

Production Deployment Guide – Entitlement Review

Author: Divyansh Singh Parihar

Target Environment: Ubuntu 22.04 + Nginx + PM2 + MongoDB + MERN stack

1 Overview

This guide details the end-to-end process for deploying the *Final_Product* MERN application in a production Linux environment, plus installation notes for Windows development workstations.

2 Prerequisites

Component	Required Version
Ubuntu Server	22.04 LTS
Node.js	18 (runtime)
VS Code	Latest stable
MongoDB Community Server	≥ 6.0
Mongosh	Latest
MongoDB Compass	Latest
PM2	Latest

3 Server Preparation (Ubuntu)

```
sudo apt update && sudo apt upgrade -y
sudo apt install ubuntu-desktop # optional GUI for server
sudo reboot
```

4 Development Tools Installation

4.1 Node.js 18

Windows

1. Download the LTS installer from <https://nodejs.org>.
2. Run the installer with default settings.
3. Verify:

```
node -v  
npm -v
```

Linux (NodeSource)

```
# remove old versions  
sudo apt remove -y nodejs npm  
  
# install curl if missing  
sudo apt update && sudo apt install -y curl  
  
# add NodeSource repo (20 LTS shown)  
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -  
  
# install Node.js  
sudo apt install -y nodejs  
  
# verify  
node -v  
npm -v
```

Alternative: nvm

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | b  
ash  
source "$HOME/.nvm/nvm.sh"  
nvm install 22
```

```
node -v # v22.x
npm -v
```

4.2 Visual Studio Code

Windows: Use the User Installer from <https://code.visualstudio.com/> with "Add to PATH" enabled.

Linux:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor
or | sudo tee /etc/apt/keyrings/packages.microsoft.gpg >/dev/null
printf "deb [arch=amd64,arm64 signed-by=/etc/apt/keyrings/packages.micro
soft.gpg] https://packages.microsoft.com/repos/code stable main" | sudo tee
/etc/apt/sources.list.d/vscode.list
sudo apt update && sudo apt install -y apt-transport-https code
```

4.3 MongoDB Compass

Windows: Download the MSI from <https://www.mongodb.com/try/download/compass> and install with defaults.

Linux:

```
wget https://downloads.mongodb.com/compass/mongodb-compass_1.45.0_a
md64.deb
sudo apt install ./mongodb-compass_1.45.0_amd64.deb
```

4.4 Mongosh

Download the MSI or DEB from <https://www.mongodb.com/try/download/shell> and install. Verify with `mongosh`.

4.5 MongoDB Community Server

Windows: Use the MSI installer, choose *Complete*, enable *Install MongoDB as a Service*. Verify:

```
mongod --version  
services.msc # service should be Running
```

Linux: Follow the official MongoDB repo instructions for Ubuntu 22.04.

5 Project Setup

5.1 Clone Repository

```
git clone https://github.com/SomethingForWork/Final_Product.git
```

5.2 Install Dependencies

```
# backend  
a. cd Final_Product/server  
b. npm install  
  
# frontend  
c. cd ../frontend  
d. npm install
```

5.3 Environment Files

Create `.env` files:

```
# server/.env  
PORT=3002  
HOST=0.0.0.0  
  
# frontend/.env  
REACT_APP_API_URL=/api
```

Rebuild the frontend whenever `.env` changes:

```
npm run build
```

6 MongoDB Configuration

6.1 Enable Authentication (Linux)

```
sudo nano /etc/mongod.conf
```

Add:

```
security:  
  authorization: enabled
```

Restart:

```
sudo systemctl restart mongod
```

6.2 Create Database Users (mongosh)

```
use restrict_app
```

```
db.createUser({  
  user: "restrict_user",  
  pwd: "<strong-password>",  
  roles: [{ role: "readWrite", db: "restrict_app" }]  
})
```

Additional admin:

```
db.createUser({  
  user: "admin_user",  
  pwd: "<admin-password>",
```

```
roles: [{ role: "dbAdmin", db: "restrict_app" }]
})
```

6.3 Connection String Format

```
mongodb://restrict_user:<password>@localhost:27017/restrict_app
```

6.4 Common Error: ECONNREFUSED

```
sudo systemctl status mongod    # check service
sudo systemctl start mongod     # start if inactive
sudo systemctl enable mongod    # auto-start on boot
```

Ensure Compass or application points to `mongodb://localhost:27017`.

7 Application Configuration

7.1 Update Backend Address in Frontend

In `server/index.js` adjust CORS:

```
app.use(cors({
  origin: "http://<SERVER_IP>:3000",
  credentials: true
}))
```

7.2 Create Initial Admin (once backend is up)

```
curl -X POST http://<SERVER_IP>:3002/create-admin \
  -H "Content-Type: application/json" \
  -d '{"name":"Admin","email":"admin@example.com","password":"<password>"}
```

8 Production Deployment Workflow

8.1 Install Global Tools

```
sudo apt update  
sudo apt install -y nodejs npm nginx  
sudo npm install -g pm2 serve
```

8.2 Backend Service (PM2)

```
cd /opt/Final_Product/server  
pm2 start index.js --name final-product-backend  
pm2 save  
pm2 startup      # generates systemd script
```

8.3 Frontend Build & Static Serve

```
cd ../frontend  
npm install  
npm run build    # ensure build/ exists
```

If `react-scripts` missing:

```
npm install react-scripts@5.0.1 --save-dev  
npm run build
```

Optional Node static server (only for quick tests):

```
serve -s build -l 3000
```

8.4 Nginx Reverse Proxy

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

```

server {
    listen 80;
    server_name <SERVER_IP>;

    root /opt/Final_Product/frontend/build;
    index index.html;

    location / {
        try_files $uri /index.html;
    }

    location /api/ {
        proxy_pass http://localhost:3002/;
        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 and reload:

```

sudo ln -s /etc/nginx/sites-available/final_product /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```

9 Validation

1. Browse to `http://<SERVER_IP>` – React app should load.
2. API endpoints (`/api/...`) must return data.
3. Verify PM2 process: `pm2 list` .
4. Check Nginx and MongoDB logs for errors.

10 Maintenance

Task	Command
View PM2 logs	<code>pm2 logs final-product-backend</code>
Restart service	<code>pm2 restart final-product-backend</code>
Update code	<code>git pull && npm install && npm run build</code>
Rotate PM2 logs	<code>pm2 flush</code>

End of Guide