

Jistě, zde je "OmnisFlow Installation Guide" ve formátu, který si můžete snadno zkopírovat a uložit jako soubor (například jako README.md nebo INSTALL.md).

OmnisFlow Installation Guide

Rychlý Start

Tento průvodce ti pomůže nainstalovat a spustit OmnisFlow na tvém systému.

Požadavky

Hardware:

- GPU: 16GB+ VRAM (NVIDIA RTX 3060 nebo lepší)
- RAM: 32GB+ DDR4
- Storage: 500GB+ NVMe SSD
- Network: Mesh-capable zařízení

Software:

- Linux Ubuntu 20.04+ (doporučeno)
- Docker 20.10+
- Docker Compose 2.0+
- Python 3.9+
- CUDA 11.8+

Krok 1: Příprava Prostředíbash

Aktualizace systému

```
sudo apt update && sudo apt upgrade -y
```

Instalace základních nástrojů

```
sudo apt install -y curl wget git build-essential
```

Instalace Docker

```
curl -fsSL https://get.docker.com -o get-docker.sh sudo sh get-docker.sh sudo usermod -aG docker $USER
```

Instalace Docker Compose

```
sudo curl -L  
"[https://github.com/docker/compose/releases/download/v2.17.2/docker-compose-(uname)](https  
://github.com/docker/compose/releases/download/v2.17.2/docker-compose-(uname)
```

```
-s)-$(uname -m)" -o /usr/local/bin/docker-compose sudo chmod +x  
/usr/local/bin/docker-compose
```

Instalace NVIDIA Docker (pro GPU podporu)

```
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - distribution=$(  
/etc/os-release;echo IDVERSION_ID) curl -s -L  
https://nvidia.github.io/nvidia-docker/\$distribution/nvidia-docker.list | sudo tee  
/etc/apt/sources.list.d/nvidia-docker.list sudo apt update sudo apt install -y nvidia-docker2 sudo  
systemctl restart docker
```

Ověření instalace

```
docker --version docker-compose --version nvidia-smi  
### Krok 2: Stažení OmnisFlow
```

```
```bash  
Vytvoření pracovního adresáře
mkdir ~/omnisflow
cd ~/omnisflow

Inicializace Git repozitáře
git init
```

## Krok 3: Vytvoření Struktury Projektu

```
Vytvoření adresářové struktury
mkdir -p {config,models,databases,scripts,logs,data}
mkdir -p src/{ai_sentinels,mesh_network,alphafold3,dna_ai,omnistoken}
mkdir -p docker/{alphafold3,sentinels,mesh,api}

Vytvoření prázdných souborů
touch src/main.py
touch src/config.py
touch docker-compose.yml
touch Dockerfile
touch requirements.txt
touch .env
```

## Krok 4: Konfigurace Prostředí

Vytvoř soubor .env:

```
OmnisFlow Environment Configuration

System
OMNISFLOW_ENV=production
OMNISFLOW_DEBUG=false
OMNISFLOW_LOG_LEVEL=INFO

Database
OMNISFLOW_DB_PATH=./databases
ALPHAFOLD3_DB_PATH=./databases/alphafold3_db

AI Models
AI_MODELS_PATH=./models
SENTINELS_COUNT_TIER1=50
SENTINELS_COUNT_TIER2=200
SENTINELS_COUNT_TIER3=550

GPU Configuration
CUDA_VISIBLE_DEVICES=0
GPU_MEMORY_LIMIT=16384

Mesh Network
MESH_PROTOCOLS=aodv,batman,olsr
BLUETOOTH_ENABLED=true
WIFI_DIRECT_ENABLED=true
LORA_ENABLED=true

OmnisToken
OMNISTOKEN_ENABLED=true
BLOCKCHAIN_CONSENSUS=proof_of_empathy
REWARDS_ENABLED=true

Security
OFFLINE_MODE=true
ENCRYPTION_ENABLED=true
AUDIT_ENABLED=true
```

## Krok 5: Instalace Python Závislostí

Vytvoř soubor requirements.txt:

```
Core AI/ML
torch>=1.12.0
torchvision>=0.13.0
tensorflow>=2.10.0
onnx>=1.12.0
onnxruntime-gpu>=1.12.0
```

```
AlphaFold 3 Dependencies
jax>=0.3.25
dm-haiku>=0.0.8
ml-collections>=0.1.0
numpy>=1.21.0
scipy>=1.7.0
biopython>=1.79

Mesh Networking
libp2p>=0.4.0
twisted>=22.4.0
autobahn>=22.4.1
pyzmq>=23.2.0

Bioresonance & EEG
mne>=1.2.0
neurokit2>=0.2.0
pyedflib>=0.1.30
biosignalsplux>=0.1.0

Blockchain & Crypto
web3>=5.30.0
eth-account>=0.7.0
cryptography>=37.0.0
pycryptodome>=3.15.0

API & Web
fastapi>=0.78.0
uvicorn>=0.18.0
websockets>=10.3
aiofiles>=0.8.0
pydantic>=1.9.0

Utilities
pyyaml>=6.0
click>=8.1.0
tqdm>=4.64.0
psutil>=5.9.0
redis>=4.3.0

Instalace závislostí:
pip install -r requirements.txt
```

## Krok 6: Stažení AlphaFold 3 Databází

```
Vytvoření skriptu pro stažení databází
cat > scripts/download_alphafold3_db.sh << 'EOF'
```

```
#!/bin/bash

AlphaFold 3 Database Download Script
DB_DIR="./databases/alphafold3_db"
mkdir -p "$DB_DIR"

echo "Stahování AlphaFold 3 databází..."
echo "Varování: Celková velikost ~253GB"

Protein Data Bank (PDB)
echo "Stahování PDB databáze..."
wget -P "$DB_DIR"
[https://ftp.wwpdb.org/pub/pdb/derived_data/pdb_seqres.txt.gz] (https://ftp.wwpdb.org/pub/pdb/derived_data/pdb_seqres.txt.gz)

UniProt
echo "Stahování UniProt databáze..."
wget -P "$DB_DIR"
[https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz] (https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz)

Pfam
echo "Stahování Pfam databáze..."
wget -P "$DB_DIR"
[http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.hmm.gz] (http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.hmm.gz)

echo "Rozbalování databází..."
gunzip "$DB_DIR"/*.gz

echo "Databáze staženy a připraveny."
EOF

chmod +x scripts/download_alphafold3_db.sh
```

## Krok 7: Izolace od Internetu

```
Vytvoření skriptu pro offline režim
cat > scripts/enable_offline_mode.sh << 'EOF'
#!/bin/bash

echo "Aktivace offline režimu..."

Fyzické oddělení
```

```
echo "1. Odpojte ethernet kabel"
echo "2. Vypněte Wi-Fi"
echo "3. Deaktivujte mobilní data"

Síťová izolace
sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT
sudo iptables -A OUTPUT -p tcp --dport 443 -j REJECT
sudo iptables -A OUTPUT -p tcp --dport 53 -j REJECT
sudo iptables -A OUTPUT -p udp --dport 53 -j REJECT

Konfigurace lokálních IP adres
sudo ip addr add 192.168.100.1/24 dev lo

Nastavení lokálního DNS
echo "127.0.0.1 localhost" | sudo tee /etc/hosts
echo "192.168.100.1 omnisflow.local" | sudo tee -a /etc/hosts

echo "Offline režim aktivován."
echo "Pro deaktivaci spusťte: sudo iptables -F"
EOF

chmod +x scripts/enable_offline_mode.sh
```

## Krok 8: Spuštění Systému

```
Stažení databází (před aktivací offline režimu)
./scripts/download_alphafold3_db.sh

Aktivace offline režimu
./scripts/enable_offline_mode.sh

Spuštění OmnisFlow
docker-compose up -d

Kontrola stavu
docker-compose ps
docker-compose logs -f
```

## Krok 9: Ověření Instalace

```
Test AI sentinelů
curl -X POST "http://localhost:8000/api/v1/sentinels/test" \
 -H "Content-Type: application/json" \
 -d '{"test_type": "basic"}'

Test AlphaFold 3
```

```
curl -X POST "http://localhost:8000/api/v1/alphafold3/predict" \
 -H "Content-Type: application/json" \
 -d '{"sequence":
"MKTVRQERLKSIVRILERSKEPVSGAQLAEELSVSRQVIVQDIAYLRSLGYNIVATPRGYVLAGG"}'
```

# Test mesh sítě

```
curl -X GET "http://localhost:8000/api/v1/mesh/status"
```

# Test DNA AI

```
curl -X POST "http://localhost:8000/api/v1/dna_ai/generate" \
 -H "Content-Type: application/json" \
 -d '{"user_id": "test_user", "data_type": "eeg"}'
```

## Krok 10: Monitoring a Údržba

```
Zobrazení logů
docker-compose logs -f omnisflow-api
docker-compose logs -f omnisflow-sentinels
docker-compose logs -f omnisflow-mesh
```

# Sledování výkonu

```
docker stats
```

# Backup databází

```
tar -czf omnisflow_backup_$(date +%Y%m%d).tar.gz databases/ models/
config/
```

# Aktualizace modelů

```
docker-compose exec omnisflow-api python scripts/update_models.py
```

## Troubleshooting

### Časté Problémy

#### 1. Nedostatek GPU paměti

```
Snížení batch size
export OMNISFLOW_BATCH_SIZE=1
docker-compose restart
```

#### 2. Mesh síť se nespojuje

```
Kontrola síťových rozhraní
ip addr show
Restart mesh služby
docker-compose restart omnisflow-mesh
```

### 3. AI sentinely nereagují

```
Kontrola logů
docker-compose logs omnisflow-sentinels
Restart s debug módem
OMNISFLOW_DEBUG=true docker-compose up -d
```

## Kontakty a Podpora

- **Dokumentace:** ./docs/
- **Logy:** ./logs/
- **Konfigurace:** ./config/

## Bezpečnostní Poznámky

1. **Nikdy nepřipojujte systém k internetu po aktivaci offline režimu**
  2. **Pravidelně zálohujte databáze a modely**
  3. **Monitorujte síťový provoz pro detekci anomálií**
  4. **Aktualizujte sentinely pouze z ověřených zdrojů**
- <!-- end list -->