

The current version is #ident
 "@(#)\$Format:LocalFoodAI_lanfr144:distributed_deployment.md:Francois
 Lange:lanfr144@school.lu:2026/06/11 08:26:59:Francois
 Lange:lanfr144@school.lu:2026/06/11
 08:26:59:1701828b122e0c319e59134ca6511a42ecad9297::\$"

Distributed Deployment Guide

This document outlines the procedure to deploy the Local Food AI stack across a mixed topology of physical Unix servers, Windows 11 subsystems, and hypervisors on the same local network.

Supported Hypervisor Topologies

You can distribute the services across any combination of:

- **Windows Subsystem for Linux (WSL 2):** Ideal for the frontend and LLM nodes.
- **Hyper-V:** Ideal for the Database node.
- **VirtualBox:** Ideal for isolated Monitoring nodes.

Native Unix/Linux Server (No Virtualization)

Alternatively, the installation can be made natively on a physical Unix/Linux server (bare-metal) without virtualization. This is ideal for production environments where services run directly on physical hardware.

Port Conflict Matrix

When deploying nodes on the same IP subnet or host machine, ensure the following ports are open on your host firewall (e.g., Windows Defender Firewall) and not conflicting with existing services:

Service Name	Default Port	Protocol	Purpose
Nginx (App)	80	HTTP	Main Application User Interface
Streamlit	8502	HTTP	Direct Application Interface
SearXNG API	8080	HTTP	AI web searching endpoint
MySQL DB	3307	TCP	Relational database port
Zabbix Web	8081	HTTP	Zabbix monitoring dashboard
Zabbix HTTPS	8444	HTTPS	Zabbix monitoring dashboard secure
Zabbix Agent	10050	TCP	Node metric scraping
Zabbix Trap	10051	TCP	Active monitoring trap receiver

Distributed Setup Procedure

1. Network Bridging

If you are using VirtualBox or Hyper-V, you **must** configure the VM network adapter to use a **Bridged Adapter** or **External Virtual Switch**. This ensures that the VMs receive an IP address on the same physical subnet as your host machine (e.g., 192.168.x.x).

For WSL 2, use `wsl --set-version <Distro> 2` and ensure `localhost` forwarding is enabled, or use a tool like `wsl-vpnkit` if you need a dedicated IP.

2. Configure the Node via Python

On each designated node, clone the repository and execute the interactive setup script.

```
python scripts/setup_deploy.py
```

The script will ask you for:

- **Node Role:** Choose whether this node is the Database, the Application Frontend, or the Monitoring hub.
- **Network IPs:** If you are setting up the Application node, it will ask you for the IP address of the Database node (e.g., the Hyper-V VM IP).
- **Credentials:** It will securely generate a local `.env` file containing your passwords (refer to the [Environment Variables Configuration Runbook](#) for variable explanations).

3. Deploy Docker

Once the script generates the role-specific `docker-compose.yml`, run:

```
docker compose up -d
```

Moving Docker Images Offline

If your Hyper-V or VirtualBox nodes do not have internet access, you can transfer the Docker images directly from a machine that does.

On the Internet-connected machine (Export):

```
docker save -o local_food_app.tar local_food_ai-app:latest nginx:latest
docker save -o local_food_db.tar mysql:8.0
docker save -o local_food_monitoring.tar zabbix/zabbix-server-mysql:ubuntu-7.0-latest
```

On the Offline Node (Import): Copy the `.tar` files via USB or SCP, then run:

```
docker load -i local_food_app.tar
```