

# Cours 5

## Monitoring et logging de services Web

Un cours de Yann Fornier

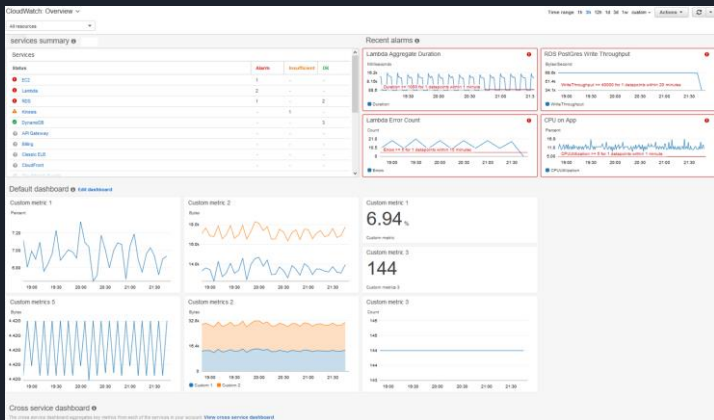


## Session 5 : Monitoring et logging des services web

- 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.



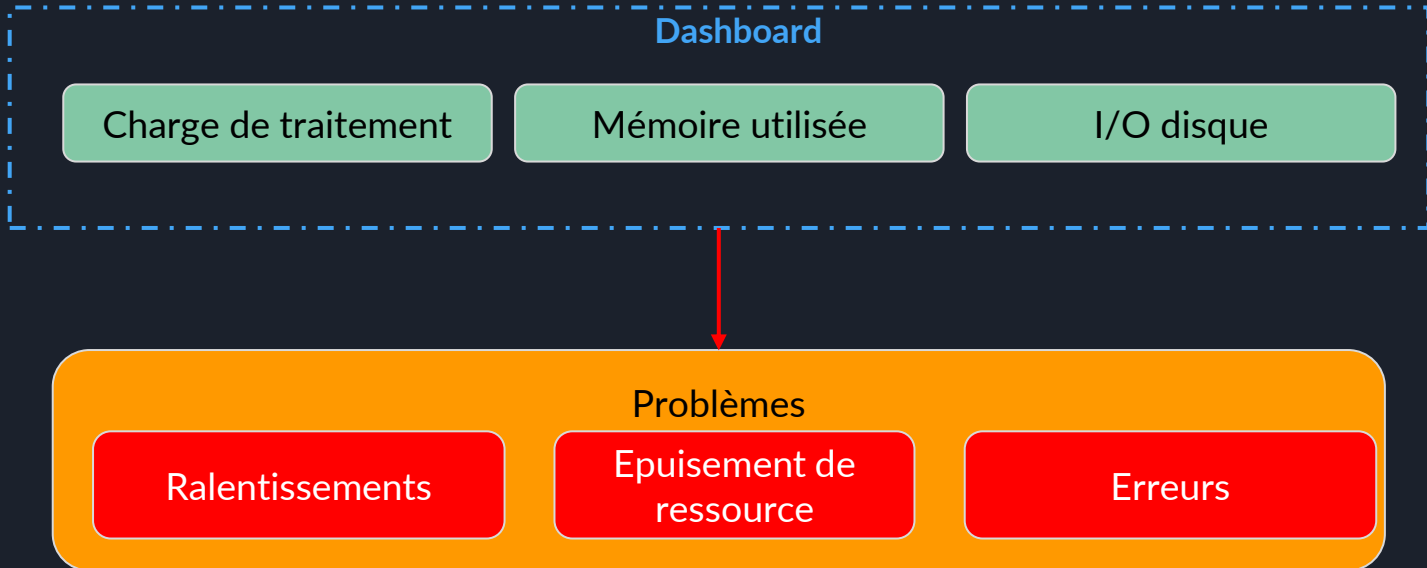
Monitoring

```
Nov 29 10:10:02 lograzer filebeat[21206]: 2020-11-29T10:10:02.278Z#011INFO#011[monitoring]#011log/lo
g.go:145#011Non-zero metrics in the last 30s#011{"monitoring": {"metrics": {"beat": {"cpu": {"system":
{"ticks":167030,"total":{"ticks":360860,"time":{"ms":3},"value":360860},"user":{"ticks":193830,"tix
e":{"ms":3}}},"handles":{"limit":{"hard":4096,"soft":1024},"open":12},"info":{"ephemeral_id":"ec16b3
47-453b-416b-8781-5932dd598d34","uptime":{"ms":118290072}},"memstats":{"gc_next":52321504,"memory_a
lloc":27085744,"memory_total":85717998872},"runtime":{"goroutines":493},"filebeat":{"harvester":{"o
pen_files":3,"running":2},"libbeat":{"config":{"module":{"running":0},"pipeline":{"clients":4,"even
ts":{"active":4118}}},"registrar":{"states":{"current":4}},"system":{"load":{"1":0,"15":0,"5":0,"nor
m":{"1":0,"15":0,"5":0}}}}}}}}
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.122Z#011INFO#011[pipeline/output.go:10
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.122Z#011INFO#011[pipeline/output.go:93#
011]attempting to reconnect to backoff(elasticsearch(http://192.168.1.114:9200)) with 1570 reconnect
attempt(s)
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:196#011retryer: send unwait-signal to consumer
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:198#011 done
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:173#011retryer: send wait signal to consumer
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:175#011 done
Nov 29 10:10:32 lograzer filebeat[21206]: 2020-11-29T10:10:32.278Z#011INFO#011[monitoring]#011log/lo
g.go:145#011Non-zero metrics in the last 30s#011{"monitoring": {"metrics": {"beat": {"cpu": {"system":
{"ticks":167030,"time":{"ms":3},"value":360870},"user":{"ticks":193840,"tix
e":{"ms":3}}},"handles":{"limit":{"hard":4096,"soft":1024},"open":12},"info":{"ephem
eral_id":"ec16b347-453b-416b-8781-5932dd598d34","uptime":{"ms":118290072}},"memstats":{"gc_next":52
521504,"memory_alloc":27487040,"memory_total":85717998880},"runtime":{"goroutines":493},"filebeat":{"
harvester":{"open_files":3,"running":2},"libbeat":{"config":{"module":{"running":0},"pipeline":{"ve
ad":{"errors":1},"write":{"bytes":125}}},"pipeline":{"clients":4,"events":{"active":4118,"retry":3}}}}
,"registrar":{"states":{"current":4}},"system":{"load":{"1":0.13,"15":0.01,"5":0.03,"norm":{"1":0.13
,"15":0.01,"5":0.03}}}}}}}}
```

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.

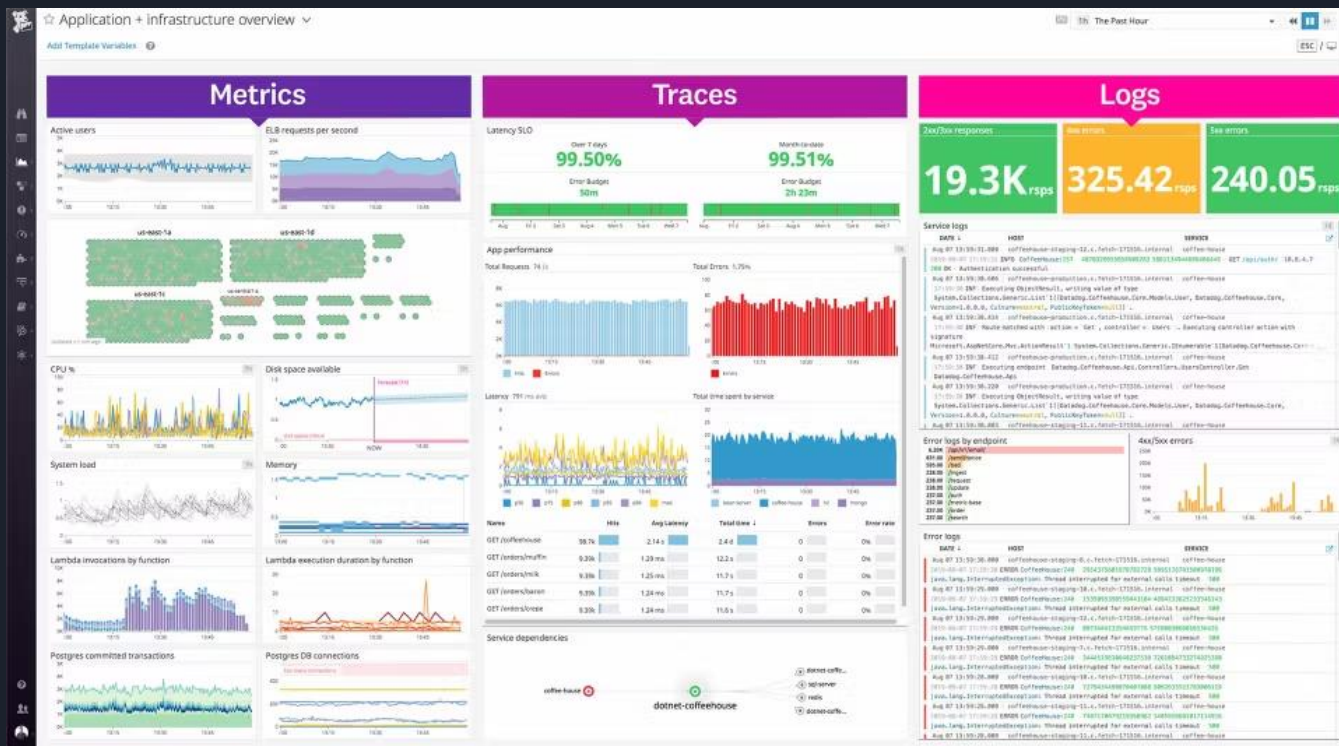
Il permet de récupérer des informations sur les erreurs et les anomalies qui se produisent, et de les utiliser pour résoudre les problèmes, comprendre les tendances et identifier les tendances à venir.

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

```
Nov 29 10:10:02 lograzer filebeat[21206]: 2020-11-29T10:10:02.278Z#011INFO#011[monitoring]#011log/lo
g.go:145#011Non-zero metrics in the last 30s#011[monitoring]: {metrics: {beat: {cpu: {system:
{ticks: 167030}, total: {ticks: 360860, time: {ms: 31, value: 360860}, user: {ticks: 193830, tim
e: {ms: 31}}, handles: {limit: {hard: 4096, soft: 1024}, open: 12}, info: {ephemeral_id: "ec16b3
47-4538-416b-a781-5892dd598d34", uptime: {ms: 1182960072}}, memstats: {gc_next: 52521504, memory_a
lloc: 27085744, memory_total: 35717398272, runtime: {goroutines: 49}}, filebeat: {harvester: {op
en_files: 3, running: 2}}, libbeat: {config: {module: {running: 0}}, pipeline: {clients: 4, even
ts: {active: 4118}}, registrar: {states: {current: 4}}, system: {load: {1: 0.13, 15: 0.01, 5: 0.03, nor
m: {1: 0.13, 15: 0.01, 5: 0.03}}}}}
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.122Z#011ERROR#011pipeline/output.go:10
0#011Failed to connect to backoff(elasticsearch(http://192.168.1.114:9200)): Get http://192.168.1.11
4:9200: EOF
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.122Z#011INFO#011pipeline/output.go:93#
011Attempting to reconnect to backoff(elasticsearch(http://192.168.1.114:9200)) with 1570 reconnect
attempt(s)
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:196#011retryer: send unwait-signal to consumer
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:198#011 done
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:172#011retryer: send wait signal to consumer
Nov 29 10:10:09 lograzer filebeat[21206]: 2020-11-29T10:10:09.123Z#011INFO#011[publisher]#011pipelin
e/retry.go:175#011 done
Nov 29 10:10:32 lograzer filebeat[21206]: 2020-11-29T10:10:32.279Z#011INFO#011[monitoring]#011log/lo
g.go:145#011Non-zero metrics in the last 30s#011[monitoring]: {metrics: {beat: {cpu: {system:
{ticks: 167030, time: {ms: 31}, total: {ticks: 360870, time: {ms: 61, value: 360870}, user: {ti
cks: 193840, time: {ms: 31}}, handles: {limit: {hard: 4096, soft: 1024}, open: 12}, info: {ephem
eral_id: "ec16b347-4538-416b-a781-5892dd598d34", uptime: {ms: 1182990072}}, memstats: {gc_next: 52
521504, memory_alloc: 27487040, memory_total: 3571779558, runtime: {goroutines: 49}}, filebeat: {
harvester: {open_files: 3, running: 2}}, libbeat: {config: {module: {running: 0}}, output: {re
ad: {errors: 1, write: {bytes: 125}}, pipeline: {clients: 4, events: {active: 4118, retry: 3}},
registrar: {states: {current: 4}}, system: {load: {1: 0.13, 15: 0.01, 5: 0.03, norm: {1: 0.13
, 15: 0.01, 5: 0.03}}}}}}}
```

Logging

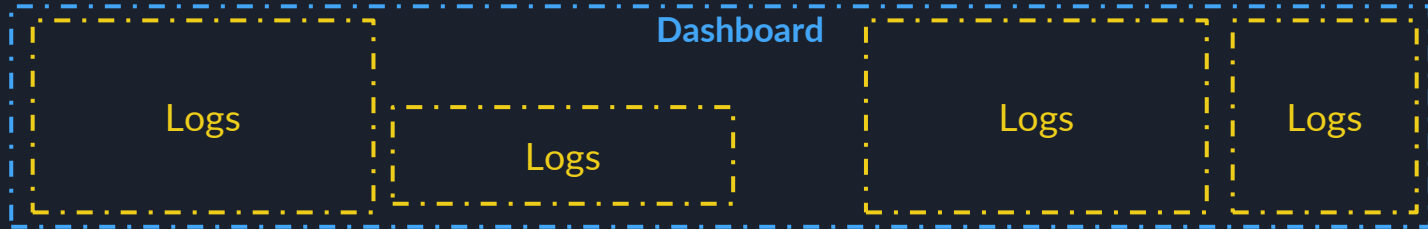
# Monitoring





# 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

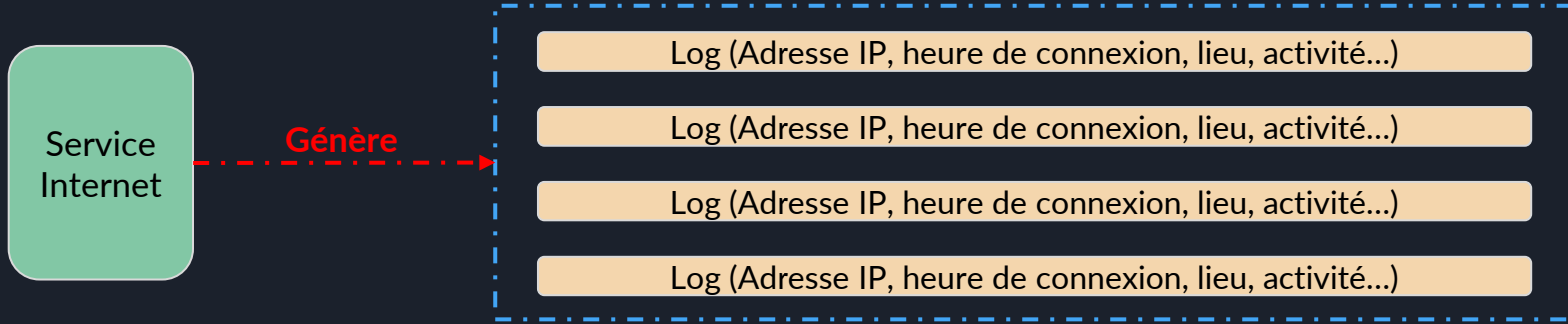
Bases de  
données

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

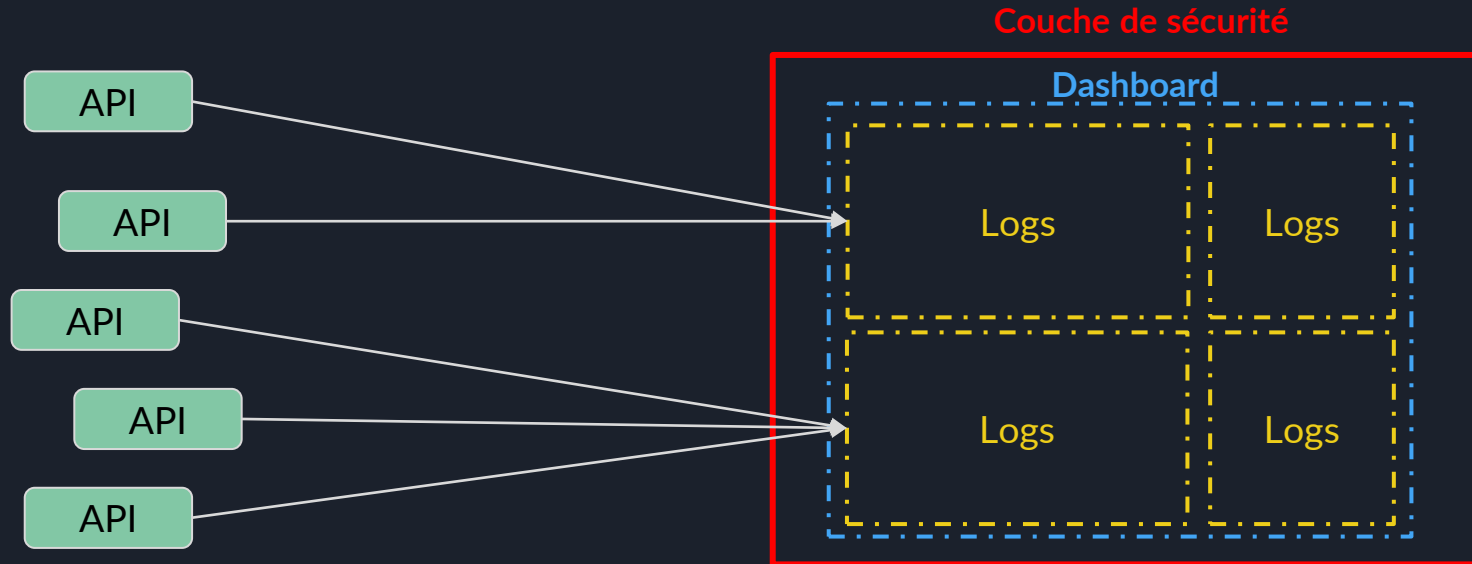


# Objectif du logging

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.

## logging (bibliothèque)

basicConfig

info

warning

error

## psutil (bibliothèque)

virtual\_memory

total

used

available

## psutil (bibliothèque)

cpu\_percent

# 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")
```





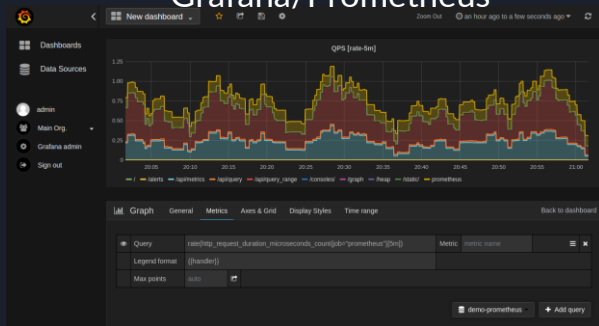
# Journalisation et monitoring combinés

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

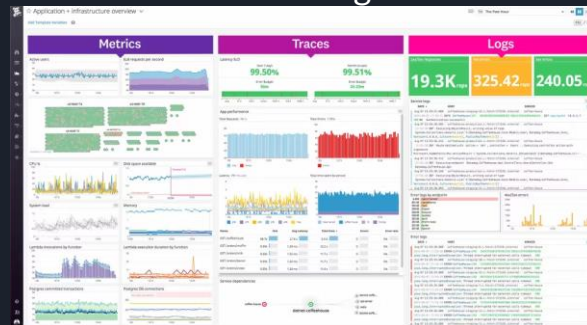
# Les outils pour du monitoring et log

Etude de cas (30 minutes)

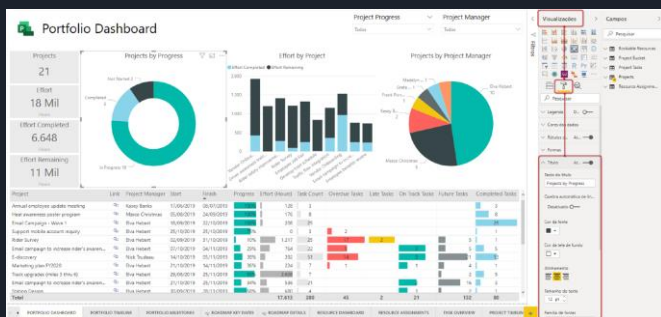
## Groupe 1 Grafana/Prometheus



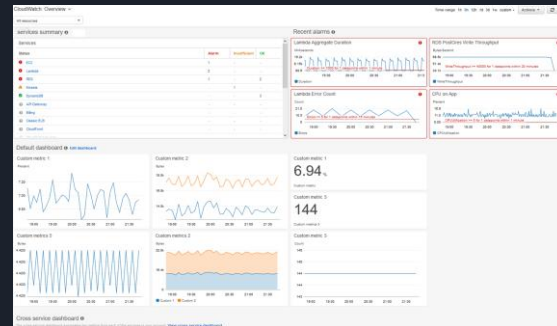
## Groupe 2 DataDog



## Groupe 3 PowerBI



## Groupe 4 AWS CloudWatch



QCM en ligne

