## NestJs & Pm2 DEPLOYMENT PROCESS

**Step 1: Prepare Your NestJS Project for Deployment**

Build the Project: Make sure your project is ready for deployment by building it. Run the following command to compile TypeScript into JavaScript:

**npm run build**

This will create the dist/ folder containing the compiled files, which will be used to run the application.

Ensure .env File is Set Up (if applicable): Your NestJS application might rely on environment variables. If you're using a .env file (for database credentials, API keys, etc.), make sure it’s properly set up and included in your project.

**Example .env file:**

**DB\_TYPE=mysql**

**DB\_HOST=localhost**

**DB\_PORT=3306**

**DB\_USERNAME=your-username**

**DB\_PASSWORD=your-password**

**DB\_DATABASE=your-database**

**Step 2: Prepare the Server (Target Machine)**

Before you deploy your NestJS application, ensure the following requirements are met on the target server.

Install Node.js and NPM: Ensure Node.js and NPM are installed on the server. You can check the versions using:

**node -v**

**npm -v**

If not installed, follow the installation guide for Node.js on your system.

Install PM2 (Optional but Recommended): PM2 is a process manager that will keep your app running even if the server is restarted. Install PM2 globally:

**npm install -g pm2**

Set up Database (if applicable): Ensure that your database (e.g., MySQL, PostgreSQL) is properly set up and accessible from the server where your app will run. Ensure that any database-specific configurations in your .env file are correct.

**Step 3: Transfer the Code to the Server**

You need to copy your project files to the server. You can use tools like SCP, FTP, or any other method you prefer.

Copy the files: Copy the entire project folder (including dist/, package.json, .env, and any other necessary files) to the server.

Example using SCP:

**scp -r /path/to/your/project user@client-server:/path/to/destination/**

Navigate to the project directory: SSH into the server and navigate to the folder where you copied your project:

**cd /path/to/destination/**

**Step 4: Install Dependencies on the Server**

Once the project is transferred to the server, you need to install the required dependencies.

Install Dependencies: Run npm install to install the dependencies as per your package.json:

**npm install**

Verify the .env file (if you're using one): Make sure your .env file is present on the server in the correct location (typically in the root of your project folder).

**Step 5: Set Up PM2 for Process Management**

Using PM2 ensures that your application runs in the background and restarts on failure or system reboot.

Create ecosystem.config.js for PM2 (Optional but recommended): This configuration file helps you manage multiple environments and other configurations (like environment variables).

**Example ecosystem.config.js:**

**module.exports = {**

**apps: [**

**{**

**name: 'nestjs-app',**

**script: 'dist/main.js',**

**instances: 'max',**

**exec\_mode: 'cluster',**

**env: {**

**NODE\_ENV: 'development',**

**DB\_TYPE: 'mysql',**

**DB\_HOST: 'localhost',**

**DB\_PORT: 3306,**

**DB\_USERNAME: 'your-username',**

**DB\_PASSWORD: 'your-password',**

**DB\_DATABASE: 'your-database',**

**},**

**env\_production: {**

**NODE\_ENV: 'production',**

**DB\_TYPE: 'mysql',**

**DB\_HOST: 'prod-db-host',**

**DB\_PORT: 3306,**

**DB\_USERNAME: 'prod-username',**

**DB\_PASSWORD: 'prod-password',**

**DB\_DATABASE: 'prod-database',**

**},**

**},**

**],**

**};**

Start the Application Using PM2: You can now start your NestJS application using PM2:

**pm2 start ecosystem.config.js --env production**

If you don’t have an ecosystem.config.js file, you can directly start the app:

**pm2 start dist/main.js --name "nestjs-app"**

Enable PM2 to Restart on Server Reboot: To ensure PM2 restarts your application if the server reboots, use:

**pm2 startup**

Then, save the PM2 process list:

**pm2 save**

**Step 6: Verify the Application is Running**

Check PM2 Process List: To verify your app is running, check the PM2 process list:

**pm2 list**

**Step 7:View Logs:**

You can view the logs for your NestJS app to ensure everything is running correctly:

**pm2 logs nestjs-app**