

# Liens serveur utiles

- [Chaine de Tonton Jo](#)
- [PROMOTHEUS & GRAFANA : Tutoriels et formations](#)
- [NETDATA & MONITORING : Tutoriels](#)
- [LOKI : Tutoriels](#)
- [Différence entre Serveur d'application et Serveur web](#)
- [Qu'est ce qu'un serveur web ?](#)
- [Formation - Comment configurer efficacement un serveur web ?](#)
- [Recevoir des notifications push grâce à Gotify](#)
- [NetBird - Créez un réseau privé sécurisé hyper facilement](#)
- [DevSecOps](#)
- [Step-By-Step Procedure To Install SSL/TLS Certificate On Nginx Web Server](#)
- [How to Receive and Install an SSL Certificate](#)
- [Monter son serveur by Belginux](#)
- [Yves Rougy - Mini infra PME](#)
- [DevOps - Stéphane ROBERT](#)
- [MTN, c'est moi qui commande by Belginux](#)
- [Backup n' Co by Belginux](#)
- [Doc lagout](#)

# Chaine de Tonton Jo

<https://www.youtube.com/c/TontonJo/featured>

# PROMOTHEUS & GRAFANA :

## Tutoriels et formations

[https://www.youtube.com/playlist?list=PLn6POgpklwWo3\\_2pj5Jflqwla62P5OI8n](https://www.youtube.com/playlist?list=PLn6POgpklwWo3_2pj5Jflqwla62P5OI8n)

# NETDATA & MONITORING :

## Tutoriels

[https://lnkd.in/dNkPTqu?trk=public\\_post\\_share-update\\_update-text](https://lnkd.in/dNkPTqu?trk=public_post_share-update_update-text)

# LOKI : Tutoriels

[https://lnkd.in/dyavkUi?trk=public\\_post\\_share-update\\_update-text](https://lnkd.in/dyavkUi?trk=public_post_share-update_update-text)

# Différence entre Serveur d'application et Serveur web

<https://waytolearnx.com/2018/09/difference-entre-serveur-dapplication-et-serveur-web.html>

---

La différence entre un **serveur d'applications** et un **serveur Web**, est la suivante: Un **serveur Web** traite exclusivement les requêtes HTTP, tandis qu'un **serveur d'applications** sert la logique métier aux programmes d'application via un nombre illimité de protocoles. Un **serveur Web** est un programme qui accepte les demandes d'informations et envoie les documents requis. Un **serveur d'applications** peut être un programme ou un ordinateur exécutant un programme dans un réseau distribué.

## Différence entre Serveur d'application et Serveur web

### Définition d'un serveur web

Un **serveur Web** gère le protocole HTTP. Lorsque le **serveur Web** reçoit une requête HTTP, il répond par une réponse HTTP, telle que renvoyer une page HTML. Pour traiter une requête, un **serveur Web** peut répondre par une image ou une page HTML statique, envoyer une redirection ou déléguer la génération de réponse dynamique à d'autres programmes tels que des scripts CGI, JSP (Pages JavaServer), servlets, ASP (Active Server Pages). ), des JavaScripts côté **serveur** ou une autre technologie côté **serveur**. Quel que soit leur but, ces programmes côté **serveur** génèrent une réponse, le plus souvent en HTML, à afficher dans un navigateur Web.

Bien qu'un **serveur Web** ne prenne pas en charge les transactions ou la gestion de connexion à une base de données, il peut utiliser diverses stratégies de tolérance aux pannes et d'évolutivité, telles que l'équilibrage de charge, la mise en cache et la mise en cluster.

### Définition d'un serveur d'application

En ce qui concerne le **serveur d'applications**, selon notre définition, un **serveur d'applications** expose la logique métier aux applications client via divers protocoles, y compris éventuellement HTTP. Alors qu'un **serveur Web** traite principalement de l'envoi de code HTML pour l'affichage dans un navigateur Web, un **serveur d'applications** permet d'accéder à la logique métier à utiliser par les programmes d'application client.

Dans la plupart des cas, le **serveur** expose cette logique métier via une API, telle que le composant EJB (Enterprise JavaBean) présent sur les **serveurs d'applications J2EE** (Java 2 Platform, Enterprise Edition). En plus, le serveur d'applications gère ses propres ressources. Ces tâches de sécurité comprennent la sécurité, le traitement des transactions, la mise en commun des ressources et la messagerie. Tout comme un serveur Web, un **serveur d'applications** peut également utiliser diverses techniques d'évolutivité et de tolérance aux pannes.

Différences clés entre un serveur Web et un serveur d'applications

- Les **serveurs Web** sont souhaitables pour le contenu statique alors que les **serveurs d'applications** sont appropriés pour le contenu dynamique.
- Les **serveurs Web** prennent en charge des langages de script tels que Perl, PHP, ASP, JSP, etc. Par contre, le **serveur d'application** assiste les langages de script ainsi que les services d'application tels que les transactions ou la gestion de connexion à la base de données
- Le **serveur d'applications** contient des conteneurs Web, EJB et un **serveur Web** en tant que partie intégrée. En revanche, un **serveur Web** contient uniquement un conteneur Web ou de servlet et peut utiliser un EJB.
- Le **serveur Web** ne prend pas en charge le multithreading, tandis que le **serveur d'applications** assiste les transactions multithreading et distribuées.
- Le **serveur Web** utilise le protocole HTML et HTTP. D'ailleurs, le **serveur d'applications** pourrait utiliser une interface utilisateur graphique et des protocoles tels que RPC / RMI, y compris HTTP.
- La limite de charge ou la capacité est supérieure dans le cas du **serveur d'applications** par rapport au **serveur Web**.
- Le **serveur Web** fournit un environnement pour exécuter une application Web et des fonctionnalités telles que la mise en cache et l'évolutivité. Par contre, le **serveur d'applications** fournit un environnement pour exécuter des applications Web avec des applications d'entreprise.

Conclusion

Un **serveur d'applications** peut contenir un **serveur Web**, ce qui signifie qu'il peut effectuer les opérations effectuées par un **serveur Web**, ainsi que des éléments et des fonctionnalités supplémentaires pour faciliter le regroupement des connexions, le regroupement d'objets, le

support des transactions, etc.

# Qu'est ce qu'un serveur web ?

<https://blog.lws-hosting.com/divers/serveur-web>

---

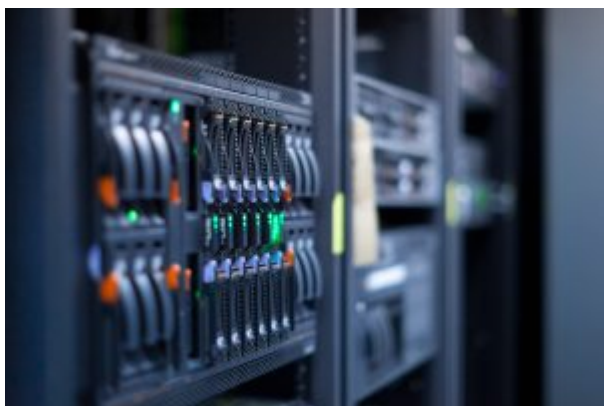
Vous êtes un peu perdu dans toutes ces appellations qui sont **serveur web**, **hébergement web**, **site web**, ... Nous allons voir dans cet article les différences de chacun et comment choisir suivant votre besoin.

Pour mieux se situer, on entend par serveur web tout type de serveur qui permet la **diffusion d'un contenu** que ce soit sur internet ou en intranet. Son but va être de diffuser des informations aux personnes lui en faisant la requête (clients). Les serveurs web sont apparus dans les années 90 grâce à Tim Berners-Lee qui est considéré comme le fondateur de **World Wide Web**.

## Comment fonctionnent les serveurs Web ?

Un **serveur web** est composé d'un logiciel permettant de communiquer avec le **protocole HTTP**. Les plus connus sont **Apache et Nginx**. Ces logiciels interprètent les requêtes HTTP des clients, les traitent en exécutant les lignes de codes du site (PHP, ...) et renvoient le contenu demandé sous forme HTML. Vous l'aurez compris, lorsque vous êtes sur un navigateur et que vous souhaitez accéder à un site web, votre navigateur envoie une demande au serveur "j'aimerais afficher cette page" et le serveur vous la renvoie en HTML/CSS/JS. Ensuite, votre navigateur interprète la réponse et vous affiche la page demandée.

Evidemment, un serveur web peut traiter plusieurs demandes simultanément suivant ses performances et c'est pour cela que le choix d'un serveur web est important afin d'éviter la surcharge et de pénaliser vos clients.



# Hébergement Web ou serveur Web ?

La distinction n'est pas toujours facile à comprendre surtout pour les novices. Nous allons donc voir cela un peu plus en détails.

Un **hébergement web** est un ensemble de services avec des quotas alloués (notamment l'espace de stockage) sur un serveur web. Vous avez donc en général **plusieurs hébergements** sur un seul et même **serveur web**. Vous l'aurez donc compris, avec un hébergement web, vous serez plus limité qu'avec un serveur web cependant ce n'est pas toujours rentable ni utile de prendre un serveur web.

Un **serveur web** est utile dans certains cas :

- Vous avez un **trafic important** et vous avez donc besoin de **plus de ressources** qu'un hébergement web ne peut vous procurer.
- Vous avez besoin d'**installer des librairies / modules php spécifiques** ou vous avez besoin d'un accès root à votre serveur.

Cependant, si vous avez un petit site vitrine pour votre entreprise avec peu de visites par exemple, il sera préférable de rester sur un hébergement web afin de ne pas vous compliquer la vie. En effet, un serveur web est un peu plus complexe à configurer qu'un hébergement web lambda. Mais il existe des logiciels permettant de simplifier la mise en place et la gestion d'un serveur web.

## Gérer un serveur Web

serveur-web

Pour **gérer un serveur web**, vous pouvez le faire en utilisant uniquement des lignes de commande mais cela est très complexe et surtout réservé aux utilisateurs avancés.

Pour éviter cela, il existe de nombreux logiciels de gestion permettant de gérer son serveur web et de le paramétrer simplement. Les plus connus sont **WHM/cPanel** mais qui est assez onéreux ainsi que **ISPConfig** qui lui est un logiciel de gestion gratuit. Voici un comparatif de ces deux [panneaux de gestion](#).

L'utilisation d'un logiciel de ce type vous permet de gérer votre serveur simplement avec une interface web. Vous pouvez configurer vos sites, vos emails, vos comptes ftp, vos modules php, etc très simplement et sans une ligne de commande.

# Trouver un serveur Web adapté

Si vous êtes novice dans l'administration de serveur web, nous vous conseillons plutôt de rester sur un [hébergement web standard](#) cependant si vous souhaitez vous lancer dans un **serveur VPS** avec un panneau de gestion **ISPConfig**, nous pouvons vous proposer ces [offres VPS](#).

Pour bien choisir votre serveur VPS, vous devez vérifier que le serveur web en question supporte bien le langage de programmation que votre site utilise. La plupart des sites actuels utilisent PHP. Vous pouvez héberger un site de ce type sur un système d'exploitation Linux grâce à Apache ou Nginx. Toutes nos [offres serveurs web](#) proposent des distributions linux.

Ensuite, vous devez vérifier que les **ressources** sont en adéquation avec vos **pré-requis techniques**. Un logiciel comme Prestashop ou WordPress n'a pas des pré-requis importants mais suivant votre trafic et vos plugins/modules, il vous faudra les augmenter avec notamment plus de **RAM/CPU**.

Concernant l'**espace de votre disque**, nos offres proposent soit **SSD** soit **HDD**. Pour la rapidité et si vous n'avez pas beaucoup à stocker sur votre serveur, nous vous conseillons un disque SSD qui permet à votre site d'être jusqu'à **10 fois plus rapide** qu'avec un HDD. Un HDD peut être intéressant dans le cas où vous utilisez votre serveur web comme un serveur de stockage, d'emails, ... ou si vous avez de nombreuses images.

Enfin, nous vous conseillons de vous faire **accompagner par un professionnel de l'hébergement** pour choisir et configurer votre serveur VPS dans le cas où vous n'avez pas un savoir-faire suffisant. Nous proposons des infogérances afin de vous aider dans la mise en place et la configuration de votre serveur web.

# Formation – Comment configurer efficacement un serveur web ?

<https://korben.info/formation-serveur-web.html>

Merci Korben (<https://x.com/Korben>) pour l'info.

# Recevoir des notifications push grâce à Gotify

<https://twitter.com/BlablaLinux/status/17436252484302850>

08

**Merci Blabla Linux pour ton travail et ta mise à disposition pour la communauté**

**<https://blablalinux.be>**

# NetBird – Créez un réseau privé sécurisé hyper facilement

[NetBird – Créez un réseau privé sécurisé hyper facilement \(korben.info\)](#)

Merci Korben (<https://x.com/Korben>) pour l'info.

# DevSecOps

[Le Site d'un Consultant DevOps | Stéphane ROBERT](#)  
[\(stephane-robert.info\)](#)

# Step-By-Step Procedure To Install SSL/TLS Certificate On Nginx Web Server

[Step-By-Step Procedure To Install SSL/TLS Certificate On Nginx Web Server! \(linkedin.com\)](#)

THESECMASTER.COM

## STEP-BY-STEP PROCEDURE TO INSTALL SSL/TLS CERTIFICATE ON NGINX WEB SERVER!

SSL/TLS CERTIFICATE | NGINX WEBSERVER | HTTP TO HTTPS

```
server {  
    listen 80;  
    listen 443 ssl;
```

```
    root /var/www/exampledomain.com/html;  
    index index.html index.htm index.nginx.debian.html;
```

```
    server_name exampledomain.com www.exampledomain.com;  
    ssl_certificate      /etc/nginx/ssl/exampledomain.com/exampledomain.com.crt;  
    ssl_certificate_key  /etc/nginx/ssl/exampledomain.com/exampledomain.com.key;  
    ssl_protocols       TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;  
    ssl_ciphers          TLS-CHACHA20-POLY1305-SHA256:TLS-AES-256-GCM-SHA384:TLS-AES-128-GCM-SHA256:HIGH:!aNULL:!MD5;
```

```
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

NGINX

OpenSSL  
Cryptography and SSL/TLS Toolkit

# How to Receive and Install an SSL Certificate

[How to Receive and Install an SSL Certificate | Support | No-IP Knowledge Base \(noip.com\)](#)

The screenshot shows the No-IP web interface for managing SSL certificates. On the left is a dark sidebar with navigation links: Dashboard, Dynamic DNS, My Services (highlighted with a red box), Managed DNS, DNS Records, Domain Registration, Email, Server Monitoring, Backup DNS, SSL Certificates (highlighted with a red box), Account, and Support Center. At the bottom of the sidebar is an orange button labeled 'Add Priority Support'. The main content area has a top header with the No-IP logo, a 'Support' dropdown, a red notification bell, a shopping cart icon with '0', and a language selector set to 'English'. Below the header, the 'SSL Certificates' section is active, showing a table with columns 'Domain', 'Expiration', and 'Status'. One entry for 'RapidSSL' is listed with a status of 'Available' and an '+ Add CSR' button. A yellow banner below the table states: 'Currently Free SSL Certificates are unavailable. No-IP is actively trying to find a replacement offering so that we can continue to help their devices.' At the bottom, a 'Help with SSL Certificates' section contains three links with gear icons: 'Generate a CSR' (Create a CSR and get started validating your SSL certificate to secure your website.), 'Set Up an SSL Certificate' (A quick overview of what to expect when setting up your SSL certificate.), and 'How to use a TXT Record for S' (Create a TXT record for SSL ver).

noip

Support

SSL Certificates

| Domain   | Expiration | Status                              |
|----------|------------|-------------------------------------|
| RapidSSL |            | Available <a href="#">+ Add CSR</a> |

Currently Free SSL Certificates are unavailable. No-IP is actively trying to find a replacement offering so that we can continue to help their devices.

Help with SSL Certificates

- [Generate a CSR](#)  
Create a CSR and get started validating your SSL certificate to secure your website.
- [Set Up an SSL Certificate](#)  
A quick overview of what to expect when setting up your SSL certificate.
- [How to use a TXT Record for S](#)  
Create a TXT record for SSL ver

Add Priority Support

# Monter son serveur by Belginux

## Table des matières

1. [La machine](#)
2. [Le système d'exploitation](#)
3. [Installer le système d'exploitation](#)
4. [Installer Docker](#)
5. [Ou installer ses applications?](#)

**Merci Belginux pour ton travail et ta mise à disposition pour la communauté - (<https://belginux.com>)**

# Yves Rougy - Mini infra PME

[Mini Infra PME - YouTube](#)

# DevOps - Stéphane ROBERT

<https://blog.stephane-robert.info/docs/>

# MTN, c'est moi qui commande by Belginux

## Table des matières

1. [Se connecter en SSH](#)
2. [Mettre à jour sa distribution](#)
3. [Se déplacer dans les dossiers](#)
4. [Créer un dossier](#)
5. [Créer un fichier](#)
6. [Créer et démarrer des conteneurs](#)
7. [Mise en pratique](#)
  1. [Se connecter au serveur en SSH](#)
  2. [Mettre à jour sa distribution](#)
  3. [Créer un dossier](#)
  4. [Se déplacer dans les dossiers](#)
  5. [Créer un fichier](#)
  6. [Créer et démarrer des conteneurs](#)
8. [Commandes supplémentaires](#)
  1. [Supprimer un fichier:](#)
  2. [Supprimer un dossier:](#)
  3. [Lister le contenu d'un dossier avec les dossiers/fichiers cachés:](#)
  4. [Renommer un fichier ou un dossier](#)
  5. [Démarrer, redémarrer, stopper un container:](#)
  6. [Afficher quels containers Docker sont actifs:](#)
  7. [Afficher tous les containers, actifs ou non.](#)

**Merci Belginux pour ton travail et ta mise à disposition pour la communauté - (**  
**<https://belginux.com>)**

# Backup n' Co by Belginux

## Table des matières

1. [Notions élémentaires pour le backup](#)
2. [Préparer son disque](#)
3. [Monter un disque dur sous Linux](#)
4. [Sauvegarder ses données avec BorgBackup](#)
5. [Sauvegarder ses données avec Restic](#)
6. [Sauvegarder ses données avec rsync](#)
7. [Sauvegarder ses données avec Proxmox Backup Server](#)
8. [Sauvegarder ses données avec Rclone](#)
9. [Envoyer ses données chiffrées en ligne sur Proxmox Backup Server et activer la 2FA](#)

**Merci Belginux pour ton travail et ta mise à disposition pour la communauté - (<https://belginux.com>)**

# Doc lagout

<https://doc.lagout.org>

Ce service propose des fichiers en tous genre. Des fichiers accessibles au travers d'un serveur web (http://) et non au travers d'un serveur FTP (ftp://). Comme son nom le laisse entendre, c'est de la bonne documentation traitant et couvrant un peu tous les domaines.

**Merci Blabla Linux pour ton travail et ta mise à disposition pour la communauté**

<https://blablalinux.be>