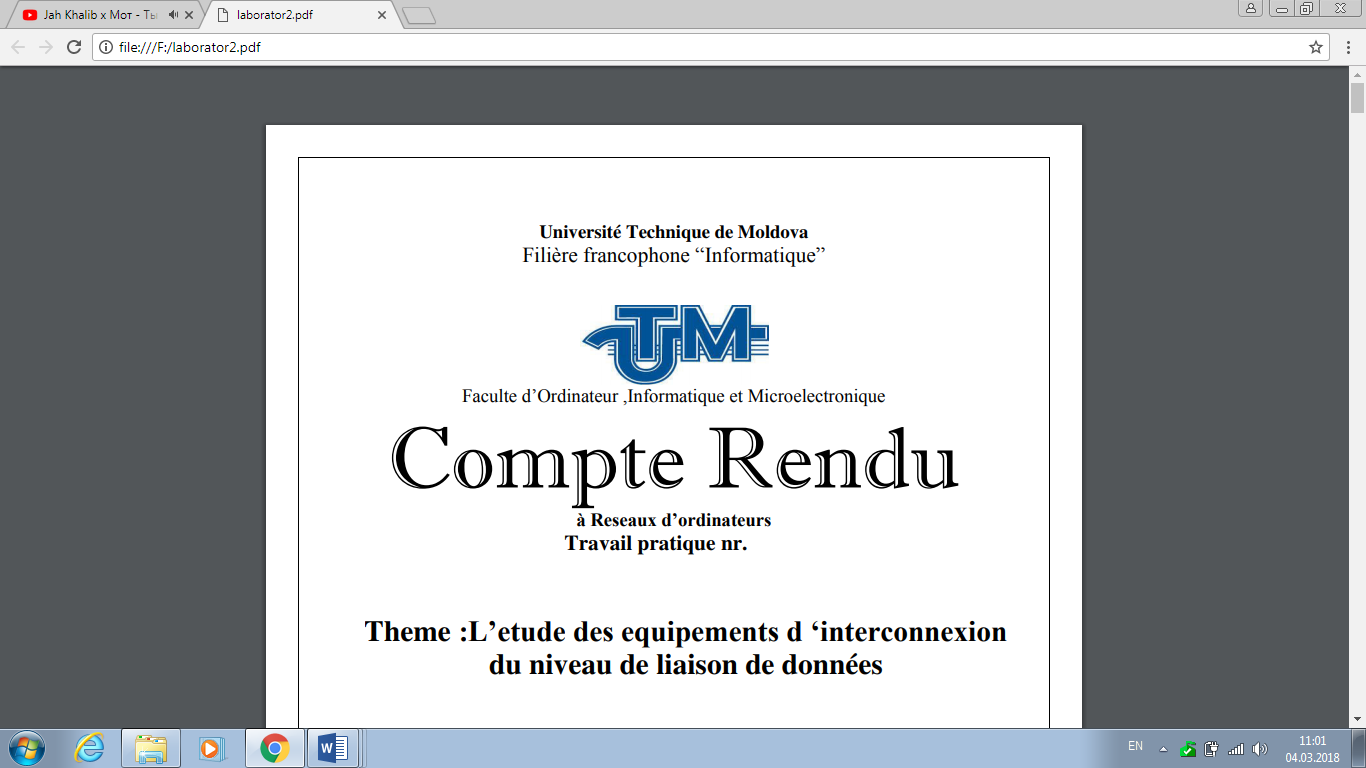
Ministère de l'éducation, de la culture et de la recherche

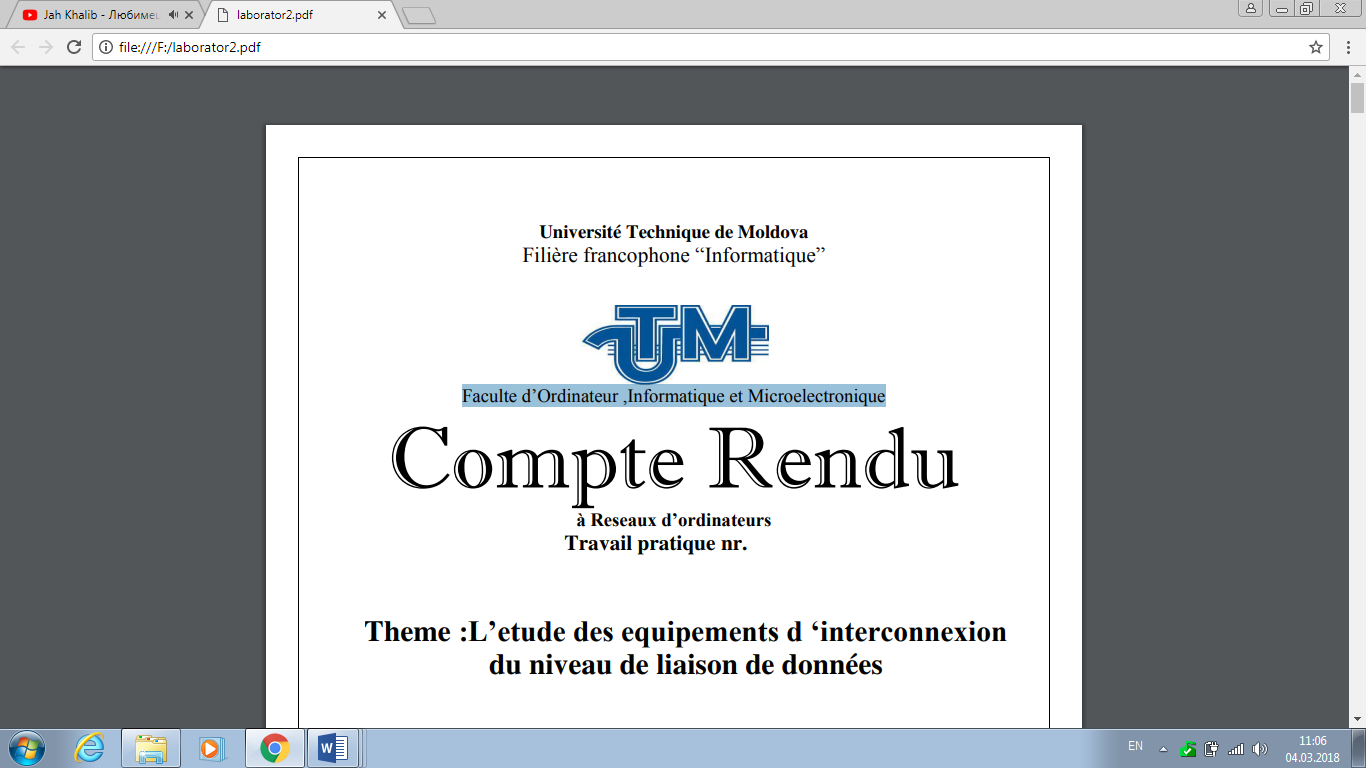
de la République de Moldova

Université technique de Moldavie

Faculté d’Ordinateur, Informatique et Microélectronique

Filière francophone “Informatique”





**TIDPP**

Travail pratique nr.6

**Thème:** **Utilizarea containere docker pentru configurarea unui mediu de lcuru.**

Effectué par l’étudiant de gr FI-181 : Damean Alexandra

Vérifié par le professeur : Rusu Viorel

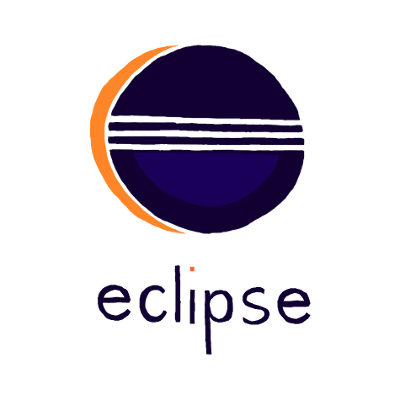
Chișinau 2020

**Tema** : Utilizarea containere docker pentru configurarea unui mediu de lcuru.

**Scopul:** installarea Docker containere pentru configuraea a cel putin 2 medii de lucru diferite pe acelasi PC/sistem de operare utilizat in procesul de dezvoltare.

***1. Se identifica mediile de lucru ce trebuie instalate (cel puțin 2) și configurate pentru asigurarea procelusului de dezvoltare a unei produs software bazat pe una tehnologia/limbajul propus:***

a) JAVA





***Concepts Docker***

***Docker*** *est une plate-forme permettant aux développeurs et aux administrateurs système de créer, d'exécuter et de partager des applications avec des conteneurs. L'utilisation de conteneurs pour déployer des applications s'appelle la conteneurisation . Les conteneurs ne sont pas nouveaux, mais leur utilisation pour déployer facilement des applications l'est.*

***La conteneurisation est de plus en plus populaire car les conteneurs sont:***

***Flexible :*** *même les applications les plus complexes peuvent être conteneurisées.*

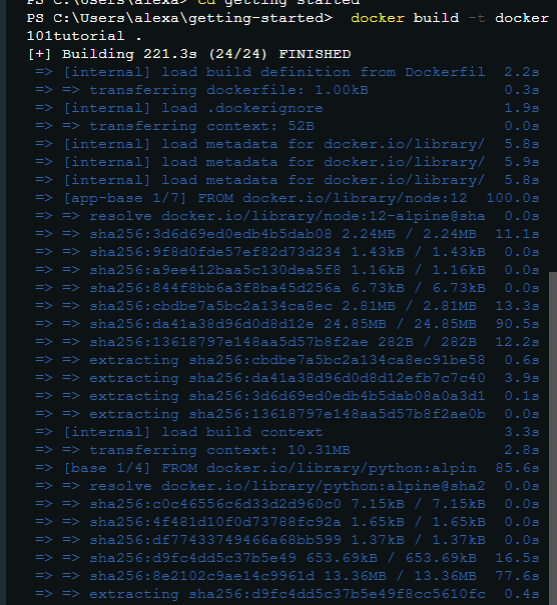
***Léger*** *: les conteneurs exploitent et partagent le noyau hôte, ce qui les rend beaucoup plus efficaces en termes de ressources système que les machines virtuelles.*

***Portable :*** *vous pouvez créer localement, déployer sur le cloud et exécuter n'importe où.*

***Lâchement couplé*** *: les conteneurs sont hautement autonomes et encapsulés, ce qui vous permet de remplacer ou de mettre à niveau l'un sans en perturber les autres.*

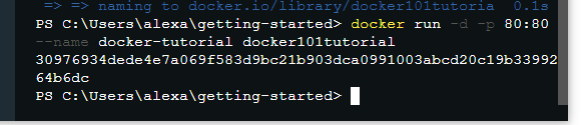
***Évolutif :*** *vous pouvez augmenter et distribuer automatiquement les réplicas de conteneurs dans un centre de données.*

***Sécurisé*** *: les conteneurs appliquent des contraintes et des isolements agressifs aux processus sans aucune configuration requise de la part de l'utilisateur.*

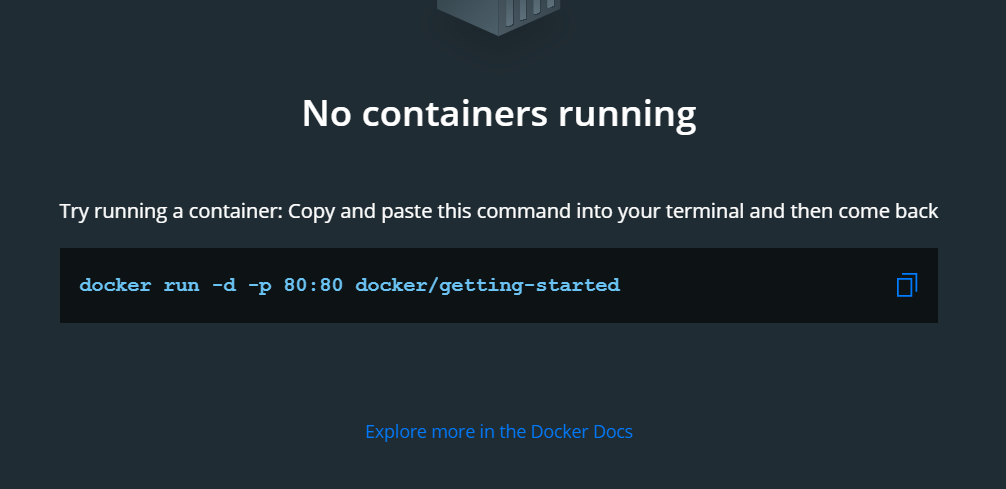
***2. Se instaleaza containerele ncesare și se descrie pe pași procesul*****

*Install Doker tutorial:*

*Run Doker:*

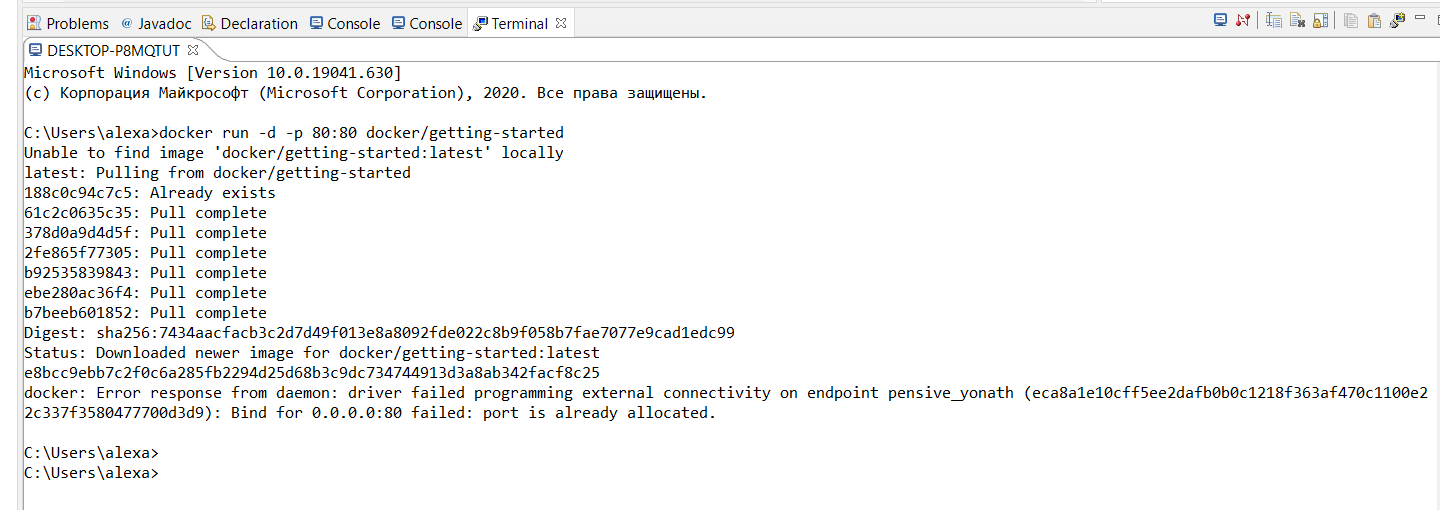


*Command for running:*

**

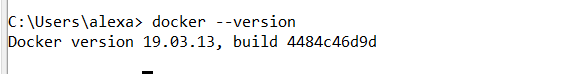
***Images et conteneurs:***

*Fondamentalement, un conteneur n'est rien d'autre qu'un processus en cours d'exécution, avec quelques fonctionnalités d'encapsulation supplémentaires qui lui sont appliquées afin de le maintenir isolé de l'hôte et des autres conteneurs. L'un des aspects les plus importants de l'isolation des conteneurs est que chaque conteneur interagit avec son propre système de fichiers privé; ce système de fichiers est fourni par une image Docker . Une image comprend tout ce qui est nécessaire pour exécuter une application - le code ou le binaire, les environnements d'exécution, les dépendances et tout autre objet du système de fichiers requis.*

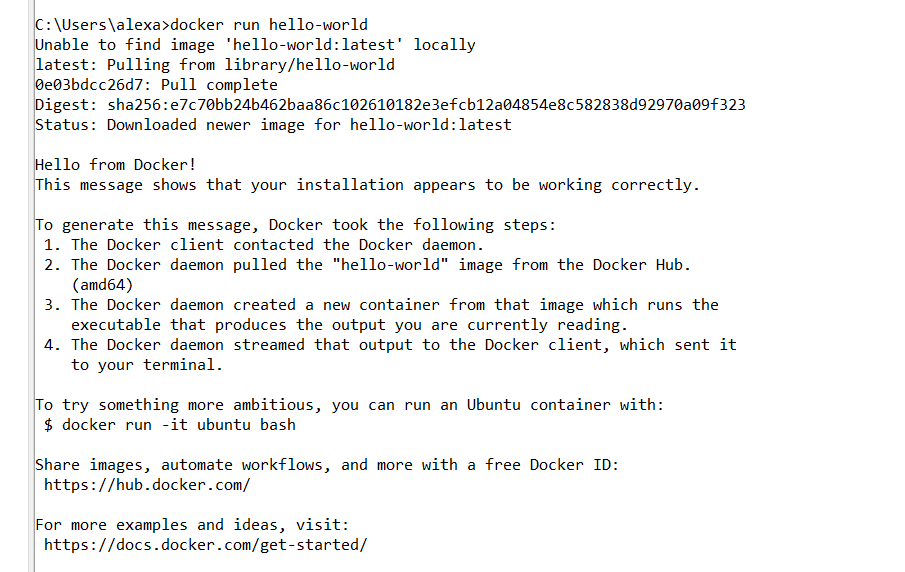


***Tester la version de Docker:***

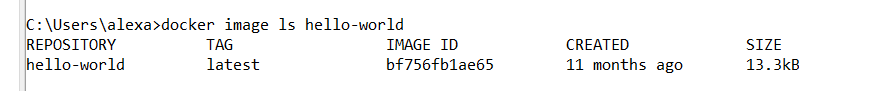
*Une fois que vous avez correctement installé Docker Desktop, ouvrez un terminal et exécutez* ***docker –version*** *pour vérifier la version de Docker installée sur votre machine.*



***Tester l'installation de Docker***



***Exécutez docker image lspour lister l' hello-worldimage que vous avez téléchargée sur votre machine.***

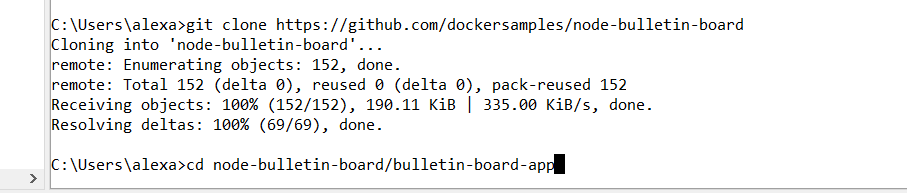


***Conclusion***

*À ce stade, nous avez installé Docker Desktop sur votre machine de développement et exécuté un test rapide pour vous assurer que vous êtes configuré pour créer et exécuter votre première application conteneurisée.*

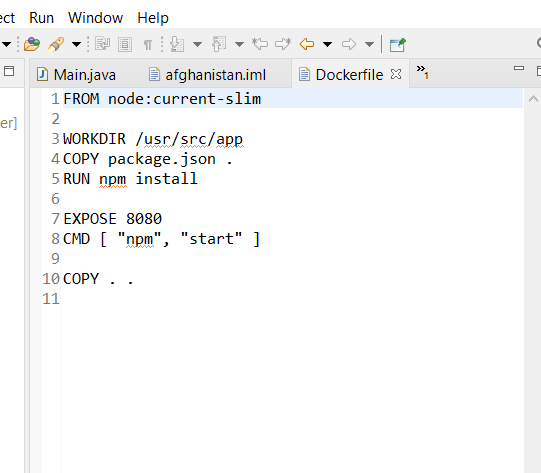
***Git***

*J’ai utilisez Git, j’ai cloner l'exemple de projet à partir de GitHub:*



***Construisez et testez votre image:***

***Maintenant que nous avez du code source et un Dockerfile, il est temps de créer notre première image et de nous assurer que les conteneurs lancés à partir de celle-ci fonctionnent comme prévu.***

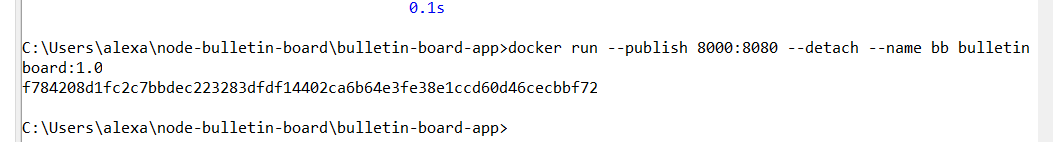


*Assurez-nous que nous êtes dans le répertoire node-bulletin-board/bulletin-board-app dans un terminal ou PowerShell à l'aide de la cdcommande. Exécutez la commande suivante pour créer notre image de tableau d'affichage:*



***Exécutez notre image en tant que conteneur:***

*Exécutez la commande suivante pour démarrer un conteneur basé sur notre nouvelle image:*

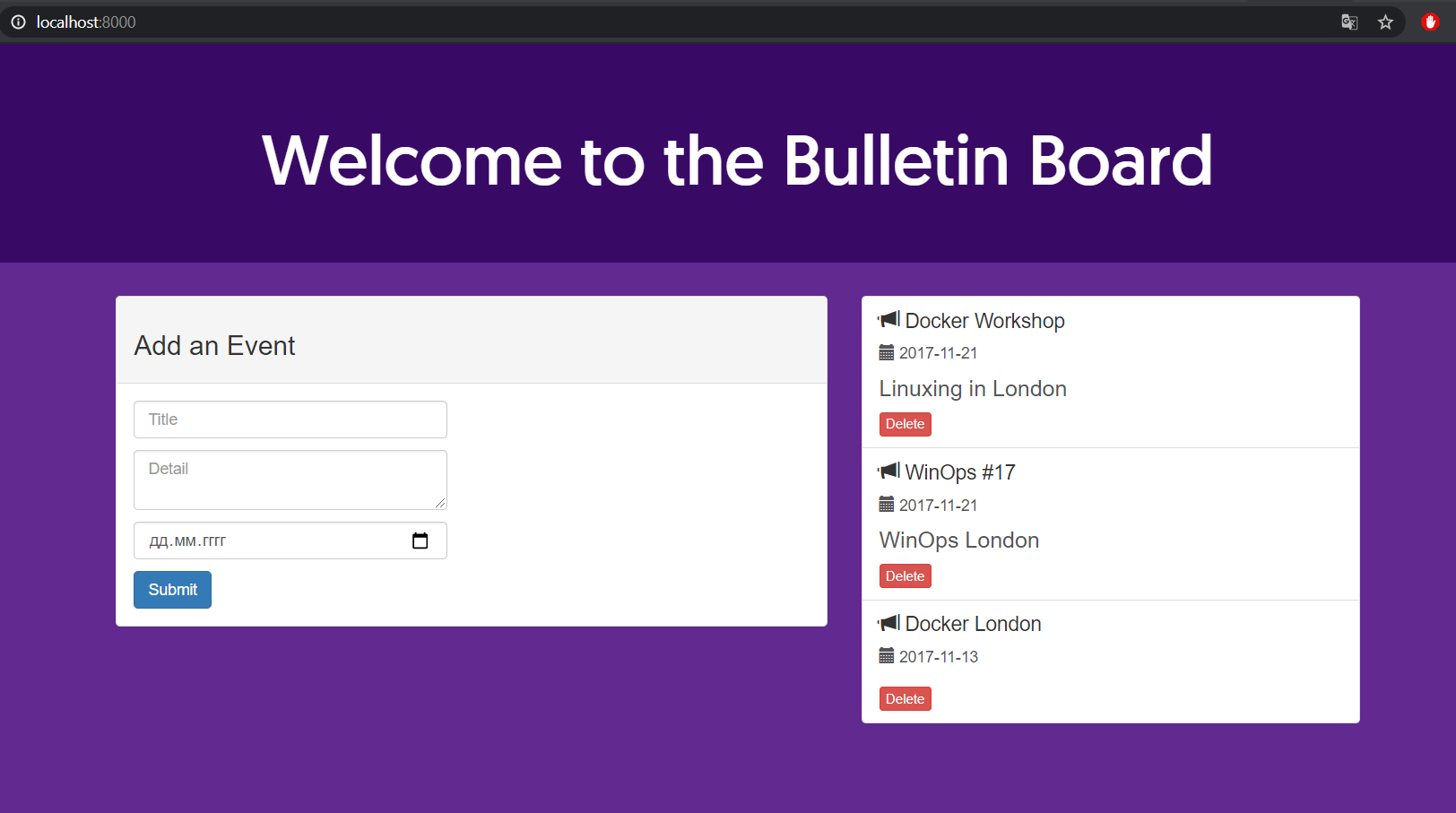


***--publish demande*** *à Docker de transférer le trafic entrant sur le port 8000 de l'hôte vers le port 8080 du conteneur. Les conteneurs ont leur propre ensemble de ports privés, donc si vous voulez en atteindre un depuis le réseau, vous devez y transférer le trafic de cette manière. Sinon, les règles de pare-feu empêcheront tout le trafic réseau d'atteindre votre conteneur, comme posture de sécurité par défaut.*

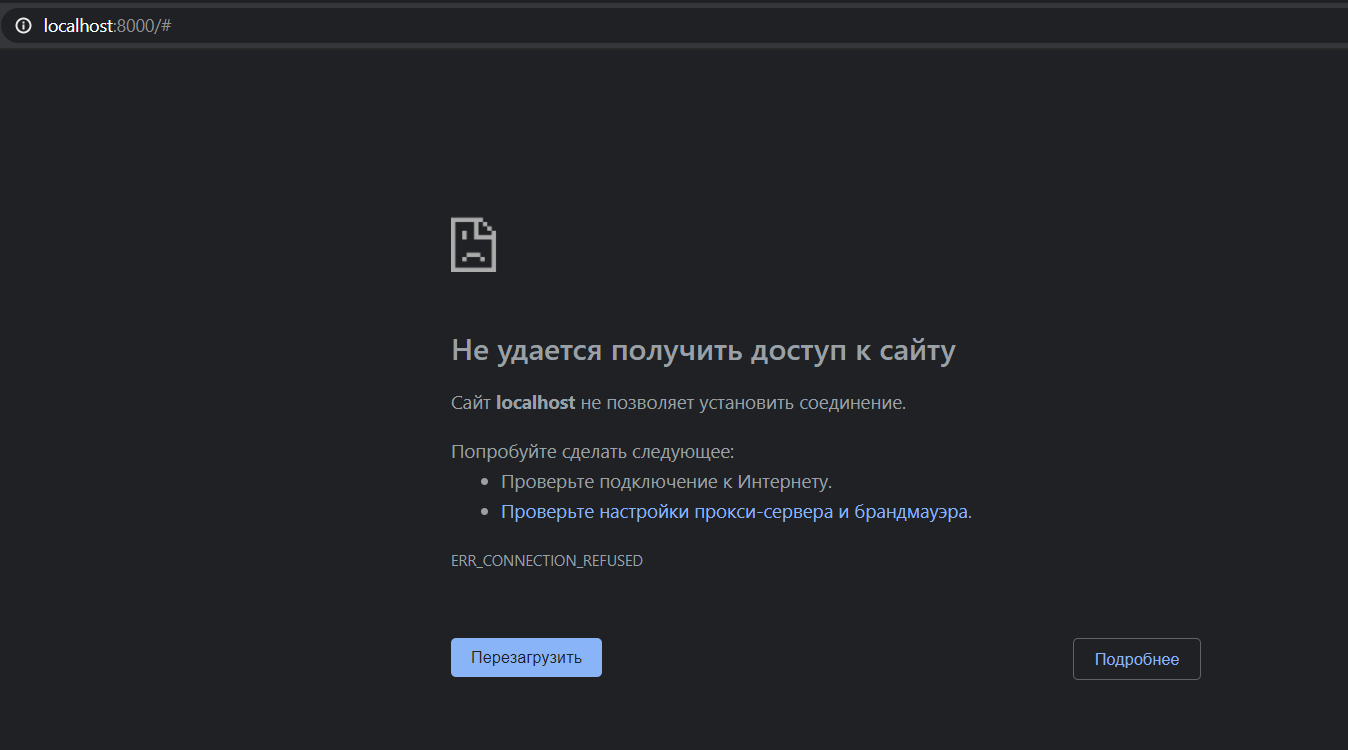
***--detach demande*** *à Docker d'exécuter ce conteneur en arrière-plan.*

***--namespécifie*** *un nom avec lequel vous pouvez faire référence à votre conteneur dans les commandes suivantes, dans ce cas* ***bb****.*

***Visitez notre application dans un navigateur à localhost:8000***



**L' –force** option arrête un conteneur en cours d'exécution, afin qu'il puisse être supprimé. Si vous arrêtez d'abord le conteneur en cours d'exécution docker stop bb, vous n'avez pas besoin de l'utiliser **–force** pour le supprimer.



Conclusion:

Dans ce laboratoire, j'ai travaillé avec Doker. J'ai installé l'environnement de travail à installer (au moins 2) et configuré pour assurer le procelus de développement d'un produit logiciel basé sur Java. J'ai installé les conteneurs nécessaires et décrit le processus par étapes. J'ai décrit comment utiliser chaque environnement installé.