

Настройка периодического резервного копирования для СКЗ по SQL

Для создания резервных копий в Postgres нужно использовать команду `pg_dump`. Пример:

```
pg_dump -U DB_USER -d DB_NAME -F p > "BACKUP_PATH"
```

-U – флаг указания пользователя,

-d – флаг указания БД,

-F p – флаг указания формата вывода и сам формат вывода, в данном случае создаст файл с расширением `.sql`.

Создадим `sh` скрипт для периодического создание РК(резервных копий) БД(базы данных) в Postgres.

Листинг `auto_dumps.sh`:

```
# Данные необходимые для подключения бд
DB_NAME="sql_bootcamp"
DB_USER="postgres"
DB_PASSWORD="root"

# Директория для РК
BACKUP_DIR="/home/adpak/DS/S21_SQL/SQL_Dumps/dumps"

# Дата
DATE=$(date +"%Y%m%d%H%M")

# Имя файла РК
BACKUP_FILENAME="db_backup_${DB_NAME}_${DATE}.sql"

# Путь к файлу РК
BACKUP_PATH="$BACKUP_DIR/$BACKUP_FILENAME"

# Установка пароля
export PGPASSWORD="$DB_PASSWORD"

# Выполнение РК
pg_dump -U $DB_USER -d $DB_NAME -F p > "$BACKUP_PATH"

# Сброс переменной с паролем(для безопасности)
unset PGPASSWORD

echo "Резервная копия базы данных $DB_NAME сохранена как $BACKUP_PATH"
```

После создания скрипта нужно сделать его исполняемым, используя команду:

```
chmod +x auto_dumps.sh
```

Затем измените метод аутентификации, открыв файл `pg_hba.conf`, находящийся в папке `/etc/postgresql/<версия>/main`

Замените данные строки

```
# "local" is for Unix domain socket connections only
local  all          all
```

peer

на

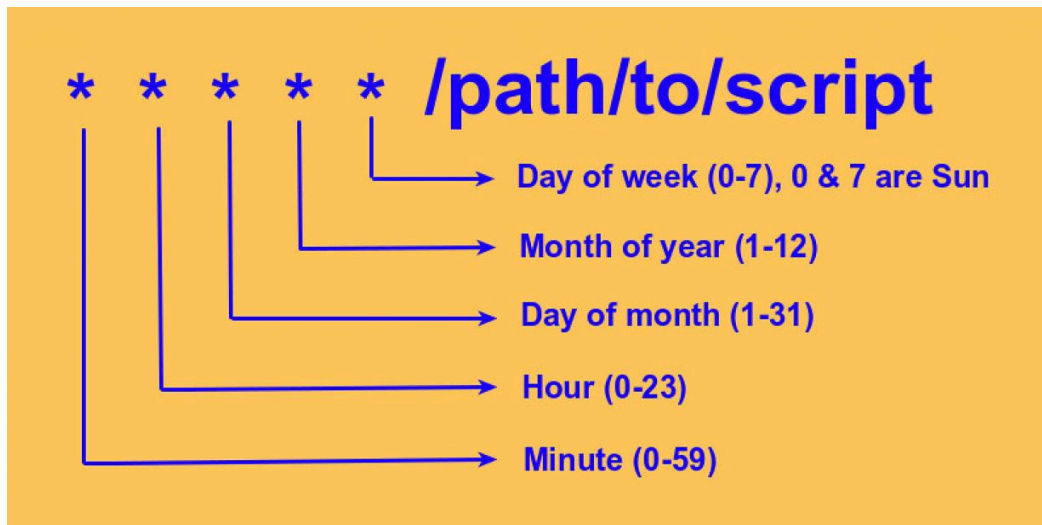
```
# "local" is for Unix domain socket connections only
local  all          all
```

password

Установите cron.

```
sudo apt-get update
sudo apt-get install cron
```

Введите `crontab -e` для редактирования файла cron для текущего пользователя. Ниже приведена структура написания задач планировщика.



Добавим в конце файла задачу:

```
00 04 * * * /home/adpak/DS/S21_SQL/SQL_Dumps/src/auto_dumps.sh
```

После сохранения изменений должно появиться сообщение:

```
crontab: installing new crontab
```