Deploying the class manager

# Requirements List

* Postgresql database <https://www.postgresql.org/download/>
* Node js 10.16 LTS <https://nodejs.org/en/download/>
* NPM or Yarn <https://yarnpkg.com/en/docs/install#windows-stable>
* Golang version 1.12 <https://golang.org/dl/>
* NGINX http://nginx.org/en/download.html

# Deployment

1. Git clone project using git clone https://github.com/emendozaspx/classmanager

## Setup Database

1. Open postgres terminal with postgres user

On Linux

sudo -u postgres psql

On Windows

Open search > sql shell (psql)

1. Enter the following command to create database CREATE DATABASE myproject;
   1. Do not forget to include the semicolon ;
   2. Replace red with stated value
2. Enter the following command to create postgres user CREATE USER user WITH PASSWORD secure\_password;
3. Enter the following command for a given user GRANT ALL PRIVALAGES ON DATABASE myproject TO user;
4. Quit the terminal with \q
5. Repeat step 1 except using the ‘user’ created instead of ‘postgres’
6. If on windows create tables using pgadmin
   1. Select the database created, the lightning bolt (Query tool) on the selection panel, it should open a window.
   2. Select the folder icon and open the file in classmanager/sql/setup.sql
   3. Press the lightning bolt icon to run sql
7. If on linux simply run \i classmanager/sql/setup.sql

## Building the frontend application

1. Run git submodule init
2. Run git submodule update
3. Change directory by cd classmanager/web
4. Install a local yarn yarn install or npm install
5. Run yarn build or npm run build

## Building application

1. If not already return to classmanager folder
2. Run go build ./cmd/server
3. Run
   1. On linux ./server
   2. On windows .\server.exe

## Deploying into production

* This only applies to linux because as of right now I am unaware of how to deploy on windows using nginx

1. Update the file config.yaml to suit your database configuration and also delete the secret key
2. Create a new file using a text editor or sudo nano /etc/sites-available/project
3. Create a nginx config file, see example provided in the installation folder
4. Replace example.com with your domain name
5. Replace localhost:8001 with the localhost with your port number
6. Optionally you can add ssl with this tutorial <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04#step-2-%E2%80%94-confirming-nginx's-configuration>
7. Run sudo nginx –t this verifies the nginx config is working
8. Run sudo systemctl reload nginx this restarts the server