

# Montée d'un serveur Publik en version Debian 11 (Bullseye)

Cette page explique comment mettre à jour un serveur hébergeant les applications Publik pour passer de la version Debian 10 (Buster) à la version Debian 11 (Bullseye).

## Préparation de l'environnement

Pensez à **vérifier les sauvegardes** . Si possible, faire une image (snapshot) ou une sauvegarde juste avant la mise à jour.

S'assurer que les outils de supervision et de gestion des configuration sont compatibles avec Debian 11.

Vérifier que la ou les machines en Debian 10 sont bien à jour :

```
# apt update
# apt full-upgrade
```

Mise à jour préalable de **PostgreSQL** : si le serveur de base de données est séparé, il doit être mis à jour **avant** les serveurs d'applications Publik. Il faut passer en **Postgresql version 13** au minimum. Voir plus bas dans ce document pour une procédure si la machine est une Debian.

## Préparation de la mise à jour

Dans le cas d'une configuration à plusieurs nœuds, faire **un serveur après l'autre**. Ne jamais faire de mise à jour simultanée sur plusieurs nœuds.

Arrêter le système de *cron* :

```
# systemctl stop cron
```

Modifier les dépôts APT pour y indiquer "bullseye" comme base. Il s'agit typiquement de mettre à jour le fichier `/etc/apt/sources.list` et ceux contenus dans le répertoire `/etc/apt/sources.list.d/`. Exemple de manipulation pour faire cette modification :

```
# sed -i 's/buster/bullseye/g' /etc/apt/sources.list
# sed -i 's/buster/bullseye/g' /etc/apt/sources.list.d/*
# sed -i 's!bullseye/updates!bullseye-security!g' /etc/apt/sources.list
```

Prise en compte de ces nouveaux dépôts :

```
# apt update
```

Installation des dépôts et fichiers de préférences APT relatifs à Publik sur les environnements de production :

```
# apt install entrouvert-repository entrouvert-repository-hotfix && sudo apt update
```

S'il s'agit d'un environnement de **test / pré-production**, et uniquement pour ceux là :

```
# apt install entrouvert-repository entrouvert-repository-testing && sudo apt update
```

## Lancement de la mise à jour

Très classiquement :

```
# apt update
# apt full-upgrade
```

## Passage en PostgreSQL version 13

Si le serveur d'application héberge aussi la base de données, après la montée de version vers Debian 11 il faut lancer la montée de version de PostgreSQL. En effet, deux clusters subsistent à la suite de la mise à jour, un actif en version 11 (venant de Debian 10) et l'autre en version 13 (installé par la mise à jour Debian 11).

Il s'agit de basculer vers le cluster en version 13 et de désactiver celui en version 11.

La documentation pour cela est sur `/usr/share/doc/postgresql-common/README.Debian.gz`. De façon pratique, la procédure revient à ces commandes :

```
# sudo -u postgres pg_dropcluster --stop 13 main
# sudo -u postgres pg_upgradecluster -m upgrade -k -j 4 11 main
# sudo -u postgres reindexdb --concurrently --all -j 4
# systemctl disable postgresql@11-main.service
```

Le 4 dans le « -j 4 » de la réindexation est à adapter selon le nombre de CPU.

**Attention** : le « -k » du `pg_upgrade` accélère la migration et fait gagner de l'espace, mais **rend le cluster 11 hors d'usage**. Si vous souhaitez conserver le cluster 11, ne l'utilisez pas. Dans tous les cas vérifiez vos sauvegardes avant la mise à jour.

## Nettoyage

À la suite de la mise à jour, un nettoyage des paquets inutiles peut être fait.

```
# apt autoremove
```

Puis un redémarrage complet de la machine est nécessaire pour passer sur le dernier noyau et réactiver les *crons*.