Quentin
SCARPA

2025 Rapport

EAP et Docker

Ma manière de travailler et les outils utilisés

Vocabulaire utile

La **conteneurisation** est une méthode qui permet d'exécuter des applications dans des environnements isolés et légers appelés *conteneurs*.
Ces conteneurs regroupent le code, les bibliothèques et les dépendances nécessaires, ce qui garantit que l'application fonctionne de la même façon sur n'importe quelle machine.

Un **EAP** (*Environnement d'Apprentissage Personnel*) est l'ensemble des outils, ressources et méthodes qu'une personne utilise pour apprendre, s'organiser et développer ses compétences.

Un **honeypot** est un système informatique volontairement vulnérable ou factice, mis en place pour attirer les cyberattaquants afin de les observer, analyser leurs méthodes et renforcer la sécurité.

[INTRODUCTION]

[Contexte]

Dans le cadre de ma reconversion professionnelle, j'ai choisi un BTS SIO que je réalise en alternance à la mairie de Caen. La formation se déroule en deux ans dans laquelle nous devrons choisir dès décembre de cette année notre spécialité.

[Objectif de ce rapport]

L'objectif de ce rapport aujourd'hui, est de montrer l'environnement dans lequel je travaille, ainsi que les outils que j'utilise et donc de démontrer leur puissance une fois utilisé correctement.
Enfin j'expliquerai le principe et l'utilisation du logiciel de **conteneurisation** ***Docker*** à l'aide d'un exemple simple.

[Mon EAP]

[Lieu et organisation]

VERSON, Vue des airs.

J'ai eu l'occasion avec ma compagne de trouver un appartement neuf dans une ville calme en périphérie de Caen. Le calme étant essentiel à mes yeux pour travailler en journée ou même tard le soir, j'estime qu'il est important de l'intégrer dans mon **EAP**.

Mon bureau :

Mon lieu de travail principal, mon bureau au travail étant assez similaire hormis le fait qu'il est en open-space. J'aime avoir de l'espace et j'ai donc pour projet de réaliser mon propre bureau afin d'avoir mon deuxième écran. Sinon, rien d'incroyable mis à part que j'aime également travailler avec une boisson chaude et un matériel de qualité et fiable :

⚙️ Configuration Hardware

💻 PC Principal

- **Processeur :** AMD Ryzen 9 7950X — 16 Cores @ 4.50 GHz
- **Mémoire vive (RAM) :** 32 Go (31,6 Go utilisables)
- **Carte graphique :** NVIDIA GeForce RTX 4080 (16 Go)

💾 Stockage

- 932 Go — Samsung SSD 980 PRO with Heatsink (1 To)
- 466 Go — Samsung SSD 850 EVO (500 Go)
- 1,82 To — Sabrent SB-RKT4P-2TB
- 932 Go — SanDisk SDSSDA-1T00

🎧 Périphériques

- **Casque :** Beyerdynamic DT 990 Pro
- **Souris :** Logitech G502 X Lightspeed
- **Clavier :** Corsair K70 Pro

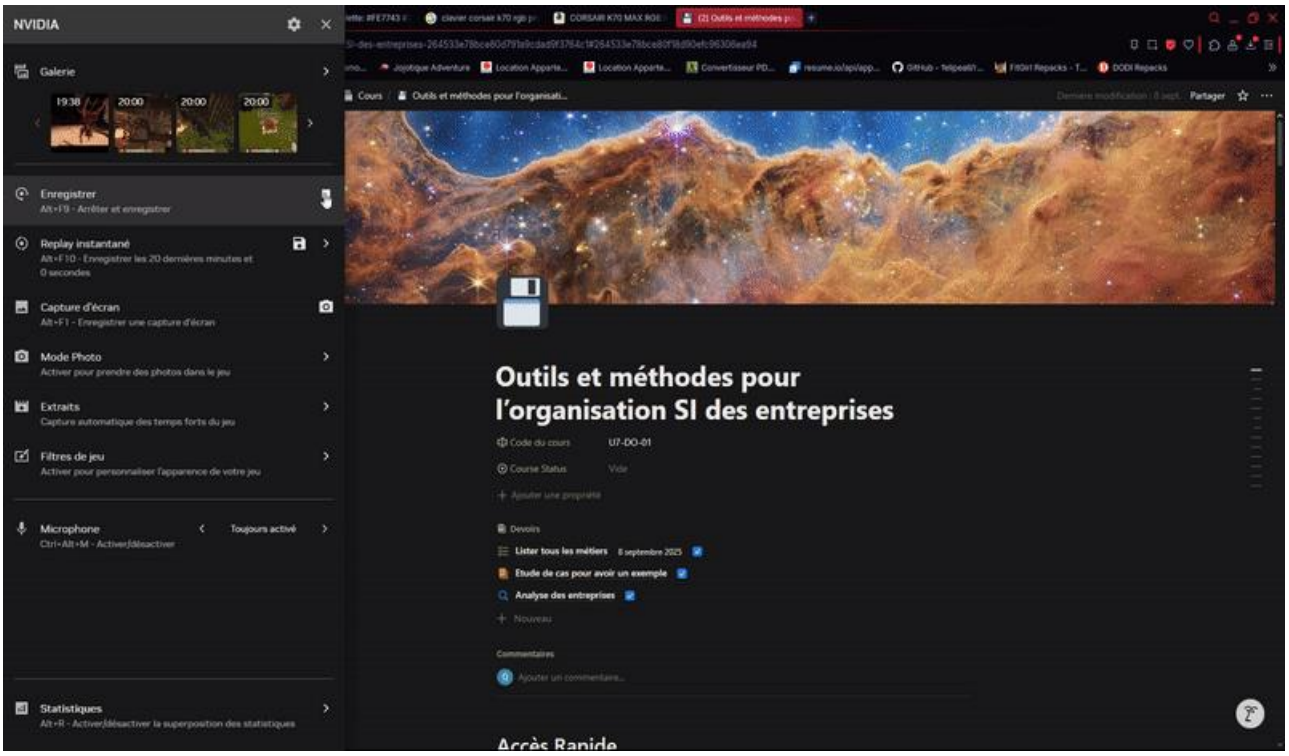
[Mes outils logiciels]

[Prise de notes]

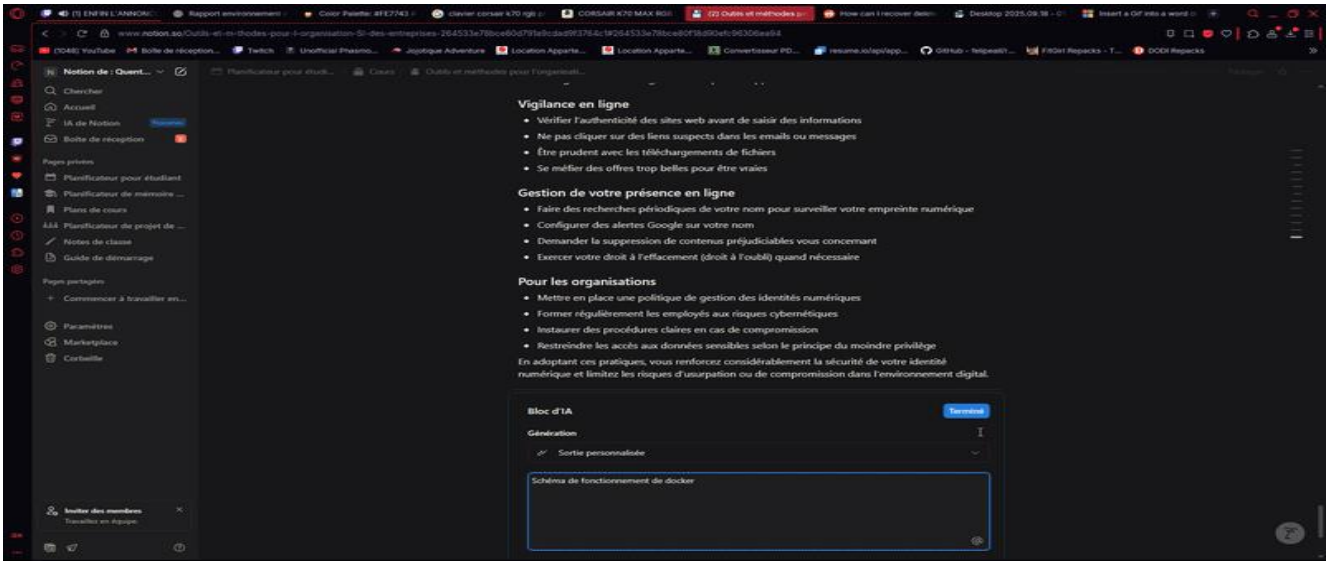
J'utilise maintenant ***Notion*** depuis la présentation de l'outil en cours. Je le trouve puissant, versatile et rapide si bien utilisé. Mais c'est quoi au juste : **Notion** est un outil de productivité tout-en-un qui permet de prendre des notes, gérer des tâches,

organiser des projets et centraliser des ressources dans une interface collaborative et personnalisable.

Notion s'organise en « Page » comme celle-ci



On peut personnaliser absolument tout pour obtenir un plan complet de nos notes et les interconnecter pour avoir un suivi des devoirs, l'accès rapide au sein de notre page et j'en passe. Enfin la fonctionnalité « Bloc ia » qui permet de demander à l'ia de Notion de réaliser des tâches simples comme l'exemple ici :



Défaut de notion : l'export pdf vers un fichier word peut-être perplexe et amener des erreurs, c'est une amélioration nécessaire à mon sens car le transfert de tableau par exemple vers un autre type de fichier peut vite devenir compliqué si mes cellules sont décalées.
Pareillement pour les accès rapides qui disparaissent complétement une fois convertis en PDF.

[Rapport et présentation]

Pour le rapport d'aujourd'hui j'utilise « Word » qu'on ne présente plus.
J'utilise également la suite Adobe, principalement les suivants :

Adobe Photoshop

- Logiciel de **retouche et création d'images**.
- Idéal pour la manipulation photo, la conception graphique et la création de visuels professionnels.

Adobe Premiere Pro

- Logiciel de **montage vidéo professionnel**.
- Utilisé pour assembler, couper et enrichir des vidéos avec transitions, effets et audio.

Adobe After Effects

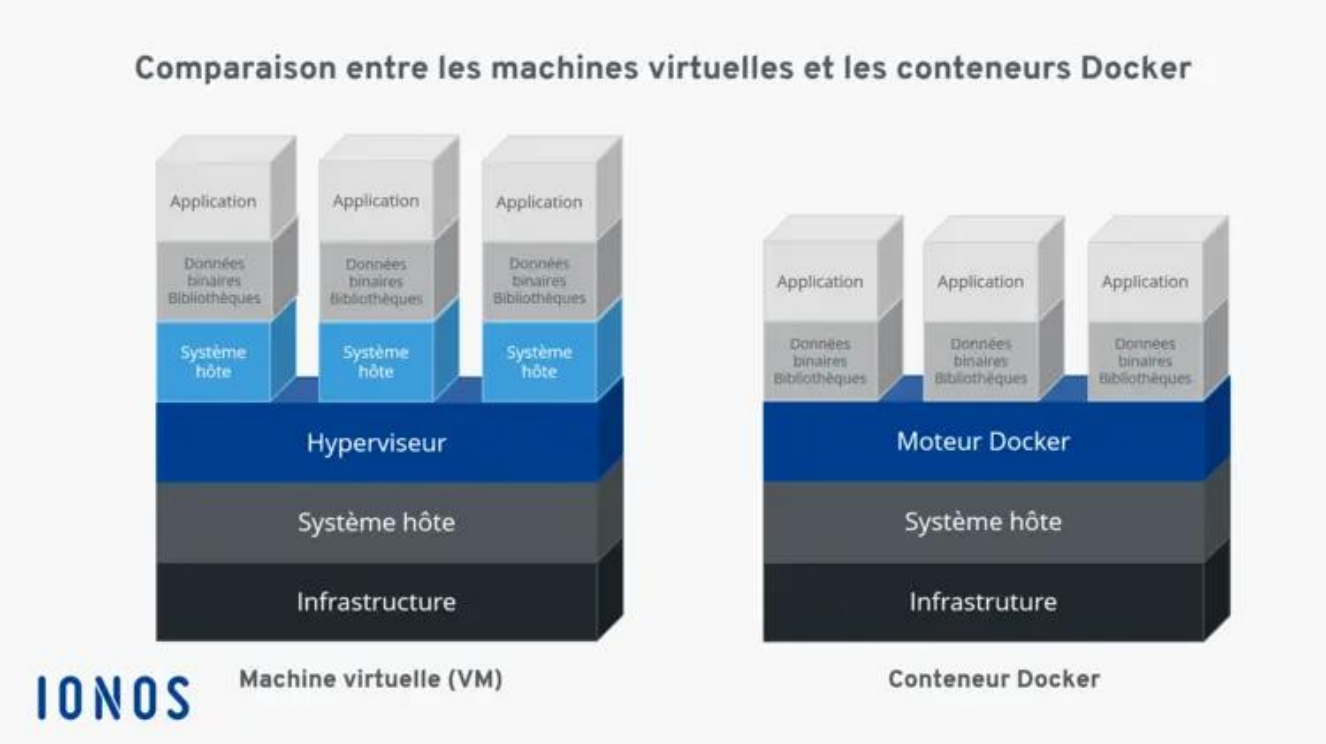
- Logiciel spécialisé en **effets visuels et animation**.
- Permet de créer des génériques, des animations 2D/3D et des effets spéciaux pour enrichir les vidéos.

[Technologie apprise : Docker]

[Définition]

Docker

Description : *Docker* est une plateforme de **conteneurisation** qui permet d'exécuter des applications dans des environnements isolés appelés *conteneurs*. Chaque conteneur embarque le code, les bibliothèques et les dépendances nécessaires, ce qui garantit que l'application fonctionne de la même manière sur n'importe quelle machine.



Méthode	Virtualisation	Conteneurisation
Définition	Création de machines virtuelles (VM) complètes avec leur propre OS sur un hyperviseur.	Exécution d'applications isolées dans des conteneurs partageant le même OS hôte.
Ressources	Plus lourde : chaque VM inclut un OS complet.	Légère : le conteneur utilise l'OS de l'hôte, moins de ressources consommées.
Démarrage	Lent : le démarrage d'une VM prend plusieurs minutes.	Rapide : un conteneur démarre en quelques secondes.
Portabilité	Limitée : les VM sont lourdes et dépendent de l'hyperviseur.	Très portable : un conteneur peut tourner sur n'importe quel système compatible Docker.
Cas d'usage	Tester des OS complets, isoler totalement des environnements.	Déployer des applications rapidement, développer et tester des services.

Pour résumer la conteneurisation permet, souvent aux développeurs, de travailler sur une app et de la tester avec exactement le même environnement sur un hôte machine complètement différent en incluant les librairies et dependencies de l'app dans un package, nommé « Docker Image » générée à l'aide d'un « Docker File ».

Attention tout de même : Une « Docker image » générée sur hôte machine ***MAC OS*** ne fonctionnera probablement pas sur un hôte machine ***WINDOWS*** et vice-versa

[Comment ai-je appris Docker ?]

Grâce à un collègue ingénieur. Il a pu répondre à deux de mes questions :
- La différence entre ***Virtualisation*** et ***Conteneurisation***.
- La possible incompatibilité d'une image entre deux hôtes avec un OS différent.

Mais également m'expliquer une utilisation concrète de Docker avec la création d'***HoneyPots*** pour tracker des possibles hackers qui chercherait à attaquer un serveur qui au final ne sert à rien.

Youtube est une source de connaissances inépuisable, encore plus quand on maîtrise l'anglais, d'où ma préférence de cet outil pour apprendre tout ce dont j'ai besoin.

[Obtention d'une Docker image]

Plusieures manières d'obtenir une image, on peut simplement chercher une image existante sur le « Docker Hub » à l'aide de la commande suivante :

```
docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
92c3b3500be6: Pull complete
ee57511b3c68: Extracting [==========================================>    ]  36.18MB/38.51MB
33791ce134bf: Download complete
cc4f24efc205: Download complete
3cad04a21c99: Download complete
486c5264d3ad: Download complete
b3fd15a82525: Download complete
```

Ici ***Nginx***, est un serveur web open source, également utilisé comme proxy inverse.
Mais on peut aussi l'avoir en la construisant nous même à partir d'un ***Docker File*** :

```
docker build [OPTIONS] PATH
```

Mais comment obtenir un ***Docker File et*** qu'est-ce qu'on y met ?
Un ***Docker file***, c'est tout simplement une suite d'instructions qui permettent de créer une image fonctionnelle à partir d'une image de base.

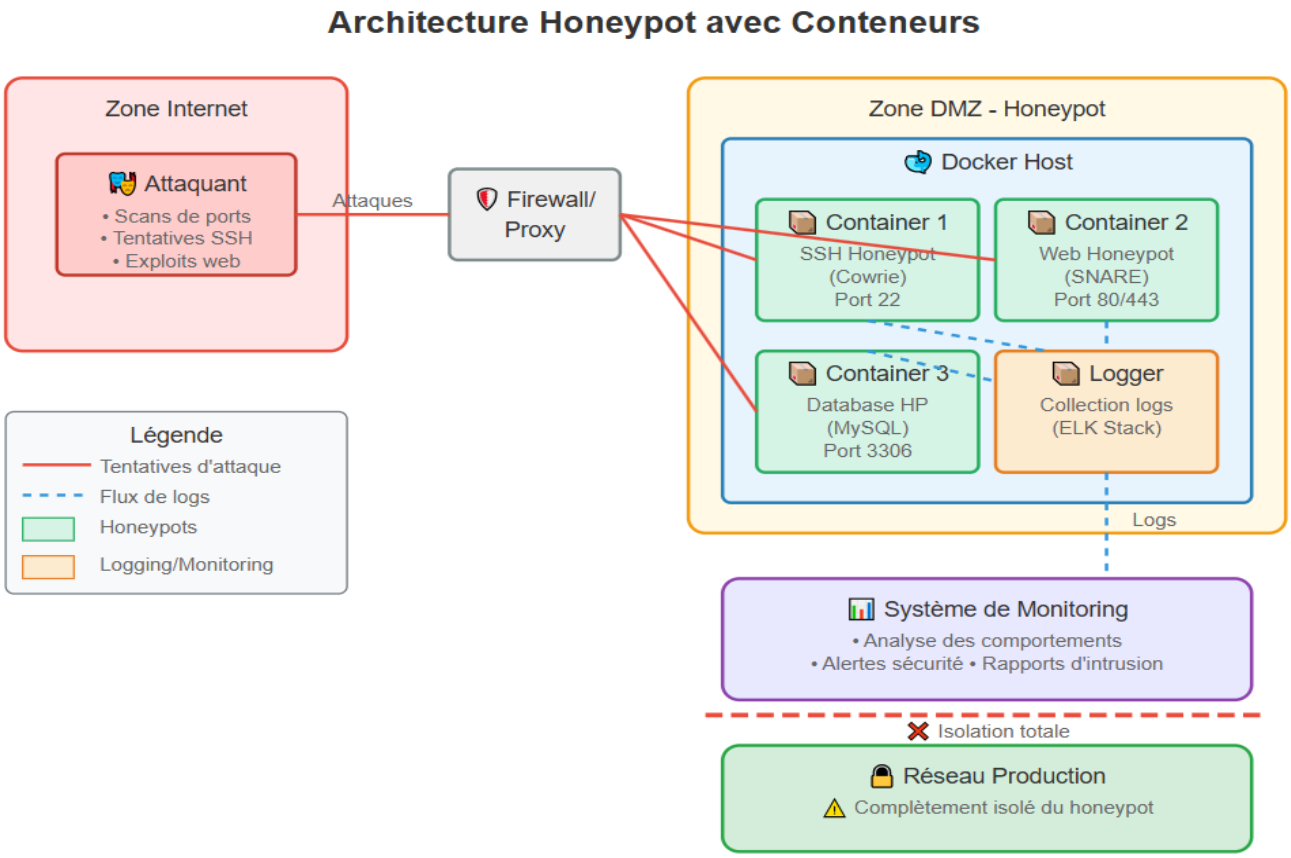
Il contient absolument tout ce qui est nécessaire au bon fonctionnement de l'app (le code, ses dépendances, configuration de l'environnement et enfin la commande pour lancer l'application)
Exemple :

```
FROM python:3.11-slim
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "main.py"]
```

Ici l'image de python utilisée avec la commande « from » est adaptée donc le nombre de commande pour installer les dépendances est moindre.
Cette installation se fait à travers « RUN pip install -r requirements.txt » où le « -r » indique que le fichier « requirements.txt » est à lire pour savoir quoi installer dans le conteneur.
Il y a plus de commandes que l'on peut utiliser dans un ***Docker File*** bien évidemment selon les besoins de l'utilisateur je ne les connais pas encore toute, bien au contraire mon apprentissage ne fait que commencer.

[Cas pratique : Le HoneyPot]

Ceci est aperçu surfacique de ce que l'on peut faire avec Docker. Je n'ai pas eu l'occasion de voir la console de commande ni les conteneurs « ***HoneyPots*** » de mon collègue de travail. En revanche avec ses explications j'ai schématisé l'utlité de son travail :



Explication du schéma :

1. L'attaquant (zone rouge) lance diverses attaques depuis Internet : scans de ports, tentatives SSH, exploits web.
2. Le firewall/proxy filtre et redirige le trafic suspect vers la zone DMZ où se trouve le honeypot.
3. Les conteneurs honeypot (zone verte) simulent différents services vulnérables :
    - Container 1 : SSH honeypot sur port 22
    - Container 2 : Serveur web honeypot sur ports 80/443
    - Container 3 : Base de données honeypot sur port 3306
4. Le système de logging (container orange) collecte toutes les tentatives d'intrusion.
5. Le monitoring (zone violette) analyse les comportements malveillants, génère des alertes et produit des rapports d'intrusion.
6. L'isolation critique : Le réseau de production est complètement isolé du honeypot. Aucune connexion n'est possible entre les deux environnements.

Avantages de cette architecture conteneurisée :

- Isolation forte entre les honeypots
- Déploiement rapide et reproductible
- Facilité de réinitialisation après compromission
- Ressources limitées pour chaque conteneur
- Monitoring centralisé des activités malveillantes

[Conclusion]

On peut remarquer qu'avec un EAP adapté ainsi que les bons outils on peut travailler plus rapidement et efficacement et ainsi réaliser des comptes rendus comme celui-ci que je pense être professionnel même si loin d'être parfait.

Les outils de notes comme, ***Notion***, permettent de mieux suivre les cours, d'organiser et centraliser tous les documents malgré une connexion internet nécessaire

A travers cet exercice j'ai également amélioré ma maîtrise de ***Word***, appris une utilisation simple de Docker et j'ai commencé à tenir un Planificateur de tâches grâce à une web app nommée ***ClickUp***.

Il me reste malgré tout à utiliser plus mon planificateur de tâches, à penser à faire des pauses et à diversifier mes connaissances sur les outils et ressources que je peux utiliser, comme Microsoft Learns, sur lequel j'ai briévement travaillé.