# **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. Install MongoDB**

wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list

sudo apt update

sudo apt install -y mongodb-org

sudo systemctl enable mongod

sudo systemctl start mongod

## **4. Configure MongoDB Authentication**

**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

**Enable authentication:** sudo nano /etc/mongod.conf

Add or edit:  
  
 security:

authorization: enabled

net:

port: 27017

bindIp: 127.0.0.1

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

**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/

sudo cp ca-bundle.crt /etc/ssl/yourdomain/ # If provided

## **7. Clone the Project**

cd /opt

sudo git clone <repo-url> ER\_Linux

sudo chown -R $USER:$USER ER\_Linux

cd ER\_Linux

## **8. Place .env Files**

* Copy your .env files to:  
  + /opt/ER\_Linux/server/.env
  + /opt/ER\_Linux/frontend/.env

## **9. Install and Start Backend**

cd /opt/ER\_Linux/server

npm install

pm2 start index.js --name er-backend

pm2 save

## **10. Build Frontend**

cd /opt/ER\_Linux/frontend

npm install

npm run build

## **11. 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.com.key;

ssl\_trusted\_certificate /etc/ssl/religare/ca-bundle.crt; # If provided

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

## **12. Update Backend CORS Origin**

In /opt/ER\_Linux/server/index.js:

app.use(cors({

origin: 'https://erapp.religare.in',

credentials: true

}));

## **13. (Optional) Enable PM2 Startup on Boot**

pm2 startup

# Follow the instructions printed to run the command as root

pm2 save

## **14. Test the Application**

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

## **15. 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