

# Deployment Guide: Secure HTTPS-Only Full Stack App on Ubuntu 22.04.5

## Overview

This guide describes how to deploy the ER\_Linux application (React frontend, Node/Express backend, MongoDB) on Ubuntu 22.04.5 with a wildcard SSL certificate ( `*.religare.in` ).

**All traffic is secured via HTTPS (port 443). No HTTP or remote MongoDB access is allowed.**

## Prerequisites

- Ubuntu 22.04.5 server with a static internal IP ( `10.91.41.16` )
- Wildcard SSL certificate ( `.religare.in` ) from your CA
- DNS record for your subdomain (e.g., `erapp.religare.in` ) pointing to your server
- Git access to your project repository
- `.env` files for backend and frontend, properly configured

## 1. System Preparation

```
sudo apt update && sudo apt upgrade -y  
sudo apt install -y git curl build-essential ufw nginx
```

## 2. Install Node.js, npm, and PM2

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -  
sudo apt install -y nodejs  
sudo npm install -g pm2
```

## 3. MongoDB 7.0 Manual Installation (.deb method)

## Step 1: Download .deb Packages (done on internet machine)

Download the following from:

<https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/7.0/multiverse/binary-amd64/>

Required:

```
mongodb-org-database_7.0.8_amd64.deb
mongodb-org-mongos_7.0.8_amd64.deb
mongodb-org-server_7.0.8_amd64.deb
mongodb-org-shell_7.0.8_amd64.deb
mongodb-org-tools_7.0.8_amd64.deb
mongodb-mongosh_2.2.4_amd64.deb
```

Upload all files to a GitHub repo folder.

## Step 2: Clone on Target Server

```
cd /opt
git clone https://github.com/your-org/mongo-offline-debs.git
cd mongo-offline-debs
```

## Step 3: Install All .deb Packages

```
sudo dpkg -i *.deb || true
sudo apt-get install -f -y
sudo dpkg -i *.deb
```

## Step 4: Enable and Start MongoDB

```
sudo systemctl enable mongod
sudo systemctl start mongod
```

## Verify Installation

```
mongod --version
mongosh
```

Both should work and show 7.0.x versions.

## 4. Configure MongoDB Authentication

### 1. Create admin and application users:

```
mongosh
```

```
use admin
db.createUser({
  user: "admin",
  pwd: "your-admin-password",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
})

use restrict_app
db.createUser({
  user: "restrict_user",
  pwd: "random",
  roles: [ { role: "readWrite", db: "restrict_app" } ]
})
exit
```

### 2. Enable authentication:

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

Add or edit:

```
security:
  authorization: enabled
net:
  port: 27017
  bindIp: 127.0.0.1
```

Note: bindIp: 127.0.0.1 ensures MongoDB is only accessible locally.

### 3. Restart MongoDB:

```
sudo systemctl restart mongod
```

## 5. Configure Firewall

```
sudo ufw allow OpenSSH
sudo ufw allow 443
sudo ufw enable
```

Only ports 22 (SSH) and 443 (HTTPS) are open. Port 27017 is NOT open.

## 6. Place SSL Certificate

```
sudo mkdir -p /etc/ssl/yourdomain
sudo cp yourdomain.com.crt /etc/ssl/yourdomain/
sudo cp yourdomain.com.key /etc/ssl/yourdomain/
```

```
/etc/ssl/religare/religare.in.crt
/etc/ssl/religare/religare.in.key
```

## 7. Clone the Project

```
cd /opt
sudo git clone <repo-url> ER_Linux
cd ER_Linux
```

## 8. Install and Start Backend

```
cd /opt/ER_Linux/server
npm install
pm2 start index.js --name er-backend
pm2 save
```

## 9. Build Frontend

```
cd /opt/ER_Linux/frontend
npm install
npm run build
```

## 10. Configure Nginx for HTTPS

```
sudo nano /etc/nginx/sites-available/erapp
```

Paste:

```
server {
    listen 443 ssl;
    server_name erapp.religare.in;

    ssl_certificate /etc/ssl/religare/religare.in.crt;
```

```

ssl_certificate_key /etc/ssl/religare/religare.in.key;

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

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

location /api/ {
    proxy_pass http://localhost:3000/;
    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/erapp /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```

## 11. Update Backend CORS Origin

In `/opt/ER_Linux/server/index.js` :

```

app.use(cors({
  origin: 'https://erapp.religare.in', // Or IP Of The Server
  credentials: true
}));

```

## 12. Test the Application

- Open: `https://erapp.religare.in`
  - Ensure the browser shows a secure (padlock) icon and the app loads.
- 

## 13. Security Notes

- Only ports 22 and 443 are open.
  - MongoDB is only accessible locally ( `127.0.0.1` ).
  - All traffic is encrypted via HTTPS.
  - No HTTP (port 80) is available.
- 

## Troubleshooting

- **Nginx errors:** Check with `sudo nginx -t` and `sudo tail -f /var/log/nginx/error.log`
  - **Backend errors:** Check with `pm2 logs er-backend`
  - **MongoDB errors:** Check with `sudo journalctl -u mongod`
- 

Shukriya 🙌 - Divyansh Singh Parihar 😊