**Formation Docker Hyper-V  
 - sys 30 – Cours & T.p.**

Michel Cabaré / www.cabare.net / michel@cabare.net

Docker Hyper-V   
 - sys 30 - Cours et Travaux Pratiques V1-00 - Janvier 2021

|  |  |
| --- | --- |
| https://WWW.CABARE.NET ©  Une image contenant objet  Description générée automatiquement |  |
|  | |
| Certification qualité délivrée au titre de la catégorie d’action suivante  actions de formation | |

**table des matiÈres**

[Docker - Containers 3](#_Toc60730946)

[Docker et les containers, linux : 3](#_Toc60730947)

[Portage sous windows : 4](#_Toc60730948)

[Spécificités windows 2016 : 4](#_Toc60730949)

# Docker - Containers

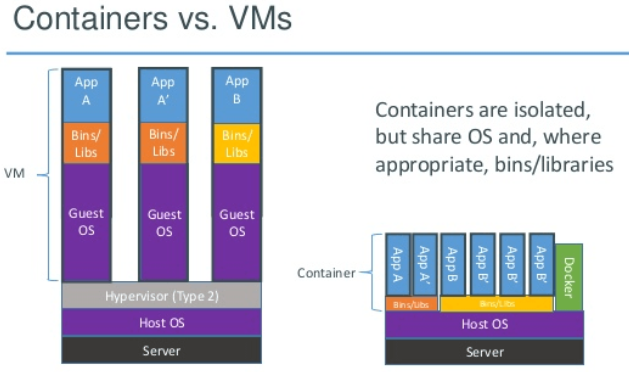
## Docker et les containers, linux :

**Docker** est une solution de virtualisation applicative originellement sous linux, autrement dit une solution qui isole l’application (avec ses dépendances) du système d’exploitation de sorte qu’elle puisse être aisément transportée d’une machine à une autre avec toutes ses configurations.

Son déploiement peut être rapide et sans surprise, et l’on peut exécuter plusieurs instances de cette application sans qu’elles n’entrent en conflit entre elles.

Cette approche normalise la mise à disposition de programmes, car elle permet d'exécuter les applications dans n'importe quel environnement tant physique que virtuel de la même manière

**Docker** permet donc de créer des environnements, appelées **containers** de manière à isoler des applications. Plus besoin de **Guest OS**, de **système invité,** un **hôte** suffit ! mais… cela fonctionne originellement uniquement **linux** !



donc il n’est plus nécessaire d’embarquer un OS complet pour chaque application virtualisée. Les applications s’exécutent directement sur l’OS de la machine physique, accédant directement aux pilotes et accélérations matérielles.

Cet Os se trouve "partagé" en quelque sorte par les applications.

Il en résulte d’importantes économies mémoires et des performances accrues comparées à une virtualisation matérielle classique, donc une encore meilleure exploitation des machines physiques

## Portage sous windows :

Dans un premier temps, on a vu naître **« boot2docker »**, un package spécial de VirtualBox embarquant une version allégée de Linux (Tiny Core Linux) destinée à simplement exécuter des conteneurs Docker

Dans un second temps, on a vu apparaître des fonctionnalités **Azure** permettant d’installer un **Host Docker** dans une **VM Linux Azure** en quelques clics simples depuis le portail du Cloud Microsoft

Actuellement : le **Docker Runtime** sous **Windows Server 2016** et **Hyper-V**

Jusqu’ici, toutes les solutions restaient fondées sur une exécution des « conteneurs » sous Linux. dans Windows Server 2016, il est possible de créer un « Docker Container » autour d’une application Windows et d’exécuter ce conteneur sous une machine Windows Server 2016.

**N.B**: Il faut bien comprendre que **Windows Server 2016** n’exécutera pas des **conteneurs Linux** au format **Docker** mais des **conteneurs Windows** au **format Docker**.

L’inverse est aussi vrai, **Linux** ne pourra jamais executer des conteneurs **Windows**. Ces limites se comprennent bien puisque **les conteneurs Docker s’appuient sur l’OS** et ne virtualisent que le système de fichiers, les configurations et les zones d’état du système. (comme la base de registres sous Windows par exemple).

Toutefois, et parce que l’implémentation Windows s’appuie sur les mêmes API, l’administration des conteneurs Linux et Windows sera strictement identique et leur fonctionnement très similaire.

Pour une entreprise, Les conteneurs **Windows** pourront être publiés dans **l’Index Docker** (le repository public d’images) et les **Docker Registrys** (les repository privés) en concomitance avec **des conteneurs Linux**

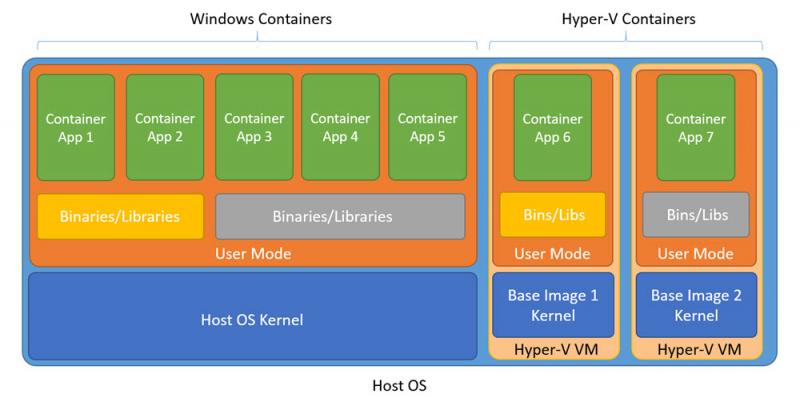
## Spécificités windows 2016 :

Les utilisateurs pourront, en plus des machines virtuelles classiques VM Hyper-V , déployer deux types de conteneurs applicatifs :

* les **Windows Server Containers**
* les **Hyper-V Containers**

le **Windows Server Containers** ne surprendra pas les habitués de Docker, car il partage le noyau de son serveur hôte (dans notre cas windows)

les **Hyper-V Containers** sont un peu différents. Contrairement à un conteneur traditionnel, chaque conteneur Hyper-V embarquera son propre noyau Windows. Une approche qui permet d’isoler davantage l’application exécutée dans le conteneur, afin de s’assurer que celle-ci n’interfère pas avec l’hôte ou avec les autres conteneurs.



Le fonctionnement d’un conteneur Hyper-V est donc très proche d’une machine virtuelle, à la différence que celui-ci peut être géré avec les solutions Docker