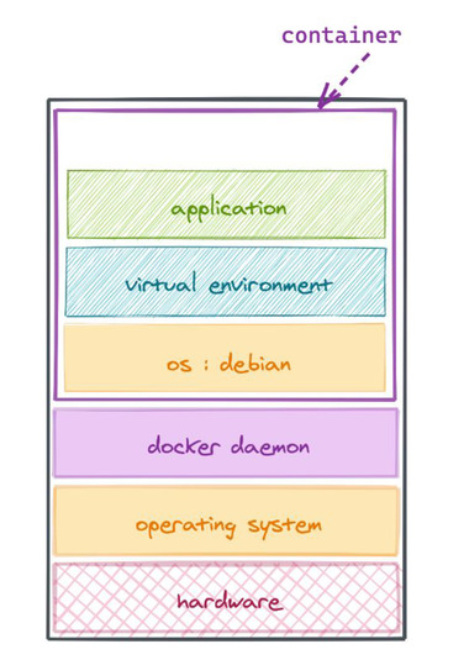
Liz BLANCA

Introduction conteneurisation

# Qu’est-ce que la conteneurisation ?



La conteneurisation est le nom donné au processus consistant à regrouper les fichiers de configurations, les bibliothèques, les dépendances et le code dans une « boîte » appelé conteneur afin de pouvoir déployer facilement et rapidement des applications sans tenir compte des différents environnements existant.

Le conteneur est avant tout un concept : on cherche à isoler l’exécution d’ensembles de processus de la machine hôte. A noter que le noyau et l’accès au matériel reste partagé (à la différence d’une VM).

Traditionnellement, pour exécuter n'importe quelle application sur votre ordinateur, vous deviez installer la version correspondant au système d'exploitation de votre machine. Par exemple, vous deviez installer la version Windows d'un *progiciel\** sur un ordinateur Windows. Toutefois, avec la conteneurisation, vous pouvez créer un progiciel unique, ou [conteneur](https://aws.amazon.com/containers), qui s'exécute sur tous les types d'appareils et de systèmes d'exploitation.

* [Vidéo explicative rapide](https://www.youtube.com/watch?v=tR1jfiEs8v4)

*\* Ensemble de logiciels munis d'une documentation, conçus pour répondre à des besoins spécifiques et permettre une utilisation autonome.*

## Prenons un exemple

|  |
| --- |
| Imaginons que vous devez expédier un meuble complexe à un ami vivant dans une autre ville. Ce meuble est composé de plusieurs pièces et son assemblage nécessite des outils spécifiques ainsi que des instructions précises. Si vous envoyez chaque pièce séparément, vous courez le risque que certaines parties soient perdues en chemin ou que votre ami ne dispose pas des outils nécessaires pour l’assemblage.  Pour éviter cela, vous décidez de tout rassembler dans une seule grande boîte. Celle-ci contient les pièces du meuble, les outils requis, les instructions détaillées, ainsi que des étiquettes indiquant l'ordre d'assemblage. Vous veillez à ce que chaque élément soit correctement emballé afin de prévenir tout mouvement durant le transport. Ainsi, une fois la boîte reçue, votre ami peut assembler le meuble sans difficulté, car tout le nécessaire est inclus et prêt à l'emploi.  Dans le domaine du logiciel, cette "boîte" représente un conteneur. Le meuble correspond à l'application, tandis que les outils et instructions symbolisent les dépendances et configurations essentielles au bon fonctionnement de l'application. |

## Quels sont les avantages des conteneurs ?

Lors du déploiement d’une même application sur différent environnement, des problèmes peuvent intervenir régulièrement. Ces problèmes surviennent généralement de différences dans la configuration requise des dépendances.

Lors de la création d’une application comprenant un grand nombre de bibliothèques et de fichiers de configurations, tout est assemblé dans un seul et unique paquet, un conteneur. Ce conteneur peut ensuite être déplacé et exécutée sur n’importe quel système sans se soucier de la comptabilité du logiciel et les conflits entre différents version.

On décompte trois avantages majeur à la conteneurisation :

### Efficacité

Contrairement aux machines virtuelles, qui nécessitent chacune leur propre système d'exploitation complet, les conteneurs partagent le même noyau du système d'exploitation hôte. Cela les rend plus légers et plus rapides, car ils consomment moins de ressources. Par exemple, si l’on fait tourner cinq conteneurs sur une machine, ils n'auront pas besoin de cinq systèmes d'exploitation distincts, ce qui permet d'économiser de la mémoire et du temps de traitement.

### Portabilité

Les conteneurs offrent un format standardisé pour l’empaquetage et permettent la portabilité de logiciels sur plusieurs environnements sans nécessité de réécrire le code du programme. Il est par exemple possible de créer une application une seule fois et la déployer sur plusieurs systèmes d'exploitation.

### Agilité

Chaque conteneur est isolé des autres, ce qui signifie qu’une application dans un conteneur ne peut pas interférer avec une autre application.  Les développeurs de logiciels peuvent résoudre les problèmes et modifier le code de l'application sans interférer avec le système d'exploitation, le matériel ou d'autres services applicatifs. Ils peuvent raccourcir les cycles de publication des logiciels et travailler rapidement sur les mises à jour avec le modèle de conteneur.

# Exemples concrets d’utilisation en entreprise (Developpement)

Imaginons que suite à mon intégration au sein de la DSI du territoire, j'ai été chargé de développer une application de type « To Do List » en C#, en utilisant le Framework .NET, à déployer sur l'ensemble des postes d'un service. Pour éviter les contraintes liées à l'installation manuelle des dépendances .NET sur chaque machine, ce qui pourrait entraîner des erreurs de configuration et un important gaspillage de temps, j'ai proposé l'utilisation de la conteneurisation. Cette approche permet de simplifier le déploiement et d'assurer un fonctionnement homogène et fiable de l'application sur toutes les machines. Pour cela je vais utiliser un VM Virtual Box sous la distribution Atomic Linux, en tirant parti des outils Distribox, et Podman tout en gérant mon code source via GitHub.

#### Premier Cas

|  |
| --- |
| Pour cette exemple nous utiliserons une machine virtuel ‘’Atomic Host’’, une distribution Linux spécialement conçue pour les conteneurs.    Débutons par le démarrage de la VM, de son terminal de commande ainsi que BoxBuddy, une solution d’interface graphique à Distribox. Mettons à jour notre VM via la commande ci-dessous.      La commande suivante permet de connaître la distribution Linux utilisé :    L’on crée ensuite notre conteneur avec Distribox. On lui donne le nom « todo-list-container » et l’on choisit une image Linux légère, ici l’on prend la dernière version d’une image Ubuntu.    Ensuite les instructions, en entrant la commande *« distribox enter todo-list-container* ». L’icône de boite en carton indique que l’on se situe bien à l’intérieur du conteneur.    Nous allons maintenant installe Git :    Puis cloner mon dépôt github sur mon conteneur :    Le Conteneur apparaît alors sur BoxBuddy et dans l’explorateur de fichier.      Nous pouvons maintenant accéder au répertoire cloné avec la commande « cd Introduction\_Container » pour par la suite ajouter d’autres éléments et les configurer si nécessaire.    J’ai ensuite installé WineZGUI (via l’interface du store) afin de pouvoir lancé l’exécutable de mon fichier « to-do-list »    On l’exécute via WineZGUI et l’application se lance !    L’application est bien installé sur ma machine grâce au conteneur et prête à être utilisé ! |

#### Second Cas

Imaginons maintenant le besoin d’installer un environnement spécifique de développement pour le service Informatique. Nous allons créer un conteneur avec Visual studio code pour l’exemple.

|  |
| --- |
| Actuellement sur une CM Atomic, Distrobox est déjà installé. Dans le cas contraire la commande  « ***sudo dnf install distrobox podman –y*** » sera à taper.  Commençons par créer notre conteneur :    L’on a utilisé une image ubuntu mais il est possible de prendre une archlinux ou encore une fedora (--image fedora :38)  L’on suit ensuite les instructions en entrant dans le conteneur :      Le conteneur apparaît bien sur le Boxbuddy  Distrobox initialise les paquets nécessaires puis le logo du carton apparaît, tout est bon.  On va maintenant mettre à jour les paquets avant d’installer n’importe quel programme :    On installe maintenant VSCode et tous les composants nécessaires avec l’aide d’un fichier .deb :    On vérifie si l’installation à réussi en vérifiant la version :    Puis l’on lance l’application :      Visual Studio Code est bien installé sur notre conteneur !    Et il est visible sur BoxBuddy et peut être ajouter au menu. |

#### Troisième Cas

Imaginons maintenant le besoin pour une entreprise d’avoir un serveur Apache afin d’héberger son site internet. En utilisant la conteneurisation l’on réduit le risque de conflit entre d’autres applications et les versions de logiciels et que le site web fonctionne de la même manière sur tous les environnements.

|  |
| --- |
| On début par créer le conteneur et se déplacer à l’intérieur :      Il apparaît bien sur BoxBuddy.  On met à jour les paquets et l’on installe Apache    Initialement j’avais plusieurs erreurs, j’ai ainsi du changer le port du serveur Apache passant du port 80 au port 8080 via les commandes :      Mes changements :      Puis l’on démarre le serveur :    On recherche sur le navigateur via l’URL : <http://localhost:8080> (port attribué au serveur)    Et l’on arrive sur la Default Page d’Apache ! Tout fonctionne parfaitement. |

# D’autres exemples concrets d’utilisation en entreprise (Developpement)

* Mise en place site web et sa Base de Donnée
* Mise en place d’un serveur Apache
* Test Automatisé (dans des environnements jetables pour assurer des conditions de tests propres)
* Création d’environnement de Démonstration
* Gestion de Sécurité (ajout de couche de sécurité et chercher des failles sans incidents)
* Collaboration de développement en partageant un environnement identique
* Déploiement en masse d’une application
* Déploiement d’un environnement de travail (développement) propre à chaque services/personnes.

## Pour Supprimer un conteneur

|  |
| --- |
| On débute par sortir du conteneur si l’on s’y trouve :    L’émoticône du carton aura normalement disparu.  On tape la commande Distribox list afin d’afficher tous les conteneurs existant :    On tape la commande « distrobox rm [nom du conteneur] –force » pour supprimer le conteneur    En faisant un distrobox list le conteneur aura disparu ! |

|  |
| --- |
|  |

#### Seconde Méthode via DOCKER

* Suivre ce [Tutoriel](https://app.lewagon.school/courses/take/intro-to-docker/presentations/45836379-containerization)

## Sources :

* <https://www.youtube.com/watch?v=tR1jfiEs8v4>
* <https://azure.microsoft.com/fr-fr/resources/cloud-computing-dictionary/what-is-a-container#:~:text=Les%20conteneurs%20permettent%20aux%20d%C3%A9veloppeurs%20de%20partager%20facilement%20les%20logiciels,les%20cycles%20de%20d%C3%A9veloppement%20logiciel>.
* <https://cyberinstitut.fr/distrobox-installer-distributions-linux-container/>
* <https://aws.amazon.com/fr/what-is/containerization/#:~:text=La%20conteneurisation%20est%20un%20processus,sur%20n'importe%20quelle%20infrastructure>.
* <https://indico.in2p3.fr/event/17124/contributions/61042/attachments/48554/61402/20180604-IN2P3-intro-conteneurs.pdf>
* <https://app.lewagon.school/courses/take/intro-to-docker/presentations/45836379-containerization>
* Cours de Monsieur Labani (Lycée Condorcet)