

PostgreSQL 安装

[FreeBSD Wiki](#) 提供了详细的安装指南。

2024年 01月 14日 在 FreeBSD 上安装 PostgreSQL 16。

以 root 用户身份运行。\$ 在终端中运行 `sudo` 命令。

安装

首先，安装 PostgreSQL 16 的服务器和客户端包。

```
$ pkg install postgresql16-server postgresql16-client
```

安装完成后，需要配置 PostgreSQL 16 的配置文件。

```
$ pkg search postgresql
```

通过 `sysrc` 命令配置 PostgreSQL 16 的启动选项。

```
$ sysrc postgresql_enable="YES"
```

配置完成后，需要启动 PostgreSQL 16 服务。

```
$ /usr/local/etc/rc.d/postgresql initdb
```

```
$ service postgresql start
```

安装完成后，可以通过 `sockstat` 命令检查 PostgreSQL 16 的监听端口。

```
$ sockstat -46 | grep 5432
```

```
<pid 用户 地址>
```

```
$ sockstat -4 | grep 5432 -> IPv4 地址
```

```
$ sockstat -6 | grep 5432 -> IPv6 地址
```

root@kali:~# postgres --help

```
$ sudo passwd postgres
```

Enter new password: foo_db
Retype new password: foo_db

```
# su - postgres (or sudo su - postgres if you are root)
$ createuser admin
$ createdb foo_db -O admin
```

root@kali:~# (psql -U postgres) postgres=# \du
Role name: admin
Type: normal role
Privileges: postgres,foo_db
root@kali:~# exit
postgres=# psql foo_db

```
$ psql foo_db
foo_db=# alter role admin with encrypted password 'yourpassword';
foo_db=# grant all privileges on database foo_db to admin;
foo_db=# exit
$ exit
```

root@kali:~# psql -U postgres -d foo_db -c 'select * from admin;'
psql: error: connection to database "foo_db" refused: no such database

```
$ dropdb foo_db
```

root@kali:~# psql -U postgres -d foo_db -c '?'

Bind

Bind

root@kali:~# psql -U postgres -d foo_db -c '(select * from admin)'
psql: error: connection to database "foo_db" refused: no such database
root@kali:~# bind address 127.0.0.1
data16: error: connection to database "foo_db" refused: no such database

```
sudo nano /var/db/postgres/data16/postgresql.conf
```

root@kali:~# listen_addresses = '127.0.0.1/24' # IP addresses to listen on.
root@kali:~# # IP address to listen on (default is #)
root@kali:~#

```
listen_addresses = '*'
```



```
sudo nano /var/db/postgres/data16/pg_hba.conf
```

```

def md5(filename):
    """Return the MD5 hash of the file at filename.

    The MD5 hash is a 32-character hexadecimal string.

    """
    import hashlib
    m = hashlib.md5()
    with open(filename, 'rb') as f:
        for chunk in iter(lambda: f.read(4096), b''):
            m.update(chunk)
    return m.hexdigest()

```

```
# TYPE DATABASE USER ADDRESS METHOD

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

# IPv4 local connections:
host all all 127.0.0.1/32 trust

# IPv6 local connections:
host all all ::1/128 trust

# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
```

1. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。
 2. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。
 3. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。

```
# Allow remote connections to foo_db as admin

host  foo_db      admin      0.0.0.0/0      md5 -> IPv4 [ ][ ][ ][ ]
host  foo_db      admin      ::/0           md5 -> IPv6 [ ][ ][ ][ ]
```

```
sudo service postgresql restart
```

我们 在 代码 中 , 使用 了 一个 叫 做 的 包 来 做 这 个 工 作 。 这 个 包 叫 做 `md5` , 它 是 在 `trust` 包 中 的 一 个 子 包 。 我 们 在 第 4 行 中 使 用 了 这 个 包 。

Revision #2

Created 14 January 2024 07:32:05 by (MeatDumpling)

Updated 14 January 2024 08:07:21 by (MeatDumpling)