



# LOG430 Architecture logicielle Itération #1

---

## Objectifs

- Configuration de votre environnement
- Déployer l'infrastructure du laboratoire
- Déployer des mécaniques de télémétrie
- Documenter la mise en place de l'infrastructure de télémétrie
- Documenter ce que la télémétrie vous révèle sur l'architecture du système
- Documenter la responsabilité détaillée de chaque composant du système
- Proposer des solutions pour améliorer l'architecture du système

## Configuration de votre environnement

- ☐ installer Dotnet runtime <https://dotnet.microsoft.com/en-us/download/dotnet/7.0>
- ☐ installer docker desktop <https://www.docker.com/products/docker-desktop/>
  - ☐ settings/Activer "Expose daemon on tcp://localhost:2375 without TLS"
- ☐ installer dockstation <https://dockstation.io/>
- ☐ création de votre compte STM <https://www.stm.info/en/about/developers>
- ☐ création de votre compte de TOMTOM <https://developer.tomtom.com/user/register>
- ☐ 19 juin 2023 - mise à jour des données de la STM (Readme à faire)

## Récupération de l'infrastructure

```
git clone git@github.com:yvanross/LOG430-STM.git
```

## Démarrer l'application avec des lignes de commande

```
cd microservices
cd DockerCompose
export APP_DATA = /home/etudiant/LOG430-STM/microservices/DockerCompose/app_data
docker-compose -f "docker-compose.yml" -p
dockercompose1041557551265095097 --ansi never up -d --build --remove-orphans
```

## Démarrer l'application avec DockerDesktop

- Utiliser le fichier docker-compose.yml dans le répertoire DockerCompose
- Démarrer dockstation
- Todo: Tout ce qu'on peut faire avec les services existant: ex: comment accéder aux données de rabbitmq

## Déployer des mécaniques de télémétrie (open telemetry)

- ☐ installer open telemetry exporter for zipkin (todo: Documentation à faire (4))

Utiliser dockstation pour trouver les images docker suivant et les intégrer dans le docker-compose.yml

- ☐ installer et configurer l'image de prometheus pour l'acquisition des Métriques:  
<https://hub.docker.com/r/prom/prometheus/>
- ☐ installer et configurer l'image de grafana pour la visualisation de la télémétrie  
<https://hub.docker.com/r/grafana/grafana>
- ☐ installer et configurer l'image d'opentelemetry collector pour le traçage distribué  
<https://hub.docker.com/r/openzipkin/zipkin/>

## Documenter ce que la télémétrie et l'analyse du code vous révèlent sur l'architecture du système

- Analyser votre architecture en termes de
  - ☐ disponibilité
  - ☐ performance
  - ☐ interopérabilité
  - ☐ modifiabilité

## Documenter la responsabilité détaillée de chaque composant du système

- ☐ réaliser une documentation (vue de type composant et connecteur) de l'infrastructure incluant les composants de télémétrie
- ☐ on veut comprendre le rôle de chaque composant du système
- ☐ faire une vue de type module pour le composant TripComparator

## Perturbation de l'infrastructure

En sachant que la variable docker NanoCpus à 500000000 est l'équivalent de 0,5 cœur de processeur, réaliser vos analyses télémétriques en fonction de cette variable à

- ☐ 0.5 cœur
- ☐ 1 cœur
- ☐ 2 cœurs
- ☐ 3 cœurs
- ☐ 4 cœurs

## Proposer des solutions pour améliorer l'architecture du système

En fonction de vos analyses et de la description sommaire des itérations à venir documentées dans le document de spécifications-itérations, proposer des solutions pour améliorer l'architecture du système en termes de

- ☐ disponibilité
- ☐ performance
- ☐ interopérabilité
- ☐ modifiabilité

## Réaliser une évaluation par les pairs

---

- ☐ réaliser une évaluation par les pairs de votre travail et spécifier les améliorations à apporter
- ☐ indiquer le pourcentage de la note que vous attribuez à chaque étudiant