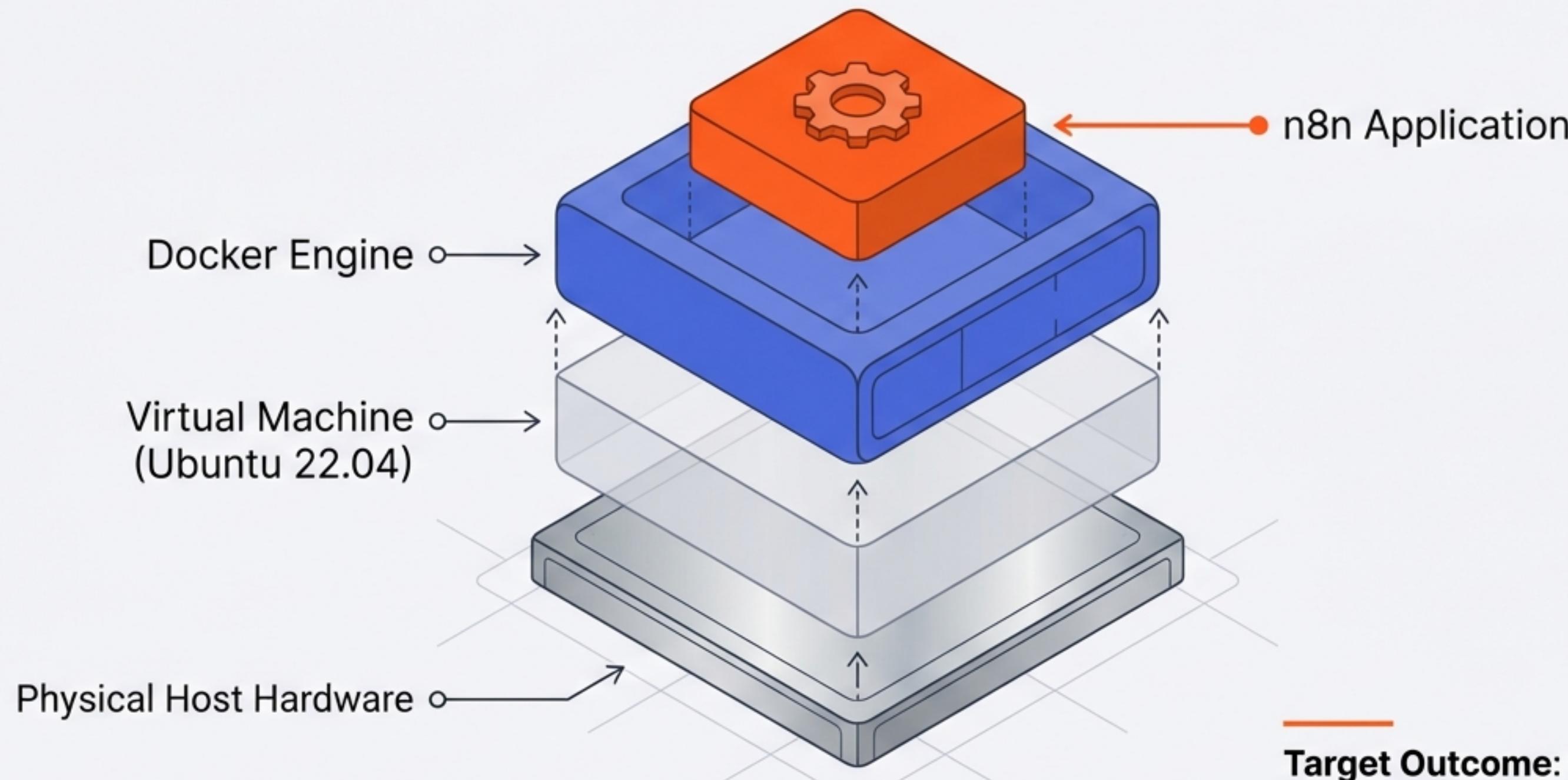


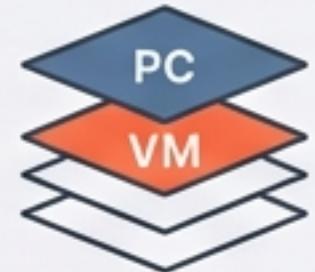
Deploying n8n via Docker on Ubuntu

A step-by-step execution guide for building a self-hosted automation stack.

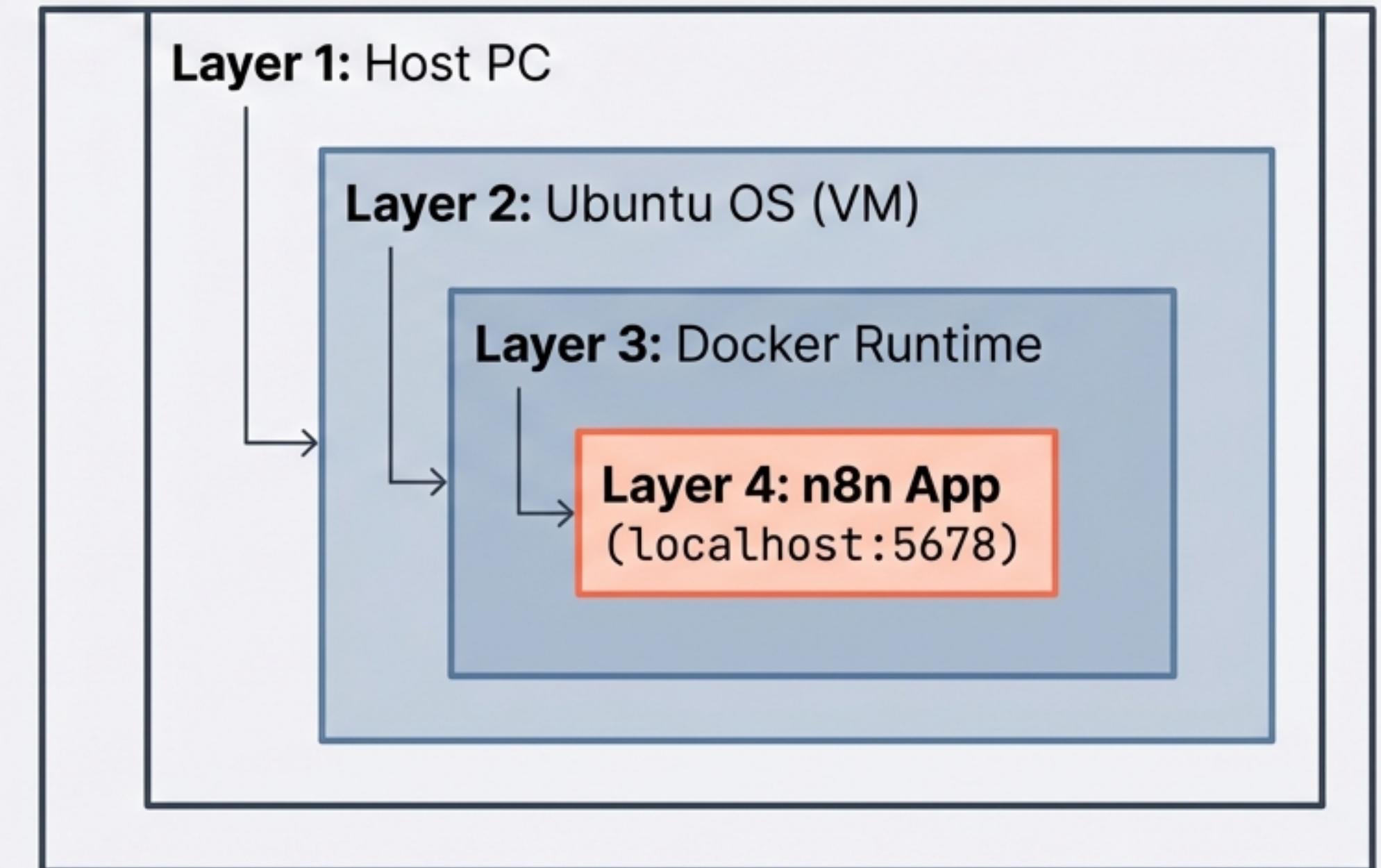


Target Outcome: A production-safe local setup on Port 5678

System Requirements & Architecture

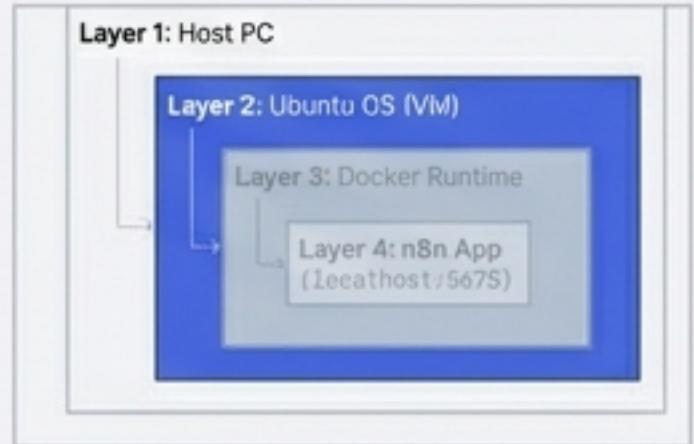


Minimum Specs	
RAM:	2 GB Minimum (4 GB Recommended)
CPU:	2 Cores
Storage:	30 GB Disk



Phase 1: Preparing the Foundation

Before installing the engine, update the OS package index to ensure system stability.

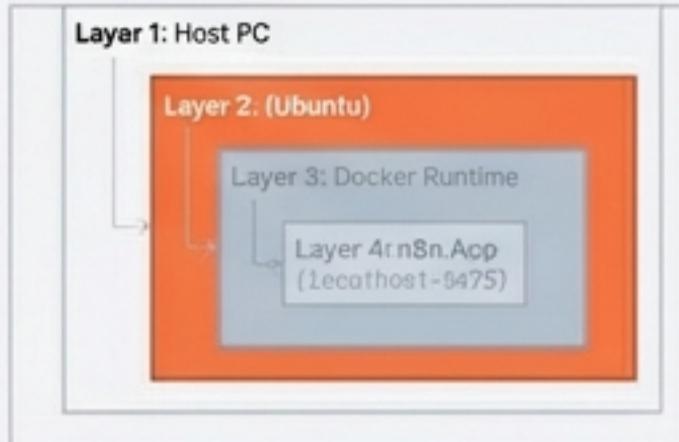


Ubuntu Terminal - Inter Regular

```
user@ubuntu:~$ sudo apt update && sudo apt upgrade -y  
[=====> ] 98%
```

Establishing Docker Trust

To install Docker securely, we must install dependencies and add the official Docker GPG key and repository to the system's keyring.



1 Install Dependencies

Ubuntu Terminal - Inter Regular

```
user@ubuntu:~$ sudo apt install -y ca-certificates  
curl gnupg lsb-release
```

2 Add GPG Key

Ubuntu Terminal - Inter Regular

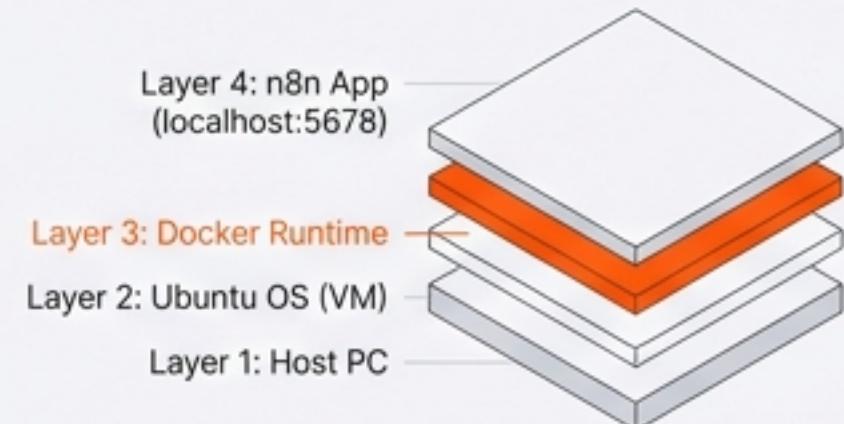
```
user@ubuntu:~$ sudo mkdir -p /etc/apt/keyrings  
user@ubuntu:~$ curl -fsSL https://download.docker.cc  
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

3 Add Repository

```
user@ubuntu:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/  
docker.list > /dev/null
```

Phase 2: Installing the Engine

Install the Docker Community Edition (CE), CLI, and the Compose plugin.

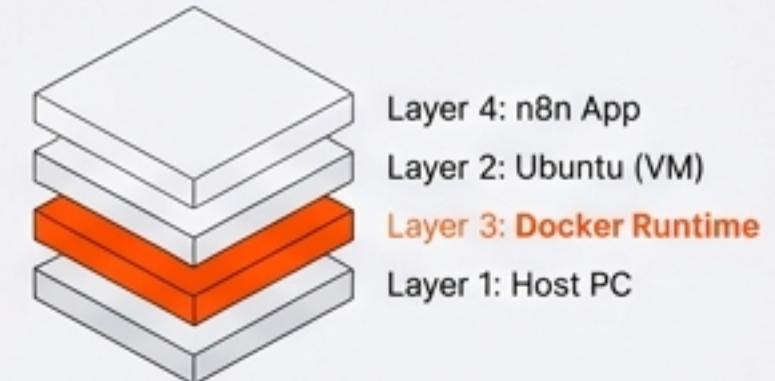


Ubuntu Terminal - Inter Regular

```
user@ubuntu:~$ sudo apt update
user@ubuntu:~$ sudo apt install -y docker-ce docker-ce-cli
containerd.io docker-compose-plugin ←
```

Crucial for
orchestration

Configuration: Enabling Docker User Permissions



⚠️ Pro Tip: By default, Docker requires root privileges. Perform this step to run Docker commands as a standard user without "sudo".

The Fix

Ubuntu Terminal - Inter Regular

```
sudo usermod -aG docker $USER  
newgrp docker
```

Hello from Docker!
This message shows that your installation appears
to be working correctly.

The Verification

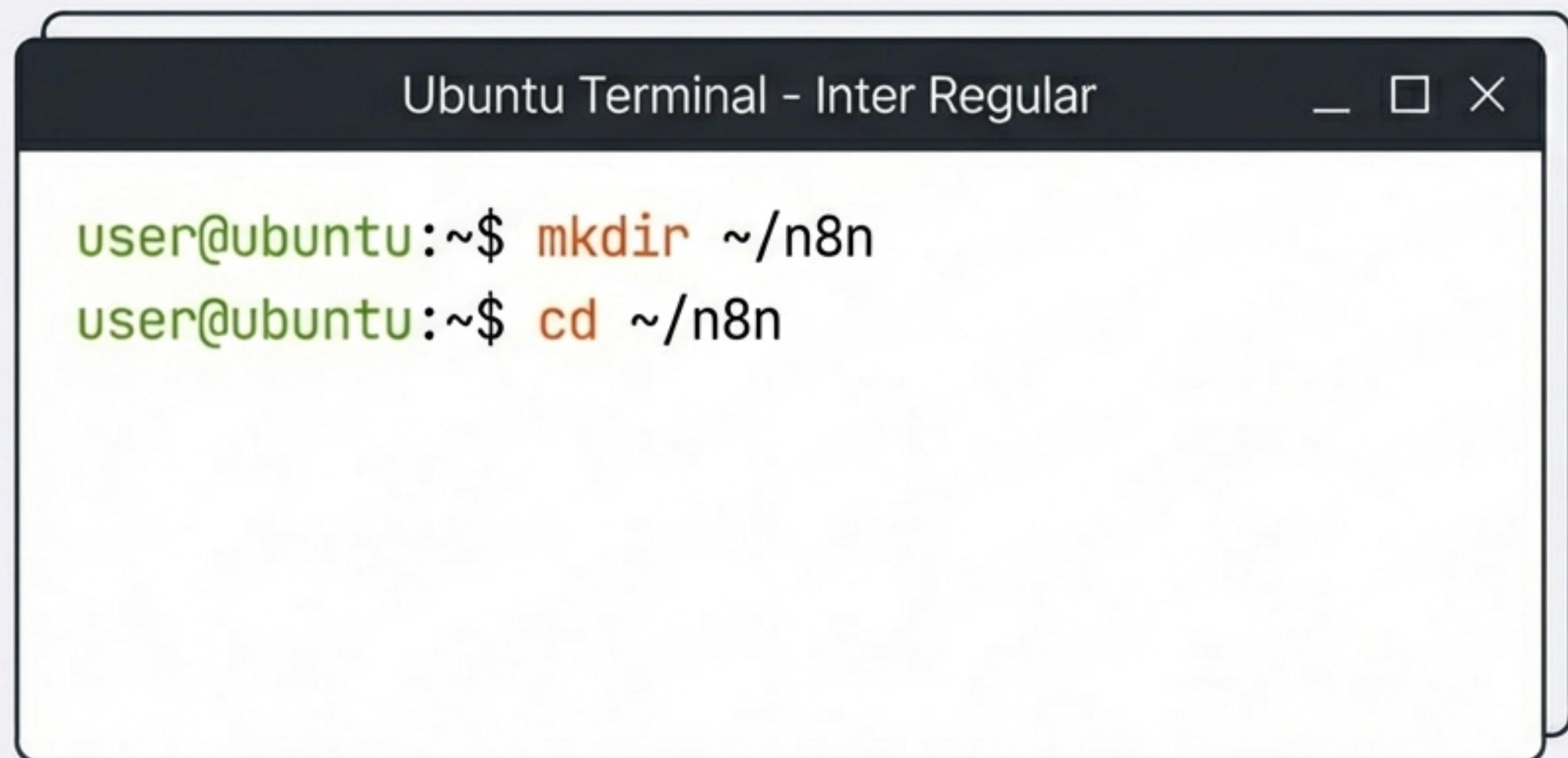
Ubuntu Terminal - Inter Regular

```
docker run hello-world
```

Hello from Docker!
This message shows that your installation appears
to be working correctly.

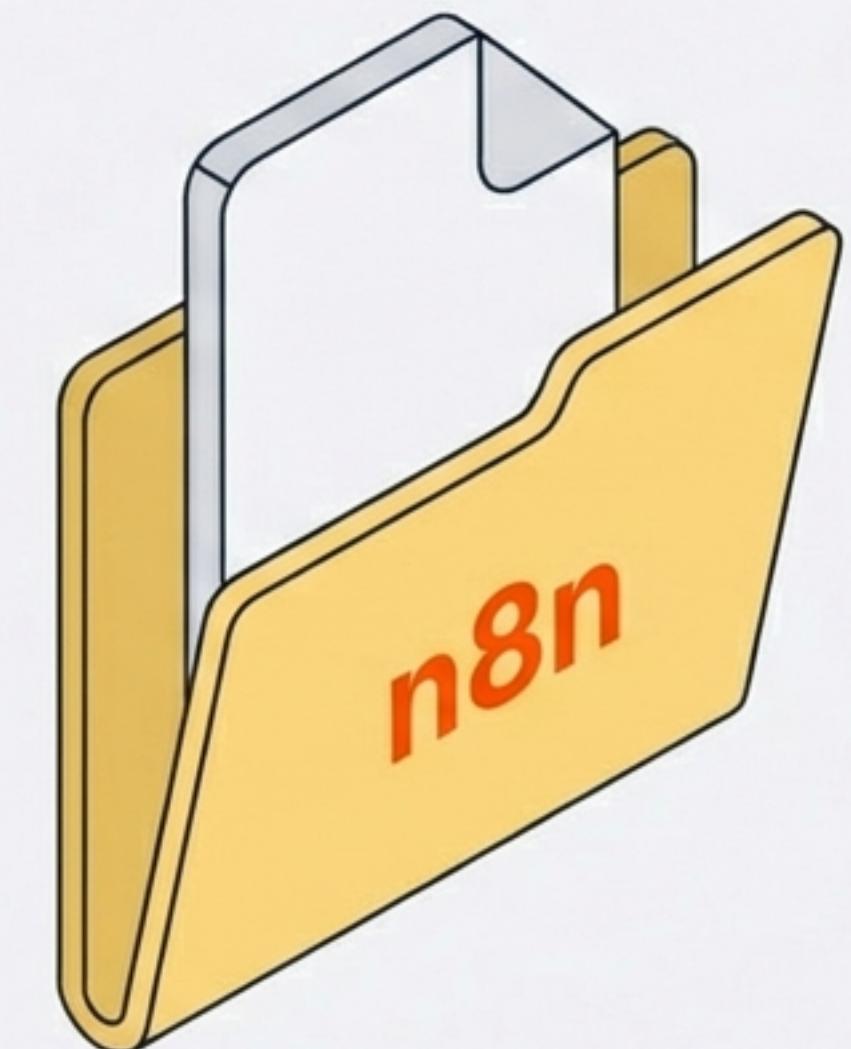
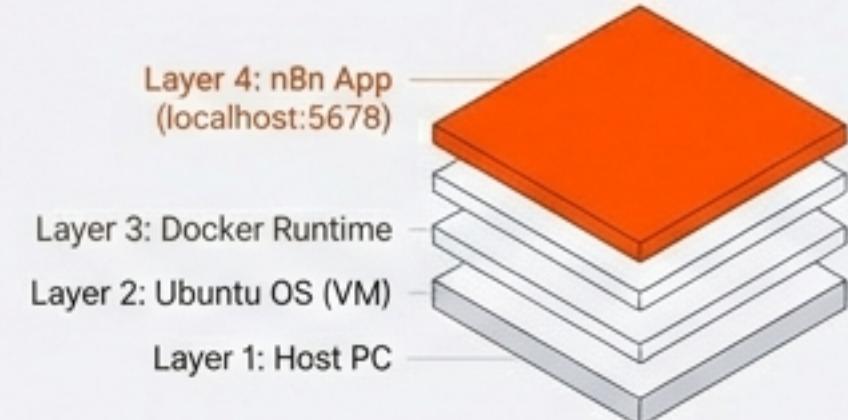
Phase 3: Project Structure

Create a dedicated directory for the n8n project. This keeps the configuration and persistent data organized.



An illustration of a terminal window titled "Ubuntu Terminal - Inter Regular". The window shows two commands entered at the prompt:

```
user@ubuntu:~$ mkdir ~/n8n
user@ubuntu:~$ cd ~/n8n
```

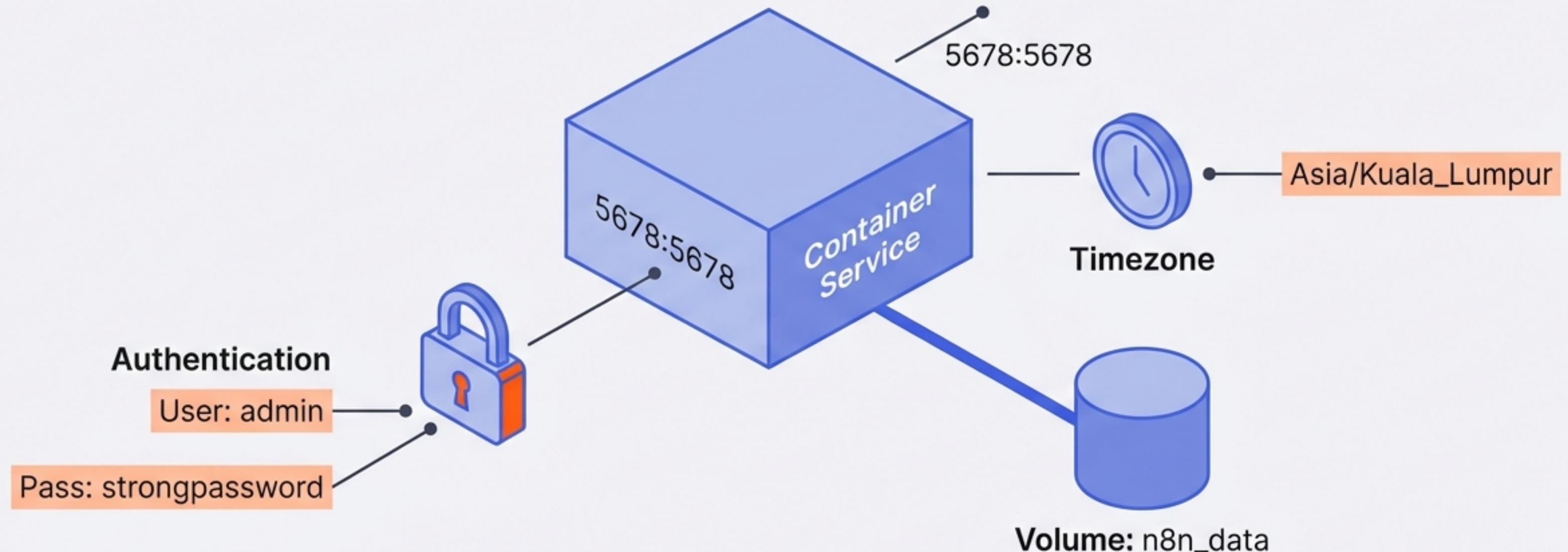
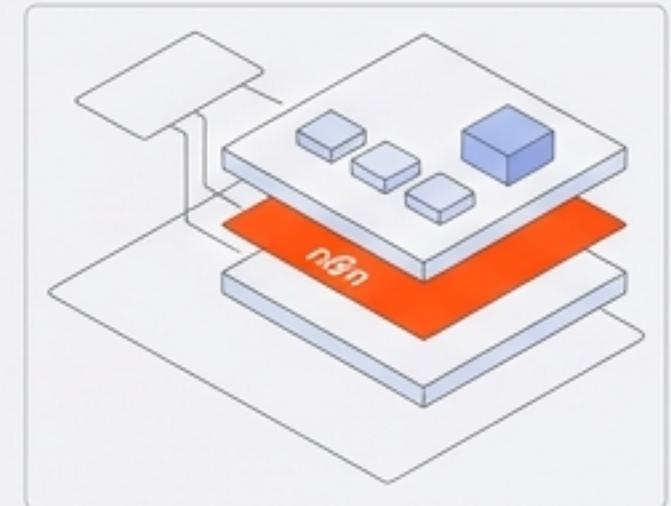


The Configuration Blueprint

We will use a docker-compose.yml file to define the service.

Prepare the file using the nano editor:

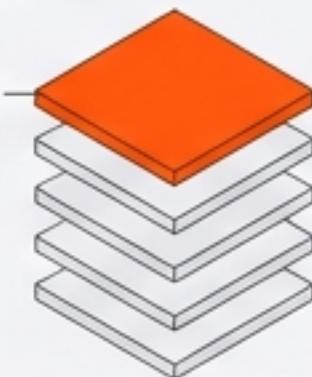
```
sudo nano docker-compose.yml
```



Architectural Overview

Highlight: n8n

Layer 4:
n8n App

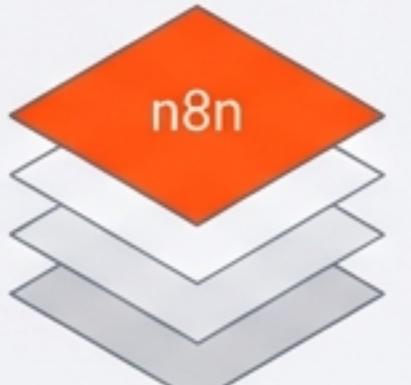


The docker-compose.yml File

Copy this exact configuration. Update the highlighted variables.

```
version: "3.8"
services:
  n8n:
    image: n8nio/n8n:latest
    container_name: n8n
    ports:
      - "5678:5678"
    environment:
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=strongpassword
      - 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: unless-stopped
volumes:
  n8n_data:
```

To Save: Press CTRL + O, Enter, then CTRL + X.



Phase 4: Launch Sequence

Initialize the container in 'detached' mode, allowing it to run in the background.

Layer 4: n8n App

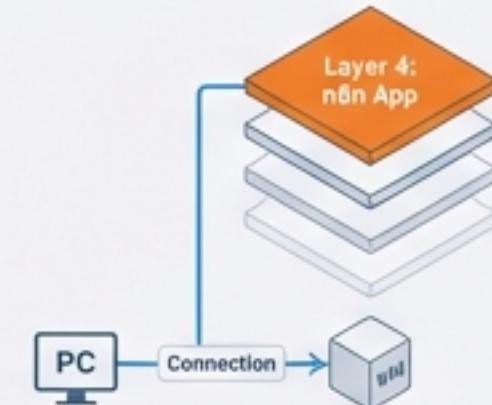
Launch

```
docker compose up -d
```

Verify Status

```
docker ps
```

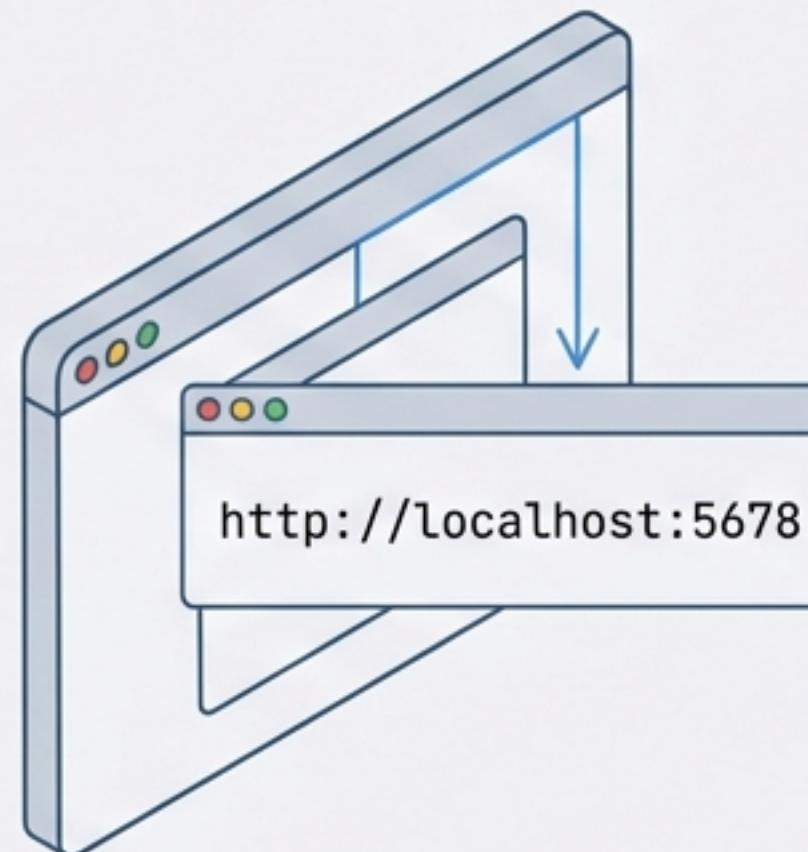
CONTAINER ID: a1b2c3d4e5 ... **STATUS:** Up 10 seconds ... **PORTS:** 0.0.0.0:5678->5678/tcp



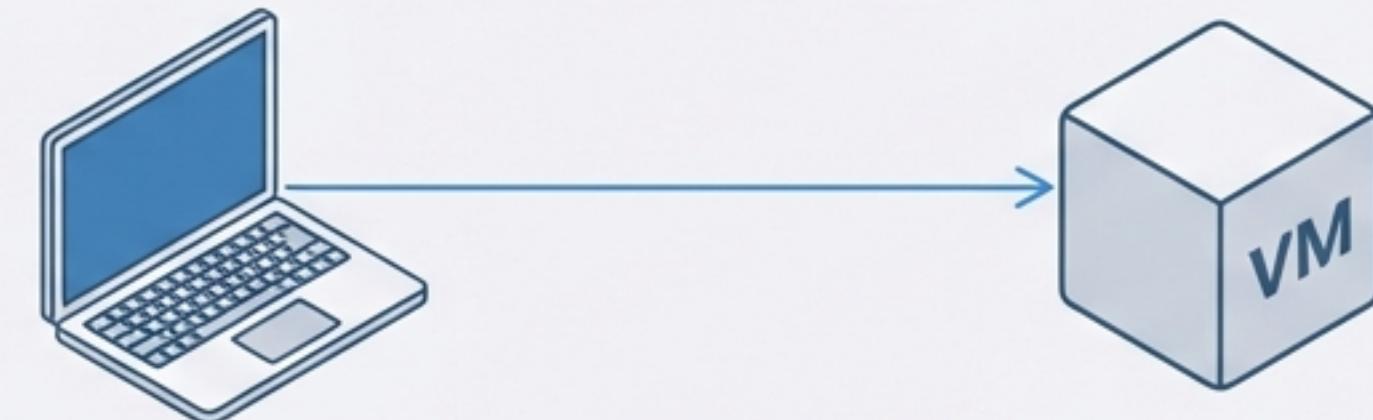
Accessing the Application

Determine how you are connecting to the service.

Internal Access



External Access (From Host) (Recommended)



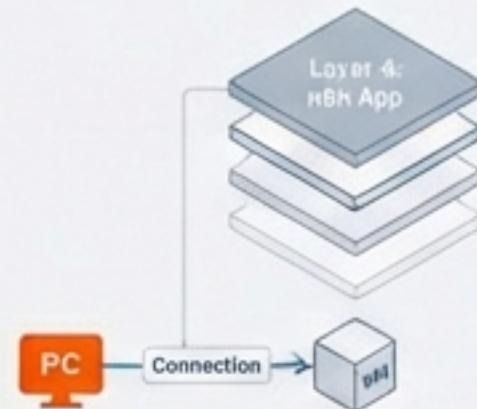
Step 1: Set VM Network to Bridged Adapter

Step 2: Find IP address with command:

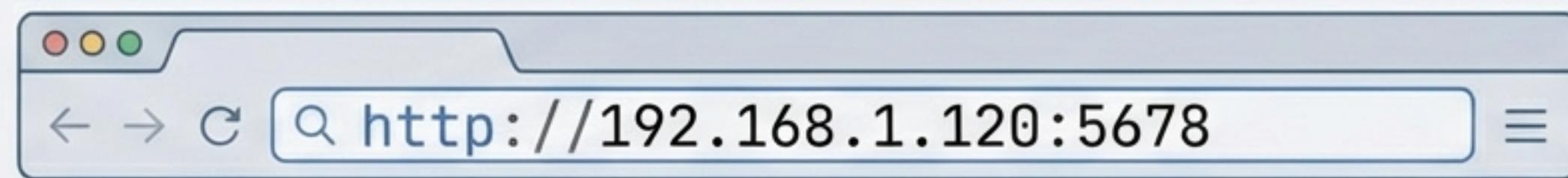
```
ip a
```

Step 3: Identify 'inet' address (e.g., 192.168.1.120)

Architectural Overview



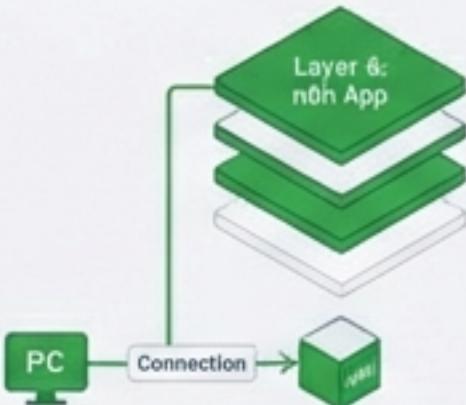
The Access Portal



A screenshot of a login form. It contains two input fields: 'Username' with the value 'admin' and 'Password' with several black dots representing the password. Below the inputs is a large blue button with the text 'Sign in'.

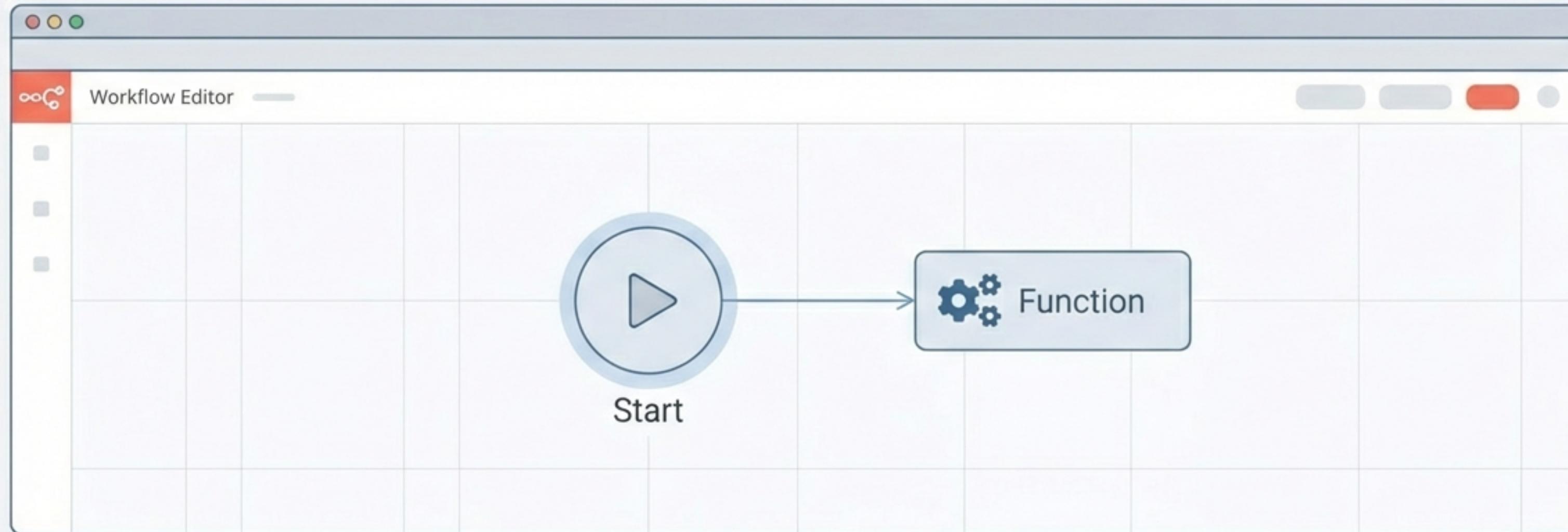
Use the credentials defined in your
docker-compose.yml file.

Architectural Overview



Mission Accomplished

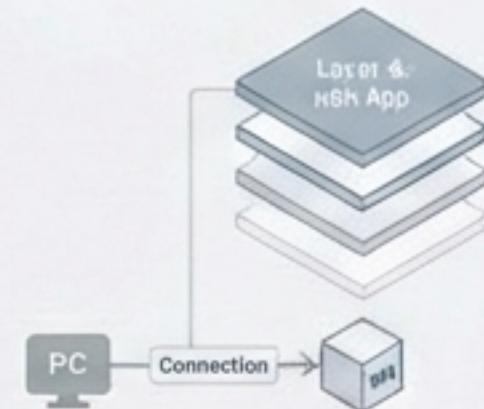
n8n is now successfully running on Docker inside the Ubuntu VM.



● Container: Running

● Data Persistence: Active (Volume)

● Security: Basic Auth Enabled



Operations & Maintenance Guide

Essential commands for managing your instance.

View Logs (Real-time)

```
docker logs -f n8n
```

Stop Service

```
docker compose stop
```

Start Service

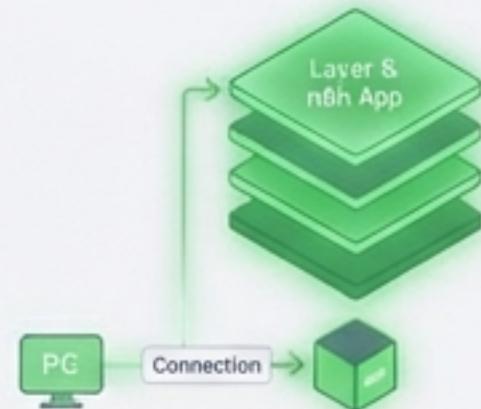
```
docker compose start
```

Safe Restart (Apply Changes)

```
docker compose down  
docker compose up -d
```

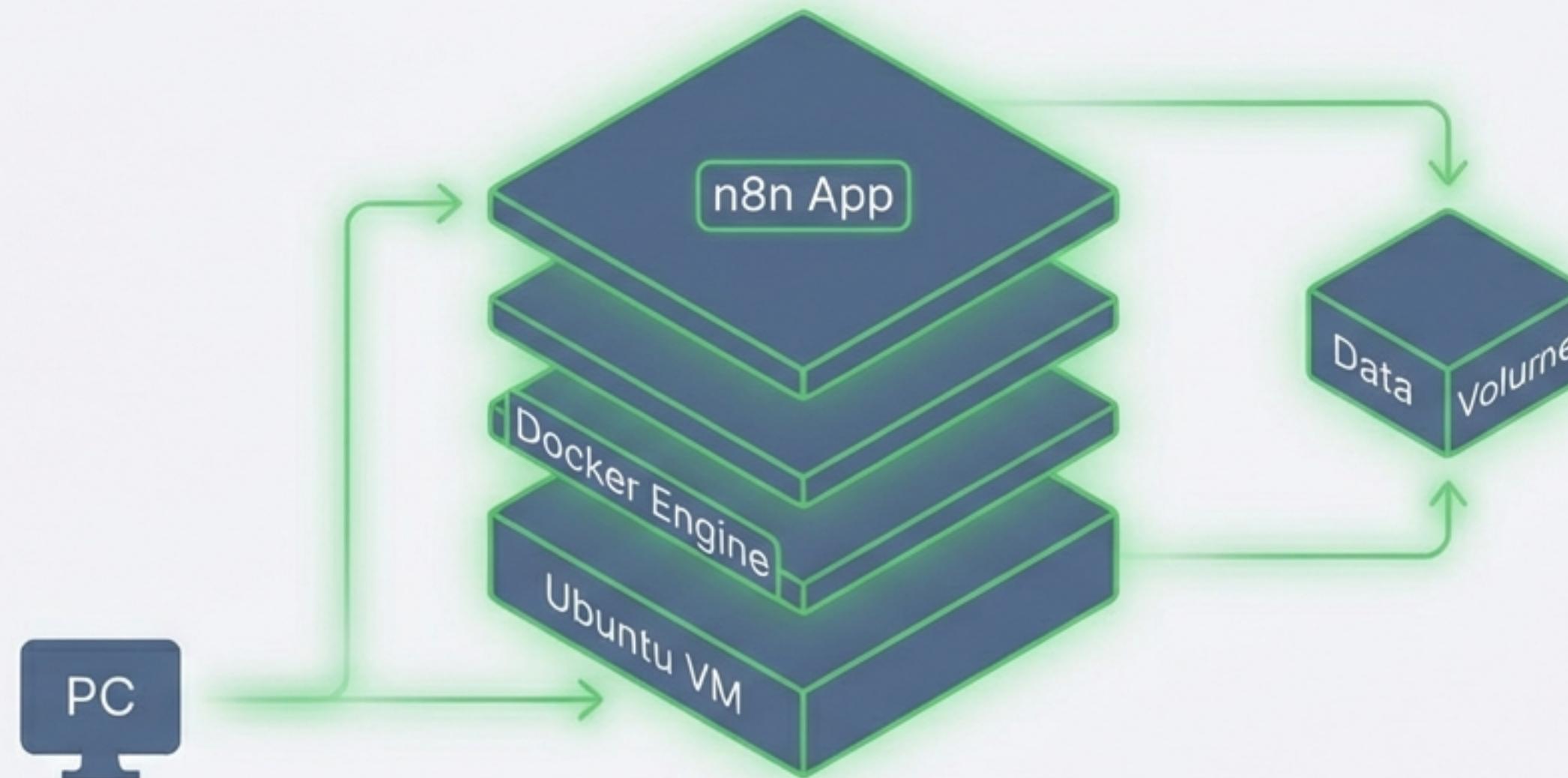
Data is preserved during restarts via the 'n8n_data' volume.

Architectural Overview



The Stack is Live.

You have deployed a resilient, scalable automation infrastructure. The Docker container manages dependencies, the Volume preserves your data, and Ubuntu provides the foundation.



Go build something amazing.