

Installation et Configuration

Procédure de déploiement Coolify

Installation initiale

Pour l'installation, il suffit d'exécuter cette commande qui va configurer la machine afin que l'installation ait le moins d'erreurs possible. C'est la méthode recommandée par Coolify :

```
curl -fsSL https://cdn.coollabs.io/coolify/install.sh | sudo bash
```

Une fois le script terminé, vous pouvez accéder à l'interface web de Coolify.

Configuration du reverse proxy hybride

Problématique

Pour que le service soit utile, j'avais besoin de rapidité. Par exemple, si je souhaite déployer un Bookstack pour le tester, je veux créer le projet, cliquer sur la création du conteneur et avoir directement la possibilité de cliquer sur un lien pour que le service s'affiche.

Ceci est possible grâce au reverse proxy intégré à Coolify. Cependant, j'ai déjà un Traefik et il était trop compliqué de mettre les deux en parallèle, surtout pour l'accès externe.

Solution mise en place : accès hybride

J'ai donc implémenté un accès hybride comprenant deux parties :

1. Accès à l'interface web Coolify

Mon reverse proxy primaire redirige le lien d'accès externe suivant :

`https://coolify.antoninpomies.fr → Interface web de Coolify`

De cette manière, je peux y accéder facilement en interne et en externe.

2. Configuration DNS pour les services déployés

J'ai créé une nouvelle zone de recherche DNS sur mon serveur : **pprod.loc**

Dans cette zone, j'ai créé deux enregistrements A :

- A coolify.pprod.loc → <ip-de-mon-coolify>
- A *.coolify.pprod.loc → <ip-de-mon-coolify>

Configuration dans Coolify

Pour configurer ce nom de domaine dans Coolify, il faut se rendre dans :

`Servers > Votre serveur > Wildcard Domain > http://coolify.pprod.loc`

De cette manière, tous les sous-domaines générés par Coolify auront un FQDN similaire à :

`service-xxxxx.coolify.pprod.loc`

Avantages de ce système

- ☐ Accès simple à l'interface web de Coolify comme l'ensemble des autres services de mon infrastructure
- ☐ Depuis l'interface, les liens générés sont au format `xxxxx.coolify.pprod.loc` et il est facile grâce à un bouton d'accéder aux services rapidement et sans action sur le reverse proxy principal
- ☐ Pas d'utilisation de solution externe, tout est natif et pris en charge par Coolify ce qui évite la dépendance à des services potentiellement plus maintenu après plusieurs années.

Inconvénients

- ⚠ Les services déployés ne sont pas accessibles depuis l'extérieur de manière simple

Ce qui n'est pas un très grave problème vu qu'il s'agit uniquement de préproduction.

Objectif et gains de productivité

Contexte initial

Je déploie souvent de nouveaux services qui sont très régulièrement dockerisables. Habituellement, je déployais de la manière suivante :

1. **Déploiement d'une VM** (5 à 10 min)
2. **Installation de Docker et création du docker-compose** avec téléchargement des images Docker nécessaires (5 à 10 min)
3. **Création manuelle d'un routeur sur mon Traefik principal** (2 min)
4. **Accès au service**

Temps total : environ 15 à 20 minutes

Processus avec Coolify

Maintenant, grâce à Coolify :

1. **Connexion à l'interface web et création d'un projet** (30s)
2. **Recherche du service** dans la bibliothèque Coolify (beaucoup de services sont déjà pris en charge) (10s)
3. **Clic sur le bouton "Deploy"** et attente du téléchargement des images (variable)
4. **Accès immédiat au service** via le lien fourni par Coolify

Temps total : environ 1 à 5 minutes

Bilan

- **Temps gagné** : environ 15 minutes par déploiement
- **Avantages supplémentaires** :
 - Réduction des procédures répétitives
 - Interface unifiée pour la gestion des services de test
 - Déploiement standardisé

Note importante

Ce système ne change pas le fait que pour une mise en production, j'aurai besoin de déployer une machine dédiée. Cependant, si le service ne me plaît pas, je n'aurai pas perdu 20 minutes pour rien. De plus je pense qu'il est possible d'intégrer de l'laC grâce à github et améliorer la préproduction.

Revision #4

Created 29 January 2026 14:56:15 by Antonin POMIES

Updated 29 January 2026 15:21:01 by Antonin POMIES