Déploiement de Chocolatey et d'un dépôt local pour FOG Project 1.6

Cette documentation décrit la mise en place d'un dépôt local **Chocolatey** (édition Open Source) hébergé sous **Docker (BaGet)** et son intégration avec **FOG Project 1.6.0.158** pour gérer le déploiement logiciel sur un parc de 500+ postes Windows.

Architecture et choix techniques

Dans FOG 1.6, le module **Software Management** s'appuie sur le client FOG (exécuté sous le compte `SYSTEM`) pour vérifier et installer des paquets Chocolatey de manière déclarative.

Pour un parc supérieur à 500 machines, l'utilisation d'un dépôt local est indispensable afin d'éviter les limitations de requêtes du dépôt communautaire public et d'accélérer les téléchargements sur le réseau local.

Critère	BaGet (Docker sous Linux)	Sonatype Nexus OSS	Chocolatey.Server (IIS)	Partage SMB / NAS
Tenue de charge (500+ PC)	Très fluide (API NuGet v3 légère)	Excellente	Correcte mais gourmande en RAM	À proscrire (verrous et lenteurs)
Coût et licence	Gratuit et Open Source	Gratuit (version OSS)	Licence Windows Server requise	Gratuit mais inadapté
Maintenance	Très faible (conteneur unique)	Modérée (interface riche, Java)	Moyenne (maintenance Windows/IIS)	Faible mais sans interface web

Solution retenue : Conteneur Docker **BaGet** sous Linux (sur le serveur FOG ou une machine virtuelle dédiée).

Prérequis

- Un serveur Linux disposant de **Docker** et **Docker Compose**.
- Un serveur **FOG Project 1.6.0.158** opérationnel avec le client FOG déployé sur les postes Windows.
- Un poste d'administration Windows (avec accès Internet temporaire) sur lequel Chocolatey CLI est installé pour préparer et pousser les paquets.
- Les ports réseau suivants ouverts entre les postes clients et les serveurs :
  - `80` / `443` vers le serveur FOG.
  - `5555` vers le serveur hébergeant BaGet.

Déploiement du serveur de dépôt local (BaGet)

Sur le serveur Linux hébergeant Docker, créer l'arborescence dédiée au dépôt :

```
mkdir -p /opt/chocolatey-repo/data
cd /opt/chocolatey-repo
```

Créer le fichier `docker-compose.yml` avec le contenu suivant :

```
services:
  baget:
    image: loicsharma/baget:latest
    container_name: chocolatey-repo
    restart: unless-stopped
    ports:
      - "5555:80"
    environment:
      - ApiKey=REMPLACER_PAR_UNE_CLE_API_COMPLEXE
      - Storage__Type=FileSystem
      - Storage__Path=/var/baget/packages
      - Database__Type=Sqlite
      - Database__ConnectionString=Data Source=/var/baget/baget.db
      - Search__Type=Database
      - AllowPackageOverwrites=true
    volumes:
      - ./data:/var/baget
```

> **Note :** Remplacer `REMPLACER_PAR_UNE_CLE_API_COMPLEXE` par un jeton secret. Cette clé sera exigée lors de l'envoi de nouveaux paquets vers le dépôt.

Démarrer le conteneur en arrière-plan :

```
docker compose up -d
```

Vérifier le bon fonctionnement du service :

- **Interface web :** `http://<IP_SERVEUR_BAGET>:5555`
- **URL de l'index NuGet (source Chocolatey) :** `http://<IP_SERVEUR_BAGET>:5555/v3/index.json`

Préparation et publication des paquets (Open Source)

Avec l'édition Open Source de Chocolatey, les installeurs (`.msi` ou `.exe`) doivent être embarqués directement à l'intérieur des archives `.nupkg` afin que les postes clients n'aient aucun accès Internet à effectuer.

Publication du paquet client Chocolatey

Ce paquet est requis pour permettre à FOG d'installer automatiquement Chocolatey sur les postes vierges.

Depuis le poste d'administration Windows (PowerShell) :

```
# Téléchargement du paquet officiel Chocolatey
New-Item -ItemType Directory -Force -Path "C:\Temp\ChocoPackages"
Invoke-WebRequest -Uri "https://chocolatey.org" -OutFile
"C:\Temp\ChocoPackages\chocolatey.nupkg"

# Envoi vers le dépôt local BaGet
choco push "C:\Temp\ChocoPackages\chocolatey.nupkg" --
source="http://<IP_SERVEUR_BAGET>:5555/v3/index.json" --api-
key="REMPLACER_PAR_UNE_CLE_API_COMPLEXE"
```

Création d'un paquet logiciel autonome

Exemple de création d'un paquet hors-ligne pour **7-Zip** :

1. Générer la structure du paquet :

```
cd C:\Temp\ChocoPackages
choco new 7zip
```

2. Copier le fichier d'installation officiel (ex. `7z2408-x64.msi`) dans le sous-dossier `C:\Temp\ChocoPackages\7zip\tools\`.
3. Éditer le fichier `C:\Temp\ChocoPackages\7zip\7zip.nuspec` pour renseigner la version exacte (ex. `<version>24.8.0</version>`) et nettoyer les métadonnées inutiles.
4. Remplacer le contenu du script `C:\Temp\ChocoPackages\7zip\tools\chocolateyinstall.ps1` par :

```
\(ErrorActionPreference = 'Stop'\)toolsDir     = "(Split-Path -parent
MyInvocation.MyCommand.Definition)"
fileLocation = Join-Path toolsDir '7z2408-x64.msi'

\$packageArgs = @{
  packageName    = \$env:ChocolateyPackageName
  fileType       = 'MSI'
  file           = \$fileLocation
  softwareName   = '7-Zip*'
  silentArgs     = "/qn /norestart"
  validExitCodes = @(0, 3010, 1641)
}

Install-ChocolateyInstallPackage @packageArgs
```

5. Compiler et publier le paquet sur le serveur BaGet :

```
cd C:\Temp\ChocoPackages\7zip
choco pack
choco push "7zip.24.8.0.nupkg" --
source="http://<IP_SERVEUR_BAGET>:5555/v3/index.json" --api-
key="REMPLACER_PAR_UNE_CLE_API_COMPLEXE"
```

Configuration de FOG Project

Hébergement du script d'amorçage (Bootstrap)

Pour que l'agent FOG installe Chocolatey sans accéder à Internet, il faut héberger une version modifiée du script officiel `install.ps1` sur le serveur web de FOG.

1. Télécharger le script depuis `https://chocolatey.org`.
2. Ouvrir le fichier et modifier la variable `$url` (ou `$chocolateyDownloadUrl`) afin qu'elle pointe vers l'URL de téléchargement direct du paquet `chocolatey.nupkg` hébergé sur BaGet :

   ```
   \$url = 'http://<IP_SERVEUR_BAGET>:5555/v3/package/chocolatey/2.3.0'
   ```

   (Adapter le numéro de version selon le paquet poussé précédemment).
3. Placer ce fichier sur le serveur FOG dans le répertoire web public :

   ```
   mkdir -p /var/www/html/choco
   # Copier le fichier install.ps1 modifié dans /var/www/html/choco/install.ps1
   chown -R www-data:www-data /var/www/html/choco
   ```

Paramétrage dans l'interface web FOG

1. Se connecter à l'interface d'administration de FOG.
2. Naviguer vers **FOG Configuration** > **FOG Settings** > **FOG Client**.
3. Renseigner les paramètres suivants :
   - **`FOG_SOFTWARE_CHOCO_BOOTSTRAP_URL`** : `http://<IP_SERVEUR_FOG>/choco/install.ps1`
   - **`FOG_SOFTWARE_CHOCO_NUPKG_URL`** : `http://<IP_SERVEUR_BAGET>:5555/v3/package/chocolatey/2.3.0`
4. Enregistrer les modifications.

Bascule des postes clients vers la source locale

Par défaut, Chocolatey conserve le dépôt communautaire public dans sa configuration. Il est nécessaire de le supprimer et de déclarer le dépôt local sur l'ensemble du parc (via un **Snapin** FOG ou une **GPO** au démarrage des ordinateurs) :

```
# Suppression de la source publique par défaut
choco source remove -n=chocolatey

# Ajout du dépôt local BaGet
choco source add -n="DepotLocal" -s="http://<IP_SERVEUR_BAGET>:5555/v3/index.json" --
priority=1
```

Déploiement d'un logiciel sur le parc

1. Dans l'interface web FOG, ouvrir le menu **Software**.
2. Cliquer sur **Create New Software**.
3. Renseigner le nom d'affichage et l'identifiant exact du paquet tel qu'il apparaît dans BaGet (ex. `7zip`).
4. Associer ce logiciel aux hôtes ou aux groupes de machines souhaités via l'onglet **Software** de chaque hôte ou groupe.

Maintenance et bonnes pratiques

- **Sauvegarde du dépôt :** Sauvegarder régulièrement le répertoire `/opt/chocolatey-repo/data` qui contient à la fois la base de données SQLite (`baget.db`) et tous les fichiers `.nupkg`.
- **Montée de version d'un logiciel :** Incrémenter la balise `<version>` dans le fichier `.nuspec`, remplacer l'installeur dans `tools\`, recompiler avec `choco pack` et renvoyer avec `choco push`. L'agent FOG mettra automatiquement à jour les postes lors de son prochain cycle de vérification.
- **Charge réseau :** Sur un parc de 500+ postes, vérifier que l'intervalle de communication du client FOG (`FOG_CLIENT_CHECKIN_TIME`) est d'au moins `300` à `600` secondes afin d'étaler les requêtes vers le serveur BaGet.

Revision #1
Created 2026-11-23 11:04:41 UTC by Axel
Updated 2026-11-23 11:05:02 UTC by Axel