Deployment Guide

Author: Dhawanit Bhatnagar

Date: 02-Aug-2025

Version: 1.0

Table of Contents

---

1. Introduction

This document provides a **step-by-step guide** to deploying the **Document Management System (DMS)**.

It covers **manual deployment**, **Docker-based deployment**, **database initialization**, **environment configuration**, and **post-deployment checks**.

---

2. Prerequisites

Before deploying, ensure the following tools and resources are available:

- **System Requirements**
  - CPU: 2+ cores

    - RAM: 4GB minimum (8GB recommended)
    - Disk Space: 10GB free
- **Installed Software:**
    - Node.js (v20+)
    - npm (v9+)
    - PostgreSQL (v15 or later)
    - Docker (for container deployment)
    - Docker Compose
- **Access Requirements:**
    - SSH access to target server
    - Permissions to create and manage Docker containers
    - Ports **3000 (backend)**, **5173 (frontend)**, **5432 (Postgres)** open on firewall

---

3. Environment Setup

3.1 Clone Repository

- git clone https://github.com/dhawanit/document-management.git
- cd document-management

3.2 Project Structure

```
document-management/
├─── backend/
│   ├─── src/
│   ├─── .env
│   └─── Dockerfile
├─── frontend/
│   ├─── src/
│   ├─── .env
│   └─── Dockerfile
├─── docs/
└─── docker-compose.yml
└─── README.md
```

3.3 Environment Variables

Create a .env file in both **backend** and **frontend** directories.

3.3.1 Backend .env

- PORT=3000
- DATABASE_URL=postgres://docuser:docpass@postgres:5432/docdb
- JWT_SECRET=supersecretkey
- JWT_EXPIRES_IN=3600s
- STORAGE_PROVIDER=local
- AWS_ACCESS_KEY_ID=
- AWS_SECRET_ACCESS_KEY=
- AWS_BUCKET_NAME=
- RANDOM_INGESTION=true

3.3.2 Frontend .env

- VITE_API_BASE_URL=http://localhost:3000

---

4. Backend Deployment

4.1 Install Dependencies

- cd backend
- npm install

4.2 Run Database seeders

- npm run seed:all

4.3 Start Backend

- npm run start:dev

Backend should be available at:

http://localhost:3000

---

5. Frontend Deployment

5.1 Install Dependencies

- cd frontend
- npm install

5.2 Build Production Files

- npm run dev

Frontend should be available at:

http://localhost:5173

---

6. Database Setup

- PostgreSQL database docdb should be created.
- Run the seeders to create:
  - **Admin User:** admin@document.com / Admin@123
  - **Sample Users:** 1000 editor/viewer accounts

---

7. Docker Deployment

7.1 Build and run containers:
  7.1.1 docker-compose up --build -d

7.2 Services:
  7.2.1 docmgmt-postgres → Database
  7.2.2 docmgmt-backend → NestJS backend
  7.2.3 docmgmt-frontend → React frontend

7.3 Auto-restart enabled:

   7.3.1 If any service crashes, Docker restarts it automatically.

7.4 Health checks:

   7.4.1 Backend: /health

   7.4.2 Database: pg_is ready

---

8. Running Seeders

Once containers are running:

docker exec -it docmgmt-backend npm run seed:all

This creates admin and test users in the database.

---

9. Health Checks & Monitoring

- **Backend:**

  Visit http://localhost:3000/health → Should return { status: "ok" }.

- **Database:**

  docker exec -it docmgmt-postgres pg_isready

- **Frontend:**

  Access UI via browser at http://localhost:5173.

- Use tools like **Docker logs** or **PM2** (for manual deployment) to monitor services.

---

10. Rollback Procedure

- To rollback to the last working container build:
    - *docker-compose down*
    - *docker-compose pull*
    - *docker-compose up -d*
