1. **Choosing a server platform**

List server platforms: AWS, Firebase, …

* Choose and create a server. OS should be Ubuntu 20.04 LTS.
* Use any terminal to connect to the server by SSH.
* After connecting to the server, run “apt update” & “apt upgrade” to update the server.
* Use “apt search postgres” command to find the latest “postgresql-<xx>” version, then run “apt install postgresql-<xx>” to install postgresql. Run “service postgresql status” to check if the postgres is running.
* Choose a web server such as: NGINX, Caddy, … Find commands to install it in ubuntu and run those commands in terminal to install to server.
* Run “apt install supervisor” command to install supervisor.

1. **Adding admin user to migrations**

* Open terminal at source and run “soda generate sql AddUserToUserTable”.
* Copy SQL command to add admin user to “.up.sql” file.
* For “.down.sql” file, can put any SQL command or not.

1. **Installing GO on the server and get code up to server**

Root terminal:

* Remote to server as “root”.
* Run “adduser <username>” and enter <password> to create new user.
* Run “usermode -aG sudo <username>” to allow new user to write command without sudo.

User terminal:

* Open another terminal and use SSH command to connect as <username>. Run “sudo ls” to enter password.
* Find the <link> for the latest GO version of Linux, run “wget <link>” to install it to server.
* Open “installation instructions” at GO website, find “Linux” version.
  + For the 1st command, copy from “tar …” and run “sudo tar …”.
  + Run “vi .profile” to open “.profile” file, press “i” key to change to insert mode, add “export PATH=$PATH:/user/local/go/bin” at the end of file, type “:wq” to save and quit.
  + Run “export PATH=$PATH:/user/local/go/bin “. Run “which go” & “go version” to check.
* Run “which git” to check and install git if necessary.
* Run “git clone <link>” to clone project to server.
* Run “ls” to check if the repo was cloned or not.

1. **Setting up the remote database and building the application on server**

User terminal:

* Run “pwd”. Run “cd /etc/postgresql/<version>/main.
* Run “sudo vi pg\_hba.conf”, type “:1000” to move the end of file. Find “IPv4” & “IPv6”, changed the final column to “trust. Run “:wq”.
* Run “sudo service postgresql stop” & “sudo service postgresql start” to restart. Run “ps ax | grep postgr” to check.

DBeaver:

* Create a new connection, choose postgres.
* Open SSH tab, check “Use SSH tunnel”, input IP address of server at Host/IP. Input Username & Password of server (which has been created by root). Implementation: JSch. Press “Test Connection” to test then press” Connect” to create the connection.
* Create a new database at the connection.

User terminal:

* Run “cd” & “cd <repo name>” to move to repo.
* Run “cp database.yml.example database.yml”.
* Run “vi database.yml”, update “user:” & “password:”, remove “test:” & “production:” sections.
* Run “go get github.com/gobuffalo/pop/…” to install gobuffalo. Run “ls ~/go/bin” to check if there is “soda” or not.
* Run “vi .profile”, add “export PATH=$PATH:~/go/bin”. Run “export PATH=$PATH:~/go/bin”. Run “which soda” to check.
* Go to repo’s directory, run “soda migrate” and check in DBeaver if there are all tables or not.
* Run “go build -o <repo’s name> cmd/web/\*.go”.

1. **Connecting the application to the web server**

Open user terminal:

* Run “cd /etc/caddy/”
* Run “sudo mv Caddyfile Caddyfile.dist”
* Run “sudo vi Caddyfile” and insert to the file

|  |
| --- |
| {  email admin@here.com  }  (static) {  @static {  file  path \*.ico \*.css \*.js \*.gif \*.jpg \*.jpeg \*.png \*.svg \*.woff \*.json  }  header @static Cache-control max-age=5184000  }  (security) {  header {  # enable HSTS  Strict-Transport-Security max-age=3156000;  # disable clients from sniffing media type  X-Content-Type-Options nosniff  # keep referrer data off of HTTP connections  Referrer-Policy no-referrer-when-downgrade  }  }  Import conf.d/\*,conf |

* Run “sudo mkdir conf.di” & “sudo mv conf.di/ conf.d” & “cd conf.d”
* Run “sudo vi book.conf”
* Open sever platform, find DNS of server (url with full texts). Copy it to VI editor of book.conf. Content of book.conf.

|  |
| --- |
| <url-text> {  encode zstd gzip  import static  import security  log {  output file /var/www/book/logs/caddy-acces.log  format single\_field common\_log  }  reverse\_proxy http://localhost:8080  } |

* Run “cd /var” & “sudo mkdir www” & “cd www”.
* Run “ls ~” to check home directory, ensure there are bookings, go & go version folders.
* Then run “sudo mv ~/bookings book” to move bookings to www folder.
* Run “cd book” & “mkdir logs”.
* Run “sudo chmod 777 logs” to check (empty is ok).
* Run “sudo service caddy restart” & “sudo service caddy status” to restart & check status.
* Run “./bookings -h” to check list input to run the application.
* Run “./bookings -dbName=bookings -dbUser=postgres -dbPass=24072001do” to run the application.
* Finally, copy the url texts and run the browser to test (the website must run).

1. **Setting up Supervisor**

*Note: Supervisor to help the web run all time (don’t need to run manually).*

Open user terminal:

* Run “cd /etc/supervisor/” & “ls”, check if there is conf.d or not.
* Run “cd conf.d” & “ls”, there should be no file.
* Run “sudo vi book.conf” and insert:

|  |
| --- |
| [program:book]  command=/var/www/book/bookings - dbName=bookings -dbUser=postgres -dbPass=24072001do  directory=/var/www/book  autorestart=true  autostart=true  stdout\_logfile=/var/www/book/logs/supervisord.log |

* Run “sudo supervisorctl”. Run “status” (should be nothing). Run “update” & “status”, there should be a “book” row. From this time, the web will run without command from terminal.

1. **Writing an update script for the server**

Open user terminal:

* Run “cd /var/www/book/”.
* Run “vi update.sh” and insert:

|  |
| --- |
| #!/bin/bash  git pull  soda migrate  go build -o bookings cmd/web/\*.go  sudo supervisorctl stop book  sudo supervisorctl start book |

* Run “chmod 777 update.sh” & “./update.sh” to run update script. From now, just run “./update.sh” to start update.