# LABORATOIRE SECURITE DES SYSTEMES D'INFORMATIONS

## PROJET :

# PROPOSER UNE SOLUTION SIEM PACKAGE POUR LES TPE/PME

Auteur : Samuel Amoedo

Rédigé le : 09/05/2024

Version : 1.0

TLP : WHITE

# Table des matières

# CONTEXTE

Dans un monde de plus en plus numérique, la sécurité des systèmes informatiques est devenue une préoccupation primordiale pour les entreprises, grandes ou petites. Cependant, pour de nombreuses petites et moyennes entreprises (TPE et PME), l'adoption de solutions de sécurité sophistiquées peut être entravée par des contraintes budgétaires et de ressources techniques. Pour répondre à ce défi, notre projet vise à proposer une solution innovante et abordable : l'automatisation de l'installation de Wazuh à l'aide de Docker et Ansible.

Wazuh, une plateforme de détection et de réponse aux menaces open source, offre une gamme complète de fonctionnalités de sécurité, allant de la surveillance des journaux à la détection d'anomalies en passant par la réponse aux incidents. En combinant la puissance de Wazuh avec la flexibilité et la facilité de déploiement offertes par Docker et Ansible, notre projet vise à fournir aux TPE et PME une solution de sécurité robuste et évolutive, sans les coûts élevés associés aux solutions traditionnelles.

# ETAT DE L'ART

Actuellement, Wazuh offre une solution complète de détection des menaces, fournissant aux entreprises un outil essentiel pour surveiller et protéger leurs infrastructures informatiques. Parmi ses principaux avantages figurent sa nature open source, qui permet une personnalisation et une adaptation aux besoins spécifiques de chaque organisation, ainsi que sa capacité à intégrer des sources de données variées telles que les journaux système, les fichiers de configuration et les événements réseau. En outre, Wazuh fournit des fonctionnalités avancées telles que la détection des menaces basée sur des règles, l'analyse de la corrélation des événements et la gestion centralisée des alertes.

Cependant, malgré ses nombreux atouts, Wazuh présente également quelques limitations. Son déploiement et sa configuration initiaux peuvent être complexes pour les utilisateurs novices, nécessitant une expertise technique significative pour une mise en place optimale. De plus, la gestion des alertes peut parfois être laborieuse, en particulier dans les environnements avec un volume élevé d'événements, nécessitant une attention particulière pour éviter les faux positifs et garantir une réponse efficace aux incidents de sécurité. Enfin, bien que Wazuh offre une gamme de fonctionnalités robustes, certains besoins spécifiques en matière de sécurité peuvent nécessiter l'intégration avec d'autres outils ou solutions, ce qui peut ajouter de la complexité au déploiement et à la gestion de l'infrastructure de sécurité.

Malgré ces défis, Wazuh reste une solution attrayante pour de nombreuses entreprises en raison de sa puissance, de sa flexibilité et de son coût relativement faible par rapport à des alternatives commerciales. En intégrant cette technologie dans un environnement Dockerisé et automatisé via Ansible, notre projet vise à simplifier et à rationaliser le processus d'installation et de gestion de Wazuh, offrant ainsi aux TPE et PME une solution de sécurité efficace et abordable.

## Wazuh-docker

Wazuh-Docker est une approche qui encapsule l'ensemble de l'écosystème Wazuh, y compris le serveur, les agents et les éléments de l'interface utilisateur, dans des conteneurs Docker. Cette méthode permet une installation et une gestion simplifiées de Wazuh, en offrant une isolation des environnements et une portabilité accrue. En utilisant Docker, les administrateurs peuvent déployer facilement des instances Wazuh sur diverses plateformes sans se soucier des dépendances logicielles ou des conflits. De plus, Docker-Ansible représente une stratégie d'automatisation qui utilise Ansible pour orchestrer le déploiement et la configuration des conteneurs Docker. Ansible permet de définir les spécifications d'infrastructure sous forme de code, facilitant ainsi la gestion des tâches répétitives telles que le déploiement de conteneurs, la configuration des réseaux et la mise à jour des services. En combinant ces deux approches, les entreprises peuvent automatiser efficacement le déploiement de Wazuh dans des environnements Dockerisés, offrant ainsi une solution de sécurité souple et évolutive.

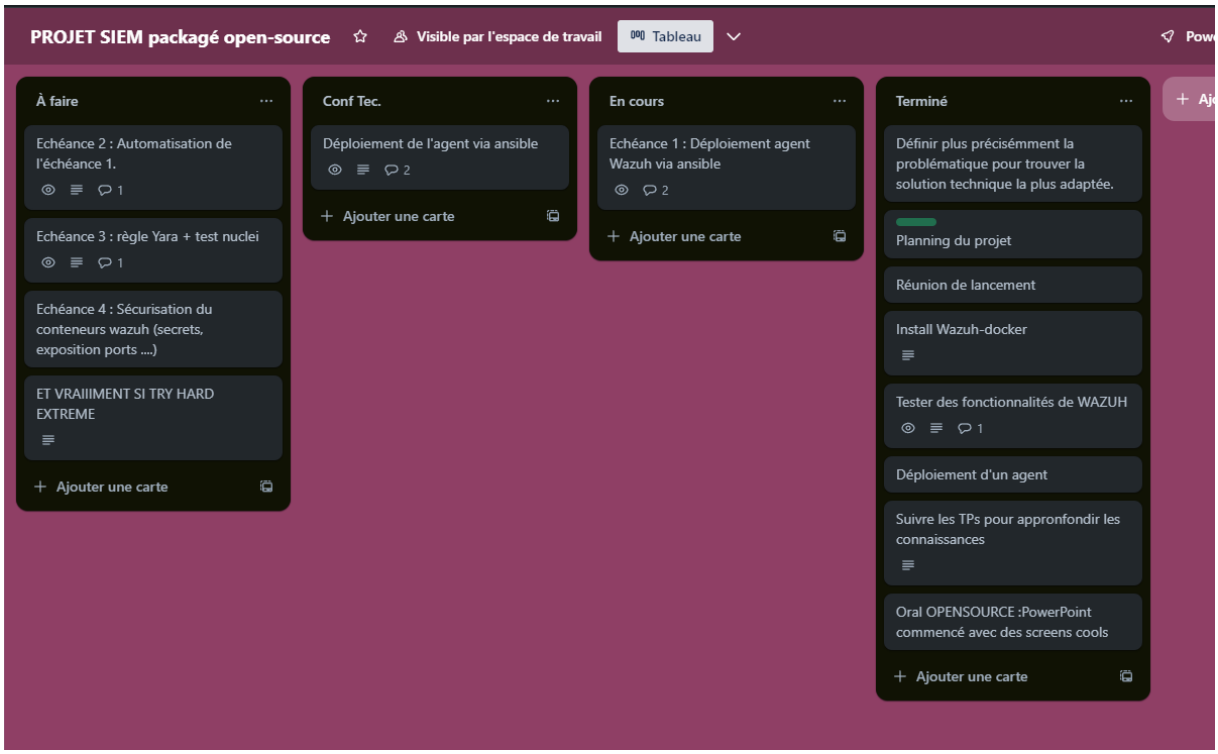
## Docker-ansible

La méthode Docker-Ansible combine la puissance des conteneurs Docker avec les capacités d'automatisation d'Ansible pour simplifier le déploiement et la gestion d'infrastructures basées sur Docker. En utilisant Ansible, les administrateurs peuvent définir l'ensemble de l'infrastructure sous forme de code, y compris la configuration des conteneurs, les réseaux, les volumes et les variables d'environnement. Cette approche permet une gestion cohérente et reproductible des environnements Docker à grande échelle, en facilitant les tâches de provisionnement, de configuration et de maintenance. En combinant Docker et Ansible, les entreprises peuvent automatiser efficacement les opérations de déploiement et de gestion des conteneurs, offrant ainsi une solution flexible et robuste pour le déploiement d'applications et de services.

# DEROULE DU PROJET

## Mise en place

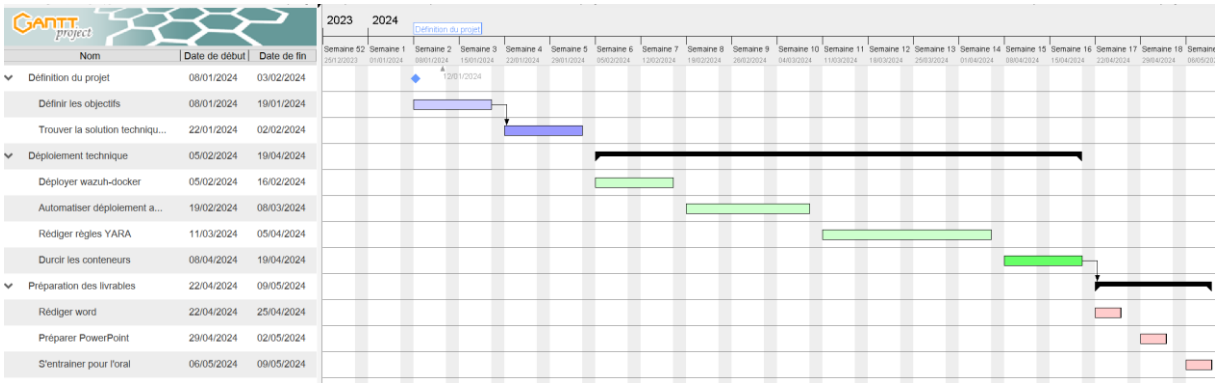
J'ai choisi d'utiliser un Trello pour organiser et visualiser mon travail. J'ai découpé mes objectifs à atteindre en « échéances » dans lesquels j'écris un détail, une manière de faire, des sources, une documentation… Et j'ai classé ces tâches en fonction de leur état d'avancement.




## Planning prévisionnel

Le planning prévisionnel a été réalisé avec GANTT et nous permet d'avoir une visualisation sur chaque tâche du projet, dans le temps.

Il va nous permettre de le comparer avec le planning réel pour voir les écarts et dégager des conclusions.



## Objectif 1 : Installation wazuh-docker

Cette première échéance consiste à installer wazuh-docker, un ensemble de conteneurs dans lesquels sont déployés les différentes parties de WAZUH : Indexer, Manager, Dashboard.

De manière descriptive, dans le fichier docker-compose.yml, les interactions entre conteneurs sont décrites grâce à l'implémentation du réseau dans ce fichier. (Voir l'extrait du fichier docker-compose.yml ci-dessous)

```yaml
# Wazuh App Copyright (C) 2017, Wazuh Inc. (License GPLv2)
version: '3.7'

services:
  wazuh.manager:
    image: wazuh/wazuh-manager:5.0.0
    hostname: wazuh.manager
    restart: always
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 655360
        hard: 655360
    ports:
      - "1514:1514"
      - "1515:1515"
      - "514:514/udp"
      - "55000:55000"
    environment:
      - INDEXER_URL=https://wazuh.indexer:9200
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - FILEBEAT_SSL_VERIFICATION_MODE=full
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/root-ca.pem
      - SSL_CERTIFICATE=/etc/ssl/filebeat.pem
      - SSL_KEY=/etc/ssl/filebeat.key
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
    volumes:
      - wazuh_api_configuration:/var/ossec/api/configuration
      - wazuh_etc:/var/ossec/etc
      - wazuh_logs:/var/ossec/logs
      - wazuh_queue:/var/ossec/queue
      - wazuh_var_multigroups:/var/ossec/var/multigroups
      - wazuh_integrations:/var/ossec/integrations
      - wazuh_active_response:/var/ossec/active-response/bin
      - wazuh_agentless:/var/ossec/agentless
      - wazuh_wodles:/var/ossec/wodles
      - filebeat_etc:/etc/filebeat
      - filebeat_var:/var/lib/filebeat
```

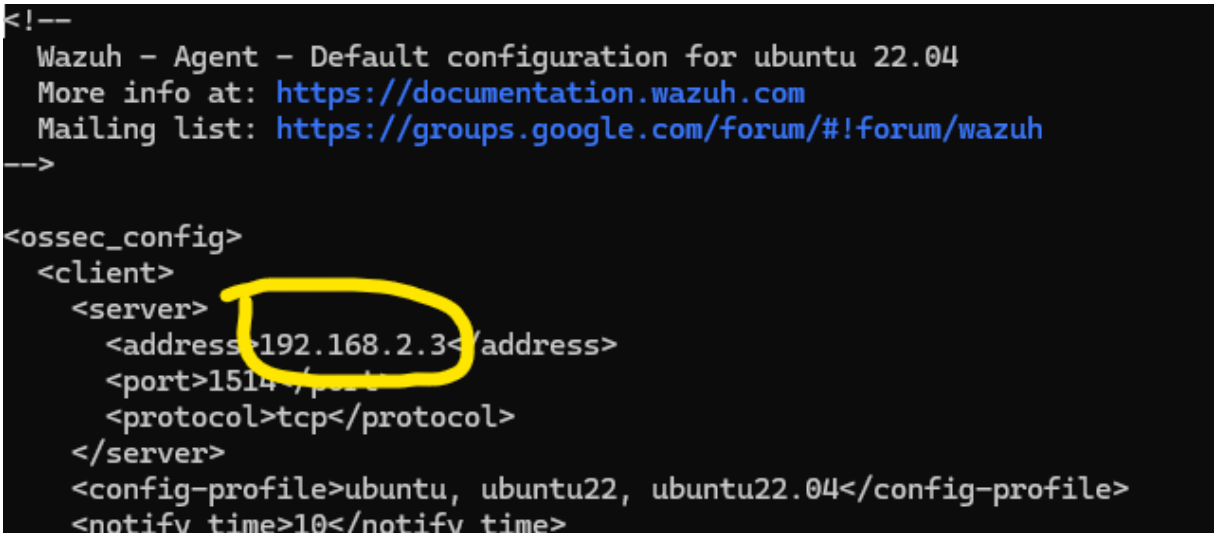
## Objectif 2 : Déploiement d'un agent Wazuh via Ansible

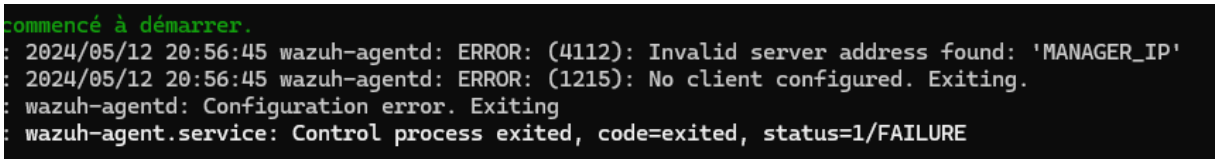
La seconde échéance fixée concerne le déploiement d'un agent Wazuh, c'est-à-dire, l'installation du logiciel qui permet la surveillance, via ansible.

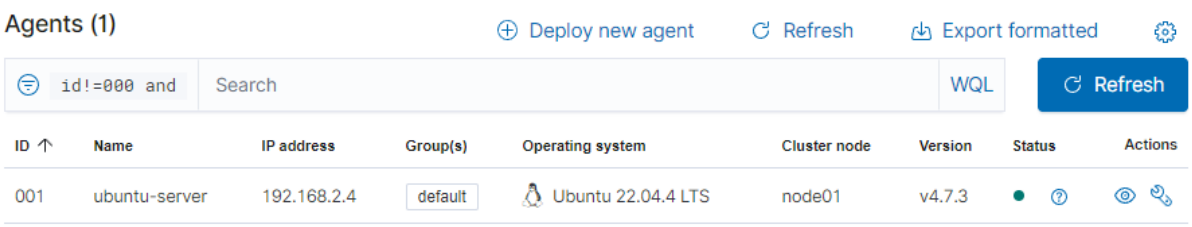
Cet agent est installé sur chaque machine à surveiller, et son rôle, de manière simplifiée, est de renvoyer les logs de la machine vers son centre de traitement qui analysera les logs.

Ansible permet de déployer des configurations via le protocole SSH. Cela permet d'automatiser certaines tâches récurrentes, d'homogénéiser la configuration des assets à surveiller.

J'ai fixé cet objectif, puisqu'il est primordial de réduire le TJM pour alléger la facture de notre TPE/PME autant que possible. Également, si l'entreprise n'a pas de SI mature, il est tout aussi important d'éviter à une ressource technique d'utiliser du temps pour déployer manuellement les agents sur toutes les nouvelles machines.



## Objectif 3 : Rédaction de règles YARA

Wazuh vient déclencher des alertes s'il reconnait dans les logs qu'il collecte, des preuves de tentatives de compromission, ou des comportements identifiés comme suspects.

Ces comportements ou ces preuves de tentatives de compromissions correspondent à des règles YARA et SIGMA qui existent déjà de base dans Wazuh et qui correspondent aux menaces déjà connues. Mais de manière non exhaustive.

Pour résumer, les règles en place ont pour point fort d'être généralistes, pouvoir correspondre à tout le monde. Le fait est qu'en correspondant à tout le monde, elle ne corresponde vraiment à personne. L'effet de bord de cela peut-être un excès de faux-positifs (chronophage pour la ressource technique en interne) ou encore, un périmètre de protection qui ne couvre pas bien le besoin du SI.

Cet objectif a été défini parce qu'il s'inscrit dans une volonté de créer des règles de détection « surmesures ».

Pour réaliser cet objectif, j'ai besoin de réussir à créer et intégrer une règle YARA et de la tester.

Dans un second temps, il serait intéressant d'utiliser des scanners de vulnérabilités, pour voir ce que Wazuh ne détecte pas avec ses règles de bases.

Dans un troisième temps, il faudrait automatiser la création et l'ajout de règles YARA en fonction des résultats du scans de la vulnérabilité.

## Objectif 4 : Durcissement des conteneurs Dockers

### *Dockers secrets*

Docker Secrets est un mécanisme permettant de gérer de manière sécurisée les données sensibles au sein d'environnements Docker. Il offre une solution centralisée pour stocker et distribuer des informations telles que des mots de passe, des clés d'API et d'autres données confidentielles nécessaires aux applications en cours d'exécution dans des conteneurs Docker. Les secrets sont chiffrés en transit et au repos, garantissant ainsi un niveau élevé de sécurité pour les informations sensibles.

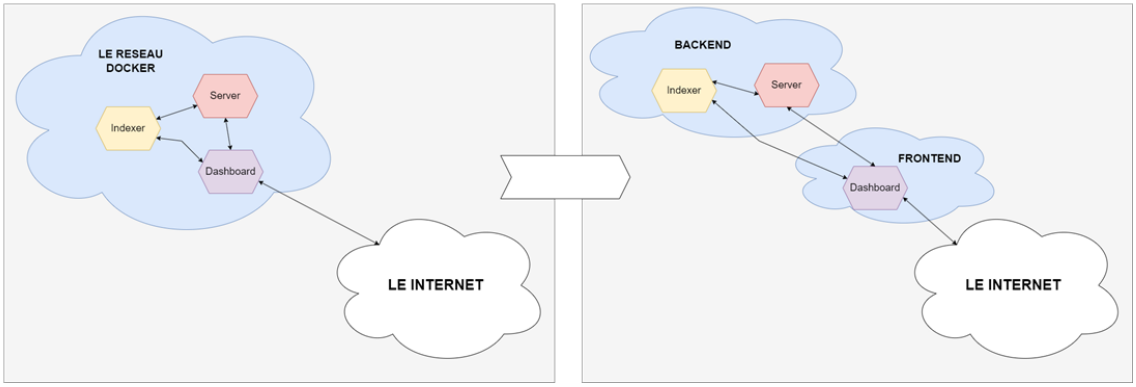
Docker Secrets permet également de limiter l'accès aux secrets uniquement aux conteneurs autorisés, renforçant ainsi la protection des données sensibles contre les accès non autorisés. En intégrant Docker Secrets dans leurs workflows, les développeurs peuvent sécuriser efficacement leurs applications tout en maintenant une architecture de conteneurisation flexible et évolutive.

*Exposition de ports*

Par défaut, le projet wazuh-docker n'applique pas le principe de sécurité du moindre privilège. C'est à dire que certaines interaction entre conteneurs sont permises alors qu'elles ne sont pas nécessaire :

- :514 → « Send collected events from syslog »
- :9200 → «Elasticsearch RESTful API »

Également, je souhaite ajouter un cloisonnement réseau en séparant le backend du frontend.




Pour appliquer ces modifications, je modifie le contenue du docker-compose comme sur les images ci-dessous.

Dans un premier temps je crée mes deux réseaux séparés et je vais préciser pour chaque composant de quel réseau il faut partie.

```
networks:
  backend:
    external: true

  frontend:
    external: true
```

```
        hard: 655360
    ports:
      - "1514:1514"
      - "1515:1515"
#     - "514:514/udp"
      - "55000:55000"
    networks:
      - backend
      - frontend
```

```
wazuh.dashboard:
  image: wazuh/wazuh-dashboard:4.6.0
  hostname: wazuh.dashboard
  restart: always
  ports:
    - 443:5601
  networks:
    - backend
    - frontend
```

```
wazuh.indexer:
  image: wazuh/wazuh-indexer:4.6.0
  hostname: wazuh.indexer
  restart: always
#   ports:
#     - "9200:9200"
  networks:
    - backend
```

Seconde étape, je viens commenter les lignes concernant les ports ouverts inutiles.

# PROBLEMES RENCONTRES

## Technique

A disposition pour ce projet, pas de ressources pécuniaires, seulement les performances de mon ordinateur personnel.

Les ressources matérielles utilisées n'étant pas équivalente à un serveur en production, j'ai rencontré plusieurs effets de bord.

Notamment, manque d'espace disque, crash de VMs.

La solution apportée : achat d'un SSD externe.

Pour le crash de VMs, croiser les doigts.

## Projet

Le souci rencontré sur le projet de manière plus globale est l'organisation du temps de travail.

Plus précisément, j'ai eu énormément de mal à diviser mon temps et mon énergie entre les multiples projets personnels, multiples projets en entreprises et les projets Ynov.

Celui-ci a alors pris beaucoup de retard et s'est fait dans l'urgence.

# BILANS

## Objectif 1 : Installation wazuh-docker

L'installation de Wazuh-docker s'est déroulé avec succès. Les autres tâches étant dépendantes de celle-ci, le commencement tardif de cette tâche a entrainé un retard sur les autres tâches techniques.

## Objectif 2 : déploiement d'un agent Wazuh via Ansible

En date du 12/05/2024, cette tâche n'est pas validée puisque je ne parviens pas à définir les variables d'environnement nécessaires à l'installation de l'agent sur le poste.

Néanmoins le prérequis pour cette tâche qui est d'acquérir la compétence : Ansible est validé. Je suis capable de comprendre de quoi je parle, d'interagir avec une personne experte sur le sujet. D'écrire des playbooks de niveau débutant.

## Objectif 3 : Rédaction de règles YARA

L'objectif est en cours de réalisation en date du 12/05/2024.

## Objectif 4 : Durcissement des conteneurs Dockers

Cet objectif comprend 2 étapes :

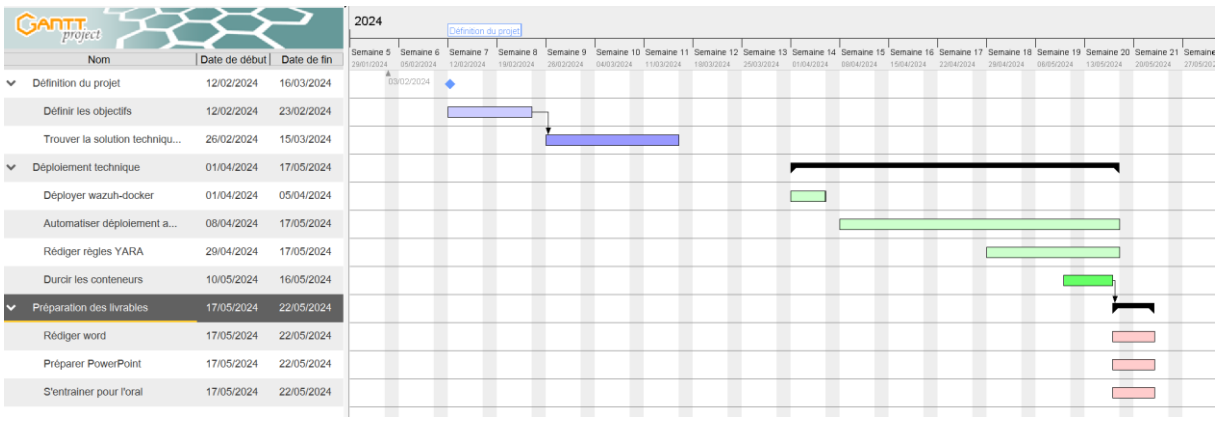
1. Mettre en place un cloisonnement réseau qui autorise strictement les échanges nécessaires entre les conteneurs (réduire l'exposition de ports notamment).
2. Rendre la gestion des secrets plus rigoureuse pour réduire le niveau de risques global du projet, notamment en passant par '*docker secrets*'.

La première partie est réalisée sans problème particulier.

La seconde partie est en cours de réalisation en date du 12/05/2024.

## Délais

Le planning réel montre des écarts importants avec le planning prévisionnel pour les raisons évoquées dans la section **« Problème rencontrés -> Projet ».**




Ce que l'on peut remarquer ce sont les tâches qui s'enchevêtrent toutes avant la date de rendu 12 mai 2024. En effet, étant la date de livraison du projet, j'ai fait le choix de proposer une piste de travail et de monter en compétences sur tous les objectifs en parallèle plutôt que de bloquer sur le 1er jusqu'à l'avoir validé.

Ce phénomène de tâches qui s'enchevêtrent a pour conséquences **: une diminution de la qualité de vie et de travail**, ce que l'on cherche à éviter au maximum dans un cas réel. Au long terme, ce mode de travail peut rendre le salarié réticent à l'idée de travailler sur le projet, voir faire naître une envie de démission.

En tant que chef de projet, il est important de trouver une solution rapidement si cela arrive.

**Solutions possibles pour détecter la situation avant le retard** : réunion, communiqué avec les membres de son équipe pour s'assurer qu'ils n'ont pas de difficultés majeures.

**Solutions possibles pour remédier si la situation n'est pas évitable** : fournir un appuie technique sur la tâche par une ressource avec plus d'expérience. Si nécessaire, faire une nouvelle répartition du travail pour alléger le salarié, voire, réaffecter la tâche.

## Compétences acquises

Pour résumer les compétences que j'ai pu ajouter à mon arc grâce à ce projet sont :

- Ansible
- Docker + Docker Secrets
- Wazuh
- Amélioration globale dans le débogage, la bonne gestion des droits …

## Retour d’expérience

Le sentiment de ne pas avoir tirer le maximum de ce que ce projet pouvait m’apporter en termes de compétences et de satisfaction, en grande partie à cause de ma priorisation qui a fait passer ce projet derrière d’autres. Je pense néanmoins poursuivre ce projet parce que je pense qu’il y a des bonnes idées et livrer quelque chose de fonctionnel pourrait être utile dans mon avenir professionnel.

# ANNEXES

## *Architecture réseau wazuh-docker*