)

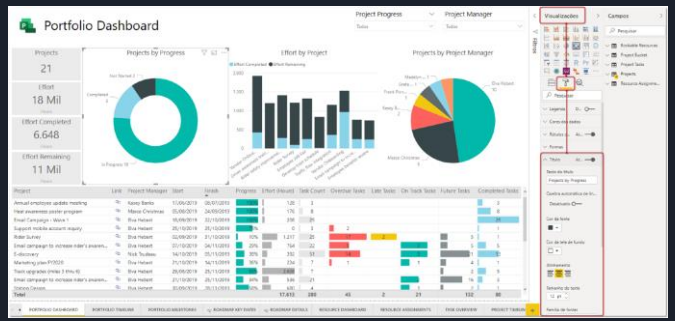
Groupe 1
Grafana/Prometheus



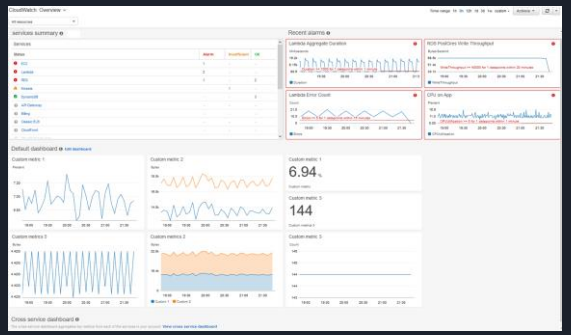
Groupe 2
DataDog



Groupe 3
PowerBI



Groupe 4
AWS CloudWatch



QCM en ligne