

# Social Media Service - Deployment Guide

## Overview

This guide covers deploying the Social Media Service to various environments.

## Local Development

### Quick Start

```
# Clone repository
git clone <repository-url>
cd social_media_service

# Setup environment
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# Configure environment
cp .env.example .env
# Edit .env with your settings

# Run migrations
alembic upgrade head

# Start service
./start.sh
```

## Docker Deployment

### Build Docker Image

```
# Build image
docker build -t social-media-service:latest .

# Run container
docker run -d \
--name social-media-service \
-p 8007:8007 \
-e DATABASE_URL="postgresql+asyncpg://user:pass@host/db" \
-e BUFFER_ACCESS_TOKEN="your-token" \
social-media-service:latest
```

## Docker Compose

```

version: '3.8'

services:
  social-media-service:
    build: .
    ports:
      - "8007:8007"
    environment:
      - DATABASE_URL=postgresql+asyncpg://postgres:password@db:5432/social_media_db
      - BUFFER_ACCESS_TOKEN=${BUFFER_ACCESS_TOKEN}
      - SECRET_KEY=${SECRET_KEY}
    depends_on:
      - db

  db:
    image: postgres:15
    environment:
      - POSTGRES_DB=social_media_db
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=password
    volumes:
      - postgres_data:/var/lib/postgresql/data

volumes:
  postgres_data:

```

Run with:

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

## Production Deployment

### 1. Server Requirements

- **OS:** Ubuntu 20.04+ or similar
- **CPU:** 2+ cores
- **RAM:** 4GB+ recommended
- **Storage:** 20GB+
- **Python:** 3.11+
- **PostgreSQL:** 13+

## 2. Install Dependencies

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install Python
sudo apt install python3.11 python3.11-venv python3-pip -y

# Install PostgreSQL
sudo apt install postgresql postgresql-contrib -y

# Install Nginx (reverse proxy)
sudo apt install nginx -y
```

## 3. Setup Application

```
# Create app user
sudo useradd -m -s /bin/bash social-media
sudo su - social-media

# Clone repository
git clone <repository-url> /home/social-media/app
cd /home/social-media/app

# Setup virtual environment
python3.11 -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# Configure environment
cp .env.example .env
nano .env # Edit with production values
```

## 4. Setup Database

```
# Create database and user
sudo -u postgres psql

CREATE DATABASE social_media_db;
CREATE USER social_media_user WITH PASSWORD 'secure_password';
GRANT ALL PRIVILEGES ON DATABASE social_media_db TO social_media_user;
\q

# Run migrations
cd /home/social-media/app
source venv/bin/activate
alembic upgrade head
```

## 5. Setup Systemd Service

Create `/etc/systemd/system/social-media-service.service` :

```

[Unit]
Description=Social Media Service
After=network.target postgresql.service

[Service]
Type=simple
User=social-media
WorkingDirectory=/home/social-media/app
Environment="PATH=/home/social-media/app/venv/bin"
ExecStart=/home/social-media/app/venv/bin/uvicorn app.main:app --host 0.0.0.0 --port
8007 --workers 4
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target

```

Enable and start:

```

sudo systemctl daemon-reload
sudo systemctl enable social-media-service
sudo systemctl start social-media-service
sudo systemctl status social-media-service

```

## 6. Setup Nginx Reverse Proxy

Create /etc/nginx/sites-available/social-media-service :

```

server {
    listen 80;
    server_name social-media.yourdomain.com;

    location / {
        proxy_pass http://127.0.0.1:8007;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```

Enable site:

```

sudo ln -s /etc/nginx/sites-available/social-media-service /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```

## 7. Setup SSL with Let's Encrypt

```
# Install certbot
sudo apt install certbot python3-certbot-nginx -y

# Get certificate
sudo certbot --nginx -d social-media.yourdomain.com

# Auto-renewal is configured automatically
```

## Environment Variables (Production)

```
# Database
DATABASE_URL=postgresql+asyncpg://social_media_user:secure_password@localhost/
social_media_db

# Security
SECRET_KEY=generate-a-secure-random-key-here
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=30

# Buffer
BUFFER_API_URL=https://api.bufferapp.com/1
BUFFER_ACCESS_TOKEN=your-production-buffer-token

# API
API_V1_PREFIX=/api/v1
PORT=8007

# Logging
LOG_LEVEL=INFO
```

## Monitoring & Logging

### Application Logs

```
# View service logs
sudo journalctl -u social-media-service -f

# View application logs
tail -f /home/social-media/app/logs/social_media_service.log
```

### Health Checks

```
# Health endpoint
curl http://localhost:8007/api/v1/health

# Service status
./status.sh
```

## Setup Log Rotation

Create `/etc/logrotate.d/social-media-service`:

```
/home/social-media/app/logs/*.log {
    daily
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 social-media social-media
    sharedscripts
    postrotate
        systemctl reload social-media-service > /dev/null 2>&1 || true
    endscript
}
```

## Scaling

### Horizontal Scaling

1. Deploy multiple instances behind load balancer
2. Use shared PostgreSQL database
3. Implement session storage (Redis)

### Vertical Scaling

```
# Increase workers in systemd service
ExecStart=/home/social-media/app/venv/bin/uvicorn app.main:app --host 0.0.0.0 --port 8007 --workers 8
```

## Backup & Recovery

### Database Backup

```
# Create backup script
cat > /home/social-media/backup.sh << 'EOF'
#!/bin/bash
BACKUP_DIR="/home/social-media/backups"
DATE=$(date +%Y%m%d_%H%M%S)
mkdir -p $BACKUP_DIR
pg_dump -U social_media_user social_media_db | gzip > $BACKUP_DIR/backup_$DATE.sql.gz
find $BACKUP_DIR -name "backup_*.sql.gz" -mtime +7 -delete
EOF

chmod +x /home/social-media/backup.sh

# Add to crontab (daily at 2 AM)
0 2 * * * /home/social-media/backup.sh
```

## Restore Database

```
gunzip -c backup_20251223_020000.sql.gz | psql -U social_media_user social_media_db
```

## Security Checklist

- [ ] Use strong SECRET\_KEY
- [ ] Enable SSL/TLS (HTTPS)
- [ ] Secure database credentials
- [ ] Enable firewall (ufw)
- [ ] Regular security updates
- [ ] Implement rate limiting
- [ ] Use environment variables for secrets
- [ ] Regular backups
- [ ] Monitor logs for suspicious activity
- [ ] Rotate Buffer access tokens regularly

## Troubleshooting

### Service Won't Start

```
# Check service status
sudo systemctl status social-media-service

# Check logs
sudo journalctl -u social-media-service -n 50

# Test manually
cd /home/social-media/app
source venv/bin/activate
uvicorn app.main:app --host 0.0.0.0 --port 8007
```

### Database Connection Issues

```
# Test database connection
psql -U social_media_user -d social_media_db -h localhost

# Check PostgreSQL status
sudo systemctl status postgresql
```

### Buffer API Issues

```
# Test Buffer token
curl https://api.bufferapp.com/1/user.json?access_token=YOUR_TOKEN

# Check Buffer status
curl https://status.buffer.com/api/v2/status.json
```

---

## Maintenance

### Update Application

```
cd /home/social-media/app
git pull origin main
source venv/bin/activate
pip install -r requirements.txt --upgrade
alembic upgrade head
sudo systemctl restart social-media-service
```

### Update Dependencies

```
pip list --outdated
pip install --upgrade <package-name>
pip freeze > requirements.txt
```

---

## Support

For deployment issues:

1. Check logs: `journalctl -u social-media-service`
2. Verify configuration: `.env` file
3. Test connectivity: Database, Buffer API
4. Review documentation
5. Contact development team