Inštalačná Dokumentácia

Tento dokument obsahuje inštalačnú dokumentáciu pre aplikáciu, vrátane krokov na migráciu databázy pomocou Django, postup zálohovania databázy s použitím cronjobu a jej manuálneho obnovenia.

#### Vytvorenie databázy:

Spustite nasledujúci príkaz na vytvorenie novej databázy s názvom new\_app\_macky:

createdb -U postgres -h localhost new\_app\_macky

* -U postgres: Určuje používateľa databázy (v tomto prípade postgres).

1. -h localhost: Určuje, že sa pripojíme k lokálnemu serveru PostgreSQL na adrese localhost.
2. new\_app\_macky: Názov novej databázy, ktorú chcete vytvoriť.

#### Aplikovanie Migrácií Databázy:

Nasledujúce príkazy slúžia na aplikovanie migrácií, čo zabezpečí správnu štruktúru databázy pre vašu aplikáciu Django:

#### Vytvorenie migrácií:

python manage.py makemigrations

Tento príkaz vytvorí migrácie na základe zmien definovaných vo vašich modeloch.

#### Aplikovanie migrácií:

python manage.py migrate

Tento príkaz aplikuje vytvorené migrácie na databázu a vytvorí alebo upraví tabuľky podľa definícií modelov vo vašej aplikácii.

#### Exportovanie základných údajov:

Predpokladá sa, že máte exportovaný .sql súbor (app\_macky\_dump\_base.sql), ktorý obsahuje základné údaje potrebné pre vašu aplikáciu. Na importovanie tohto súboru do databázy použite nasledujúci príkaz:

psql -U postgres -h localhost -d app\_macky -f app\_macky\_dump\_base.sql

* -U postgres: Určuje používateľa databázy (v tomto prípade postgres).
* -h localhost: Určuje, že sa pripojíme k lokálnemu serveru PostgreSQL na adrese localhost.
* -d app\_macky: Určuje názov existujúcej databázy (app\_macky), do ktorej chcete importovať údaje.
* -f app\_macky\_dump\_base.sql: Určuje názov a cestu k SQL súboru, ktorý obsahuje údaje na importovanie do databázy.

Týmto spôsobom sa vytvorí a naplní vaša databáza s potrebnými údajmi pre správny chod vašej Django aplikácie. Uistite sa, že názvy databázy a súbory sú správne a zodpovedajú vašej konfigurácii.

#### Nastavenie aplikácie

V súbore settings.py v konfigurácii premenných DATABASES nastavíme potrebné parametre na pripojenie k databáze a tiež na určenie IP adresy, na ktorej bude server Django bežať. Tu je zjednodušený príklad nastavenia:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'app\_macky',

'USER': 'postgres',

'PASSWORD': 'postgres',

'HOST': 'localhost',

'PORT': '5432',

}

}

Popis parametrov:

* 'ENGINE': 'django.db.backends.postgresql': Určuje použitý databázový engine. V tomto prípade sa používa PostgreSQL.
* 'NAME': 'app\_macky': Určuje názov databázy, do ktorej sa server Django pokúsi pripojiť.
* 'USER': 'postgres': Určuje používateľské meno, ktoré sa použije na autentifikáciu pri pripájaní k databáze.
* 'PASSWORD': 'postgres': Určuje heslo používateľa, ktoré sa použije na autentifikáciu pri pripájaní k databáze.
* 'HOST': 'localhost': Určuje adresu (IP alebo doménové meno), na ktorej beží databázový server. V tomto prípade je nastavené na localhost, čo znamená, že server databázy beží na tom istom počítači ako Django server. Ak je databázový server na inom stroji, môžete tu zadať IP adresu tohto stroja.
* 'PORT': '5432': Určuje port, na ktorom databázový server počúva na pripojenia. Predvolený port pre PostgreSQL je 5432.

Poznámka:

* Ak chcete server Django spustiť na inej IP adrese ako localhost (napríklad na verejnej IP adrese alebo na IP adrese v rámci Docker kontajnera), zmeňte hodnotu parametra 'HOST' na príslušnú IP adresu (napríklad 147.175.150.172).
* Uistite sa, že údaje v settings.py sú správne a zodpovedajú konfigurácii vašej databázy (napríklad názov databázy, používateľské meno, heslo a IP adresa servera databázy).

#### Build kontajnera:

sudo docker build -t app\_macky\_django .

Tento príkaz vytvorí nový Docker obraz s názvom app\_macky\_django na základe súboru Dockerfile v aktuálnom adresári.

#### Spustenie kontajnera:

sudo docker run -v /home/melikant/BP/aplikacia\_macky/media/:/app/media/ -p 80:80 app\_macky\_django

Tento príkaz spustí kontajner z vytvoreného obrazu app\_macky\_django:

* -v /home/melikant/BP/aplikacia\_macky/media/:/app/media/: Tento parameter zabezpečí mapovanie priečinku /home/melikant/BP/aplikacia\_macky/media/ na priečinok /app/media/ vo vnútri kontajnera. Toto je užitočné pre zdieľanie dát medzi hostiteľským strojom a kontajnerom a v našom prípade mapujeme priečinok s obrázkami.
* -p 80:80: Tento parameter mapuje port 80 z hostiteľského stroja na port 80 vo vnútri kontajnera. To umožní prístup k aplikácii, ktorá beží vo vnútri kontajnera na portu 80 zvonka.
* app\_macky\_django: Toto je názov obrazu, ktorý sa má spustiť vo forme kontajnera.

Po spustení tohto príkazu by mala vaša aplikácia bežať vo vnútri Docker kontajnera app\_macky\_django a byť prístupná na porte 80 hostiteľského stroja. Pred spustením uistite sa, že ste postavili obraz pomocou príkazu docker build a že máte správne nastavené mapovanie priečinkov a portov podľa potreby vašej aplikácie.

#### Zastavenie Kontajneru a Čistenie Databázy (Voliteľné):

Po úspešnej obnove môžete zastaviť kontajner a odstrániť databázu "new\_app\_macky", ak už nie je potrebná.

Najprv zistíme aké ID má náš kontainer:

sudo docker ps -l

teraz použijeme {CONTAINER ID} pre zastavenie procesu

docker stop {CONTAINER ID}

#### Zálohovanie s Cronjobom (Každý 14. Deň v mesiaci)

1. Konfigurácia Cronjobu:

Pridajte úlohu cronjobu na automatické zálohovanie databázy každý 14. deň.

crontab -e

1. Pridanie Záznamu do Cronu:

Vložte nasledujúci záznam do cronu pre zálohovanie databázy.

0 0 \*/14 \* \* sudo pg\_dump -U postgres -h localhost -d app\_macky -W > /mnt/data/app\_macky\_dump\_$(date +\%Y\%m\%d).sql

* 0 0 \*/14 \* \*: Tento časový záznam definuje, že príkaz sa spustí o polnoci (00:00) každých 14 dní (hoci \*/14 zabezpečuje, že sa to stane vždy, kedy je delitelne 14 a teda toto znamená každých 14 dní)
* sudo pg\_dump -U postgres -h localhost -d app\_macky -W: Toto je príkaz na vytvorenie zálohy databázy. pg\_dump je nástroj na zálohovanie PostgreSQL databáz. -U postgres definuje používateľa databázy (postgres v tomto prípade), -h localhost určuje, že databáza beží na lokálnom hostiteľovi, -d app\_macky zvolí konkrétnu databázu s názvom app\_macky a -W vyžiada heslo (zobraziť heslo, ktorý sa snaží získat bez použitia hesla)

Tento záznam spustí príkaz `pg\_dump` každých 14. deň v mesiaci a uloží zálohu do súboru s dátumom.

Týmto spôsobom môžete nainštalovať aplikáciu "app\_macky\_django", aplikovať migrácie databázy pomocou Django, zálohovať a obnoviť databázu s pomocou cronjobu. Upravte cesty a údaje podľa potreby vašej konfigurácie.

#### Obnova Databázy zo Zálohy:

Obnovte databázu "new\_app\_macky" zo zálohy `app\_macky\_dump.sql`.

psql -U postgres -h localhost -d new\_app\_macky -f app\_macky\_dump.sql