

Nodered

- Créer un container Nodered sous forme de stack

Créer un container Nodered sous forme de stack

Un container Nodered peut se créer de deux manières :

- directement en tant que Container
- sous forme de stack Docker, au sens Portainer.

Une stack sert à configurer un ou plusieurs containers liés entre eux, sur la base d'un fichier docker-compose.

Pour en savoir plus sur les fichiers docker-compose : <https://datascientest.com/docker-compose-tout-savoir>

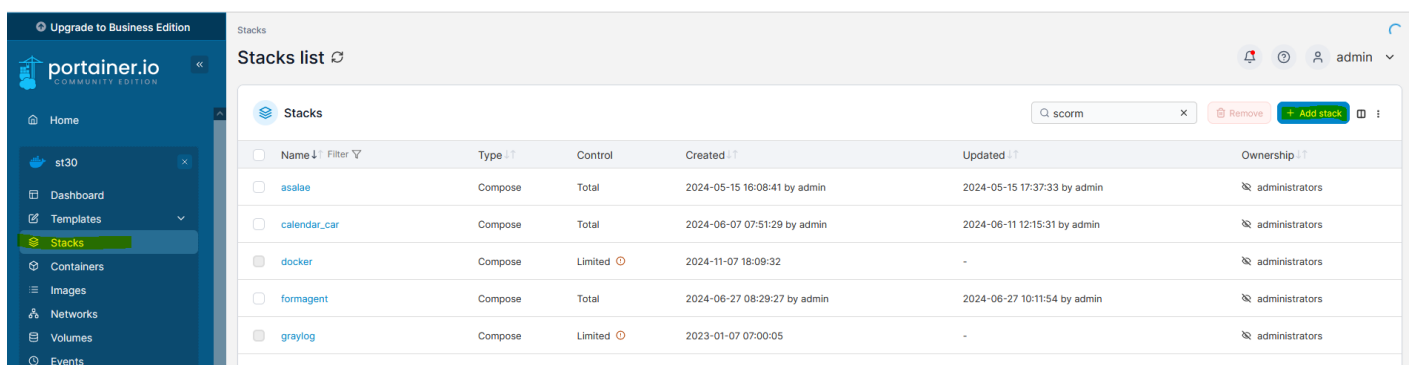
Dans le cas présent, nous automatisons la configuration d'un container Nodered :

- création d'un réseau propre
- définition du port d'accès au container
- labels traefik pour avoir un accès en https
- création d'un volume et lien avec le container

Tutoriel pour créer une nouvelle stack : <https://docs.portainer.io/user/docker/stacks/add>

Pas à pas dans le cas de Nodered :

1 - Sélectionner "Stacks" puis "Add stack"



2 - Choisir un nom pour la stack et compléter le champ "Name"

Name

This stack will be deployed using `docker compose`.

Build method

Web editor Upload Repository Custom template

Web editor Ctrl+F for search

You can get more information about Compose file format in the [official documentation](#).

Define or paste the content of your docker compose file here Copy to clipboard

```
1
```

Copier la configuration ci-dessous dans le block "Webeditor"

```
version: "3.8"

services:
  nodered-test:
    image: registry.vincennes.fr/images-docker/nodered-custom:latest
    restart: always
    env_file:
      - stack.env
    labels:
      - traefik.http.routers.nodered-${app_name}.rule=Host(`${host}`) && PathPrefix(`/nodered-${app_name}/`)
      - traefik.http.middlewares.nodered-${app_name}_middleware.striprefix.prefixes=/nodered-${app_name}/
      - traefik.enable=true
      - traefik.http.routers.nodered-${app_name}.tls=true
      - traefik.http.services.nodered-${app_name}.loadbalancer.server.port=1880
      - traefik.http.routers.nodered-${app_name}.middlewares=nodered-${app_name}_middleware@docker
    ports:
      - ${app_port}:1880
```

```
volumes:  
  - nodered_data: /data
```

```
networks:  
  - reseau
```

```
volumes:  
  nodered_data:  
networks:  
  reseau:  
  driver: bridge
```

Changer "nodered-test" par le nom souhaité.

3 - Dans la partie "Environment variables", cliquer sur "Advanced mode"

Environment variables

You may use [environment variables in your compose file](#). The environment variable values set below will be used as substitutions in the compose file. Note that you may also reference a `stack.env` file in your compose file. A `stack.env` file contains the environment variables and their values (e.g. `TAG=v1.5`).

stack.env file operation

When deploying via **Repository**, the `stack.env` file must already reside in the Git repo.
When deploying via **Web editor**, **Upload** or **Custom template deployment**, the `stack.env` file is auto created from what you set below.

Advanced mode

Switch to advanced mode to copy & paste multiple variables

+ Add an environment variable

Load variables from .env file

Dans la fenêtre qui apparaît :

Environment variables

You may use [environment variables in your compose file](#). The environment variable values set below will be used as substitutions in the compose file. Note that you may also reference a `stack.env` file in your compose file. A `stack.env` file contains the environment variables and their values (e.g. `TAG=v1.5`).

stack.env file operation

When deploying via **Repository**, the `stack.env` file must already reside in the Git repo.
When deploying via **Web editor**, **Upload** or **Custom template deployment**, the `stack.env` file is auto created from what you set below.

Simple mode

Switch to simple mode to define variables line by line, or load from .env file

e.g. `key=value`

Copy to clipboard

1

Copier les valeurs suivantes :

```
app_name=test  
app_port=1835  
host=st30.vincennes.fr
```

app_name => nom de l'application liée à Nodered

app_port => port utilisé par Nodered

host => serveur sur lequel le nodered est installé.

4 - Cliquer sur "Deploy Stack"

5 - Ajouter le réseau créé dans la liste des réseau Traefik

6 - Vérifier le bon fonctionnement du container

Le container apparaît dans la liste sous cette forme :



nodered-test-nodered-test-1 healthy nodered-test nodered/node-red:latest 2024-12-17 17:25:53 10.0.17.2 1835:1880

Il est accessible via le port choisi et en https sous la forme :

https://<host>/nodered-<app_name>/

<https://st30.vincennes.fr/nodered-test/>