

n8n s'utilise à partir de votre navigateur préféré. Si vous souhaitez faire un test sans pour autant déployer votre instance locale, sachez que la version Cloud dispose d'un essai de 14 jours.

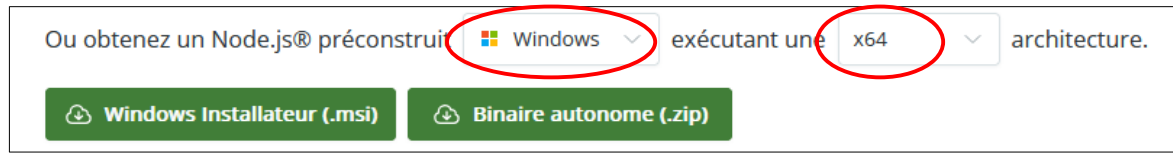
Table des matières

1, Installer n8n sur Windows sans Docker :.....	2
2, Installer n8n sur Linux sans Docker :.....	3
3, Utiliser PM2 pour le démarrage automatique et la gestion des processus n8n sous Linux :.....	4
4, Installer n8n sur un NAS Synology avec Docker :.....	5
5, Installer n8n sur un système Linux avec Docker :.....	8
<i>Etape 1</i> : Prérequis.....	8
<i>Etape 2</i> : Configurer l'environnement Docker et personnaliser n8n avec Docker Compose sous Linux.....	8
<i>Etape 3</i> : Démarrer et gérer les conteneurs Docker pour n8n sur système Linux.....	9
<i>Etape 4</i> : Configurer un reverse proxy Nginx pour sécuriser et exposer n8n en HTTPS sur Linux.....	9
6, Premiers pas dans n8n : accéder à l'interface.....	10
<i>Etape 1</i> : création du compte administrateur.....	11
<i>Etape 2</i> : Activation de la clé de licence.....	12
7, Optimiser et pérenniser son installation n8n sur Windows et Linux sans Docker.....	12
8, Cas pratiques et exemples d'intégration sans Docker avec n8n.....	13
<i>Etape 1</i> : création d'un workflow.....	13
<i>Etape 2</i> : Cas pratiques.....	15
<i>Etape 3</i> : Bibliothèque.....	15

1, Installer n8n sur Windows sans Docker :

Etape 1 :

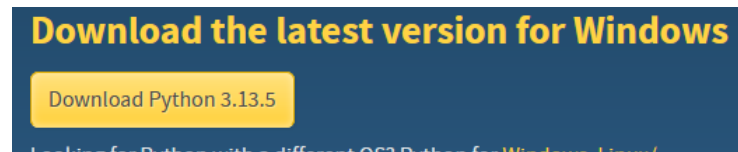
Commençons par télécharger « **Node.js** » via ce lien : <https://nodejs.org/fr/download>



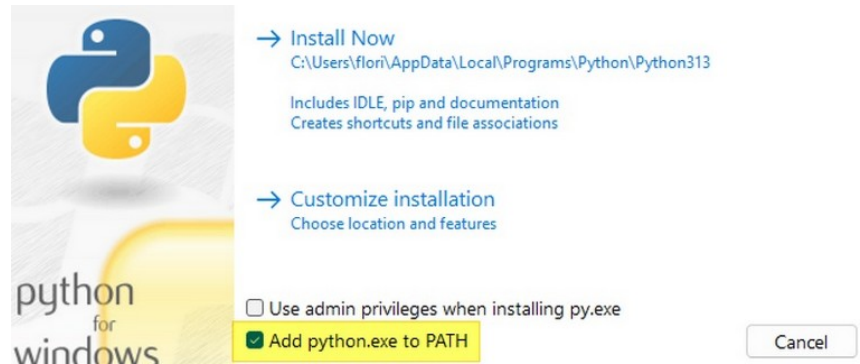
Dans notre exemple, on a choisit : « **Windows** » version « **x64** ». Cliquez sur le bouton "**Windows Installer (.msi)**". Lancez l'installation de Node.js et suivez l'assistant. Conservez les choix par défaut.

Etape 2 :

Commençons par télécharger « **Python** » via ce lien : <https://www.python.org/downloads/>



Quand vous allez procéder à l'installation de Python, veillez à cocher l'option "**Add python.exe to PATH**". Cela permet d'ajouter le chemin où se trouve le binaire de Python à la variable d'environnement **PATH** de Windows. Ainsi, Python peut être appelé depuis n'importe quel emplacement du système. Effectuez une installation classique via le bouton "**Install Now**".



Puis patientez pendant l'installation.

Etape 3 :

Ouvrir une console PowerShell ou une invite de commande (Terminal) :
Taper la commande : « **npm install -g n8n** ».

```
PS C:\Users\Utilisateur> npm install -g n8n
npm : Impossible de charger le fichier D:\Program\node.js\npm.ps1, car l'exécution de scripts est désactivée sur ce système. Pour plus d'informations, consultez about_Execution_Policies à l'adresse https://go.microsoft.com/fwlink/?LinkID=135170.
Au caractère Ligne:1 : 1
+ npm install -g n8n
+ ~~~~
+ CategoryInfo          : Erreur de sécurité : (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Si vous avez ce message en rouge, c'est que l'exécution de scripts est désactivée.
Pour l'activer, la ligne de commande est : « **Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force** »

```
PS C:\Users\Utilisateur> Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force
PS C:\Users\Utilisateur> npm install -g n8n
^
```

Puis recommencer à installer **n8n**

```
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated infisical-node@1.3.0: Package no longer supported. Contact us for more info.

added 1951 packages in 3m

228 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 10.9.2 -> 11.4.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
npm notice To update run: npm install -g npm@11.4.2
npm notice
PS C:\Users\Utilisateur>
```

Exécutez la commande qui suit après le texte "To update run" :
Dans notre exemple, c'est : « **npm install -g npm@11.4.2** »

Mise à jour terminée.

Vous pouvez lancer « **n8n** ».

```
PS C:\Users\Utilisateur> npm install -g npm@11.4.2

added 1 package in 10s

25 packages are looking for funding
  run `npm fund` for details
PS C:\Users\Utilisateur> n8n
```

```
[license SDK] Skipping renewal on init because
OIDC login is disabled.
Version: 1.102.4

Editor is now accessible via:
http://localhost:5678

Press "o" to open in Browser.
```

L'application sera alors accessible via l'adresse :
http://localhost:5678

Puis, appuyer sur la touche du clavier la lettre « **o** » pour l'ouvrir.

Sachez que pour le moment, n8n ne se lancera pas automatiquement au démarrage de Windows, il conviendra alors de rejouer cette commande (des solutions existent). Sinon, lancez n8n par l'intermédiaire d'une tâche planifiée qui se lance au démarrage de Windows.

2, Installer n8n sur Linux sans Docker :

Etape 1 :

Lancer un Terminal sur votre machine Linux. Se connecter en tant qu'administrateur pour effectuer l'installation :

Mettre à jour le cache des paquets && mettre à jour les paquets :

sudo apt update && sudo apt upgrade

Installation de l'environnement Python :

sudo apt install build-essential python3 python3-pip python3-venv

Pour installer la prise en charge native des pilotes NVIDIA CUDA afin d'utiliser le GPU :

sudo apt install nvidia-cuda-toolkit nvidia-cudnn

Etape 2 :

installation de Node.js Version Manager :

sudo curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh | bash

source ~/.bashrc

nvm install 20

nvm use 20

nvm alias default 20

node -v

```
flo@Ubuntu2404:~$ nvm install 20
Downloading and installing node v20.19.3...
Downloading https://nodejs.org/dist/v20.19.3/node-v20.19.3-linux-x64
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v20.19.3 (npm v10.8.2)
Creating default alias: default -> 20 (-> v20.19.3)
flo@Ubuntu2404:~$ nvm use 20
Now using node v20.19.3 (npm v10.8.2)
flo@Ubuntu2404:~$ nvm alias default 20
default -> 20 (-> v20.19.3)
flo@Ubuntu2404:~$ node -v
v20.19.3
flo@Ubuntu2404:~$
```

Etape 3 :

Passons à l'installation de n8n :

npm install -g n8n

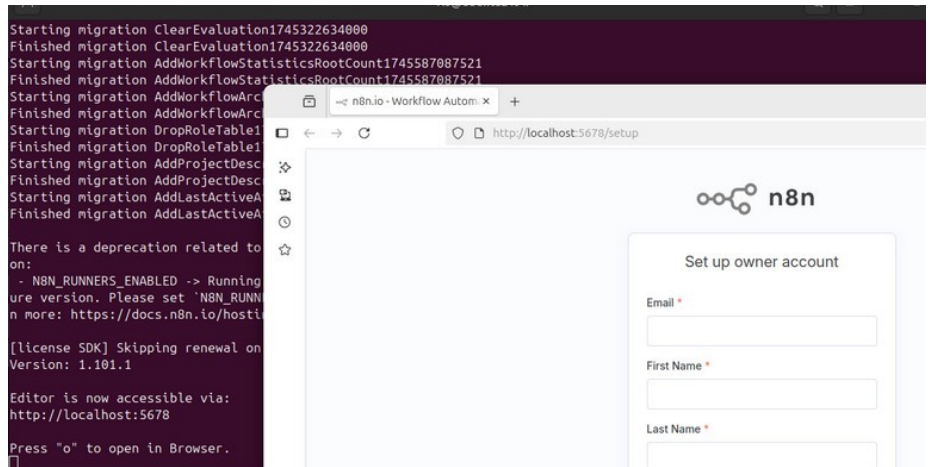
Une fois installé, vérifiez la version :

n8n --version

Pour démarrer l'application, il suffit de lancer :

n8n

Le serveur est accessible ensuite sur **http://localhost:5678**.



3, Utiliser PM2 pour le démarrage automatique et la gestion des processus n8n sous Linux :

PM2 est un gestionnaire de processus conçu pour maintenir les applications Node.js en fonctionnement continu, même en cas de redémarrage de la machine. Il permet aussi de configurer des redémarrages planifiés, ce qui est très utile pour pérenniser l'intégration et limiter les interruptions dans les workflows. Voici comment procéder :

Installez PM2 globalement :

sudo npm install -g pm2

Démarrez n8n avec une option de redémarrage journalier (ici à minuit) :

pm2 start n8n --cron-restart="0 0 * * *" -- start

Enregistrez la configuration :

pm2 save

Générez le script de démarrage automatique au boot :

pm2 startup

Après avoir exécuté pm2 startup, une commande additionnelle est affichée : elle doit être lancée avec sudo, par exemple :

sudo env PATH=\$PATH:/home/user/.nvm/versions/node/v20.x.x/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u user --hp /home/user

N'hésitez pas à adapter la commande selon votre nom d'utilisateur et le chemin de l'installation NVM.

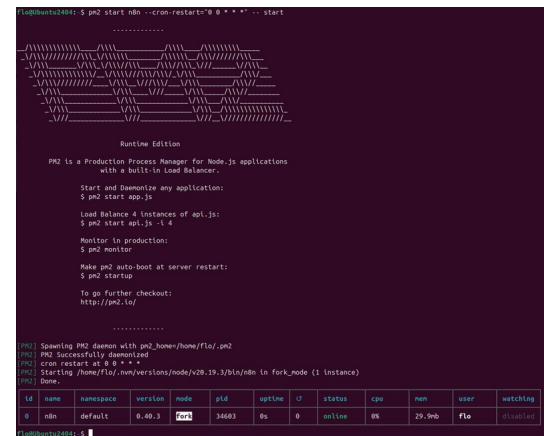
Autre exemple : **sudo env PATH=\$PATH:/home/flo/.nvm/versions/node/v20.19.3/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u flo --hp /home/flo**

Cette gestion fine garantit que n8n soit toujours prêt à répondre aux requêtes, rendant l'automatisation robuste même dans un environnement Linux multi-utilisateur ou serveur.

Pour s'assurer que n8n fonctionne bien après un reboot, tapez :

pm2 list

Celui-ci affichera le statut actif de l'application dans le gestionnaire PM2.



4, Installer n8n sur un NAS Synology avec Docker :

Etape 1 :

Ouvrir



« panneau de configuration »,



« Accès externe »,

puis aller vérifier que vous avez un nom de domaine dans l'onglet « **DDNS** » avec le statut **Normal**, il servira de référence pour le nom d'hôte à n8n.

QuickConnect	DDNS	Configuration du routeur	Avancé	
Ajouter	Modifier	Supprimer	Mettre à jour maintenant	Personnaliser le fournisseur
Fournisseur de service	Nom d'hôte	Adresse externe	Statut	Dernière heure de mise à jour
Synology	xtrem3vasion.synology.me		Normal	17-07-2025 17:58

Etape 2 :

Toujours dans le **panneau de configuration**, on va sur



« **Portail de connexion** », puis « **Avancé** » afin d'accéder au « **Proxy inversé** ».

Dans l'onglet «Général» :

Nom du proxy inversé : **n8n**

Protocole : **HTTPS**

Nom d'hôte : **n8n.xtrem3vasion.synology.me**

Port : **5443** (n'oubliez pas d'ajouter l'accès à ce port sur votre routeur).

Cocher « **Activer HSTS** ».

Nom d'hôte : **localhost**

Port : **5678**

Général	En-tête personnalisé	Paramètres avancés
Nom du proxy inversé :	n8n	
Source		
Protocole :	HTTPS	
Nom d'hôte :	n8n.xtrem3vasion.synology.me	
Port :	5443	
<input checked="" type="checkbox"/> Activer HSTS		
Profil de contrôle d'accès :	Non configuré	
Destination		
Protocole :	HTTP	
Nom d'hôte :	localhost	
Port :	5678	

Dans l'onglet « En-tête personnalisé » :

Cliquer sur « **Créer** », puis sur « **WebSocket** » afin d'afficher 2 lignes.

Cliquer sur le bouton « **Sauvegarder** ».

Général	En-tête personnalisé	Paramètres avancés
Créer	Supprimer	
Créer		Valeur
WebSocket		\$http_upgrade
Connection		\$connection_upgrade

Proxy inversé

Créer Modifier Supprimer

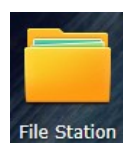
Description	Source	Destination
n8n	https://n8n.xtrem3vasion.synology.me:5443	http://localhost:5678

Fermer la fenêtre Proxy inversé ainsi que celle du panneau de configuration. ❌

Etape 3 :

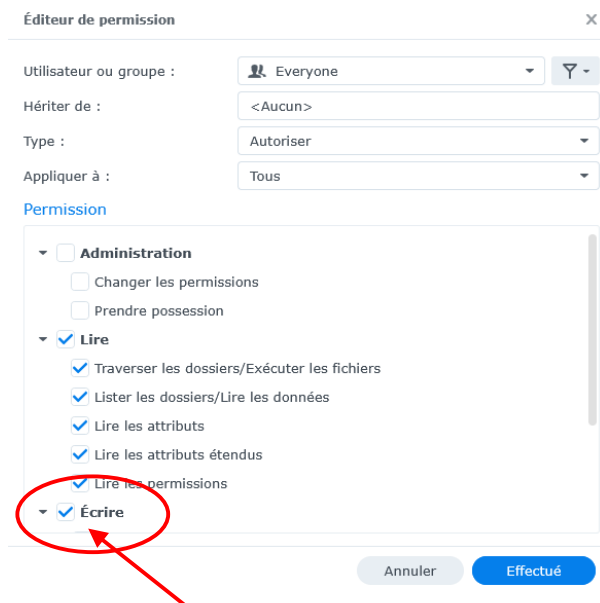
Pour la suite, il faut que Docker soit déjà installé sur le NAS, sinon voir le « **tutoriel – Docker** » pour l'installer.

Ouvrir « **File Station** », dans le dossier « **docker** », créer 3 sous-dossiers : « **data** », « **files** » et « **postgresql** »

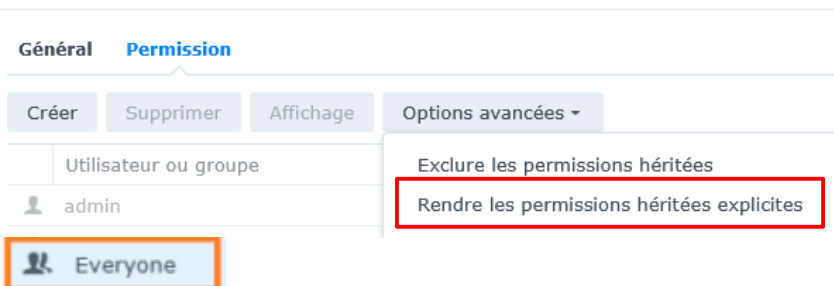


Nom
data
files
postgresql

Clic droit sur le dossier « **data** »,
 aller dans les **propriétés**,
 onglet « **Permission** »,
 puis le bouton « **Options avancées** »,
 Sélectionner « **Rendre les permissions héritées explicites** ».



Propriétés



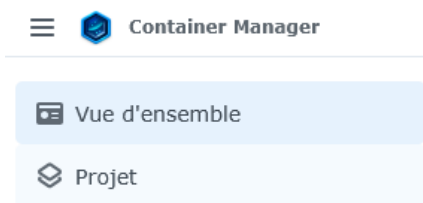
Double clic sur le groupe « **Everyone** » pour modifier ses permissions afin qu'ils puissent « **écrire** » en cochant simplement la case « **Ecrire** ».

Cliquez sur « **Effectué** », puis sur le bouton « **Sauvegarder** ».

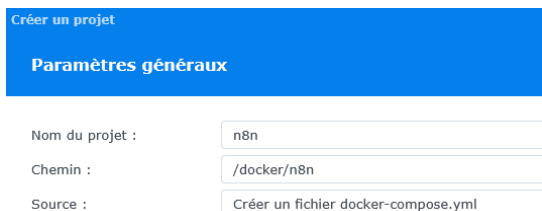


Etape 4 :

Ouvrir « **Container Manager** », clic gauche sur les 3 traits horizontaux en haut à gauche de la fenêtre qui vient de s'afficher, puis sur « **Projet** ».

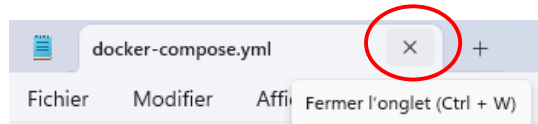


Bouton « **Créer** » un nouveau projet, puis mettre le nom du projet : **n8n**
 Chemin (exemple) : **/docker/n8n** (définir le chemin)
 Source : **Charger un fichier docker-compose.yml**



Revenir dans l'explorateur de fichiers Windows pour créer un nouveau document texte avec comme nom : « **docker-compose.yml** », puis clic droit sur « **Modifier dans Bloc-notes** », coller toutes les lignes de code ci-dessous dans le fichier, n'oubliez pas de modifier selon votre chemin d'accès et votre nom d'hôte avant de fermer et d'enregistrer.

```
services:
  postgres:
    image: postgres:17
    container_name: n8n-postgresql
    hostname: n8n_xtrem3vasion
    security_opt:
      - no-new-privileges:true
    healthcheck:
      test: ["CMD", "pg_isready", "-q", "-d", "2025-n8n", "-U", "2025-n8n-Dimitri"]
      timeout: 60s
      interval: 5s
      retries: 3
    restart: always
    environment:
      POSTGRES_DB: 2025-n8n
```



```

POSTGRES_USER: 2025-n8n-Dimitri
POSTGRES_PASSWORD: motdepasse
TZ: Europe/Paris
volumes:
  - /volume1/docker/n8n/postgresql:/var/lib/postgresql/data:rw
n8n:
  image: docker.n8n.io/n8nio/n8n
  container_name: n8n
  security_opt:
    - no-new-privileges:true
  restart: always
  ports:
    - "5678:5678"
  environment:
    N8N_HOST: n8n.xtrem3vasion.synology.me
    N8N_PORT: 5678
    N8N_PROTOCOL: https
    N8N_SECURE_COOKIE: true
    N8N_RUNNERS_ENABLED: true
    NODE_ENV: production
    WEBHOOK_URL: https://n8n.xtrem3vasion.synology.me:5443
    GENERIC_TIMEZONE: Europe/Paris
    TZ: Europe/Paris
    DB_TYPE: postgresdb
    DB_POSTGRESDB_HOST: n8n_xtrem3vasion
    DB_POSTGRESDB_DATABASE: 2025-n8n
    DB_POSTGRESDB_PORT: 5432
    DB_POSTGRESDB_USER: 2025-n8n-Dimitri
    DB_POSTGRESDB_PASSWORD: motdepasse
  volumes:
    - /volume1/docker/n8n/data:/home/node/.n8n:rw
    - /volume1/docker/n8n/files:/files:rw
  depends_on:
    postgres:
      condition: service_healthy

```

Créer un projet X

Paramètres généraux

Nom du projet :

Chemin : Définir le c...

Source : ▼

Fichier : Parcourir

```

1 services:
2   postgres:
3     image: postgres:17
4     container_name: n8n-postgresql
5     hostname: n8n_xtrem3vasion
6     security_opt:
7       - no-new-privileges:true
8     healthcheck:
9       test: ["CMD", "pg_isready", "-q", "-d", "2025=n8n", "-U", "2025=n8n=Dimitri"]
10      timeout: 60s
11      interval: 5s
12      retries: 3
13      restart: always

```

Suivant

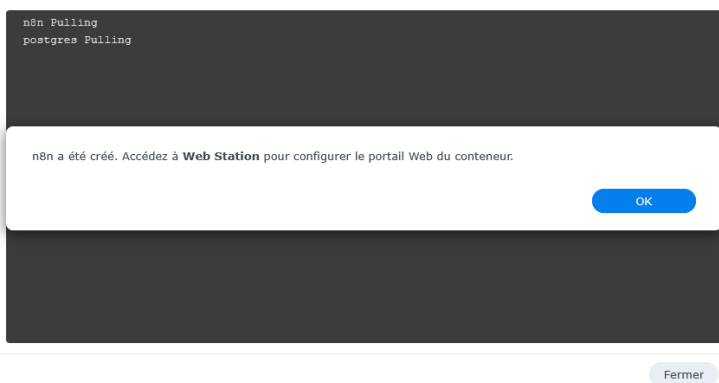
« Suivant »,

Activez cette option pour fournir des services HTTP/HTTPS au conteneur via Web Station. Seuls les ports exposés peuvent être utilisés pour Web Station.

Configurer le portail Web via Web Station

Cocher seulement si vous voulez que n8n est accès au Web Station sur votre NAS.

« Suivant », puis « Effectué »



Fermer

« **OK** » pour modifier le portail web du conteneur :

Modifier le portail du serveur de paquets

Service :

Statut : Normal

Type de portail :

Nom d'hôte :

Port : 80 / 443

HTTP

HTTPS

Paramètres HTTPS : HSTS

Profil de contrôle d'accès : i

Profil de page d'erreur : i

Activer les journaux d'accès i

Créer un raccourci dans le menu principal

Nom :

Utilisateurs autorisés :

Annuler

Appliquer

« **Appliquer** » pour valider les modifications.

5. Installer n8n sur un système Linux avec Docker :

Etape 1 : Prérequis

Voici la liste des prérequis indispensables :

- **Serveur Linux à jour** (Debian 11+, Ubuntu 22.04+ recommandés)
- **Installation de Docker et Docker Compose** avec versions compatibles (Docker 20.10+, Compose 2.x)
- **Reverse proxy installé** (Nginx ou Traefik conseillé)
- **Nom de domaine pointant vers le serveur** pour faciliter l'accès et la configuration SSL
- **PostgreSQL configuré** via Docker pour la gestion de la base de données n8n
- **Configuration réseau adaptée** : ports 80, 443 (HTTP/S) et 5678 (n8n) ouverts et redirigés si besoin

Tout débutant sur Linux gagnera à comprendre ce que représentent ces éléments : Docker encapsule des applications dans des conteneurs légers, isolés du système hôte, tandis qu'un reverse proxy comme Nginx redirige et sécurise les accès entrants vers n8n. PostgreSQL est une base relationnelle performante à privilégier pour une instance en production, du fait de sa robustesse et compatibilité avec le framework Node.js, sur lequel est basé n8n.

Etape 2 : Configurer l'environnement Docker et personnaliser n8n avec Docker Compose sous Linux

La magie du déploiement de n8n sur Linux repose sur la simplicité d'usage de Docker Compose. Ce fichier YAML décrit l'ensemble des conteneurs et leurs interactions dans un format lisible et modifiable. La configuration initiale consiste à créer une arborescence dédiée à n8n, puis à définir deux fichiers essentiels : **.env** pour les variables d'environnement, et **docker-compose.yml** pour orchestrer les services.

Un aspect souvent négligé est la ségrégation des données persistantes. Trois dossiers locaux doivent être créés : un pour les données PostgreSQL, un pour les fichiers générés par n8n, et un dernier pour les fichiers temporaires ou d'échange. Ces répertoires seront montés dans les conteneurs respectifs, ce qui évite la perte de données en cas de redémarrage ou mise à jour. Exemple de création des répertoires :

- `mkdir -p /opt/docker-compose/n8n/{postgresql,n8n_data,n8n_files}`
- `chmod 777 /opt/docker-compose/n8n/n8n_data` (nécessaire pour éviter les erreurs de permission)

Le fichier **.env** contient des variables clés :

- **N8N_HOST** : nom d'hôte que l'interface affichera et utilisera pour générer les URLs
- **N8N_WEBHOOK_URL** : URL accessible publiquement pour que les webhooks fonctionnent correctement
- **POSTGRES_DB, POSTGRES_USER, POSTGRES_PASSWORD** : informations de connexion à la base de données PostgreSQL
- **N8N_GENERIC_TIMEZONE** : fuseau horaire, impactant l'exécution des workflows

Le **docker-compose.yml** comprend deux services principaux :

- **postgres** : basé sur l'image officielle PostgreSQL avec réglage des variables d'environnement, volumes pour la persistance et contrôle de santé
- **n8n** : image docker officielle n8n, configuration des variables d'environnement pour connecter l'application à la base, gestion des ports, volume pour persister la data et fichiers, dépendance santé sur PostgreSQL.

En configurant via ces deux fichiers, vous créez un environnement **automatisé, reproductible et facilement modifiable**. C'est la force combinée du logiciel libre, Docker, et Linux. Cette méthode vous prémunit aussi contre les erreurs de dépendances ou incompatibilités entre versions.

Etape 3 : Démarrer et gérer les conteneurs Docker pour n8n sur système Linux

Une fois l'environnement prêt, il faut s'assurer que les permissions sur les dossiers montés soient correctes afin d'éviter que le conteneur n8n ne rencontre de difficultés à écrire ses fichiers. Dans la majorité des cas, un réglage `chmod 777` sur le dossier destiné à contenir les données de n8n s'avère nécessaire, bien que cette approche soit permissive et demande de bien comprendre les risques associés.

Lancer les conteneurs avec la commande suivante permet d'initier et déployer les services en arrière-plan :

- **sudo docker compose up -d**

Surveillez l'état des conteneurs via :

- **sudo docker compose ps** pour voir la liste des conteneurs actifs et leur statut
- **sudo docker compose logs -f** pour visualiser en temps réel les journaux de sortie, indispensable en cas de problème

Le conteneur PostgreSQL doit passer en « healthy » (état sain) pour que n8n puisse se connecter correctement et démarrer sans erreur. Les logs afficheront des messages tels que :

“Editor is now accessible via: <https://n8n.it-connect.local>”

Ce message confirme que le service est opérationnel et prêt à l'usage à travers le port 5678, accessible depuis votre navigateur via le nom de domaine configuré. L'utilisation d'un gestionnaire de logs simplifie grandement le diagnostic lorsque l'on déploie des solutions complexes reposant sur Node.js, base de données PostgreSQL et middleware.

Etape 4 : Configurer un reverse proxy Nginx pour sécuriser et exposer n8n en HTTPS sur Linux

Un reverse proxy est une brique fondamentale pour sécuriser l'accès à n8n, surtout si vous exploitez des webhooks ou exposez votre automation à travers Internet. Nginx s'impose comme un choix optimal sur Linux pour sa légèreté, fiabilité et facilité de configuration. Au lieu d'exposer directement le port Docker 5678, Nginx joue l'intermédiaire, permettant d'appliquer un certificat SSL/TLS et de gérer redirections et entêtes HTTP essentielles.

Voici les étapes-clés :

- Installer Nginx via le gestionnaire de paquets, par exemple « **sudo apt install nginx** » sur Debian/Ubuntu.
- Créer un répertoire pour stocker le certificat SSL auto-signé ou généré par Let's Encrypt.
- Générer un certificat avec OpenSSL (ou via Certbot pour Let's Encrypt) :

« **sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /etc/nginx/ssl/n8n.key -out /etc/nginx/ssl/n8n.crt** »

La configuration Nginx doit inclure deux blocs serveur :

- Un bloc écoutant sur le port 80, qui fera une redirection 301 vers HTTPS.
- Un bloc HTTPS, avec certificat, gestion des protocoles TLS 1.2 et 1.3, et configuration du proxy_pass sur **http://127.0.0.1:5678** où tourne n8n.

Le fichier de configuration essentiel ressemblera à ceci (nom de domaine à adapter) :

Extrait de configuration Nginx :

```
server {
    listen 80;
    server_name n8n.example.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name n8n.example.com;

    ssl_certificate /etc/nginx/ssl/n8n.crt;
    ssl_certificate_key /etc/nginx/ssl/n8n.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    location / {
        proxy_pass http://127.0.0.1:5678;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        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_buffering off;
        proxy_read_timeout 300s;
    }
}
```

Activation de ce site via un lien symbolique dans `/etc/nginx/sites-enabled/` puis rechargement de Nginx est indispensable :

```
sudo ln -s /etc/nginx/sites-available/n8n.example.com /etc/nginx/sites-enabled/
```

```
sudo systemctl reload nginx
```

En accédant à `https://n8n.example.com`, vous découvrirez l'interface d'administration n8n, protégée par HTTPS. Un avertissement de certificat SSL peut survenir si le certificat est auto-signé, mais ce point est réglable via une CA reconnue.

6, Premiers pas dans n8n : accéder à l'interface

Une fois n8n installé et accessible, la configuration des workflows permet d'exploiter pleinement l'automatisation. n8n est construit sur Node.js et offre une prise en charge native de centaines de services via leurs API, simplifiant les chaînes d'automatisation complexes.

Parmi les concepts clés :

- **Workflows** : séquences de tâches automatisées pouvant inclure des scripts, appels API, et déclencheurs divers.
- **Webhooks** : points d'entrée HTTP permettant de déclencher un workflow depuis l'extérieur, crucial pour réagir à des événements externes.
- **Intégrations** : connecteurs natifs pour MySQL, PostgreSQL, Redis, et bien d'autres, facilitant l'interconnexion des données.

Exemple pratique : Synchroniser automatiquement les données d'un CRM avec une base MySQL après réception d'un webhook déclenché par un formulaire client. Cette orchestration évite les erreurs manuelles, réduit le délai de traitement et libère des ressources humaines.

n8n propose aussi une interface graphique intuitive et des modèles prêt-à-l'emploi, accessibles via GitHub ou le site officiel, accélérant la mise en route même pour les novices. L'utilisation de Redis comme cache intermédiaire est recommandée pour améliorer la gestion des files d'attente dans des workflows volumineux en production.

La synergie entre Docker, Linux et n8n simplifie donc des déploiements traditionnellement complexes. Cette architecture modulaire facilite les mises à jour, le débogage et offre une montée en charge maîtrisée pour accompagner la croissance des usages d'automatisation.

Etape 1 : création du compte administrateur

À partir de votre machine, accédez à l'adresse suivante (si vous êtes en local) : **http://localhost:5678**. Vous arrivez alors sur la page de **setup**, où vous devez créer un compte administrateur de la plateforme. Complétez le formulaire, c'est obligatoire pour sécuriser l'accès à vos workflows et protéger votre automatisation.

The screenshot shows the 'Set up owner account' form on the n8n website. It includes fields for Email, First Name, Last Name, and Password. Below the password field, there is a note: '8+ characters, at least 1 number and 1 capital letter'. There is also a checkbox for 'I want to receive security and product updates' and a 'Next' button at the bottom.

E-mail
Prénom
Nom
Mot de passe

Cocher la case pour être avertit en cas de mise à jour.
Puis « Next »

« Get started »

The screenshot shows the 'Customize n8n to you' form. It contains several dropdown menus with the following questions: 'What best describes your company?', 'Which role best describes you?', 'Who will your automations mainly be for?', 'How big is your company?', and 'How did you hear about n8n?'. There is a 'Get started' button at the bottom right.

Get paid features for free (forever)

Receive a free activation key for the advanced features below - lifetime access.

- Workflow history**
Review and restore any workflow version from the last 24 hours
- Advanced debugging**
Easily fix any workflow execution that's errored, then re-run it
- Execution search and tagging**
Search and organize past workflow executions for easier review
- Folders**
Organize your workflows in a nested folder structure

Enter email to receive your license key *

d.isselin@hotmail.fr

Included features may change, but once unlocked, you'll keep them forever. [More info](#)

Skip

Send me a free license key

traduction :

Bénéficiez de fonctionnalités payantes gratuitement (pour toujours)

Recevez une clé d'activation gratuite pour les fonctionnalités avancées ci-dessous, accès à vie.

- Historique du workflow**
Examinez et restaurez n'importe quelle version du workflow des dernières 24 heures
- Débogage avancé**
Corrigez facilement toute exécution de workflow erronée, puis réexécutez-la
- Recherche et étiquetage des exécutions**
Recherchez et organisez les exécutions de workflow passées pour une révision plus facile
- Dossiers**
Organisez vos workflows dans une structure de dossiers imbriqués

Saisissez votre adresse e-mail pour recevoir votre clé de licence *

d.isselin@hotmail.fr

Les fonctionnalités incluses peuvent changer, mais une fois déverrouillées, vous les conservez pour toujours. Plus d'informations

Ignorer

M'envoyer une clé de licence gratuite

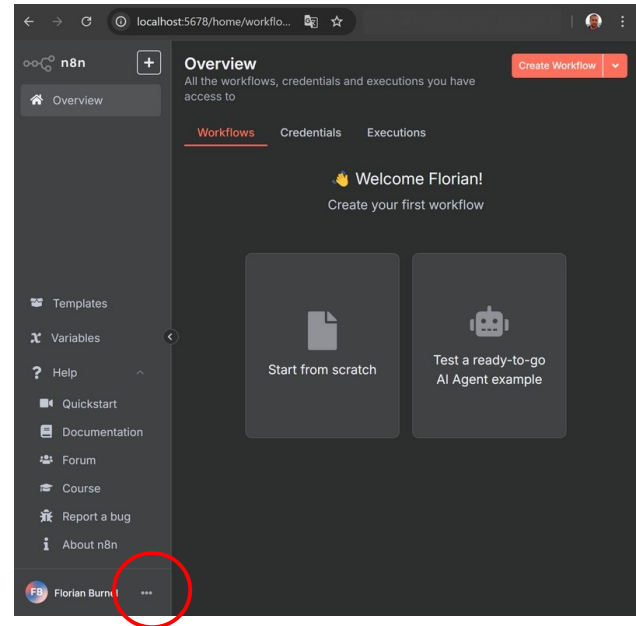
À partir de cette interface web intuitive, il est possible de créer, modifier et superviser un ensemble puissant de workflows connectant diverses applications et API. Cette modularité est l'un des points forts de n8n, car elle permet de déployer des intégrations personnalisées et adaptées à chaque cas d'usage, tout en restant dans un environnement open source.

Les workflows reposent sur un principe simple : connecter des blocs ou nœuds représentant des actions spécifiques, comme l'envoi d'un email, l'extraction de données via une API, ou encore l'exécution d'un script personnalisé. Chaque workflow peut être déclenché de multiples façons, selon des horaires, des événements externes ou à la demande.

- Relier des services via des API
- Automatiser des scripts et traitements
- Surveiller les exécutions et exceptions
- Collaborer en partageant les workflows

Pour les débutants, la bibliothèque de templates intégrée est une véritable aubaine : elle propose des modèles préconçus pour les intégrations les plus courantes, facilitant l'apprentissage et la mise en œuvre rapide. On peut aussi importer/exporter les workflows pour les partager au sein d'une équipe d'administrateurs ou de développeurs.

Les premières expérimentations avec n8n sont souvent révélatrices : un utilisateur peut, par exemple, automatiser l'envoi quotidien d'un rapport de logs serveur, synchroniser des données entre un CRM et une base de données, ou créer un agent intelligent combinant IA et automatisation.



Vous voilà désormais sur l'interface de n8n ! Le plus dur reste à faire : créer vos premiers Workflows, voire même votre premier agent IA.

Etape 2 : Activation de la clé de licence

En bas à gauche de la fenêtre, cliquez sur les 3 petits points (...), puis « **Settings** » puis « **Usage and plan** ».

1. Sélectionnez « **Déverrouiller** » pour entrer votre e-mail, puis sélectionnez « **Envoyez-moi une clé de licence gratuite** ».
2. Vérifiez votre adresse électronique pour le compte que vous avez saisi.

Une fois que vous avez une clé de licence, activez-la en cliquant sur le bouton dans l'e-mail de licence ou en visitant « **Settings** » puis « **Usage and plan** » et sélectionnez « **Enter activation key** ».

A screenshot of a dialog box titled 'Enter activation key'. It features a text input field containing the license key '8a7d7-7a-4-bdc7-703e00'. Below the input field are two buttons: 'Cancel' and 'Activate'.

Coller la clé de licence, puis « **Activate** ».

Vous avez débloquées des fonctionnalités :
- l'historique du flux de travail, débogage dans l'éditeur, dossiers et recherche d'exécution personnalisée.

Une fois activée, votre licence n'expirera pas. Vous faites partis dorénavant de la communauté n8n.

7, Optimiser et pérenniser son installation n8n sur Windows et Linux sans Docker

La stabilité et la pérennité d'une installation n8n hors Docker dépendent en grande partie de la maintenance régulière et de l'optimisation des routines d'exécution. Voici quelques conseils techniques clés :

- **Surveillance des logs** : n8n génère des logs détaillés accessibles en console. Surveillez-les pour détecter les erreurs éventuelles ou les ralentissements de workflows complexes.
- **Mise à jour du système** : gardez votre système Windows ou Linux à jour, notamment Node.js et Python, pour éviter les incompatibilités et bénéficier des dernières améliorations.
- **Gestion mémoire et ressources** : pensez à vérifier que n8n ne monopolise pas trop les ressources, en particulier quand plusieurs workflows sont actifs simultanément.

- **Backup régulier des workflows** : exportez régulièrement vos configurations et sauvegardez-les dans des dépôts Git ou sur des supports externes pour prévenir toute perte accidentelle.
- **Automatisation du démarrage** : sur Linux, l'utilisation de PM2 est un excellent moyen d'assurer que n8n tourne en permanence. Sous Windows, configurer une tâche planifiée est conseillé pour démarrer automatiquement le service.

L'approche sans Docker privilégie une installation légère, permettant un contrôle fin du système. Cependant, elle demande une vigilance accrue sur la maintenance du serveur et sur la sécurisation de la plate-forme, notamment dans des environnements multi-utilisateurs ou exposés sur Internet.

8, Cas pratiques et exemples d'intégration sans Docker avec n8n

Au-delà de l'installation, la vraie valeur de n8n réside dans sa capacité à faire dialoguer divers services et API. Sans la couche Docker, l'approche reste tout aussi efficace mais plus directe, car elle utilise directement le système hôte. Cela peut devenir un atout dans des environnements où la légèreté est un critère fort, comme les serveurs personnels ou les machines dédiées au développement.

Voici quelques exemples concrets pour inspirer votre exploitation de n8n en local :

- **Synchronisation entre un service mail IMAP et un CRM** : n8n peut superviser la boîte mail à la recherche de nouveaux messages, les analyser via un script Python, puis créer ou mettre à jour les contacts dans un CRM.
- **Automatisation de rapports système** : à partir des logs Linux, un script lancé par n8n compile un résumé quotidien qui est envoyé par email à une équipe opérationnelle via SMTP.
- **Agent intelligent pour réponses automatisées** : en intégrant une API d'intelligence artificielle, n8n orchestre la compréhension des requêtes utilisateurs et rédige des réponses adaptées, optimisant ainsi un service client ou helpdesk.

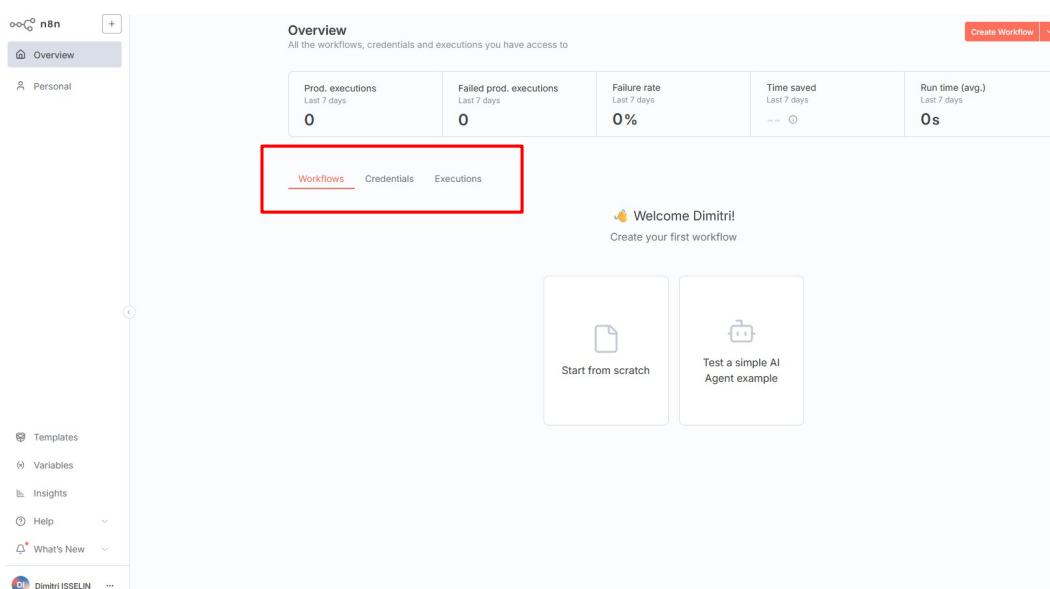
Chaque scénario suppose une bonne maîtrise des scripts, mais aussi de la logique des flux de données. n8n facilite cette orchestration avec une interface graphique intuitive et le support des API REST, ce qui rend l'ajout ou la modification de nœuds accessible même sans être expert dans le développement.

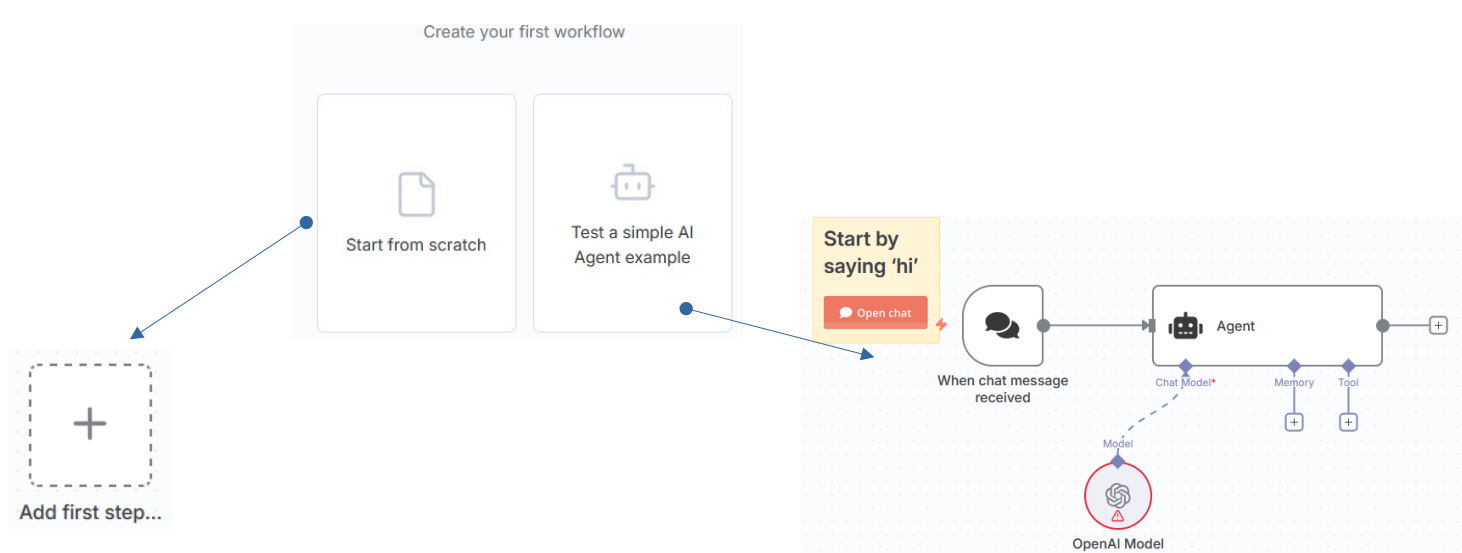
Etape 1 : création d'un workflow

Notre objectif sera le suivant : créer un workflow pour vérifier l'état du site web IT-Connect. Si le code HTTP est incorrect (autre que 200), c'est qu'il y a un problème ! Une alerte par e-mail sera envoyée.

Sur l'interface de n8n, nous pouvons déjà constater la présence de trois onglets sur la partie centrale :

- **Workflows** : créer, modifier, activer ou supprimer des scénarios d'automatisation.
- **Credentials** : gérer les identifiants pour accéder aux services externes (API, bases de données, e-mails, applications, etc.).
- **Executions** : consulter l'historique des exécutions de workflows pour analyser, déboguer ou relancer.

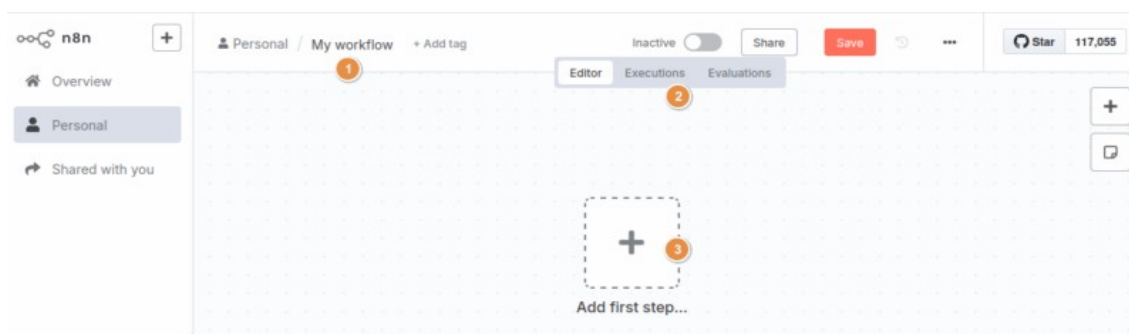




Et, surtout, nous avons le bouton "**Create Workflow**" en haut à droite (ou "**Start from scratch**" au centre) pour créer un nouveau workflow de toute pièce, puis cliquer sur « **Add first step...** ». C'est ce que vous pouvez faire, mais ce n'est pas obligatoire, car il existe de nombreux modèles sur Internet. Vous pouvez partir d'une base et la faire évoluer, selon vos besoins.

J'attire votre attention sur plusieurs éléments :

- 1 - Le nom de votre workflow, vous n'avez qu'à cliquer dessus pour le renommer.
- 2 - Basculez entre l'éditeur, l'historique des exécutions de ce workflow et le système de tests (évaluations).
- 3 - La zone centrale, c'est votre terrain de jeu, là où vous allez construire votre workflow et ajouter des étapes.



La première étape à ajouter correspond à l'élément déclencheur :

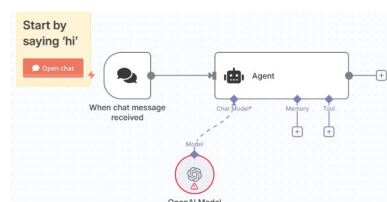
- Un déclencheur externe (webhook, email, fichier déposé...),
- Une planification (tous les jours à 8h, chaque lundi, toutes les 5 minutes...),
- Un événement dans une application,
- Un message reçu dans le chat,
- La soumission d'un formulaire,
- Etc...

Pour exécuter le workflow toutes les minutes, choisissez "**On a schedule**".

Choix du « **Start from scratch** » :



Choix du « **Test a simple AI Agent example** » :



Qu'est-ce qui déclenche ce flux de travail ?

Un déclencheur est une étape qui démarre votre flux de travail

Rechercher des nœuds...

Déclencher manuellement

Exécute le flux en cliquant sur un bouton dans n8n. Idéal pour démarrer rapidement

Sur l'événement de l'application

Exécute le flux lorsque quelque chose se produit dans une application comme Telegram, Notion ou Airtable

Selon un horaire

Exécute le flux tous les jours, toutes les heures ou à un intervalle personnalisé

Lors d'un appel webhook

Exécute le flux à la réception d'une requête HTTP

Sur le formulaire de soumission

Générer des formulaires Web dans n8n et transmettre leurs réponses au flux de travail

Lorsqu'il est exécuté par un autre workflow

Exécute le flux lorsqu'il est appelé par le nœud Exécuter le flux de travail à partir d'un autre flux de travail

Sur le message de chat

Exécute le flux lorsqu'un utilisateur envoie un message de discussion. À utiliser avec les nœuds AI

Lors de l'exécution de l'évaluation

Exécutez un ensemble de données via votre flux de travail pour tester les performances

D'autres moyens...

Exécute le flux sur les erreurs de workflow, les modifications de fichiers, etc.

Que se passe-t-il ensuite ?

Rechercher des nœuds...

AI



Construisez des agents autonomes, résumez ou recherchez des documents, etc.



Action dans une application



Faites quelque chose dans une application ou un service comme Google Sheets, Telegram ou Notion



Transformation des données



Manipuler, filtrer ou convertir des données



Couler



Branchez, fusionnez ou bouclez le flux, etc.



Cœur



Exécutez du code, effectuez des requêtes HTTP, définissez des webhooks, etc.



L'humain dans la boucle



Attendez l'approbation ou l'intervention humaine avant de continuer



Ajouter un autre déclencheur



Les déclencheurs démarrent votre flux de travail. Les workflows peuvent avoir plusieurs déclencheurs.



Etape 2 : Cas pratiques

Voir « [n8n – cas pratiques Volume 1](#) » pour avoir quelques idées.

Etape 3 : Bibliothèque

Pour débuter, vous pouvez aussi piocher dans la [bibliothèque de modèles n8n](#) et les réadapter, ou simplement vous amuser pour faire quelques tests.

Fin du tutoriel