Récemment, nous avons été confrontés à la problématique de gestion de TP au sein de l’ESI.

On veut concevoir un système qui permet aux enseignants de partager les applications utilisées dans les Travaux pratiques avec leurs étudiants, d’une manière plus facile et plus efficace.

L'objet de cette démonstration sera la première partie de notre solution: la conteneurisation avec Docker.

On va commencer par justifier le choix de notre solution et des outils de conteneurisation utilisés.

Ensuite on va:

* expliquer la procédure de la mise en place en détails
* montrer comment les images créées peuvent être exploitées par les enseignants.
* et démontrer les résultats de notre projet en termes de fonctionnement à travers un scénario de tests, et en termes de performance et de consommation des ressources.

pour les 2 applications qu’on a choisies

et à la fin on va citer les limites de cette solution, qui motive l’utilisation des outils d’orchestrations.

**INTRO:**

La solution jusqu'à maintenant est de mettre en place des machines virtuelles, d'installer les applications sur ces machines et d'effectuer les configurations nécessaires sur chaque VM qui va être utilisées après par plusieurs étudiants.

Le problème avec cette approche est qu'elle utilise énormément de ressources, le processus d'installation du système d'exploitation et de toutes les applications et leurs dépendances sur chaque VM est fastidieux et répétitif et nécessite une connection internet la plupart du temps, de plus le système en VMs peut être difficile à maintenir, surtout quand différents utilisateurs doivent pouvoir accéder au même machines.

La solution à ce problème est la conteneurisation.

A travers les conteneurs, on peut limiter la virtualisation au niveau des applications en regroupant uniquement l'exécutable d'une application avec ses dépendances, et en créant un environnement d'exécution facilement partageable, duplicable et compatible avec les différentes infrastructures matérielles et logicielles.

Il existe plusieurs outils de conteneurisation, parmi les plus populaires et les plus utilisés on trouve: LXD, Docker, RKT.

Dans notre cas, on a choisi d'utiliser Docker, car:

1. il est compatible avec tous les 3 OS: linux,windows et mac
2. il peut être utilisé avec des outils d'orchestration comme Kubernetes,
3. il facilite le partage et la manipulation des conteneurs,
4. et il est fiable.

**Conclusion:**

Ayant résolu le problème des performances, des ressources et de la portabilité. On se trouve confronté à une nouvelle problématique: comment pouvons-nous automatiser le déploiement et la gestion de ces conteneurs? Comment peuvent-ils partager les ressources disponibles entre eux tous d’une manière optimale? et comment assurer la mise à l'échelle?

La solution cette fois-ci est d’utiliser un outil d’orchestration: Kubernetes qui sera le sujet de notre prochaine démonstration.