Installation et Configuration du Reverse Proxy NGINX + Deux Containers Dolibarr

Dernière mise à jour : 10/06/2024

Documentation Parthenos

Sommaire

1. Installation et Mise en place de l'environnement

1.1. Explication infrastructure et adressage

Pour la mise en place de deux serveurs (containers) Web hébergeant le pro logiciel Dolibarr (ayant chacun une fonction différente, ils ne sont pas liés), nous aurons besoin de créer deux containers dans notre serveur dédié au sein de l'hyperviseur Proxmox.

Notre serveur dédié en lui-même jouera le rôle de reverse Proxy par le biais de NGINX ainsi que de passerelle (le serveur dédié possède une carte réseau avec une @IP publique ainsi qu'une autre pour l'@IP et le réseau privée) pour que nos containers puissent accéder à Internet.

Les serveurs Web (containers) utiliseront également le service NGINX, en plus du gestionnaire de bases de données MariaDB, et le langage PHP pour l'hébergement et le fonctionnement de Dolibarr.

Voici leurs adressages (Les informations à mettre dans le fichier de configuration des interfaces est donné un peu plus bas) :

<table>
<tr><td>SRV-Dédié</td><td>
Interface eno0

@IP : <Celle spécifiée dans le panel OVH, section « adresse IP » de notre serveur>/24

Passerelle : <Celle spécifiée dans le panel OVH, section « passerelle » de notre serveur >
<table>
<tr><th>Name</th><th>Type</th><th>Active</th><th>Autostart</th><th>VLAN a…</th><th>Ports/Slaves</th><th>Bond Mode</th><th>CIDR</th><th>Gateway</th></tr>
<tr><td>eno0</td><td>Network Device</td><td>Yes</td><td>Yes</td><td>No</td><td></td><td></td><td>5.39.90.204/24</td><td>5.39.90.254</td></tr>
</table>
Dans cet exemple, l'@IP publique que nous a donné OVH pour notre serveur est 5.39.90.204 et sa passerelle (« Gateway » en anglais) est 5.39.90.254

Interface vmbr0

@IP : 10.0.0.1/24 (/24 en CIDR signifie 255.255.255.0 en masque de S/R)

Passerelle : NON
<table>
<tr><td>vmbr0</td><td>Linux Bridge</td><td>Yes</td><td>Yes</td><td>No</td><td>10.0.0.1/24</td></tr>
</table>
</td></tr>
<tr><td>Premier Container</td><td>
Interface eth0

@IP : 10.0.0.2/24

Passerelle : 10.0.0.1
<table>
<tr><th>ID</th><th>Name</th><th>Bridge</th><th>Firewall</th><th>VLAN Tag</th><th>MAC address</th><th>IP address</th><th>Gateway</th></tr>
<tr><td>net0</td><td>eth0</td><td>vmbr0</td><td>Yes</td><td></td><td>BC:24:11:82:…</td><td>10.0.0.2/24</td><td>10.0.0.1</td></tr>
</table>
</td></tr>
</table>

Deuxième Container	Interface eth0 @IP : 10.0.0.3/24 Passerelle : 10.0.0.1

Lors de la création de vos Containers, vous serez invité à saisir l'adresse IP + CIDR ainsi que sa passerelle. Cela vous évite de passer par le fichier « interfaces » pour la configuration.

1.2. Configuration du fichier /etc/network/interfaces

Pour mettre à jour les informations réseaux de notre serveur, rendez-vous sur le shell Proxmox du serveur dédié ou bien sur votre console SSH et ouvrer le fichier de configuration des interfaces réseaux se situant dans /etc/network/interfaces :

```
root@NGINX:~# nano /etc/network/interfaces
```

Si vous n'êtes pas connecté en tant que root, ajouter « sudo » avant toute commande

Copier le contenu suivant dans votre fichier (contenant des règles iptables pour activer le nat notamment) :

```
auto lo
inet lo inet loopback

auto eno0
iface eno0 inet static
        address 5.39.90.204/24
        gateway 5.39.90.254

auto vmbr0
iface vmbr0 inet static
        address 10.0.0.1/24
        bridge-ports none
        bridge-stp off
        bridge-fd 0

        # Laisser l'adresse MAC écrite par défaut dans votre fichier,
chaque serveur à sa propre adresse, ne recopiez pas celle présente
dans l'exemple de cette documentation !

        hwaddress 00:22:4D:83:F1:E0

iface vmbr0 inet6 static
        address 2001:41d0:8:9ccc::1/128
```

```
        gateway 2001:41d0:8:9cff:ff:ff:ff:ff


post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '10.0.0.0/24' -o eno0 -j
MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '10.0.0.0/24' -o eno0 -j
MASQUERADE

post-up iptables -t raw -I PREROUTING -i fwbr+ -j CT --zone 1
post-down iptables -t raw -D PREROUTING -i fwbr+ -j CT --zone 1
```

Une fois recopier, enregistrer le fichier, puis relancer vos deux interfaces réseaux (eno0 & vmbr0) :

```
root@NGINX:~# ifdown eno0 && ifup eno0
root@NGINX:~# ifdown vmbr0 && ifup vmbr0
```

Vous pouvez désormais vérifier votre configuration à l'aide de la commande « ip a » :

```
root@NGINX:~# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever

2: eno0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 00:22:4d:83:f1:e0 brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    inet 5.39.90.204/24 scope global eno0
       valid_lft forever preferred_lft forever
    inet6 fe80::222:4dff:fe83:f1e0/64 scope link
       valid_lft forever preferred_lft forever

3: vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 00:22:4d:83:f1:e0 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 scope global vmbr0
       valid_lft forever preferred_lft forever
    inet6 2001:41d0:8:9ccc::1/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::222:4dff:fe83:f1e0/64 scope link
       valid_lft forever preferred_lft forever
```

Si tout est OK, vous devriez avoir cette même configuration (c'est normal si vous trouvez d'avantages d'informations ou d'interfaces réseaux après le saisi de cette commande).

Comme écrit un peu plus haut en ce qui concerne les containers, vous pouvez directement spécifiaient vos @IP au moment de les créer, cela vous fera faire gagner du temps sans pour autant passer par le fichier « interfaces ».

2. Configuration du Reverse Proxy avec NGINX

2.1. Installer NGINX et Le paquet SSL/TLS

Pour que notre serveur sous NGINX puisse transmettre les requêtes clients à nos serveurs WEB (comme s'il jouait le rôle d'un intermédiaire), il nous faut installer le paquet NGINX pour pouvoir ensuite commencer la configuration de ce dernier. Mais avant tout chose, on s'assure que notre serveur et tous ses paquets soient à jour, pour cela :

```
root@NGINX:~# pveam update
root@NGINX:~# apt update && apt upgrade -y
```

Maintenant qu'on se soit assuré de cela, on peut désormais installer NGINX :

```
root@NGINX:~# apt install nginx -y
```

Une fois l'installation terminé, ouvrer votre navigateur internet et tapez l'@IP publique de votre serveur dans la barre d'URL. Si vous avez une page web qui indique « Welcome to NGINX ! », cela signifie que le service est correctement installé.

Vous pouvez à tout moment vérifier l'état de NGINX :

```
root@NGINX:~# systemctl status nginx
```

Avant de commencer la configuration de NGINX, on va installer le paquet permettant d'ajouter un certificat SSL Let's Encrypt à nos noms de domaine, ainsi que la dépendance NGINX pour que cela fonctionne :

```
root@NGINX:~# apt install certbot -y
root@NGINX:~# apt install python3-certbot-nginx -y
```

Après l'installation, il se peut que le fichier « options-ssl-nginx.conf » ne soit pas présent dans le répertoire **/etc/letsencrypt**. Si en effet, il n'est pas présent, exécuter cette commande :

```
root@NGINX:/etc/letsencrypt# wget
https://raw.githubusercontent.com/certbot/certbot/master/certbot-
nginx/certbot_nginx/_internal/tls_configs/options-ssl-nginx.conf
```

Cela va récupérer le fichier de configuration SSL pour NGINX manquant et ne vous produira pas d'erreur lors du redémarrage de ce dernier.

2.2. Configurer NGINX

On va commencer par générer le certificat SSL pour nos deux noms de domaine :

```
root@NGINX:~# certbot --nginx -d dolibarr-test1.egide-vault.fr
```

Faite de même pour le deuxième nom de domaine.

Le fichier de configuration par défaut se situe dans le répertoire « /etc/nginx/sites-available/ », pour nos besoins, nous allons créer xun nouveau fichier dans ce même répertoire :

```
root@NGINX:~# touch /etc/nginx/sites-available/<nom_hote>
```

Une fois crée, ouvrez-le et coller cette configuration :

```
root@NGINX:~# nano /etc/nginx/sites-available/<nom_hote>
```

```
server {
        listen 80;
        server_name dolibarr-test1.egide-vault.fr;

        location / {
                return 301 https://$server_name$request_uri;
        }
}

server {
        listen 443 ssl;
        server_name dolibarr-test1.egide-vault.fr;

        ssl_certificate /etc/letsencrypt/live/dolibarr-test1.egide-
vault.fr/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/dolibarr-
test1.egide-vault.fr/privkey.pem; # managed by Certbot
        include /etc/letsencrypt/options-ssl-nginx.conf; # managed by
Certbot
```

```
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by
Certbot

        location / {
        proxy_connect_timeout       600;
        proxy_send_timeout          600;
        proxy_read_timeout          600;
        send_timeout                600;
        proxy_pass http://10.0.0.2:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $remote_addr;

      }
}
```

Bien sûr, n'oubliez pas de remplacer le nom de domaine qui se trouve après « server_name » par le vôtre, faite le également pour les champs ssl_certificate ou l'on peut retrouver votre nom de domaine dans le chemin.

Les lignes proxy se terminant par « time out » permettent de laisser le temps à notre serveur de reverse proxy de lire et interpréter les requêtes échangées entre lui et le serveur Web en question lors de l'installation Web de Dolibarr sur un temps maximum de 10 minutes (600 secondes comme indiqué), pour éviter une « Erreur 504 : Gateway Timeout ».

Pour appliquer cette configuration à NGINX, il faut crée un lien symbolique :

```
root@NGINX:~# ln -s /etc/nginx/sites-available/<nom> /etc/nginx/sites-
enabled<nom>
```

Vous pouvez aussi supprimer le fichier de configuration par défaut :

```
root@NGINX:~# rm /etc/nginx/sites-enabled/default
```

Il faut maintenant reproduire la même étape avec un second fichier de configuration pour notre deuxième serveur Web tout en modifiant les informations pour que cela corresponde. Cette fois-ci, il faut le créer dans le répertoire dans **/etc/nginx/conf.d** :

> Le « .conf » à la fin du fichier est obligatoire. Autrement, ce fichier de configuration ne sera pas pris en compte par NGINX

```
root@NGINX:~# touch /etc/nginx/conf.d/<nom_hote>.conf
root@NGINX:~# nano /etc/nginx/sites-available/<nom_hote>
```

```
server {
        listen 80;
        server_name dolibarr-test2.egide-vault.fr;

        location / {
                return 301 https://$server_name$request_uri;
        }
}

server {
        listen 443 ssl;
        server_name dolibarr-test2.egide-vault.fr;

        ssl_certificate /etc/letsencrypt/live/dolibarr-test2.egide-
vault.fr/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/dolibarr-
test2.egide-vault.fr/privkey.pem; # managed by Certbot
        include /etc/letsencrypt/options-ssl-nginx.conf; # managed by
Certbot
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by
Certbot

        location / {
        proxy_connect_timeout       600;
        proxy_send_timeout          600;
        proxy_read_timeout          600;
        send_timeout                600;
        proxy_pass http://10.0.0.3:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $remote_addr;

      }
}
```

Redémarrer le service :

```
root@NGINX:~# systemctl restart nginx
```

3. Configuration des serveurs Web

3.1. Installations et configuration des prérequis au fonctionnement de Dolibarr

Pour que Dolibarr puisse être opérationnels, nos deux serveurs (containers) Web ont besoin de plusieurs services et paquets pour finalement devenir des serveurs LEMP (Linux, Engine-X (prononciation), MariaDB, PHP).

> Toutes les étapes qui suivent doivent être appliquées aux deux serveurs Web (pour chaque chapitres)

Après la création de vos containers Serveur Web dans Proxmox, assurez-vous que leurs configurations IP est correcte en vous rendant dans le fameux fichier **/etc/network/interfaces** (se référer au chapitre 1.1 pour l'adressage) :

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.0.2/24
gateway 10.0.0.1
```

Assurez-vous également qu'elles soient opérationnelles grâce à la commande « ip a » et vérifier qu'il puisse bien ping vers internet « ping 1.1.1.1, ping google.fr ».

Comme pour notre serveur, il faut mettre à jour nos paquets et notre machine :

```
root@Dolibarr-Instance1:~# apt update && apt upgrade -y
```

3.1.1. Installation et Configuration de MariaDB (Base de données)

Commençons par installer le système de notre base de données, nous utiliserons MariaDB :

```
root@Dolibarr-Instance1:~# apt install mariadb-server -y
```

Exécuter la commande pour la configuration de base de MariaDB, puis laisser vous guidez par les instructions

```
root@Dolibarr-Instance1:~# mysql_secure_installation
```

Une fois cela fait, connecter-vous à votre base de données en tant qu'utilisateur « root » et avec votre mdp

```
root@Dolibarr-Instance1:~# mysql -u root -p
```

Créer une base de données propre à Dolibarr :

```
MariaDB [(none)]> create database db_dolinstance1
```

Ensuite, nous allons attribuer à cette base de données, un utilisateur qui aura tous les privilèges sur cette dernière pour que Dolibarr puisse communiquer et réaliser des actions sans problèmes.

```
MariaDB [(none)]> CREATE USER 'doluser1'@'localhost' IDENTIFIED BY
'mdptest';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON db_dolinstance2.* TO
'doluser1'@'localhost'
```

Vous pouvez maintenant quitter l'interface SQL en tapant la commande **exit**

3.1.2. Installation de PHP et de ses extensions

Le backend de Dolibarr fonctionne avec le langage coté serveur PHP et les nombreuses fonctionnalités proposées par le pro logiciel demande à faire appel à de nombreuses extensions PHP qui doivent être installés sur notre système pour fonctionner.

Nous allons simplement les installaient à l'aide d'une seule et simple commande :

```
root@Dolibarr-Instance1:~# apt install php php-cli php-common php-curl
php-gd php-intl php-json php-mbstring php-mysql php-soap php-xml php-
xmlrpc php-zip php-fpm php-imap
```

Version de PHP utilisée dans cette documentation : **8.2**

3.1.3. Installation et Configuration de NGINX

Dans nos serveurs Web, NGINX ne va pas jouer le rôle de reverse Proxy mais bien de serveur HTTP (le contenu du site sera hébergé sur ce dernier).

```
root@Dolibarr-Instance1:~# apt install nginx -y
```

On peut faire la commande **curl** pour s'assurer que NGINX soit bien installé sur nos serveurs du réseau privé et ce depuis n'importe quelle machine.

```
root@Dolibarr-Instance1:~# curl 10.0.0.2
root@Dolibarr-Instance1:~# curl 10.0.0.3
```

Une fois que tout est bon, on va éditer le fichier de configuration par défaut de NGINX qui se trouve dans le répertoire « **/etc/nginx/sites-available/default** »

Une fois à l'intérieur, voici les éléments à modifier (vous pouvez supprimer l'entièreté de la configuration par défaut et coller celle-ci-dessous) :

1) Remplacer le chemin root par le chemin où se trouverons les fichiers HTML, PHP et de code du site de Dolibarr, en l'occurrence pour notre cas, nous le mettrons dans le répertoire suivant : **/var/www/dolibarr/htdocs**
2) Vu que Dolibarr fonctionne avec PHP, il est important de rajouter index.php à la ligne ou se trouve tous les index (mettre vers le début).
3) Ajouter tout le bloc PHP. Au niveau du **fastcgi_pass unix :[…]**, pensez à spécifier la version de PHP installé sur votre machine. Par exemple, si votre version est 8.2, la ligne correcte sera :

   `fastcgi_pass unix:/run/php/php8.2-fpm.sock;`

   `Fasctcgi_read_timeout 400` permet de laisser le temps à notre serveur Web (plus précisément PHP-FPM) de lire et d'échanger les requêtes lors de l'installation Dolibarr pour les serveurs les plus lents. En bref, la même raison que spécifié dans le chapitre 2.2 concernant l'erreur timeout 504.

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/dolibarr/htdocs;

        # Add index.php to the list if you are using PHP
        index index.html index.php index.htm index.nginx-debian.html;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                 try_files $uri $uri/ =404;
        }

       location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/run/php/php8.2-fpm.sock;
```

```
                fastcgi_read_timeout 400;
        }
}
```

Vous pouvez maintenant vérifier que la configuration est correcte :

```
root@Dolibarr-Instance1:~# nginx -t
```

Si c'est OK et Successful, vous pouvez redémarrer NGINX :

```
root@Dolibarr-Instance1:~# systemctl restart nginx
```

3.2. Installation et Configuration de Dolibarr

3.2.1. Téléchargement et Droits

Tous nos prérequis sont installés et configurés, nous pouvons maintenant passer à la partie principale. En fonction de la période à laquelle vous regardez cette documentation, la version de Dolibarr la plus récente ou « stable » à changer, veuillez donc à bien vérifier cette version-là plus récente.

Pour cela, rendez-vous sur le SourceForge du pro logiciel qui va vous répertorier les dernières versions. Vous n'aurez plus qu'à modifier le numéro de version dans l'URL présent dans le **wget** donné ci-dessous.

Avant tout chose, assurez-vous d'être dans le répertoire prévu pour le téléchargement de Dolibarr (et dans notre fichier de configuration NGINX), c'est-à-dire :

```
root@Dolibarr-Instance1:~# cd /var/www
```

Nous pouvons maintenant exécuter la commande, on va recevoir un fichier .tgz de Dolibarr au sein du répertoire « **/var/www** »

```
root@Dolibarr-Instance1:/var/www# wget
https://sourceforge.net/projects/dolibarr/files/Dolibarr%20ERP-
CRM/19.0.2/dolibarr-19.0.2.tgz
```

Une fois reçu, décompresser-le :

```
root@Dolibarr-Instance1:/var/www# tar xvf dolibarr-19.0.2.tgz
```

Vous pouvez maintenant supprimer le fichier compresser en .tgz et renommer le Dolibarr décompressé en « dolibarr ».

```
root@Dolibarr-Instance1:/var/www# rm -r dolibarr-19.0.2.tgz
root@Dolibarr-Instance1:/var/www# mv dolibarr-19.0.2 dolibarr
```

Pour des raisons de sécurité, nous allons appliquer des droits permettant la lecture seule des fichiers pour l'utilisateur sur lequel tourne le serveur.

```
root@Dolibarr-Instance1:/var/www# chmod -R 755 /var/www/dolibarr
root@Dolibarr-Instance1:/var/www# chown -R www-data:www-data
/var/www/dolibarr
```

On va ensuite devoir créer un fichier de configuration vide en donnant des droits propriétaires (lectures, écritures, etc…) à l'utilisateur du serveur Web pour pouvoir procéder à l'installation Web.

3.2.2. Configuration

Si tout à été configuré comme il faut, en tapant l'un de vos deux noms de domaines (inscrits dans les deux fichiers configurations NGINX du serveur Reverse-proxy) avec l'alias **/install** dans votre navigateur Web (exemple : dolibarr-test1.egide-vault.fr/install), vous devriez arriver sur la page d'installation de Dolibarr.

Les étapes à suivre, une fois arrivée sur la page d'installation :

1) Appuyer sur « étape suivante ».
2) Si tout est coché en vert, vous devriez pouvoir cliquer sur « démarrer ».
3) Vous n'avez pas besoin de toucher à la configuration du dessus au niveau des répertoires, les réglages par défaut nous conviennent. **Cependant, spécifier bien https et non http au niveau de l'URL racine**

   Au niveau des informations des bases de données, vous n'avez plus qu'à spécifier les informations renseignées lors de la configuration de ces dernières. En l'occurrence dans notre cas :

Nom de la base de données	**db_dolinstance1**
Identifiant	doluser1
Mot de passe	mdptest

Nous n'avons plus qu'à attendre quelques minutes pour que l'installation puisse se faire. Le temps dépend bien entendu des ressources et des performances de vos machines.

A la fin, vous serez invité à créer un utilisateur Superadmin (personne ayant tous les droits).

Et c'est ce qui conclut votre installation de Dolibarr ! Vous pouvez désormais vous rendre sur la page d'accueil de votre site pour vous connectez et commencer à personnaliser votre tableau de bord, vos informations, plug-ins, etc…