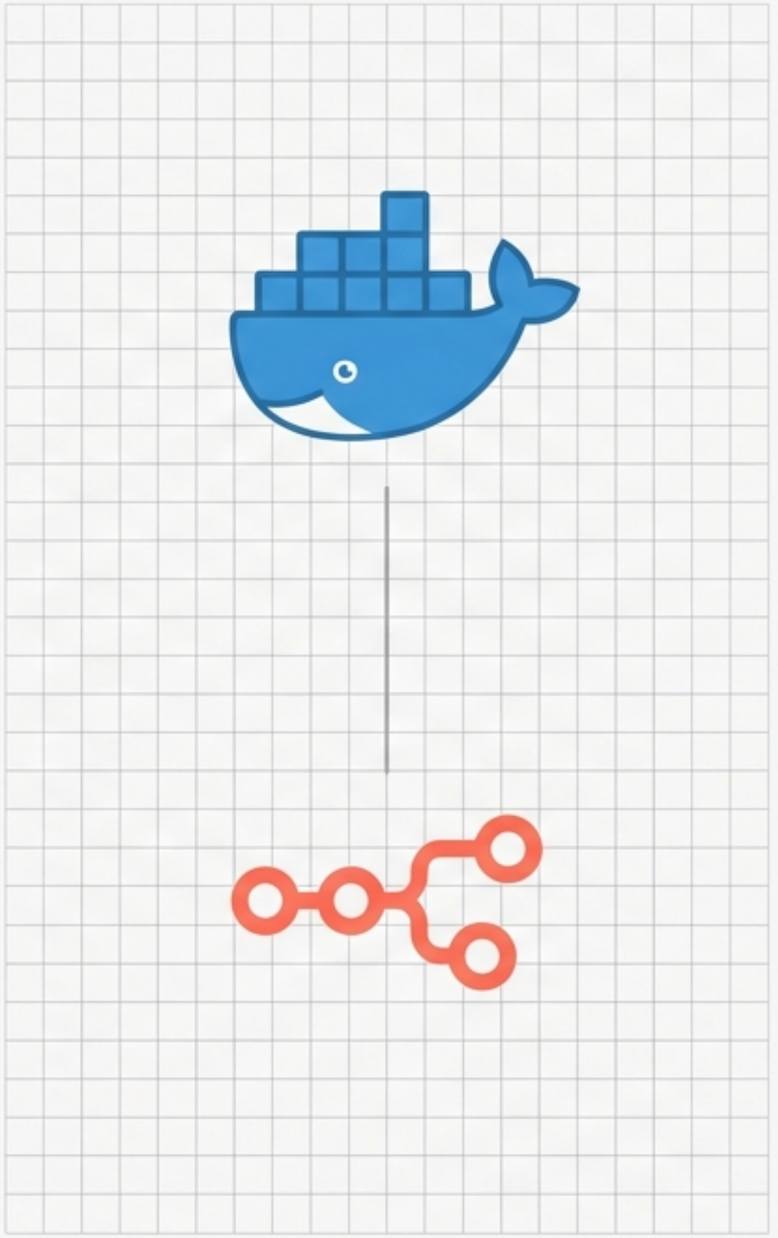
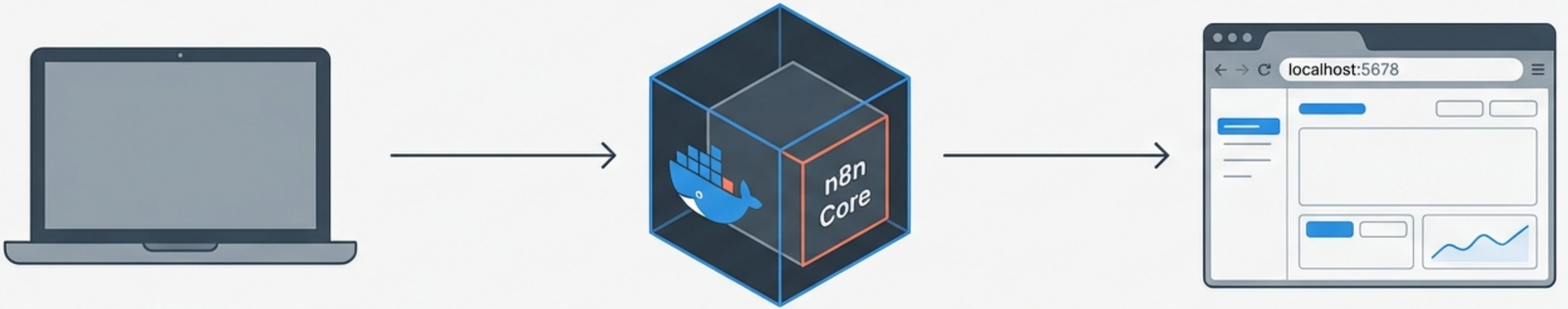
Local Automation Mastery: Deploying n8n with Docker Compose

A comprehensive guide to self-hosting your workflow automation engine.



GUIDE V1.0 // LOCALHOST DEPLOYMENT

The Objective: A Persistent, Self-Hosted Environment



Local Machine

Containerized Service

localhost:5678

Definition of Done

We are building a local instance of n8n that is robust, persistent, and accessible. By the end of this guide, you will have a containerized application running on your machine, ready for development without external dependencies.

Key Outcome

A running n8n instance accessible via `http://localhost:5678` with persistent data storage volumes.

System Prerequisites and Tooling

01. Docker Desktop

Required for Windows, macOS, or Linux. Handles the virtualization engine.

https://www.docker.com/products/docker-desktop/

02. Verification

Basic familiarity with Command Line Interface (CLI) is required.

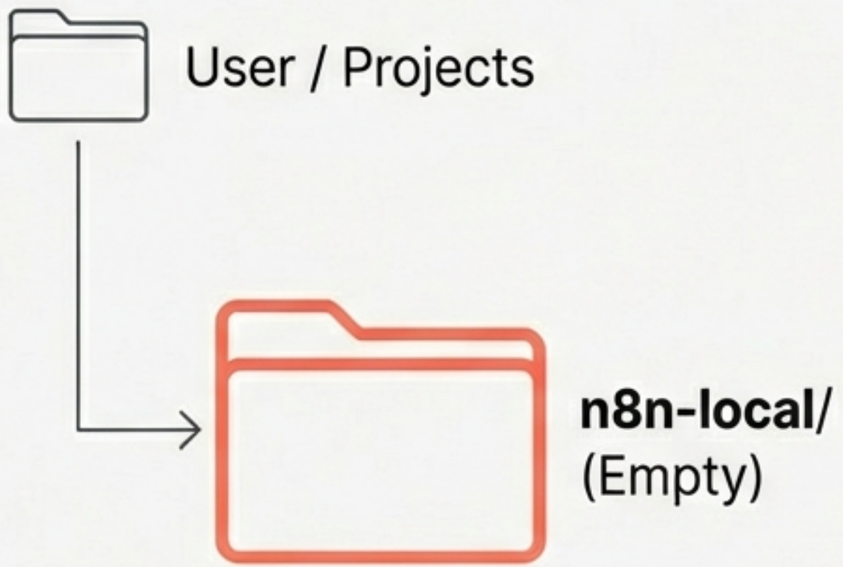
Ensure Docker is installed and running.

```
$ docker --version
Docker version 20.10.21, build baeda1f
```

Phase 1: Initializing the Workspace Directory

Create a dedicated directory to house your configuration and data. This ensures your project is contained and easy to manage.

```
$ mkdir n8n-local
$ cd n8n-local
```



The Architect’s Blueprint: docker-compose.yml

Create a file named `docker-compose.yml` inside your new directory. This file defines the infrastructure as code.

docker-compose.yml

```yaml
version: '3.8'
services:
  n8n:
    image: n8nio/n8n
    container_name: n8n
    ports:
      - '5678:5678'
    environment:
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=admin123
      - N8N_HOST=localhost
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - WEBHOOK_URL=http://localhost:5678/
      - GENERIC_TIMEZONE=Asia/Kuala_Lumpur
    volumes:
      - n8n_data:/home/node/.n8n
    restart: always
volumes:
  n8n_data:
```

Anatomy I: Defining the Service & Network Interface

The Container Image

We pull the official image '**n8nio/n8n**' and assign it the name '**n8n**' for easy reference.

Port Mapping

5678:5678 maps the Host Port (Left) to the Container Port (Right). This bridges the gap between your laptop and the Docker environment.

Reliability

Ensures the service automatically restarts if it crashes or the daemon reboots.

docker-compose.yml

```
    environment: {}
    volumes:
      - /usr/n8n/n8npratnic/volume
services:
  n8n:
    image: n8nio/n8n
    container_name: n8n
    ports:
      - '5678:5678'
    restart: always
```

Anatomy II: Configuring Authentication & Environment

Authentication

Basic Auth is set to TRUE. We define the user as 'admin' and password as 'admin123'. This secures the UI.

Connectivity

Variables like WEBHOOK_URL tell n8n how to construct its own URLs for external triggers.

Localization

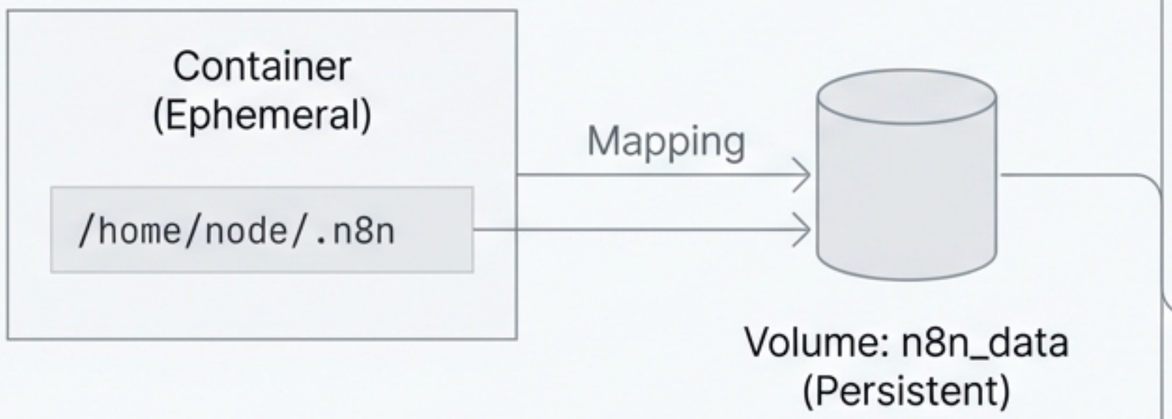
GENERIC_TIMEZONE defines the schedule for time-based nodes (e.g., cron jobs).

```
docker-compose.yml
version: '1.0'
services:
  n8n:
    image: n8n.compose
    container_name: n8n.compose
    ports:
      - nIT72.0.0.0:5678
    environment:
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=admin123
      - N8N_HOST=localhost
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - WEBHOOK_URL=http://localhost:5678/
      - GENERIC_TIMEZONE=Asia/Kuala_Lumpur
    volumes:
      - ./nome/n8n_admin/docker-Kuala_Lumpur
    restart: enabled
```

Anatomy III: Ensuring Data Persistence with Volumes

Why Volumes?

Docker containers are ephemeral—when they stop, their internal data vanishes. To prevent losing your workflows, we map a persistent volume.



docker-compose.yml

```yaml
version: '1.0'
services:
  n8n:
    image: n8nio/n8n
    container_name: n8n
    ports:
      - "5678:5678"
    environment:
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=admin123
      - N8N_HOST=localhost
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - WEBHOOK_URL=http://localhost:5678/
      - GENERIC_TIMEZONE=Asia/Kuala_Lumpur
    volumes:
      - n8n_data:/home/node/.n8n
    restart: always

volumes:
  n8n_data:
```

Phase 3: Executing the Deployment

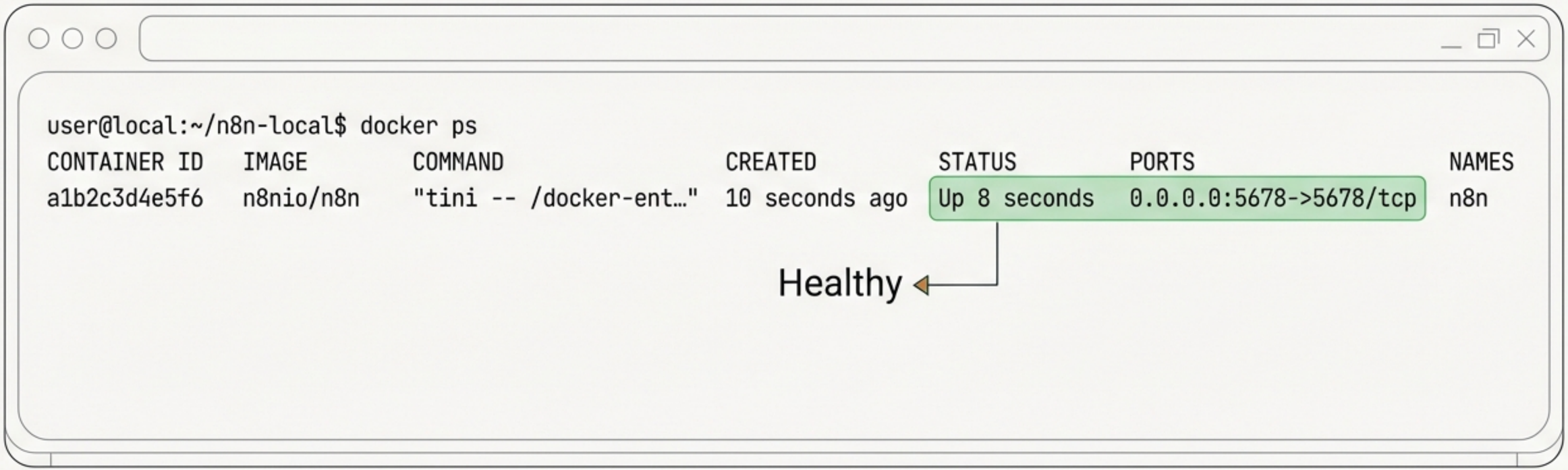
```
user@local:~/n8n-local$ docker compose up -d
```

Technical Insight

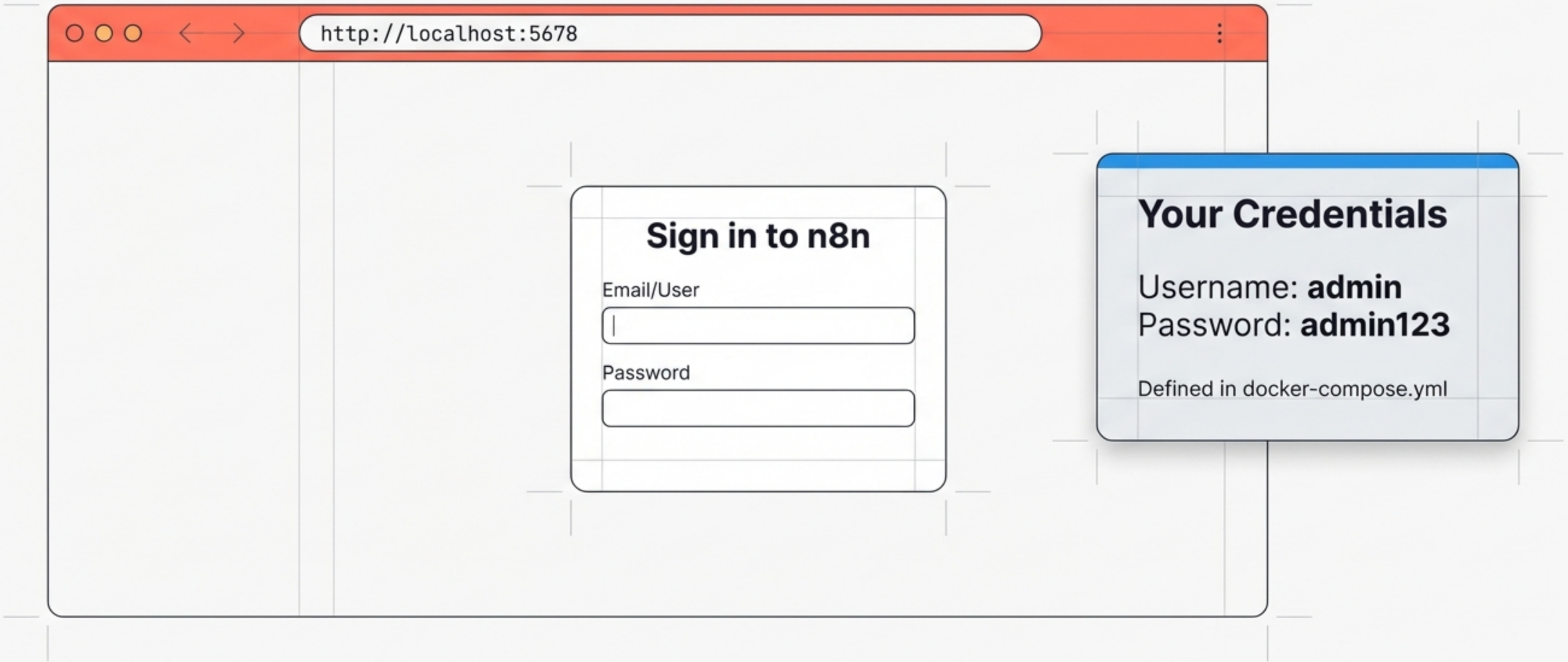
The '-d' flag stands for Detached Mode. This instructs Docker to run the container in the background, returning control of the terminal to you immediately.

Verification: Confirming Container Status

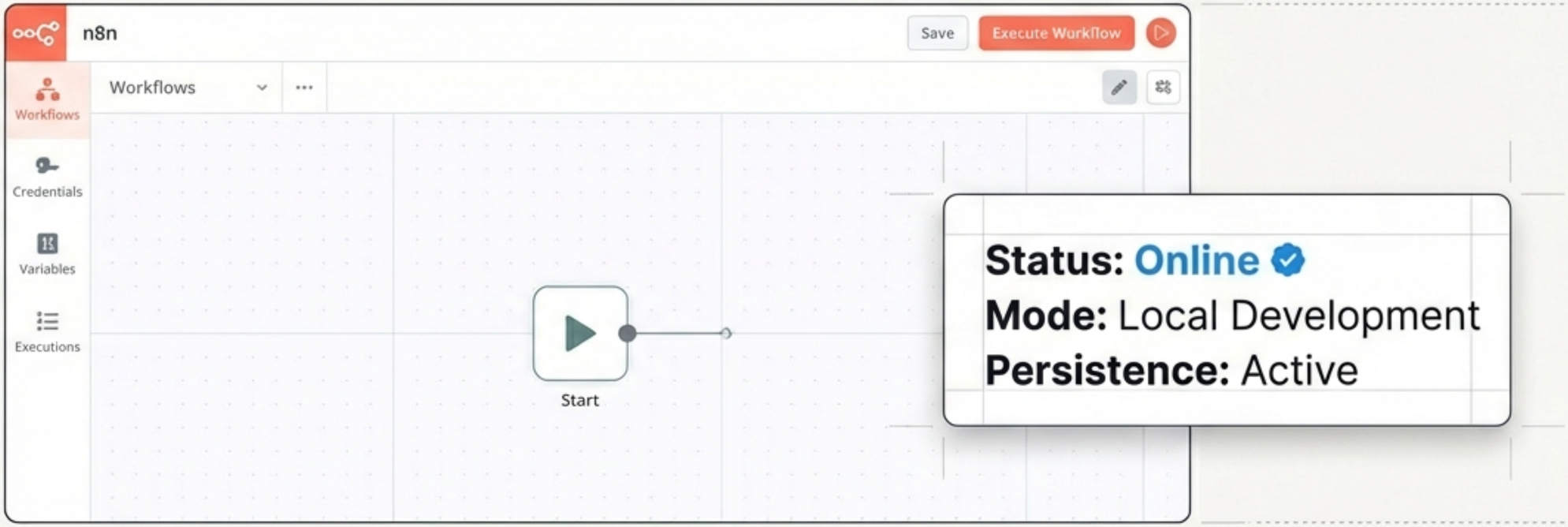
Trust, but verify. Use the process status command to ensure the container is healthy.



Accessing the n8n Interface



Mission Accomplished



n8n is now running locally on your machine. Your environment is persistent, secure, and ready for automation.

Start Building Workflows →