Inštalačná Dokumentácia

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

Pre všeobecné premenné budem používať <premenná>, čo znamená, že <premenná> je označenie, ktoré sa používa na reprezentáciu akéhokoľvek typu premenných alebo hodnôt v rámci daného kontextu. Táto notácia nám umožňuje používať jednotný symbol (v tomto prípade <premenná>) na odkazovanie sa na rôzne premenné, názvy, hodnoty alebo iné entity, ktoré môžu byť ľubovoľného typu alebo obsahu. Pri použití tohto označenia je dôležité si uvedomiť, že ide len o symbolický názov a konkrétna premenná alebo hodnota by mala byť nahradená konkrétnym obsahom, ktorý je relevantný pre danú situáciu alebo úlohu.

Obsah

1 Vytvorenie databázy:....2
1.1 Nastavenie postgresql:....2
1.1.1 Zistenie umiestnenia hlavného konfiguračného súboru:....2
1.1.2 Úprava konfiguračného súboru postgresql.conf....2
1.1.3 Nastavenie parametra listen_addresses:....2
1.1.4 Úprava súboru pg_hba.conf:....3
1.1.5 Reštartujte službu PostgreSQL pre aplikovanie zmien:....3
1.2 Nastavenie .pgpass súboru pre automatické pripojenie k databáze....3
1.2.1 Vytvorenie .pgpass súboru....3
1.2.2 Pridanie informácií o pripojení....3
1.3 Exportovanie základných údajov:....4
2 Nastavenie aplikáci....4
2.1 Zostavenie kontajnera:....5
2.2 Spustenie kontajnera:....5
2.3 Zastavenie Kontajneru a Čistenie Databázy (Voliteľné):....6
2.4 Zálohovanie (Každých 7 dní)....6
2.4.1 Priraďte práva na spustiteľnosť:....6
2.4.2 Konfigurácia crontab pre pravidelné spúšťanie skriptu:....6
2.4.3 Vytvorenie zálohovacieho adresára:....6
3 Obnova Databázy zo Zálohy:....6

1 Vytvorenie databázy:

Spustite nasledujúci príkaz na vytvorenie novej databázy:

```
createdb -U <postgresql meno> -h localhost <názov databázy>
```

-U <postgresql meno> určuje používateľa databázy.

-h localhost určuje, že sa pripojíme k lokálnemu serveru PostgreSQL na adrese localhost.

1.1 Nastavenie postgresql:

Pre úspešné fungovanie aplikácie potrebujeme spustiť a nakonfigurovať postgresql server.

Spustenie postgresql:

```
sudo systemctl start postgresql
```

1.1.1 Zistenie umiestnenia hlavného konfiguračného súboru:

Použite nasledujúci príkaz na zistenie umiestnenia hlavného konfiguračného súboru pre PostgreSQL:

```
sudo psql -U <postgresql meno> -h localhost -c 'SHOW config_file'
```

Tento príkaz vypíše cestu k hlavnému konfiguračnému súboru.

1.1.2 Úprava konfiguračného súboru postgresql.conf

Otvorte konfiguračný súbor postgresql.conf v textovom editore:

```
sudo nano <cesta_k_konfiguracnemu_suboru>
```

Namiesto <cesta_k_konfiguracnemu_suboru> dosadte cestu k súboru získanú pomocou príkazu SHOW config_file.

1.1.3 Nastavenie parametra listen_addresses:

```
listen_addresses = '*'
```

Tento parameter umožní PostgreSQL počúvať na všetkých rozhraniach.

1.1.4 Úprava súboru pg_hba.conf:

nájdite cestu k súboru `pg_hba.conf`, ktorý obsahuje pravidlá autentifikácie.

Otvorte súbor `pg_hba.conf` v textovom editore:

```
sudo vim <cesta_k_pg_hba_conf>
```

Pridajte nasledujúci riadok na koniec súboru:

```
host    all       all      0.0.0.0/0        md5
```

Tento riadok povolí prístup pre všetkých používateľov z ľubovoľnej IP adresy (0.0.0.0/0) pomocou autentifikácie typu md5.

1.1.5 Reštartujte službu PostgreSQL pre aplikovanie zmien:

```
sudo systemctl restart postgresql
```

Týmto spôsobom môžete úspešne upraviť konfiguráciu PostgreSQL na povolenie počúvania na všetkých rozhraniach a povoliť prístup z ľubovoľnej IP adresy cez autentifikáciu md5.

1.2 Nastavenie .pgpass súboru pre automatické pripojenie k databáze

1.2.1 Vytvorenie .pgpass súboru

Vytvorte súbor s názvom `.pgpass` v domovskom adresári. Súbor by mal mať nastavené povolenia `600`, aby bol prístupný len vám.

```
touch .pgpass
chmod 600 .pgpass
```

1.2.2 Pridanie informácií o pripojení

Otvorte `.pgpass` súbor v textovom editore a pridajte nasledujúci riadok:

```
localhost:5432:<postgresql meno>:<názov databázy>:<postgresql
heslo>
```

1.3 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 použite nasledujúci príkaz:

```
psql -U postgres -h localhost -d <názov databázy> -f
app_macky_dump_base.sql
```

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

2 Nastavenie aplikáci

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 spustený. Tu je zjednodušený príklad nastavenia:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': '<názov databázy>',
        'USER': '<postgresql meno>',
        'PASSWORD': '<postgresql heslo>',
        'HOST': '<IP ADRESA>',
        'PORT': '5432',
    }
}
```

Popis parametrov:

`'ENGINE': 'django.db.backends.postgresql'` určuje použitý databázový engine. V tomto prípade sa používa PostgreSQL.

`'USER': '<postgresql meno>'` určuje používateľské meno, ktoré sa použije na autentifikáciu pri pripájaní k databáze.

'PASSWORD': '<postgresql heslo>' určuje heslo používateľa, ktoré sa použije na autentifikáciu pri pripájaní k databáze.

'HOST': '<IP ADRESA>' určuje adresu (IP alebo doménové meno), na ktorej je spustený databázový server.

'PORT': '5432' určuje port, na ktorom databázový server počúva na pripojenia. Predvolený port pre PostgreSQL je 5432.

2.1 Zostavenie kontajnera:

sudo docker build -t <docker obraz názov>

Tento príkaz vytvorí nový docker obraz s názvom <docker obraz názov> na základe súboru dockerfile v aktuálnom adresári.

<docker obraz názov> je názov obrazu, ktorý sa má spustiť vo forme kontajnera.

V tomto súbore potrebujeme taktiež nastaviť port, na ktorom bude aplikácia spustená:

CMD ["python3", "manage.py", "runserver", "0.0.0.0:<port>"]

2.2 Spustenie kontajnera:

sudo docker run -v <MEDIA>:/app/media/ -p <port>:<port> -d <docker obraz názov>

Tento príkaz spustí kontajner z vytvoreného obrazu <docker obraz názov>:

- <MEDIA>: premenná predstavuje absolútnu cestu do adresára kde sa nachádza adresár s obrázkami

-v <MEDIA>:/app/media/: Tento parameter zabezpečí mapovanie priečinku <MEDIA> na priečinok /app/media/ vo vnútri kontajnera. Toto je potrebné pre zdieľanie dát medzi hostiteľským strojom a kontajnerom a v našom prípade mapujeme priečinok s obrázkami.

-p <port>:<port>: 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 zvonku.

-d spôsobí že docker bude bežať na pozadí.

2.3 Zastavenie Kontajneru a Čistenie Databázy (Voliteľné):

Po úspešnej obnove môžete zastaviť kontajner, ak už nie je potrebná.
Najprv zistíme aké ID má náš kontajner:

```
sudo docker ps -l
```

teraz použijeme `<CONTAINER ID>` pre zastavenie procesu

```
docker stop <CONTAINER ID>
```

2.4 Zálohovanie (Každých 7 dní)

súbor cronjob.sh slúži na periodické zálohovanie databázy a manipuláciu súborov v pravidelných intervaloch. Zálohy sa vytvárajú každý druhý deň od špecifikovaného začiatočného dátumu.

2.4.1 Priraďte práva na spustiteľnosť:

```
chmod +x cronjob.s
```

2.4.2 Konfigurácia crontab pre pravidelné spúšťanie skriptu:

Otvorenie crontab editora:

```
crontab -e
```

pridáme nasledujúci záznam:

```
00 00 * * * <cronjob.sh cesta>
```

2.4.3 Vytvorenie zálohovacieho adresára:

Vytvorte adresár pre ukladanie záloh databázy.

```
mkdir <zálohovací adresár>
```

Umožnite zápis do adresára, aby bol skript schopný uložiť zálohy.

```
sudo chmod 777 <zálohovací adresár>
```

Nakoniec prepíšte aj premenné pre cieľové adresáre v súbore cronjob.sh.

3 Obnova Databázy zo Zálohy:

Najprv musíme zmazať aktuálnu databázu a následne ju znova vytvoríme aby bola prázdna:
Je to hlavne z dôvodu že zálohovaný .sql súbor nemá kontrolu pre existenciu tabuliek.

```
dropdb -U <postgresql meno> -h localhost <názov databázy>
createdb -U <postgresql meno> -h localhost <názov databázy>
```

Ak máme vytvorené databázu, vykoname príkaz, ktorý obnoví údaje zo zálohy.

```
psql -U postgres -h localhost -d <názov databázy> -f <záloha>.sql
```