Search Service - Deployment Guide

Overview

This guide covers deploying the Search Service in various environments.

Table of Contents

Production Deployment

Prerequisites

- Ubuntu 20.04+ / Debian 11+
- PostgreSQL 14+
- Python 3.11+
- Nginx (reverse proxy)
- SSL Certificate

Step 1: System Setup

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

# Install dependencies
sudo apt install -y postgresql postgresql-contrib python3.11 python3.11-venv nginx cer
tbot

# Install PostgreSQL extension
sudo -u postgres psql -c "CREATE EXTENSION IF NOT EXISTS pg_trgm;"
```

Step 2: Database Setup

```
# Create database and user
sudo -u postgres psql << EOF
CREATE DATABASE search_service_prod;
CREATE USER search_user WITH PASSWORD 'secure_password';
GRANT ALL PRIVILEGES ON DATABASE search_service_prod TO search_user;
\c search_service_prod
CREATE EXTENSION pg_trgm;
EOF
```

Step 3: Application Setup

```
# Create application user
sudo useradd -m -s /bin/bash search_service
sudo su - search_service

# Clone repository
git clone <repository_url> /home/search_service/app
cd /home/search_service/app

# Create virtual environment
python3.11 -m venv venv
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt

# Configure environment
cp .env.example .env
nano .env
```

Step 4: Environment Configuration

Production `.env`:

```
# Database
DATABASE_URL=postgresql://search_user:secure_password@localhost/search_service_prod

# Service URLs
CONTENT_SERVICE_URL=https://content.missionengadi.org
PARTNERS_SERVICE_URL=https://partners.missionengadi.org
PROJECTS_SERVICE_URL=https://projects.missionengadi.org
SOCIAL_MEDIA_SERVICE_URL=https://social.missionengadi.org

# Security
SECRET_KEY=your_secure_secret_key_here
JWT_SECRET=your_jwt_secret_here
ALLOWED_HOSTS=search.missionengadi.org

# Search Settings
DEFAULT_PAGE_SIZE=10
MAX_PAGE_SIZE=100
MAX_SEARCH_RESULTS=1000

# Performance
DATABASE_POOL_SIZE=20
DATABASE_MAX_OVERFLOW=10

# Logging
LOG_LEVEL=INFO
LOG_FILE=/var/log/search_service/app.log
```

Step 5: Database Migration

```
# Run migrations
alembic upgrade head

# Verify tables
psql -U search_user -d search_service_prod -c "\dt"
```

Step 6: Systemd Service

Create `/etc/systemd/system/search_service.service`:

```
[Unit]
Description=Search Service
After=network.target postgresql.service

[Service]
Type=notify
User=search_service
Group=search_service
WorkingDirectory=/home/search_service/app
Environment="PATH=/home/search_service/app/venv/bin"
ExecStart=/home/search_service/app/venv/bin/gunicorn app.main:app \
    --workers 4 \
    --worker-class uvicorn.workers.UvicornWorker \
    --bind 0.0.0.0:8011 \
    --access-logfile /var/log/search_service/access.log \
    --error-logfile /var/log/search_service/error.log \
    --log-level info
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Enable and start:

```
# Create log directory
sudo mkdir -p /var/log/search_service
sudo chown search_service:search_service /var/log/search_service

# Enable service
sudo systemctl enable search_service
sudo systemctl start search_service
sudo systemctl status search_service
```

Step 7: Nginx Configuration

Create `/etc/nginx/sites-available/search_service`:

```
upstream search_service {
    server 127.0.0.1:8011;
}

server {
    listen 80;
    server_name search.missionengadi.org;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name search.missionengadi.org;

    ssl_certificate /etc/letsencrypt/live/search.missionengadi.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/search.missionengadi.org/privkey.pem;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    client_max_body_size 10M;

    location / {
        proxy_pass http://search_service;
        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;
        proxy_connect_timeout 60s;
        proxy_send_timeout 60s;
        proxy_read_timeout 60s;
    }

    location /docs {
        proxy_pass http://search_service/docs;
        proxy_set_header Host $host;
    }

    location /health {
        proxy_pass http://search_service/health;
        access_log off;
    }
}
```

Enable site:

```
sudo ln -s /etc/nginx/sites-available/search_service /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

Step 8: SSL Certificate

```
# Get SSL certificate
sudo certbot --nginx -d search.missionengadi.org

# Auto-renewal
sudo certbot renew --dry-run
```

Docker Deployment

Dockerfile

Create `Dockerfile`:

```
FROM python:3.11-slim

WORKDIR /app

# Install system dependencies
RUN apt-get update && apt-get install -y \
    postgresql-client \
    && rm -rf /var/lib/apt/lists/*

# Install Python dependencies
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# Copy application
COPY . .

# Create logs directory
RUN mkdir -p logs

EXPOSE 8011

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8011"]
```

Docker Compose

Create `docker-compose.yml`:

```yaml
version: '3.8'

services:
  postgres:
    image: postgres:14
    environment:
      POSTGRES_DB: search_service_db
      POSTGRES_USER: search_user
      POSTGRES_PASSWORD: secure_password
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    ports:
      - "5432:5432"

  search_service:
    build: .
    ports:
      - "8011:8011"
    environment:
      DATABASE_URL: postgresql://search_user:secure_password@postgres/
search_service_db
      CONTENT_SERVICE_URL: http://content_service:8007
      PARTNERS_SERVICE_URL: http://partners_service:8009
      PROJECTS_SERVICE_URL: http://projects_service:8010
    depends_on:
      - postgres
    volumes:
      - ./logs:/app/logs
    restart: always

volumes:
  postgres_data:
```

Create `init.sql`:

```sql
CREATE EXTENSION IF NOT EXISTS pg_trgm;
```

Deploy:

```
docker-compose up -d
docker-compose logs -f search_service
```

Performance Tuning

PostgreSQL Optimization

Edit `/etc/postgresql/14/main/postgresql.conf`:

```
# Memory
shared_buffers = 256MB
effective_cache_size = 1GB
maintenance_work_mem = 64MB
work_mem = 16MB

# Query Planner
random_page_cost = 1.1
effective_io_concurrency = 200

# WAL
wal_buffers = 16MB
min_wal_size = 1GB
max_wal_size = 4GB

# Checkpoints
checkpoint_completion_target = 0.9

# Connections
max_connections = 100
```

Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

Database Maintenance

Create maintenance script `/home/search_service/maintenance.sh`:

```
#!/bin/bash

# Vacuum and analyze
psql -U search_user -d search_service_prod << EOF
VACUUM ANALYZE search_index;
VACUUM ANALYZE search_query;
VACUUM ANALYZE search_suggestion;
REINDEX TABLE search_index;
EOF
```

Schedule via cron:

```
# Run daily at 3 AM
0 3 * * * /home/search_service/maintenance.sh
```

Application Optimization

Use Gunicorn with multiple workers:

```
gunicorn app.main:app \
    --workers 4 \
    --worker-class uvicorn.workers.UvicornWorker \
    --bind 0.0.0.0:8011 \
    --worker-connections 1000 \
    --max-requests 1000 \
    --max-requests-jitter 100
```

Monitoring

Health Checks

```
# Application health
curl http://localhost:8011/health

# Database health
psql -U search_user -d search_service_prod -c "SELECT 1;"
```

Logging

Application logs:

```
tail -f /var/log/search_service/app.log
tail -f /var/log/search_service/access.log
tail -f /var/log/search_service/error.log
```

PostgreSQL logs:

```
tail -f /var/log/postgresql/postgresql-14-main.log
```

Metrics

Monitor key metrics:

- Search query latency
- Index size growth
- Query throughput
- Error rates
- Database connections

Backup and Recovery

Database Backup

Create backup script `/home/search_service/backup.sh`:

```
#!/bin/bash

BACKUP_DIR="/backups/search_service"
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_FILE="$BACKUP_DIR/search_service_$DATE.sql.gz"

mkdir -p $BACKUP_DIR

# Backup database
pg_dump -U search_user search_service_prod | gzip > $BACKUP_FILE

# Keep only last 7 days
find $BACKUP_DIR -name "*.sql.gz" -mtime +7 -delete

echo "Backup completed: $BACKUP_FILE"
```

Schedule daily backups:

```
0 2 * * * /home/search_service/backup.sh
```

Recovery

```
# Restore from backup
gunzip -c /backups/search_service/search_service_20241225.sql.gz | \
    psql -U search_user search_service_prod

# Reindex after restore
POST /api/v1/indexing/reindex/content
```

Troubleshooting

Service won’t start

```
sudo systemctl status search_service
sudo journalctl -u search_service -f
```

Database connection issues

```
# Test connection
psql -U search_user -d search_service_prod

# Check PostgreSQL status
sudo systemctl status postgresql
```

High memory usage

```
# Check process memory
ps aux | grep gunicorn

# Reduce workers if needed
# Edit /etc/systemd/system/search_service.service
```

Security Checklist

- [ ] SSL/TLS enabled
- [ ] Strong database passwords
- [ ] JWT secrets rotated
- [ ] Firewall configured
- [ ] Rate limiting enabled
- [ ] Regular security updates
- [ ] Backup encryption
- [ ] Access logs monitored

Support

For deployment support, refer to:
- README.md (./README.md)
- API_DOCUMENTATION.md (./API_DOCUMENTATION.md)
- INTEGRATION_GUIDE.md (./INTEGRATION_GUIDE.md)