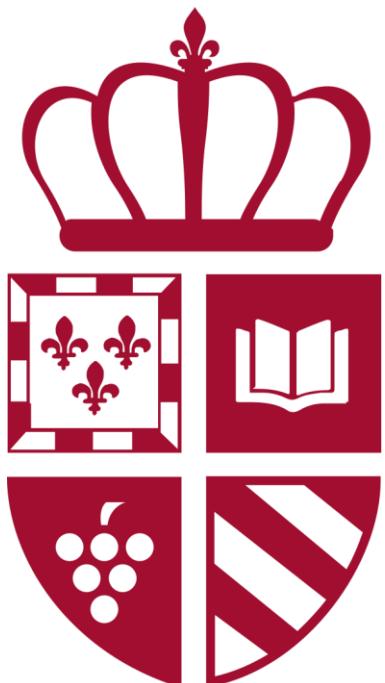


Du 10/03/25 au 11/04/25

Guillaume Emorine



**UNIVERSITÉ
BOURGOGNE
EUROPE**

Guide à destination des utilisateurs
concernant l'outil d'automatisation n8n,
son installation, et son utilisation sur
Windows

Les outils mis en œuvre

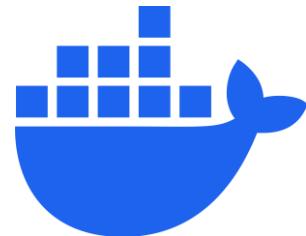
Voici les programmes et autres outils qui sont mis en œuvre lors d'une installation de n8n, qui sont bon à savoir.

Node.js est un environnement d'exécution pour le langage JavaScript, disponible sur Windows, macOS et Linux. Pour Windows, on peut directement récupérer la dernière version avec support étendu (LTS) sur leur [site](#) avant de l'installer. Si l'installateur .msi a été choisi, il installe aussi npm, qui est nécessaire pour installer et exécuter n8n.



npm est le gestionnaire de paquets par défaut pour Node.js, et il est inclus avec Node.js par défaut depuis 2011. C'est avec npm qu'on installe n8n, bien qu'on puisse éventuellement l'installer via Docker.

Docker est un puissant outil qui permet de mettre des conteneurs en place, avec leurs propres systèmes d'exploitation, ainsi que certaines applications prêtes à l'emploi. Il existe une image Docker pour n8n, et Docker possède ses avantages : en installant n8n via Docker, l'instance n8n tournera de manière totalement isolée par rapport à la machine physique.



Enfin, **n8n**, prononcé *n-eight-n*, est un outil d'automatisation de tâches dit “low-code”, qui va prioriser une approche simple comme le glisser-déplacer, tout en offrant une interface visuellement plaisante à son utilisateur.



Installer n8n

Une fois Node.js et npm installés, on peut désormais installer n8n de deux façons différentes, bien qu'on puisse techniquement l'utiliser sur leurs services cloud.

Via npm :

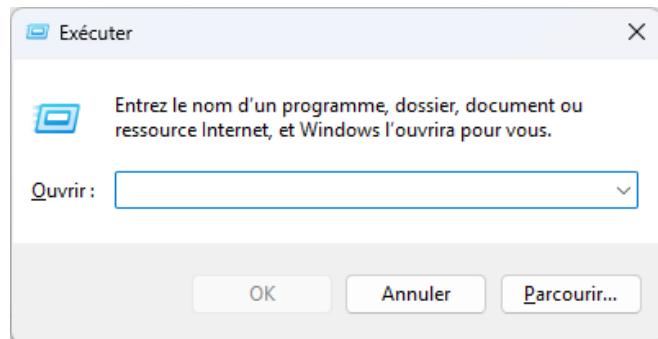
La première méthode pour exécuter n8n localement (et la moins complexe) est de l'installer globalement via **npm**.

Afin d'installer n8n avec npm, il faut se procurer npm. Pour cela, on a deux méthodes : récupérer le .msi de Node.js qui va installer npm automatiquement, ou le récupérer manuellement en exécutant ces commandes dans PowerShell, bien que ce soit moins pratique :

```
winget install Schniz.fnm
```

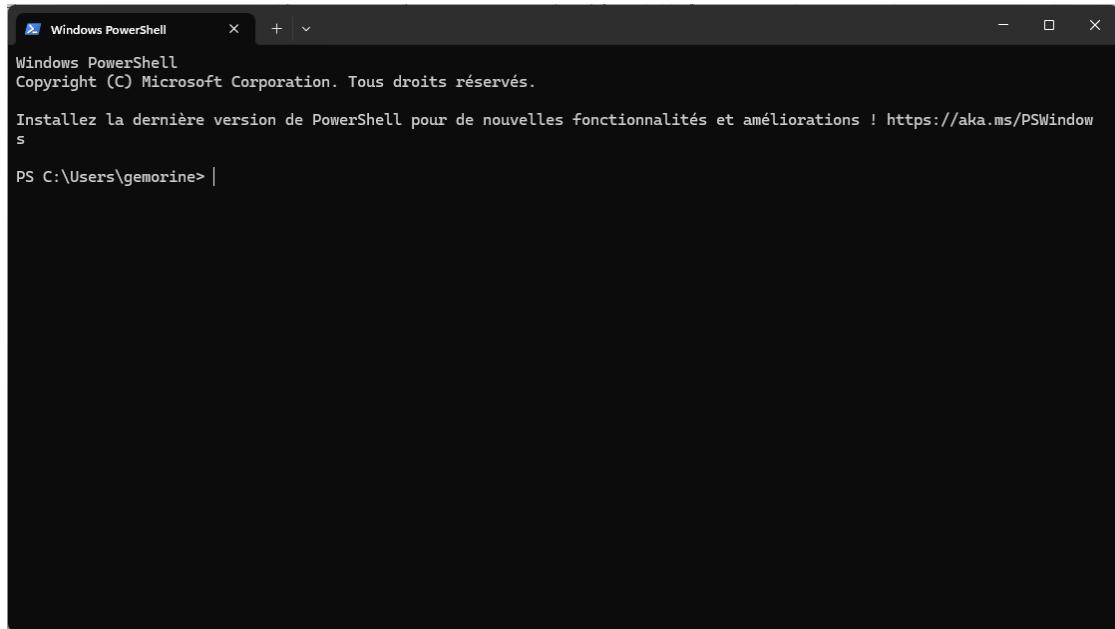
```
fpm install 22
```

Pour ouvrir PowerShell, il faut juste appuyer sur les touches **Windows** et **R** en même temps, puis taper “powershell” sans guillemets dans la fenêtre qui va s'ouvrir. Si c'est fait correctement, on devrait d'abord avoir une fenêtre de ce type :



Avec cette fenêtre d'ouverte, on peut ensuite y écrire “powershell” et appuyer sur Entrée (ou cliquer sur OK), ce qui devrait lancer PowerShell (un interpréteur de

commandes) automatiquement, nous laissant devant cette fenêtre :



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the standard PowerShell opening text: "Windows PowerShell", "Copyright (C) Microsoft Corporation. Tous droits réservés.", and "Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows". Below this, it shows the command "PS C:\Users\gemorine> |" followed by a large black rectangular redaction box.

Pas de panique si la vôtre est différente, car il est probable que Windows ait décidé de lancer PowerShell en dehors de leur nouvelle application, le Terminal Windows. Malgré tout, vous devriez pouvoir voir une fenêtre à fond bleu, avec un PS précédant la ligne où vous pouvez entrer des commandes.

Une fois dans PowerShell, on peut lancer la commande suivante afin d'installer n8n :
`npm install n8n -g`

Il est possible qu'un message d'erreur apparaisse par rapport à une "ExecutionPolicy", et dans ce cas, il faut lancer la commande suivante :

`Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser`

Ensuite, on laisse n8n s'installer, et après qu'il a terminé, on peut exécuter la commande suivante pour le lancer :

`n8n start`

Après avoir lancé cette commande, on attend qu'un message spécifique apparaisse :

```
Editor is now accessible via:  
http://localhost:5678  
  
Press "o" to open in Browser.
```

Quand ce message apparaît, on peut appuyer sur la touche "O" et un onglet devrait s'ouvrir dans le navigateur défini par défaut.

On se retrouve ensuite devant une interface de création de compte. Une fois le compte créé en renseignant une adresse mail, son nom, son prénom, et un mot de passe, on

entre enfin dans l'interface principale de n8n. Il faut noter que le mot de passe ne **PEUT PAS** encore être réinitialisé, donc gardez le précieusement ! Une autre chose à noter est que le fuseau horaire de n8n est défini sur “Amérique/New York” par défaut. Ce paramètre ne se change pas globalement, mais on peut le modifier pour chaque workflow dans les paramètres de celui-ci, accessibles en cliquant sur “...” en haut à droite de l'interface.

Via Docker :

Afin d'installer n8n via Docker, il faut tout d'abord installer Docker. Pour les ordinateurs sous macOS ou Windows, il existe une application nommée **Docker Desktop** qui permet d'avoir une interface graphique pour modifier et utiliser Docker. Il faut noter que l'installation de Docker Desktop demande des droits d'administration. Lorsque Docker Desktop a fini de s'installer, il faut redémarrer sa machine, et on peut désormais lancer l'application.

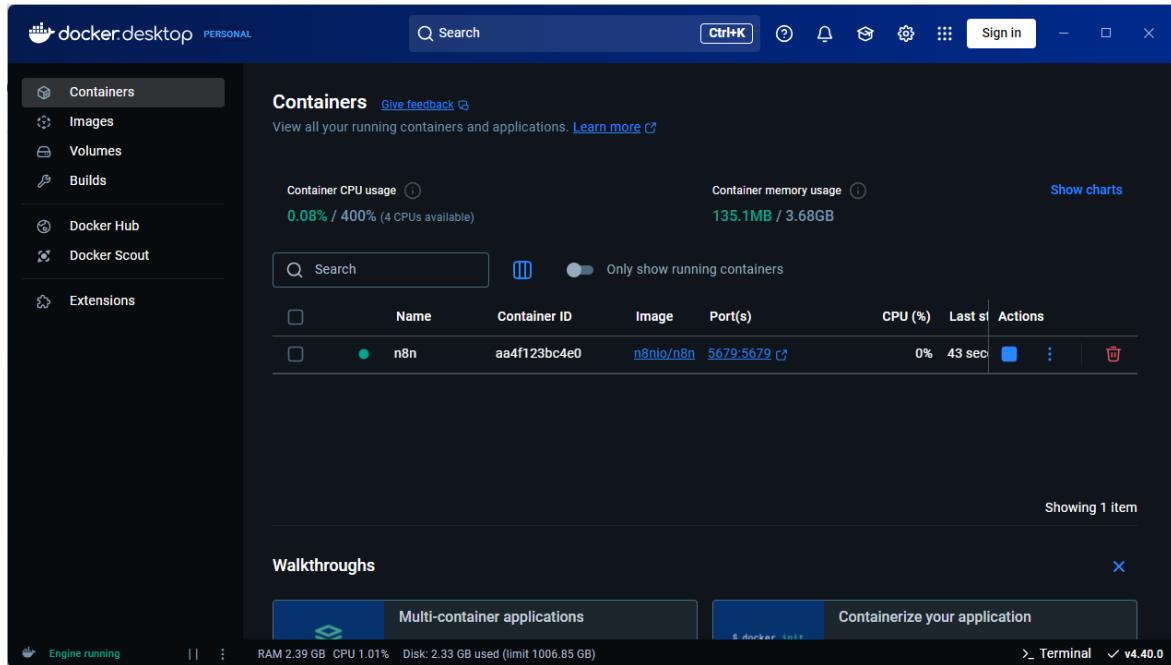
Une fois Docker Desktop (et le sous-système Linux pour Windows, ou **WSL**), on peut lancer une fenêtre PowerShell en appuyant sur les touches **Windows** et **R** simultanément, et en tapant “powershell”.

Ensuite, on peut lancer un conteneur Docker avec n8n en lançant les commandes suivantes :

```
docker volume create n8n_data
```

```
docker run -it --rm --name n8n -p 5678:5678 -e  
GENERIC_TIMEZONE="Europe/Paris" -v n8n_data:/home/node/.n8n  
docker.n8n.io/n8nio/n8n
```

Une fois ces commandes exécutées, on peut voir que notre image/conteneur Docker tourne, et n8n est désormais accessible quand on regarde sa fenêtre PowerShell.



The screenshot shows the Docker Desktop application interface. On the left, a sidebar menu includes 'Containers' (selected), 'Images', 'Volumes', 'Builds', 'Docker Hub', 'Docker Scout', and 'Extensions'. The main area is titled 'Containers' with a search bar and a toggle for 'Only show running containers'. It displays a table with one row:

	Name	Container ID	Image	Port(s)	CPU (%)	Last seen	Actions
<input type="checkbox"/>	n8n	aa4f123bc4e0	n8nio/n8n	5679:5679	0%	43 sec	

Below the table, it says 'Showing 1 item'. At the bottom, there's a 'Walkthroughs' section with links to 'Multi-container applications' and 'Containerize your application'. The status bar at the bottom shows 'Engine running', system resources (RAM 2.39 GB, CPU 1.01%), and Docker version v4.40.0.

Bien sûr, quand on accède à n8n pour la première fois, il faudra créer un compte pour pouvoir y accéder. Il faut noter que le mot de passe ne **PEUT PAS** encore être réinitialisé, donc gardez le précieusement !

L'interface utilisateur

The screenshot shows the n8n Overview page. At the top, there's a search bar and a sorting dropdown set to "Sort by last updated". Below this, a list of workflows is displayed:

- Workflow n8n Veyon
- Workflow avec LLM
- Workflow veille technico
- Workflow n8n Anaconda
- Workflow n8n Qgis
- Workflow n8n Fiji
- Recherche et obtention auto des Maj LM Studio pour WAPT
- Workflow n8n GeoDa
- Workflow n8n Jamovi
- Workflow n8n Agent GLPI

Each workflow entry includes a "Personal" and "Inactive" toggle switch and a three-dot menu icon.

Voici la page d'accueil de n8n. Ici, on peut voir plusieurs onglets (quand on a déjà créé un “workflow”), ainsi qu'un bouton rouge pour en créer un.

Quand un workflow est créé, on se retrouve avec l'interface suivante :

The screenshot shows the n8n Editor page. At the top, there's a toolbar with icons for "My workflow" (+ Add tag), "Editor" (selected), "Executions", "Inactive" (switch), "Share", and "Save". Below the toolbar is a large workspace area with a central dashed box containing a plus sign (+). Below this box, the text "Add first step..." is visible. At the bottom of the workspace are four small icons: a grid, a magnifying glass, a search icon, and a refresh icon.

En cliquant sur le +, on peut ajouter le déclencheur, qui va se charger de démarrer l'ensemble de la suite des tâches. On peut faire en sorte que le déclencheur soit hebdomadaire, quotidien, ou même qu'il soit déclenché manuellement quand on clique sur le bouton “Test workflow”.

Ensuite, une fois le déclencheur placé, on peut rajouter une tâche/une étape en cliquant sur le “+” en haut à droite ou sur le “+” relié à une étape existante.

What happens next?

Search nodes...

-  **Advanced AI**
Build autonomous agents, summarize or search documents, etc. →
-  **Action in an app**
Do something in an app or service like Google Sheets, Telegram or Notion →
-  **Data transformation**
Manipulate, filter or convert data →
-  **Flow**
Branch, merge or loop the flow, etc. →
-  **Core**
Run code, make HTTP requests, set webhooks, etc. →
-  **Human in the loop**
Wait for approval or human input before continuing →
-  **Add another trigger**
Triggers start your workflow. Workflows can have multiple triggers. →

Voici la liste qui apparaît quand on clique sur le "+" : celle-ci regroupe plusieurs possibilités, regroupées par catégories. On peut manipuler les données qui proviennent d'étapes précédentes, effectuer une requête HTTP vers un site, exécuter une commande sur sa machine, ou même interagir avec des applications grâce à des modules spécialisés fait pour ça.

Par exemple, en fournissant à n8n de quoi se connecter à un serveur de mails, je peux faire en sorte qu'il m'envoie un mail tous les jours avec le message de mon choix. On peut aussi faire en sorte que le "workflow" vérifie ma disponibilité pour certaines dates, tant que j'ai fourni les éléments nécessaires à l'étape qui le fait. n8n possède donc des modules pour interagir directement avec des applications et sites web afin d'y effectuer des actions spécifiques, comme ajouter des

lignes à un tableau Excel sur Microsoft 365.

Avec n8n, on peut donc automatiser autant de choses qu'on le souhaite, tant qu'on arrive à déterminer ce qu'on veut faire, et qu'on sait décomposer en plusieurs étapes précises. Par exemple, un tutoriel (en anglais) pour apprendre les bases d'un workflow peut se trouver [ici](#), sur le site officiel pour la documentation de n8n.

Exemple de workflow

Ce workflow va permettre de faire une tâche très basique : il va récupérer la date et l'heure au moment de son exécution afin de l'écrire dans un fichier texte. Tout d'abord, il faut ajouter un déclencheur, et le tout premier dans la liste est celui qu'on veut :

The screenshot shows a dark-themed user interface for n8n. At the top, the title "What triggers this workflow?" is displayed in white. Below it, a subtitle "A trigger is a step that starts your workflow" is shown. A search bar with the placeholder "Search nodes..." is present. A red vertical bar on the left indicates the current section. On the right, there is a card with a blue arrow icon labeled "Trigger manually" and the text "Runs the flow on clicking a button in n8n. Good for getting started quickly".

Ensuite, on peut modifier le fuseau horaire du workflow avant d'ajouter une autre étape reliée au déclencheur en cliquant sur le "+". L'étape suivante qu'il nous faut est dans la section "Data transformation", plus précisément l'outil "Edit fields (Set)".

Ensuite, cliquez sur le bouton "Execute previous nodes" à gauche, et déroulez la liste "Variables and context" pour retrouver la date et l'heure.

On peut ensuite cliquer deux fois sur la zone avec "Drag input fields here or Add field", et glissez-déposez la variable \$now dans le champ du bas. On peut aussi donner un nom pour que les données qui sortiront aient un nom spécifique plutôt qu'un chiffre

The screenshot shows the 'Edit Fields' interface with 'Parameters' selected. It displays four field configurations:

- name**: A String, value = `{{ $now }}`, resulting in [DateTime: 2025-04-09T10:03:39.630+02:00]
- name**: A String, value = `{{ $now }}`, resulting in [DateTime: 2025-04-09T10:03:49.442+02:00]
- date**: A String, value = `{{ $now.format('DDDD') }}`, resulting in mercredi 9 avril 2025
- heure**: A String, value = `{{ $now.format('HH:MM') }}`, resulting in 10:04

A red box highlights the first 'name' field's configuration.

comme 0 ou 1.

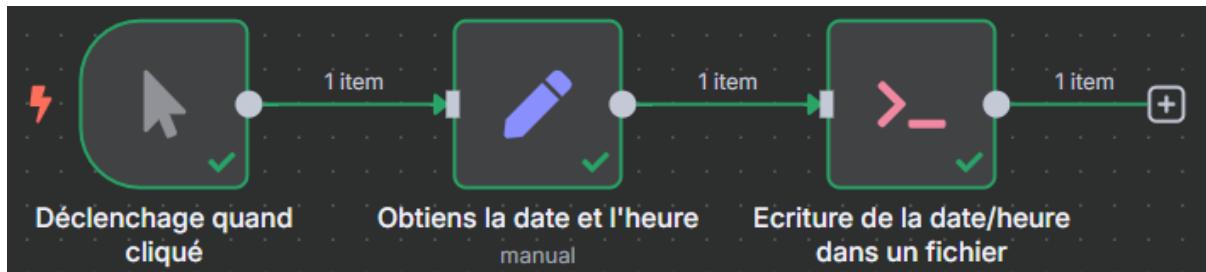
Pour pouvoir reformater l'heure et la date, on peut utiliser une fonction, `format()`, pour pouvoir obtenir la date d'un côté et l'heure de l'autre.

Avec ces instructions de formatage, on peut voir qu'on se retrouve avec la date détaillée et l'heure. On peut donc tester notre étape, pour ensuite retourner vers le “canvas” afin d’ajouter la dernière étape : il faut choisir l’étape “Execute command” dans la section “Core”. Dedans, on peut mettre la commande suivante :

```
echo {{ $json.date }}{{ $json.heure }}>test.txt
```

Enfin, on peut tester notre workflow, et sur Windows, on retrouvera notre document test.txt dans le répertoire C:\Utilisateurs\<nom d'utilisateur>.

Le résultat final, après avoir renommé nos étapes, ressemble à ça :



Même s'il n'y a pas beaucoup d'utilité à ce workflow, il montre quand même certaines bases comme la manipulation des données et l'exécution d'une commande en local sur le poste.

Avantages et inconvénients

n8n possède des avantages et des inconvénients comme tout outil.

Avec n8n, on peut automatiser un nombre incalculable de tâches, mais il faut savoir la décomposer étape par étape, tout en sachant quoi faire et quelles “nodes” (étapes) employer. Un inconvénient peut être le manque de traductions en français pour le moment, mais cela peut éventuellement changer.

Sachant qu'on peut héberger n8n en local, on peut donc automatiser des tâches chez soi sans aucune dépendance aux serveurs externes de n8n. De plus, on peut utiliser n8n gratuitement, mais on perd l'accès à certaines fonctionnalités qui peuvent éventuellement servir en entreprise. Utiliser n8n via docker offre la possibilité de définir un fuseau horaire de manière globale, un avantage par rapport à la méthode via npm.

Parfois, il faut savoir utiliser des éléments de JavaScript afin d'obtenir les résultats escomptés, ce qui demande donc un minimum de savoir, et de connaissance sur la logique des langages de programmation.

En somme, n8n est un très bon outil qu'on peut utiliser chez soi afin d'automatiser de multiples tâches comme la recherche de mises à jour logicielles, par exemple.