

Proxmox

Proxmox est une plateforme de **virtualisation open-source** qui permet de gérer des environnements virtuels.

- [Update package database error](#)
- [Helper-script](#)
- [Reverse proxy avec VPS frontend](#)

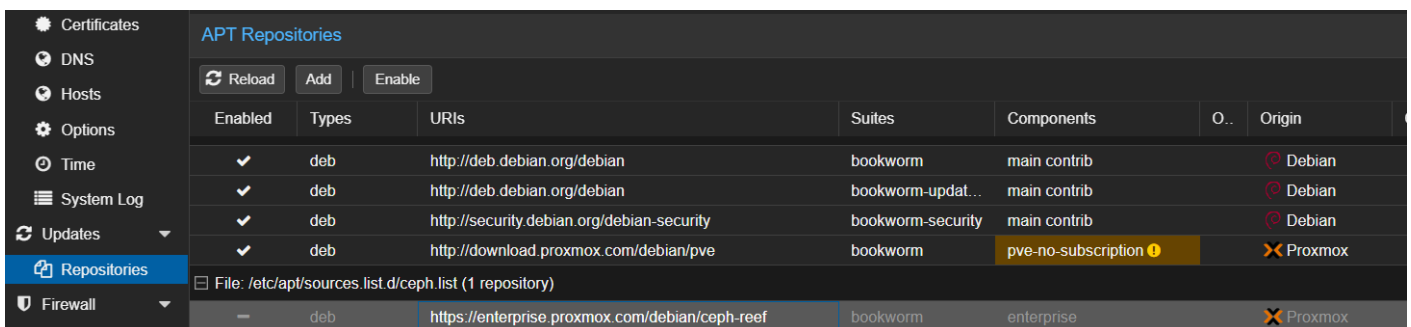
Update package database error

❏ Erreur rencontrée :

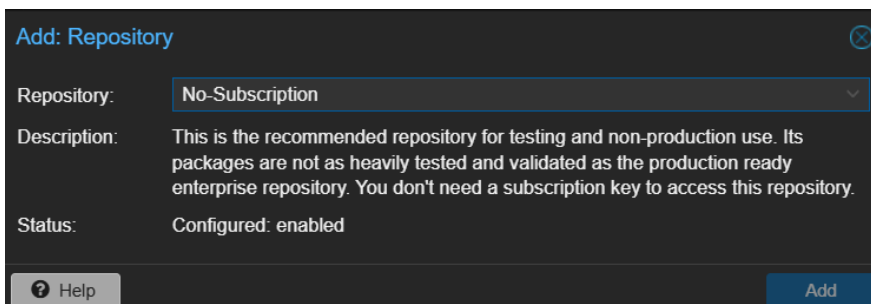
Code d'erreur : TASK ERROR: command 'apt-get update' failed: exit code 100

❏ Solution :

- Sélectionnez l'hôte concerné.
- Allez dans l'onglet **Mises à jour** puis **Dépôts**.
- Repérez le dépôt **Enterprise** et **désactivez-le**.
- Si un second dépôt **Enterprise** est présent, désactivez-le également.



Cliquez sur **Ajouter**, puis changez le dépôt en "**No Subscription**", et ajoutez-le.

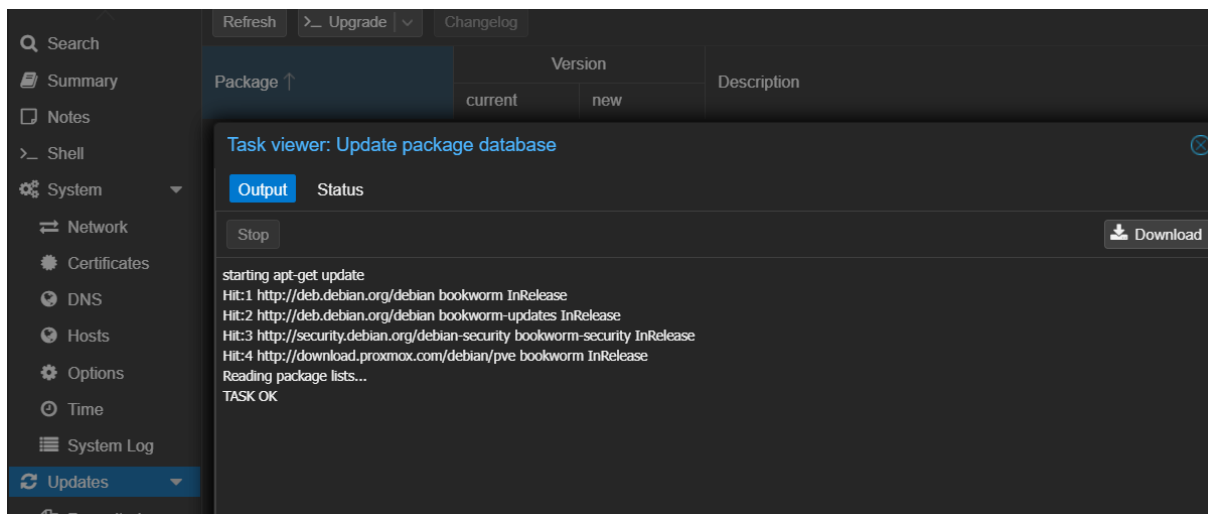


❏ Vérification :

- Vous devriez voir une **coche verte** et le message :
"You get updates for Proxmox VE".

❏ Finalisation :

- Retournez dans l'onglet **Mises à jour**.
- Cliquez sur **Actualiser**.
- La tâche devrait maintenant se compléter **sans erreur**.



Helper-script

Source : <https://community-scripts.github.io/ProxmoxVE/>

Proxmox VE Helper-Scripts :

- **Collection d'outils** : Ces scripts sont conçus pour simplifier la configuration et la gestion de l'environnement Proxmox Virtual Environment (VE).
- **Plus de 300 scripts** : Le dépôt communautaire propose une vaste gamme de scripts pour aider à gérer les machines virtuelles, les conteneurs, les clusters hautement disponibles, le stockage et les réseaux.
- **Interface conviviale** : Les utilisateurs peuvent créer des conteneurs Linux ou des machines virtuelles de manière interactive, avec des options pour des configurations simples ou avancées.

Ces outils sont très utiles pour optimiser l'expérience de gestion de Proxmox VE.

Reverse proxy avec VPS frontend

📋 Objectif

- Utiliser un VPS OVH comme passerelle d'accès
- Tunnel des services locaux via FRP (Fast Reverse Proxy)
- Reverse proxy HTTPS avec Caddy sur le VPS
- Accès via des sous-domaines personnalisés

📋 Prérequis

- Un VPS avec Ubuntu/Debian
- Un nom de domaine chez OVH avec DNS pointés vers le VPS
- Un serveur local (Proxmox) avec :
 - APP1 sur 192.168.X.X:8080
 - APP2 sur 192.168.X.X:8080/exemple
 - PROXMOX sur 192.168.X.X:8662

Différents cas d'usage sont donnés ici. APP1 étant une application classique sans configuration particulière. APP2 est une application utilisant une redirection vers un lien spécifique, ici le dossier "exemple". Et enfin une configuration adapté pour PROXMOX.

📁 Partie 1 : Installer FRP

Sur le VPS (serveur FRP)

```
cd /opt
wget https://github.com/fatedier/frp/releases/download/v0.58.0/frp_0.58.0_linux_amd64.tar.gz
tar -xzf frp_0.58.0_linux_amd64.tar.gz
mv frp_0.58.0_linux_amd64 frp
nano /opt/frp/frps.ini
```

Contenu :

```
[common]
```

```
bind_port = 6000
```

Lancer manuellement (ou voir plus bas pour le service) :

```
/opt/frp/frps -c /opt/frp/frps.ini
```

Créer le service systemd :

```
sudo nano /etc/systemd/system/frps.service
```

Contenu :

```
[Unit]
```

```
Description=FRP Server
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=/opt/frp/frps -c /opt/frp/frps.ini
```

```
Restart=always
```

```
User=user
```

```
LimitNOFILE=65535
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Puis :

```
sudo systemctl daemon-reexec
```

```
sudo systemctl enable --now frps
```

Sur le serveur local (client FRP)

```
cd /opt
```

```
wget https://github.com/fatedier/frp/releases/download/v0.58.0/frp_0.58.0_linux_amd64.tar.gz
```

```
tar -xzf frp_0.58.0_linux_amd64.tar.gz
```

```
mv frp_0.58.0_linux_amd64 frp
```

```
nano /opt/frp/frpc.ini
```

Contenu :

```
[common]
server_addr = IP_DU_VPS
server_port = 6000
```

```
[APP1]
type = tcp
local_ip = 192.186.X.X
local_port = 8080
remote_port = 9000
```

```
[APP2]
type = tcp
local_ip = 192.168.X.X
local_port = 8080
remote_port = 9001
```

```
[PROXMOX]
type = tcp
local_ip = 192.168.X.X
local_port = 8662
remote_port = 9002
```

Si vous réalisez une modification/ajout dans le fichier ci-dessus, pensez bien à redémarrer FRPC `systemctl restart frpc`

Créer le service FRP client :

```
sudo nano /etc/systemd/system/frpc.service
```

Contenu :

```
[Unit]
Description=FRP Client
After=network.target

[Service]
ExecStart=/opt/frp/frpc -c /opt/frp/frpc.ini
Restart=always
User=user
LimitNOFILE=65535
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Et activer :

```
sudo systemctl daemon-reexec
```

```
sudo systemctl enable --now frpc
```

☐ Partie 2 : Installer Caddy sur le VPS

```
sudo apt install -y debian-keyring debian-archive-keyring curl
```

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o  
/usr/share/keyrings/caddy.gpg
```

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy.list
```

```
sudo apt update
```

```
sudo apt install caddy
```

☐ Partie 3 : Configuration du Caddyfile (côté VPS)

```
sudo nano /etc/caddy/Caddyfile
```

Contenu recommandé :

Ce contenu comporte différents types de configuration. Prenez le plus adapté à votre cas d'usage.

```
proxmox.domaine.com {  
  reverse_proxy https://localhost:9000 {  
    transport http {  
      tls_insecure_skip_verify  
      versions h1  
    }  
    header_up Host {host}  
    header_up X-Real-IP {http.request.remote}  
    header_up X-Forwarded-For {http.request.remote}  
    header_up X-Forwarded-Proto {http.request.scheme}  
  }  
}
```



```
app1.domaine.com {  
    reverse_proxy localhost:6001  
}  
  
app2.domaine.com {  
    @exemple_path path /*  
    rewrite @exemple_path /exemple{path}  
  
    reverse_proxy localhost:9002 {  
        transport http {  
            versions h1  
        }  
        header_up Host {host}  
        header_up X-Forwarded-Proto https  
    }  
}
```

Si vous réalisez une modification/ajout dans le fichier ci-dessus, pensez bien à redémarrer CADDY `systemctl reload caddy`

Puis :

```
sudo systemctl reload caddy
```

☐ Test des services

- Proxmox : <https://proxmox.domaine.com>
- APP1 : <https://app1.domaine.com>
- APP2 : <https://app2.domaine.com> (mais affiché sans /exemple)

Tester en navigation privée pour éviter les cookies HTTPS précédents.

☐ Et voilà !

Tu as maintenant une exposition sécurisée de tes services locaux via un VPS, sans exposer ton IP personnelle, avec HTTPS et Caddy.