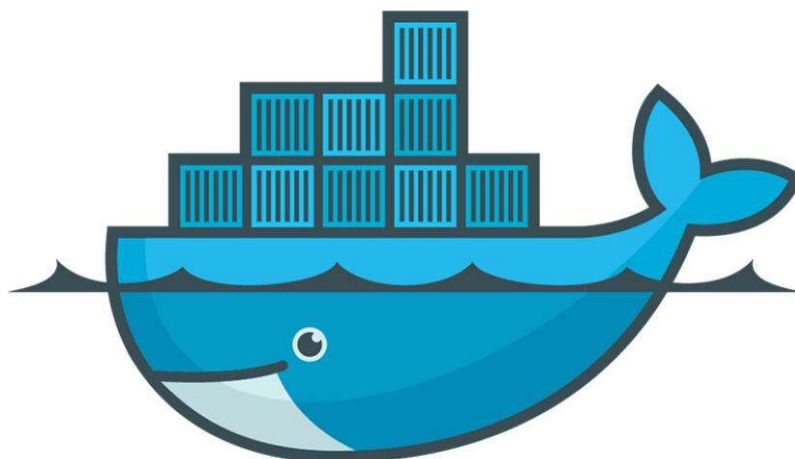

1 PROCÉDURE

- Les apprentis travaillent individuellement.
- Le cahier des charges est présenté, commenté et discuté.
- Les apprentis sont entièrement responsables de la sécurité de leurs données.
- En cas de problèmes graves, les apprentis avertissent leur chef de projet au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Les informations utiles à l'évaluation de ce projet sont disponibles au chapitre 3.2.

2 TITRE : Docker Network Toolkit

Objectif du projet (SMART) :

- Dans 5 semaines, à la fin de ce projet, chaque élève sera capable de créer un conteneur Docker personnalisé à partir d'une image de base et de documenter ce processus en utilisant un fichier Dockerfile, tout en respectant les bonnes pratiques de Docker.



1 SUJET

Bienvenue dans le projet "**P_Bulle DockerNetworkToolkit**". Ce projet vise à vous initier aux concepts de la virtualisation par conteneur, en vous permettant de créer, gérer et orchestrer des conteneurs Docker. À la fin de ce projet, vous serez en mesure de comprendre les principes de base de Docker, de créer des conteneurs individuels et de les orchestrer en utilisant Docker Compose.

Les attentes sont la création d'une boîte à outil pour l'informaticien orienté infrastructure sous la forme d'un ensemble de conteneurs ayant des outils spécifiques aux tests réseaux ou de système.

2 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC de laboratoire standard.
- Microsoft Office Professionnel.
- VirtualBox
- Notepad++, Visual Studio Code, Visual Studio Community.

3 CAHIER DES CHARGES

3.1 Temps à disposition

- 40 périodes

3.2 Éléments évalués

- Dockerfile :
 - Votre Dockerfile doit démontrer votre aptitude à personnaliser un container.
- Docker compose :
 - Un fichier docker-compose.yml est produit et permet l'orchestration de plusieurs services.
- Documentation et rapport technique :
 - Les étapes clés de votre projet sont documentées de telle sorte à ce qu'un autre informaticien puisse reproduire votre projet sans connaissance préalable de Docker.
 - Le schéma réseau doit être exhaustif et montrer les éléments permettant la communication inter-containers et la communication avec l'hôte.
- Présentation orale :
 - Lors de la dernière séquence du projet, vous présenterez à l'aide d'un PowerPoint les éléments saillants de votre projet et les apprentissages obtenus.

1 Procédure

1. Création d'un premier conteneur : Les élèves devraient créer un conteneur Docker simple à partir d'une image de base (par exemple, une image Linux) et le personnaliser en y installant un logiciel de leur choix. Ils devraient également comprendre les commandes Docker de base telles que `docker build`, `docker run`, `docker stop`, `docker exec`, etc.
2. Introduction à Docker Compose : Comment se faciliter la vie en ayant un début d'IaC (Infrastructure as Code). L'enseignant explique comment docker compose permet de définir et d'orchestrer plusieurs conteneurs, et comment rédiger un fichier `docker-compose.yml`.
3. Création d'une composition Docker : Les élèves devraient travailler en équipe pour créer une composition Docker qui inclut au moins deux conteneurs. Par exemple, un conteneur pour une application web et un autre pour une base de données. Ils doivent définir les services, les réseaux et les volumes nécessaires dans le fichier `docker-compose.yml`. Un schéma réseau est attendu pour expliquer comment Docker permet la communication entre les conteneurs et/ou avec l'hôte. Les adresses IP en jeu doivent être mentionnées.
4. Déploiement et test : Les élèves devraient déployer leur composition Docker et s'assurer que les conteneurs communiquent correctement entre eux. Ils devraient également tester l'application web pour vérifier son bon fonctionnement.
5. Documentation : Chaque équipe doit documenter leur projet en expliquant comment fonctionne leur composition Docker, quelles applications ils ont mis en place, les commandes utilisées, et les problèmes rencontrés et résolus.
6. Présentation : Les équipes présentent leur projet à la classe, en expliquant leur composition Docker, les défis rencontrés et les leçons apprises.