

29/08/2026



IT-CONNECT

Actualités IT & Tech, tutoriels en administration
système, réseau et cybersécurité.



IT-Connect » Cours - Tutoriels » Virtualisation, Conteneurs & DevOps » Proxmox VE » Proxmox VE : comment créer des conteneurs Docker ?



PROXMOX VE

Proxmox VE : comment créer des conteneurs Docker ?

📅 12/03/2026 👤 Florian BURNEL 💬 0 commentaire 💎 Docker, Proxmox, Virtualisation

Exécuter des conteneurs Docker sur un environnement Proxmox VE, c'est une action que beaucoup d'entre vous cherchent à réaliser. Quelles sont les options possibles pour utiliser Docker sur Proxmox VE ? Quels sont les avantages et les inconvénients de ces méthodes ?

Voici l'essentiel à savoir à ce sujet.

Pour approfondir le sujet des conteneurs sur un environnement Proxmox VE, je vous invite à lire mes deux précédents articles à propos des conteneurs LXC :

- [Proxmox VE : bien débuter avec les conteneurs LXC](#)
- [Proxmox VE : exécuter des images OCI \(Docker\) nativement dans LXC](#)

Sommaire [-]

L'option à éviter : Docker sur l'hôte Proxmox VE
Option n°1 : installer Docker dans une machine virtuelle
Option n°2 : exécuter Docker dans un conteneur LXC
La gestion au quotidien avec Dockhand (ou Portainer)
Docker sur Proxmox VE : quelle méthode choisir ?
Conclusion

L'option à éviter : Docker sur l'hôte Proxmox VE

Commençons par ce qu'il est préférable d'éviter : l'installation de Docker directement sur l'hôte Proxmox VE en lui-même ! Cela revient à positionner Docker au même niveau que Proxmox VE, à savoir directement sur le système d'exploitation de l'hôte (Debian).

Je vous déconseille d'adopter cette méthode, je trouve que ce n'est pas propre. Docker doit venir

se positionner sur une couche supérieure vis-à-vis de l'hyperviseur. De plus, Proxmox VE modifie Debian pour en faire un hyperviseur, ce qui pourrait donc créer des conflits avec Docker.

Par exemple, Docker modifie la configuration réseau de la machine sur laquelle il est installé. Cela pourrait donc interférer avec la gestion du réseau opérée par Proxmox VE.

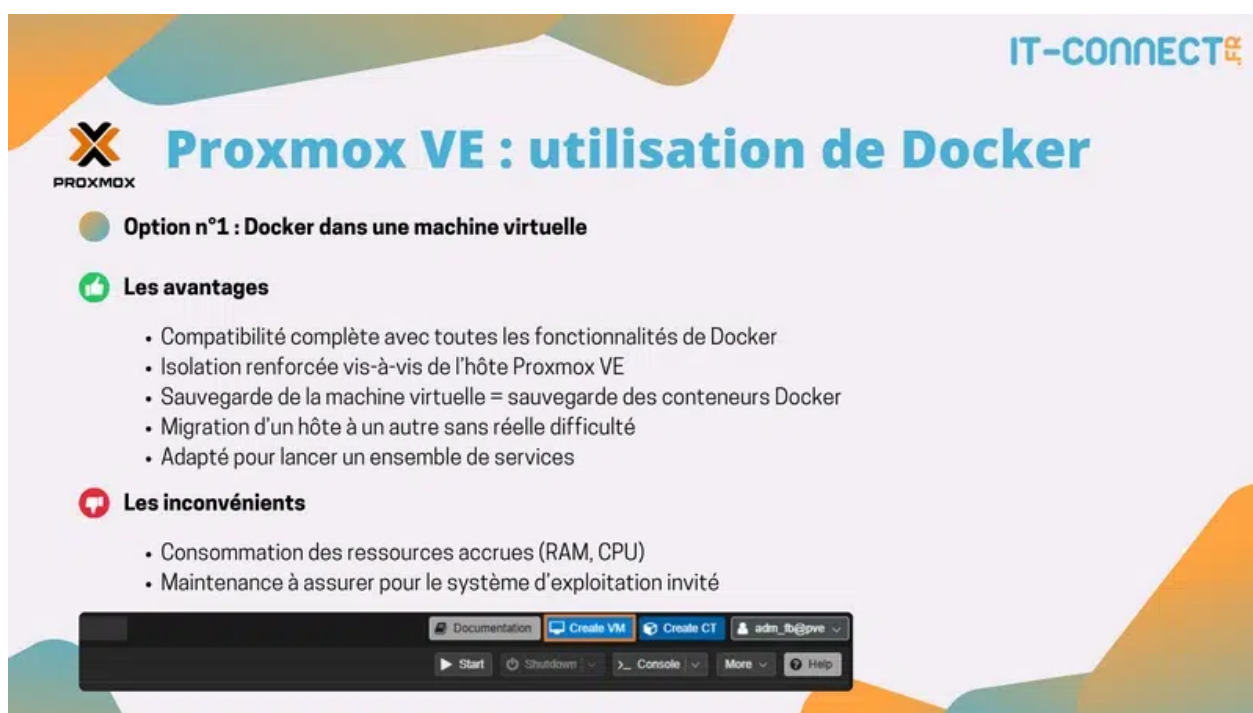
Option n°1 : installer Docker dans une machine virtuelle

Exécuter Docker au sein même d'une machine virtuelle Proxmox VE, c'est l'option la plus robuste et qui assure une compatibilité complète avec l'ensemble des fonctionnalités de Docker. L'idée est la suivante : vous installez une machine virtuelle, sous Debian ou Ubuntu, par exemple, et vous installez les paquets Docker (et Docker Compose).

La machine virtuelle est isolée vis-à-vis de l'hôte Proxmox VE et chaque partie dispose de son propre noyau Linux. Sur cette machine virtuelle, vous pouvez installer tous les conteneurs dont vous avez besoin.

L'inconvénient de cette méthode, c'est qu'elle implique une augmentation de la consommation des ressources : plus de RAM, plus de processeur, car au-delà des conteneurs, il faut des ressources pour exécuter l'OS invité. C'est une différence notable vis-à-vis des conteneurs LXC réputés pour leur légèreté.

Vous devez aussi assurer la maintenance du système d'exploitation invité (celui de la VM), comme pour n'importe quelle machine virtuelle.



IT-CONNECT

Proxmox VE : utilisation de Docker

Option n°1 : Docker dans une machine virtuelle

Les avantages

- Compatibilité complète avec toutes les fonctionnalités de Docker
- Isolation renforcée vis-à-vis de l'hôte Proxmox VE
- Sauvegarde de la machine virtuelle = sauvegarde des conteneurs Docker
- Migration d'un hôte à un autre sans réelle difficulté
- Adapté pour lancer un ensemble de services

Les inconvénients

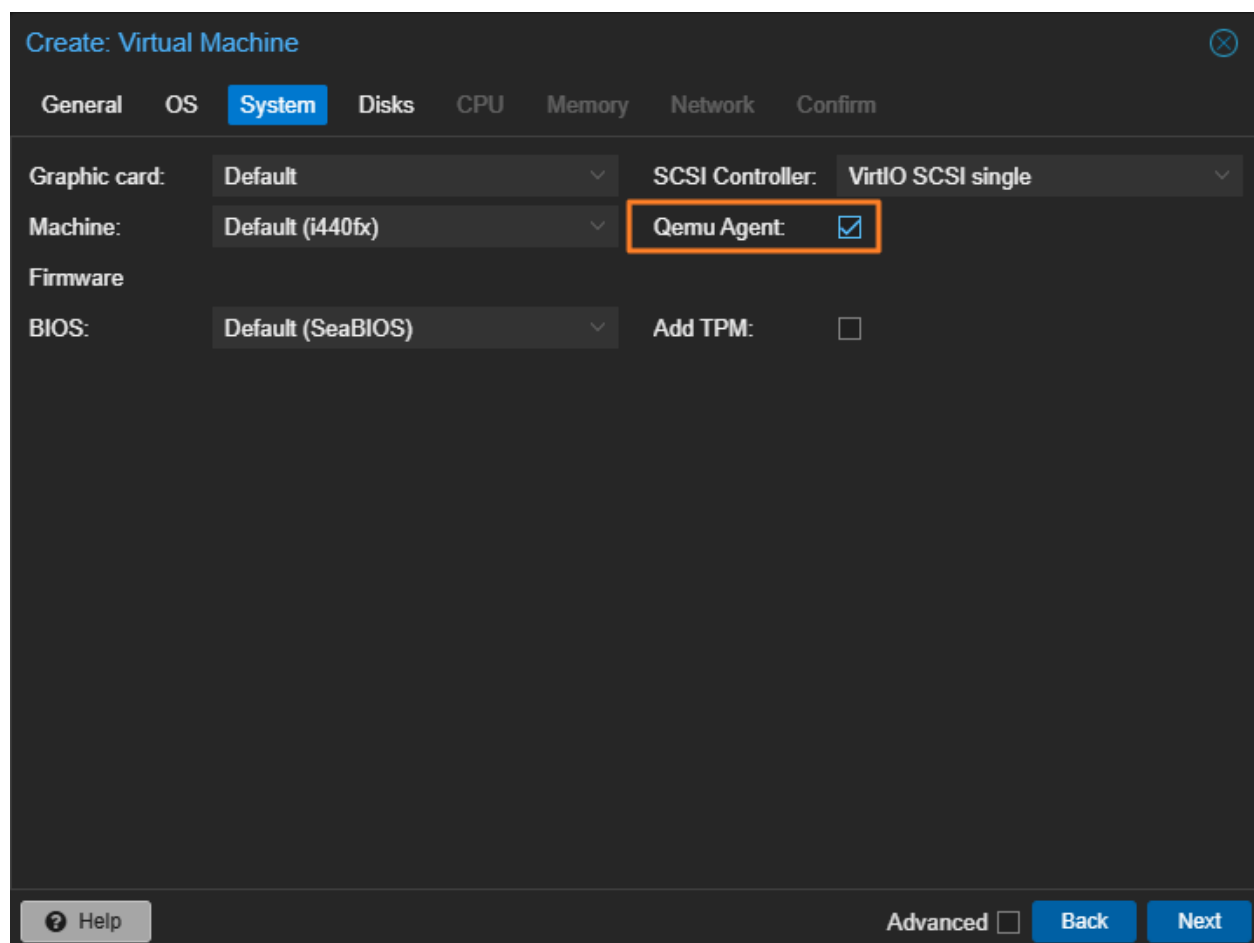
- Consommation des ressources accrues (RAM, CPU)
- Maintenance à assurer pour le système d'exploitation invité

Documentation Create VM Create CT admin@pve Start Shutdown Console More Help

Pour la production, c'est l'approche que je préfère, et elle ressemble aussi à ce que l'on peut faire avec toutes les plateformes de virtualisation. Cette machine virtuelle avec Docker peut être facilement sauvegardée, restaurée, voire même migrée vers un autre nœud Proxmox VE.

Lors de la création de cette machine virtuelle, veillez à :

- Activer l'agent QEMU, tout en sachant qu'il sera installé par défaut si vous utilisez une image Debian 13
- Choisissez un adaptateur réseau de type VirtIO
- Bien que ce soit facultatif, vous pouvez ajouter un second disque qui sera dédié au stockage des données des conteneurs Docker. Vous pouvez le monter dans `/opt` , ce qui permet ensuite de stocker vos données de projets sous `/opt/docker-compose/` .



The screenshot shows the 'Create: Virtual Machine' window in Proxmox VE, with the 'System' tab selected. The 'Qemu Agent' checkbox is checked and highlighted with an orange box. Other settings visible include 'Graphic card: Default', 'Machine: Default (i440fx)', 'SCSI Controller: VirtIO SCSI single', 'Firmware: BIOS: Default (SeaBIOS)', and 'Add TPM: [unchecked]'. The bottom of the window has a 'Help' button, an 'Advanced' checkbox, and 'Back' and 'Next' buttons.

Une fois la machine créée et le système d'exploitation installé, effectuez l'installation de Docker. Vous pouvez utiliser la méthode "manuelle" via les dépôts officiels ou exécuter le script d'installation. Pour la première méthode, consultez mon tutoriel d'[installation de Docker sur Debian](#). Pour la méthode basée sur le script, exécutez simplement ceci :

```
apt update && install curl
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
```

Voilà, vous n'avez plus qu'à créer vos premiers conteneurs (mais lisez la suite de cet article avant cela).

Option n°2 : exécuter Docker dans un conteneur LXC

Déployer Docker à l'intérieur d'un conteneur LXC sur Proxmox VE (via le mécanisme d'imbrication ou nesting) est une seconde option. C'est une pratique courante associée à un avantage majeur : elle consomme beaucoup moins de ressources qu'une machine virtuelle (VM).

Le problème, c'est que certaines applications conteneurisées exécutées via Docker ne fonctionneront pas si ce dernier est exécuté sur un conteneur LXC. L'explication est la suivante : contrairement à une VM, un conteneur LXC utilise le noyau Linux de l'hôte Proxmox. Les conteneurs Docker à l'intérieur du LXC partagent donc ce même noyau, et les restrictions sur les conteneurs LXC peuvent ne pas plaire à toutes les applications...

Pour les applications légères et ce qui s'apparente à des applications Web, cela fonctionnera bien. Je vous donne quelques exemples : [Doozzle](#), [Vaultwarden](#), [IT-Tools](#), [Traefik](#), ou encore [Homer](#). À l'inverse, il y a des chances d'avoir des problèmes et des limitations (ou besoin de tuning spécifique) avec les applications qui ont besoin de faire un lien avec le matériel, au niveau réseau via WireGuard, par exemple. On peut citer aussi Home Assistant.



Pour contourner certaines limitations, vous pourriez être amené à configurer un conteneur LXC en mode privilégié. Néanmoins, il vaut mieux éviter, car cela signifie que l'utilisateur root dans le conteneur a des permissions qui s'étendent au niveau de l'hôte Proxmox VE. Pas top niveau sécurité.

En cas de difficultés, privilégiez l'utilisation d'une machine virtuelle au sein de laquelle vous installez Docker. Si vous optez pour un conteneur LXC avec Docker, la méthode propre est de rester en **Unprivileged** et d'activer les fonctionnalités **Nesting** et **Keyctl**.

Dans un précédent article, je vous ai expliqué comment créer un conteneur LXC en ligne de commande via la commande `pct`. Nous pouvons utiliser cette technique pour créer un conteneur LXC correctement configuré pour Docker. Il est important d'activer les options `keyctl=1` et `nesting=1` (comme précisé ci-dessus).

```
CTID=306
CTTEMPLATE="local:vztmpl/debian-13-standard_13.1-2_amd64.tar.zst"
CTHOSTNAME="lxc-docker"

pct create $CTID $CTTEMPLATE \
  --hostname $CTHOSTNAME \
  --unprivileged 1 \
  --cores 2 --memory 1024 --swap 512 \
  --net0 name=eth0,bridge=vbr0,firewall=1,ip=dhcp,tag=10,type=veth \
  --rootfs local-lvm:8 \
  --features keyctl=1,nesting=1 \
  --password VotrePassword
```

Cette commande crée un conteneur avec 2 cœurs de CPU, 1 Go de RAM et un disque de 8 Go. C'est largement suffisant pour faire tourner certaines applications conteneurisées. Vous pourrez alors créer d'autres conteneurs LXC sur le même principe, ou avec un peu plus de ressources, selon les besoins.

Une fois le conteneur déployé, installez Docker à l'intérieur, sur le même principe que pour la machine virtuelle.

La gestion au quotidien avec Dockhand (ou Portainer)

Proxmox VE n'intègre pas d'outils natifs pour gérer les conteneurs Docker. C'est normal, ce n'est pas son rôle. De ce fait, vous pouvez déployer une solution comme Dockhand ou [Portainer](#) pour administrer vos conteneurs au quotidien. Ayant une préférence pour le premier cité, voyons comment l'installer rapidement.

Ce qui suit doit être effectué sur la machine virtuelle Docker ou le conteneur LXC Docker.

Commencez par créer un dossier nommé `dockhand` sous `/opt/docker-compose`, afin de stocker les données de ce projet. Au sein de ce répertoire dédié au projet Dockhand, il conviendra de créer également un sous-dossier nommé `data`. Il servira à stocker les données de l'application

de façon persistante (dont la base de données SQLite).

```
sudo mkdir -p /opt/docker-compose/dockhand/data
```

Dans le dossier `/opt/docker-compose/dockhand/`, créez le fichier `docker-compose.yml`.

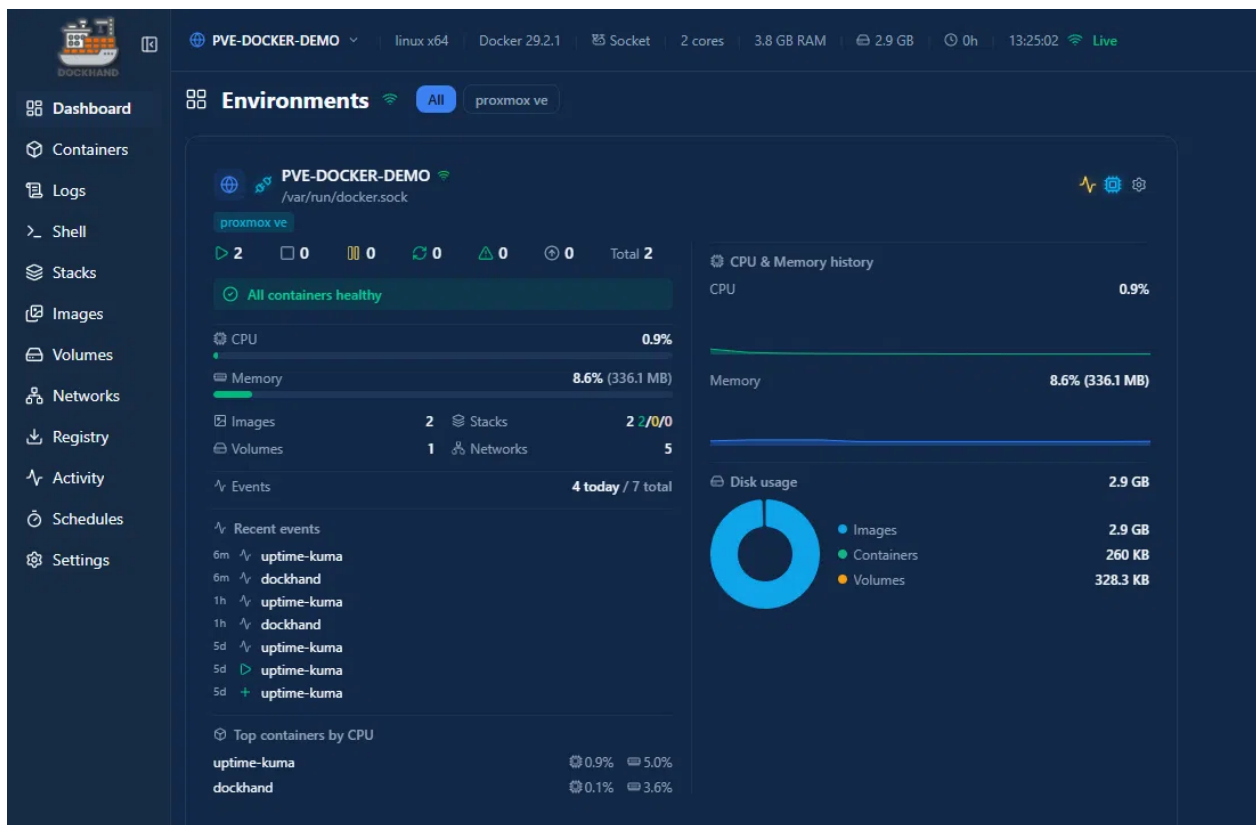
Ci-dessous, le code à insérer dans votre fichier Docker Compose. Par défaut, Dockhand s'appuie sur une base SQLite. Si vous avez besoin de meilleures performances, vous pouvez configurer le conteneur pour utiliser une base PostgreSQL.

```
services:
  dockhand:
    image: fnsys/dockhand:latest
    container_name: dockhand
    restart: unless-stopped
    ports:
      - "3000:3000"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - ./data:/app/data
```

Lancez la construction du projet :

```
docker compose up -d
```

Vous pourrez alors vous connecter à votre instance Dockhand via l'adresse IP de la VM/du conteneur où Docker est installé. Précisez le port `3000` puisque Dockhand est accessible sur ce port par défaut. L'interface de Dockhand se présente à vous, vous n'aurez qu'à ajouter l'environnement Docker local pour commencer à l'administrer.



Pour apprendre à utiliser Dockhand, découvrez mon guide complet :

- [Tutoriel - Prise en main de Dockhand pour l'administration de Docker](#)

Ainsi, vous disposez de l'interface de Proxmox VE pour administrer vos machines virtuelles, les conteneurs LXC et la plateforme virtualisée en elle-même, puis de Dockhand pour l'administration des environnements Docker. Une seule instance Dockhand peut se connecter à X moteurs Docker.

Docker sur Proxmox VE : quelle méthode choisir ?

Si vous avez les ressources (RAM/CPU), privilégiez la VM. C'est la méthode "zéro souci", mais elle ne permet pas d'optimiser l'utilisation des ressources. Sinon, vous pouvez mixer entre une machine virtuelle et des conteneurs LXC avec Docker pour vous adapter aux besoins des applications à déployer.

Autrement dit, vous pouvez utiliser des conteneurs LXC (plus légers, plus rapides) autant que possible, tout en ayant une machine virtuelle pour certains projets plus exigeants et incompatibles avec cette approche.

On peut aussi voir la situation de cette façon :

- Machine virtuelle à privilégier en entreprise pour la stabilité et la sécurité (isolation plus forte),
- Conteneur LXC à privilégier pour un Homelab ou s'il y a énormément de conteneurs, pour optimiser l'utilisation des ressources, en particulier sur du matériel modeste.

Conclusion

Vous pouvez utiliser l'une de ces deux méthodes ou mixer les deux pour exécuter des conteneurs Docker sur Proxmox VE. Vous pouvez aussi regarder du côté des images OCI (utilisées aussi par Docker), même si cette approche manque encore de maturité à l'heure actuelle (Proxmox VE 9.1) et n'a pas encore la même souplesse que de passer directement par Docker.

Si vous avez l'habitude d'exécuter des applications conteneurisées via Docker sur Proxmox VE, n'hésitez pas à partager votre retour d'expérience en commentaire. Cela sera enrichissant pour les autres lecteurs de cet article.

- [Documentation - Proxmox VE](#)



Ajouter IT-Connect à mes
sources préférées



Florian BURNEL | Co-founder of IT-Connect

Cofondateur d'IT-Connect et Microsoft MVP "Cloud and Datacenter Management". Mon obsession depuis près de 15 ans ? Rendre l'administration système et la cybersécurité accessibles, que vous soyez junior ou confirmé. Plus qu'un métier, l'IT est pour moi une véritable passion. J'accompagne au quotidien les sysadmins et les professionnels de l'IT dans leur montée en compétences et leur veille technique. [Voir la bio complète](#) >

Open Source

Windows Server

PowerShell



Partagez cet article





PATCH DE MARS 2026 POUR VEEA
M BACKUP & REPLICATION : 5 FAI
LLES CORRIGÉES, DONT 3 CRITIQU
UES

NOUVEAUTÉ LET'S ENCRYPT : CER
TBOT GÉNÈRE ENFIN DES CERTIFI
CATS TLS POUR VOS ADRESSES IP
!



👍 Vous pourrez aussi aimer



Proxmox VE : bien débiter avec les
conteneurs LXC

📅 09/03/2026 💬 0



Proxmox VE : enrôler les cer
Secure Boot Microsoft 2023

📅 05/08/2026 💬 0

Laisser un commentaire

Votre adresse e-mail ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Commentaire *

Nom *

E-mail *

Site web

☐

Enregistrer mon nom, mon e-mail et mon site dans le navigateur pour mon prochain commentaire.

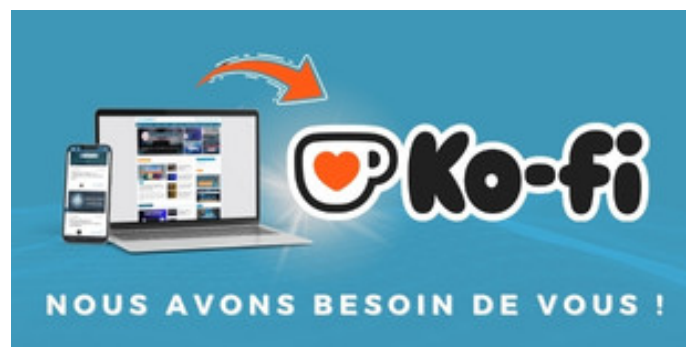
Laisser un commentaire

Ce site utilise Akismet pour réduire les indésirables. [En savoir plus sur la façon dont les données de vos commentaires sont traitées.](#)

 Newsletter : 1 e-mail par semaine !

Votre adresse e-mail*

S'abonner



Guides & Tutos



SELF-HOSTING & OPEN SOURCE

SERVEUR WEB

GoAccess : analyser les logs de votre serveur web en temps réel

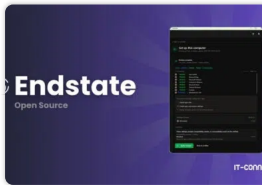
 25/08/2026  Florian BURNEL  0 commentaire

Installez GoAccess pour analyser les logs de vos serveurs web Nginx, Apache ou Traefik : tableau de bord terminal, rapport HTML temps réel et compatible Docker.



NIS 2 : pourquoi la connexion à Windows reste l'angle mort de vos accès

📅 25/08/2026 💬 2 commentaires



Endstate : sauvegarder et restaurer ses applications Windows (et leurs configs) sur un nouveau PC

📅 21/08/2026 💬 3 commentaires

★ Veille IT & Bons Plans

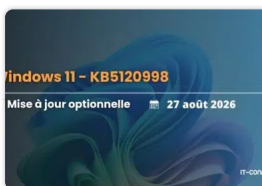


ENTREPRISE

Windows Autopilot : Microsoft inscrit l'appartenance de vos PC directement dans l'UEFI

📅 28/08/2026 👤 Florian BURNEL 💬 0 commentaire

Nouveauté Windows Autopilot : Device association lie un PC Windows 11 à votre tenant via un marqueur UEFI attesté par le TPM, avant l'inscription dans Intune.



Windows 11 KB5120998 : la barre des tâches redevient déplaçable, mais attention aux bugs

📅 28/08/2026 💬 0 commentaire



GPUThor : l'attaque Rowhammer qui casse l'ECC des GPU NVIDIA

28/08/2026 0 commentaire



Offres de la rentrée GoDeal24 : Office 2024 Pro Plus à 19.99€ et Windows 11 Pro à 12.25€

28/08/2026 8 commentaires



Next.js : deux failles critiques permettent une exécution de code à distance sans authentification

28/08/2026 0 commentaire



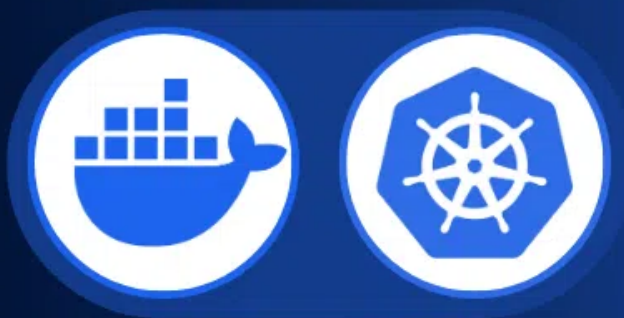
Windows 11 : Microsoft a décidé de bloquer le pilote qui fait planter les jeux

28/08/2026 0 commentaire

 Nos derniers cours gratuits

Docker et Kubernetes

De l'initiation à la mise en production



IT-CONNECT

COURS EN LIGNE

16 chapitres

★★★★★

Docker et Kubernetes : de l'initiation à la mise en production



Jérémy GAK

Terraform pour Microsoft Azure

Infrastructure as Code

**IT-CONNECT****COURS EN LIGNE**

18 chapitres

★★★★★

L'Infrastructure as Code avec Terraform pour Microsoft Azure



Luc BRETON

Vous cherchez quelque chose ?

Recherche



Découvrir IT-Connect

A propos

Comment contribuer sur IT-Connect ?

Contact

Espace annonceurs

L'Équipe

Offres d'emploi

Politique de confidentialité

Puis-je réutiliser le contenu publié sur IT-Connect ?

Soutenir IT-Connect

Espace personnel

Se connecter

Créer un compte

Flux des publications

Changer de langue



Recommandations

Blogmotion

Tech2tech

Générateur de mots de passe

Powered by
 OVHcloud

IT-Connect - Copyright © 2026 | Creative Commons License BY-NC 4.0