

# Instrukcja Uruchomienia Serwera — System Primus 2026

To repozytorium odpowiada za orkiestrację usług, zarządzanie siecią i bezpieczeństwo danych (TLS/Szyfrowanie).

## Struktura Projektu

- `dev/` [DOMYŚLNE]: Zawiera konfigurację deweloperską (buduje obrazy lokalnie z `context: .`).
- `docker-compose.yml` : Konfiguracja produkcyjna (używa gotowych obrazów z Docker Hub).
- `docker-compose-nvidia.yml` : Dodatek produkcyjny dla NVIDIA GPU.
- `docker-compose-amd.yml` : Dodatek produkcyjny dla AMD GPU.

## Zalecane - Wdrażanie Produkcyjne (Gotowe Obrazy)

Wersja produkcyjna pobiera gotowe obrazy z Docker Hub, co oszczędza czas (brak komplikacji na docelowej maszynie).

## Lokalne Budowanie

Jeśli chcesz, aby zmiany w kodzie były uwzględniane w kontenerach, musisz zbudować obrazy lokalnie.

1. **Konfiguracja Env:** `cp .env.example .env`
2. **Wejdź do katalogu dev:**

```
cd dev
```

3. **Dalej postępuj zgodnie z poniższymi instrukcjami**

## ☐ Konfiguracja NVIDIA (GPU)

Zalecana dla systemów z kartami graficznymi NVIDIA.

### Instalacja NVIDIA Container Toolkit:

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor  
-o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \  
  && curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-  
  container-toolkit.list | \  
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-  
  keyring.gpg] https://#g' | \  
    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list  
  
sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit  
sudo nvidia-ctk runtime configure --runtime=docker  
sudo systemctl restart docker
```

### Uruchomienie:

```
docker compose -f docker-compose-nvidia.yml up -d
```

## □ Konfiguracja AMD (GPU)

### Tryb Hybrydowy (WSL2)

1. **Ollama Windows:** \$env:OLLAMA\_VULKAN="1"; ollama serve
2. **Konfiguracja .env:** OLLAMA\_URL=http://host.docker.internal:11434/api/generate
3. **Uruchomienie:** docker compose up --build -d

### Tryb Natywny (Linux ROCm)

```
docker compose -f docker-compose-amd.yml up -d
```

## ○ Konfiguracja z zewnętrznym Ollama/OpenAI (CPU)

Zalecana dla systemów bez dedykowanej karty graficznej, z niekompatybilną kartą graficzną lub do szybkich testów.

### Uruchomienie:

```
docker compose up -d
```

*[!IMPORTANT] W pliku .env należy ustawić wszystkie wymagane zmienne, w tym adresy serwerów i ewentualne klucze API.*

## □ Certyfikaty SSL

System generuje certyfikaty self-signed przy pierwszym uruchomieniu  
( scripts/generate\_certs.sh ).

### Własne Certyfikaty

Podmień pliki w nginx/certs/ : nginx.crt , nginx.key , rootCA.pem . Następnie: docker compose restart nginx redis mosquitto

## Dokumentacja Projektu

Pełna dokumentacja architektury, modelu danych i modułów znajduje się w osobnym repozytorium:  
[primus-docs](#)