

The Problem: A "Stuck" Configuration

Your project on Render is "stuck" thinking it needs a Dockerfile. Since you can't change the setting directly, trying to fix it on the Render dashboard is a dead end.

The Solution: Tell Render What to Do from the Beginning

We will force Render to recognize your project as a **Node.js application** from the moment it connects to your repository. We will do this by adding a single configuration file to your GitHub project.

This approach bypasses Render's faulty auto-detection and gives it explicit, undeniable instructions.

The Definitive Fix: Using a `render.yaml` File

You are going to create a new file named `render.yaml` in your project. This file is like a master instruction manual that Render will read *before* it tries to guess anything.

Here is your clear, step-by-step action plan:

Step 1: Create the `render.yaml` File

1. In your `nexus-7.0-live` project folder on your computer, create a new file at the very root level (in the same directory as your `package.json`).
2. Name the file exactly: `render.yaml`
3. Copy and paste the following code into this new file. This code has been written specifically for your project.

Generated yaml

```
# =====
# == BWGA Nexus 7.0 - Definitive Render Blueprint
# == This file tells Render exactly how to build and deploy the app.
# =====

services:
  # This is your main Next.js Web Service
  - type: web
    name: nexus-7-0-live
    # IMPORTANT: This tells Render to use its native Node.js environment
```

```

runtime: node
plan: free # Explicitly set the plan
branch: main # Or master, depending on your GitHub branch name
buildFilter:
  paths:
    - "app/**"
    - "prisma/**"
    - "public/**"
    - "package.json"
    - "next.config.js"
    - "tsconfig.json"
# The commands to build your Next.js application
buildCommand: "npm install && npx prisma generate && npm run build"
# The command to start your application
startCommand: "npm start"
envVars:
  - key: DATABASE_URL
    fromDatabase:
      name: nexus-db # This must match the name of your database service below
      property: connectionString
  - key: NEXTAUTH_URL # Important for authentication
    value: https://nexus-7-0-live.onrender.com # Use your actual Render URL
  - key: NEXTAUTH_SECRET
    generateValue: true # Let Render generate a secure secret for you

# This is your PostgreSQL Database Service
- type: psql
  name: nexus-db
  plan: free # Explicitly set the plan
# No other configuration needed for the database

```

content_copydownload

Use code [with caution](#).Yaml

Step 2: Push the New File to GitHub

1. Save the `render.yaml` file.
2. Commit this new file to your GitHub repository and push the changes.

Generated bash

```

git add render.yaml
git commit -m "Add Render Blueprint for Node.js deployment"
git push

```

content_copydownload

Use code [with caution](#).Bash

Step 3: Create a New Project on Render Using the Blueprint

Since your current Render service is "stuck," the cleanest and most reliable method is to create a new one using your new blueprint.

1. **Delete the old service on Render:** Go to your Render dashboard, select the stuck `nexus-7-0-live` service, go to "Settings," scroll to the bottom, and delete it. This ensures a clean slate.
2. **Create a New Project:** On the main Render dashboard, click "**New +**" and then select "**Blueprint.**"
3. **Connect Your Repository:** Select your `nexus-7-0-live` repository from the list. Render will automatically detect the `render.yaml` file.
4. **Approve the Plan:** Render will show you the two services it's about to create: the `nexus-7-0-live` web service and the `nexus-db` database. Click "Approve" or "Create."