

Préparation :

1. Créez une **nouvelle VM Ubuntu ou CentOS**. Travail par équipe de deux/individuel. Merci de donner les noms des équipes.
2. **Documentation explicative (Si besoin)**

Votre première tâche : Installer Docker dans votre VM :

1. Démarrez la VM Ubuntu
2. Installez-y docker
3. Installer un conteneur Web avec Ubuntu
4. Installer un conteneur en utilisant CentOS
5. Installer et utiliser git et docker

Tâche-1 (30%) : Créer un fichier Dockerfile ubuntu.

1. Créez un Dockerfile dans [~/TP1-1](#).
 - 1.1. Utilisez l'image ubuntu
 - 1.2. Seront installés dans cette image (dockerFile) un serveur Web sécurisé de site avec votre nom et prénom, vim et lynx et gimp
2. Créez l'image et [donnez-lui l'étiquette qui portera votre Ubuntuprénom](#).
3. Exécutez l'image Ubuntu[votreprénom](#).
4. Capturez l'image de votre conteneur dans lequel vous exécutez la commande : "lynx @IP_votre_index". Intégrez le fonctionnement de la partie dans votre document
5. Mettez également votre **Dockerfile** dans la documentation.

Tâche-2 (20%) : créer un fichier Dockerfile

1. Créez un Dockerfile dans [~/TP1-2](#).
 - 1.1. Utilisez l'image centos latest
 - 1.2. Seront installés dans cette image (dockerFile) un serveur Web httpd sécurisé, net-tools
2. Créez l'image et [donnez-lui l'étiquette qui portera votre Centosprénom](#).
3. Exécutez l'image Centos[votreprénom](#).
4. Capturez l'image de votre conteneur dans lequel vous exécutez la commande : "lynx @IP_votre_index". Intégrez le fonctionnement de la partie dans votre document
5. Mettez également votre **Dockerfile** dans la documentation.

Tâche-3 (30%) : créer un fichier Docker compose.

1. Mettez en place avec docker compose avec les services (fichier docker-compose.yml)
 - 1.1. Une application base de données Mysql qui utilise une image MYSQL du repository
 - 1.2. Une application serveur web qui utilise votre image nginx de la tâche-1.
 - 1.2.1. Configuration sur le port 5000
 - 1.2.2. Ajout du volume persistant au conteneur qui sera sur un répertoire de votre système
 - 1.2.3. Limiter l'utilisation de CPU à 250m et la RAM à 400M
 - 1.3. Les applications [jenkins](#) sur le port 9005 et [sonarqube](#) sur le port 9006
 - 1.3.1. Limiter l'utilisation de CPU à 250m et la RAM à 1 GO
2. Personnaliser les adresses IP pour offrir des adresses personnalisées à chacune de vos applications

3. Faites un ping de votre conteneur Web sur vos services de base de données pour prouver que vos conteneurs communiquent.

Tâche -4 (recherche) (10%). Supposons que vos applications conteneurisées nécessitent un serveur DHCP pour louer des adresses IP aux conteneurs exécutés sur le même réseau. Mettez en place un serveur DHCP "Dockerisé" pour simplifier l'attribution des adresses IP à vos autres conteneurs, le déploiement et la gestion de vos applications conteneurisées.

Tâche-5. (10%) En utilisant les lignes de commandes, déposer vos fichiers dockerFiles dans votre compte docker hub ou dans votre git (serveur git du département : https://git.dti.crosemont.quebec/users/sign_in)