Flexbo-en Dokku Deployment Guide

Flexbo-en Deployment with Dokku (Buildpacks + Procfile, No Dockerfile)

This guide documents the **complete deployment process** for the `flexbo-en` app on a VPS using **Dokku v0.35.20**, with **Buildpacks**, a **Procfile**, and **Lets Encrypt SSL**. It also includes all the advanced debug steps taken to resolve internal Dokku plugin issues.

Architecture Overview

Components

Component	Description
Frontend	React + Vite (built to `/dist`)
Backend	Node.js (Express server for `/api/forward`)
Static files	Served from `/media` (persistent mount)
Email	Handled via Resend API

Dockerfile: Not Used

- The Dockerfile was **renamed to `Dockerfile.bak`**.
- Instead, **Buildpacks** + **Procfile** architecture was used.

Why Buildpacks + Procfile

Reason	Benefit
Clean separation	Node server handles both API and React
Dokku-native behavior	Better integration with plugins
Easier SSL/NGINX integration	Uses internal NGINX generation

Step-by-Step Deployment

1. Create the App

```bash
dokku apps:create flexbo-en
```

2. Set Buildpack

```bash
dokku buildpacks:set flexbo-en https://github.com/heroku/heroku-buildpack-nodejs
dokku builder:set flexbo-en selected herokuish
```

3. Remove Dockerfile

```bash
mv Dockerfile Dockerfile.bak
git commit -am "Disable Dockerfile for Buildpacks"
```

4. Ensure `Procfile` exists

```
web: node server_resend/server.js
```

5. Set Environment Variables

```bash
dokku config:set flexbo-en NODE_ENV=production
dokku config:set flexbo-en RESEND_API_KEY=your_api_key_here
```

6. Mount Persistent `/media`

```bash
dokku storage:mount flexbo-en /var/lib/dokku/data/storage/flexbo-en-media:/media
```

7. Proxy Port Mapping

```bash
dokku config:set flexbo-en DOKKU_PROXY_PORT_MAP="http:80:5000 https:443:5000"
```

8. Push to Deploy

```bash
git push dokku main
```

Set Up HTTPS with Let's Encrypt

```bash
dokku letsencrypt:enable flexbo-en
dokku letsencrypt:cron-job --add
```

Debug Log: What Went Wrong and Fixes

Problem 1: `nginx.conf` not generated

- Cause: Dokkus `nginx-vhosts` plugin was corrupted

- Fix:
  ```bash
  mv /var/lib/dokku/plugins/available/nginx-vhosts{,.broken}
  git clone https://github.com/dokku/dokku.git dokku-core-tmp
  cp -r dokku-core-tmp/plugins/nginx-vhosts /var/lib/dokku/plugins/available/
  ln -s ../available/nginx-vhosts /var/lib/dokku/plugins/enabled/
  ```

Problem 2: `PR_CONNECT_RESET_ERROR`

- Cause: Missing NGINX config and SSL proxy misrouting
- Fix: Reinstall NGINX plugin + regenerate `nginx.conf`

Problem 3: `builder selected: empty`

- Fix: Explicitly set builder
  ```bash
  dokku builder:set flexbo-en selected herokuish
  ```

After Fixes:

- `nginx.conf` correctly generated at `/home/dokku/flexbo-en/nginx.conf`
- SSL worked
- React + Node app deployed over HTTPS

Final Notes

- Dokku v0.35.20 can misbehave if plugins are corrupted
- Always verify:
  - `dokku report flexbo-en`
  - `builder selected: herokuish`
  - `nginx.conf` exists
  - `lets_encrypt:report` shows active cert

CI/CD with GitHub Actions

# CI/CD Deployment with GitHub Actions

## Setup `.github/workflows/deploy.yml`

```yaml
name: Deploy to Dokku

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Push to Dokku
        env:
          DOKKU_HOST: ${{ secrets.DOKKU_HOST }}
          DOKKU_APP: flexbo-en
          DOKKU_KEY: ${{ secrets.DOKKU_KEY }}
        run: |
          mkdir -p ~/.ssh
          echo "$DOKKU_KEY" > ~/.ssh/id_rsa
          chmod 600 ~/.ssh/id_rsa
          ssh-keyscan -H "$DOKKU_HOST" >> ~/.ssh/known_hosts
          git remote add dokku dokku@$DOKKU_HOST:$DOKKU_APP
          git push dokku main -f
```

## GitHub Secrets to Add

- `DOKKU_HOST` your VPS IP or domain (e.g. 209.145.61.113)
- `DOKKU_KEY` your private SSH key with Dokku access

This workflow will deploy your app automatically on every push to `main`.