AI-Powered Resume Screening System

Production Readiness & Deployment Guide

1. Environment Setup

Production requires robust servers unlike the dev 'runserver'.

- **Web Server**: Nginx (Reverse Proxy, SSL, Static Files)
- **App Server**: Gunicorn (WSGI) or Uvicorn (ASGI)
- **Database**: MongoDB Atlas (Managed) or Replica Set

2. Containerization (Docker)

```
# Dockerfile Optimized
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
CMD ["gunicorn", "resume_screener.wsgi:application"]
```

3. CI/CD Pipeline (GitHub Actions)

1. Push to 'main' branch.
2. Automated Tests (pytest, flake8).
3. Build Docker Image & Push to Registry.
4. Deploy to Production Server (SSH/Kubernetes).

4. Security Best Practices

- **SSL/TLS**: Force HTTPS using Let's Encrypt.
- **Secrets**: Use .env files, never commit keys.
- **Rate Limiting**: Protect API endpoints from abuse.
- **Validation**: Strict file type checks (PDF/DOCX only).

5. Scaling Strategy

Async Processing (Celery): Offload heavy ML tasks (Screening) to background workers to keep the API responsive.

Horizontal Scaling: Add more App Server instances behind a Load Balancer.