Pgbouncer installation guide and related TSG

Last updated by | Pedro Acevedo | Dec 29, 2021 at 1:18 PM PST

Install PGbouncer on a VM:

1. Install Pgbouncer

```
sudo apt update
```

```
sudo apt install pgbouncer -y
```

2. If you are going to use SSL, download the certificate:

```
sudo wget https://www.digicert.com/CACerts/BaltimoreCyberTrustRoot.crt
```

```
sudo openssl x509 -inform DER -in BaltimoreCyberTrustRoot.crt -text -out /etc/root.crt
```

3. Open the VI editor and change the following lines in file /etc/pgbouncer/pgbouncer.ini:

```
sudo vi /etc/pgbouncer/pgbouncer.ini
```

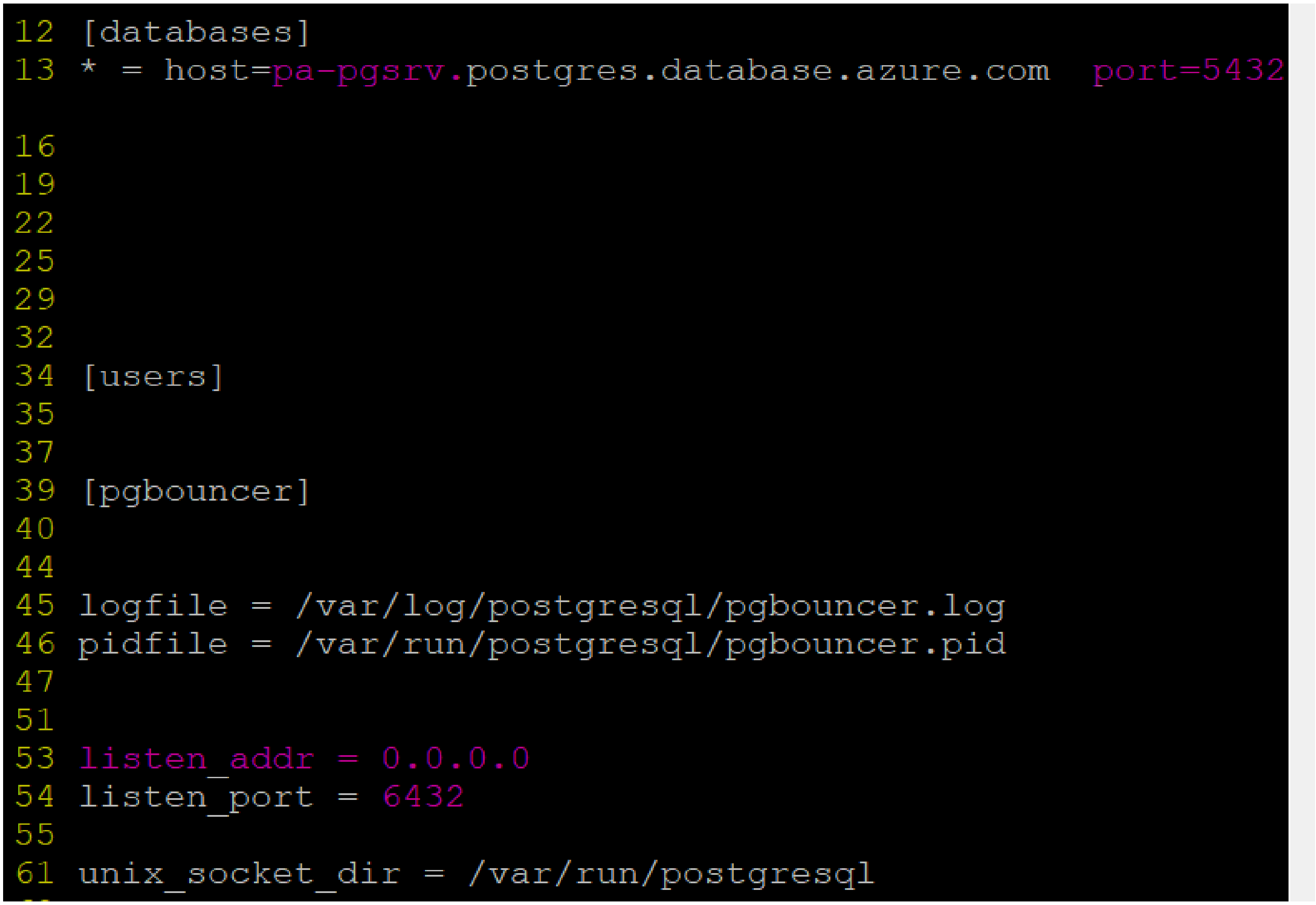
```

  = host=<servername>.postgres.database.azure.com  port=5432

  listen_addr = 0.0.0.0 #leave it as 127.0.0.1 if app runs on the same VM and there won't be any incoming connec

  listen_port = 6432 #leave as it is, or change to another port. Make sure connection string from app to pgbounc

   ignore_startup_parameters = extra_float_digits #uncomment this line if the customer is using JDBC driver
```



```
[databases]
* = host=pa-pgsrv.postgres.database.azure.com  port=5432


[users]


[pgbouncer]


logfile = /var/log/postgresql/pgbouncer.log
pidfile = /var/run/postgresql/pgbouncer.pid


listen_addr = 0.0.0.0
listen_port = 6432

unix_socket_dir = /var/run/postgresql
```

Other default parameters can be kept. If customer notices connectivity issues, they can further edit pgbouncer parameters according to the official documentation.

4. Edit file /etc/pgbouncer/userlist.txt with the required users:

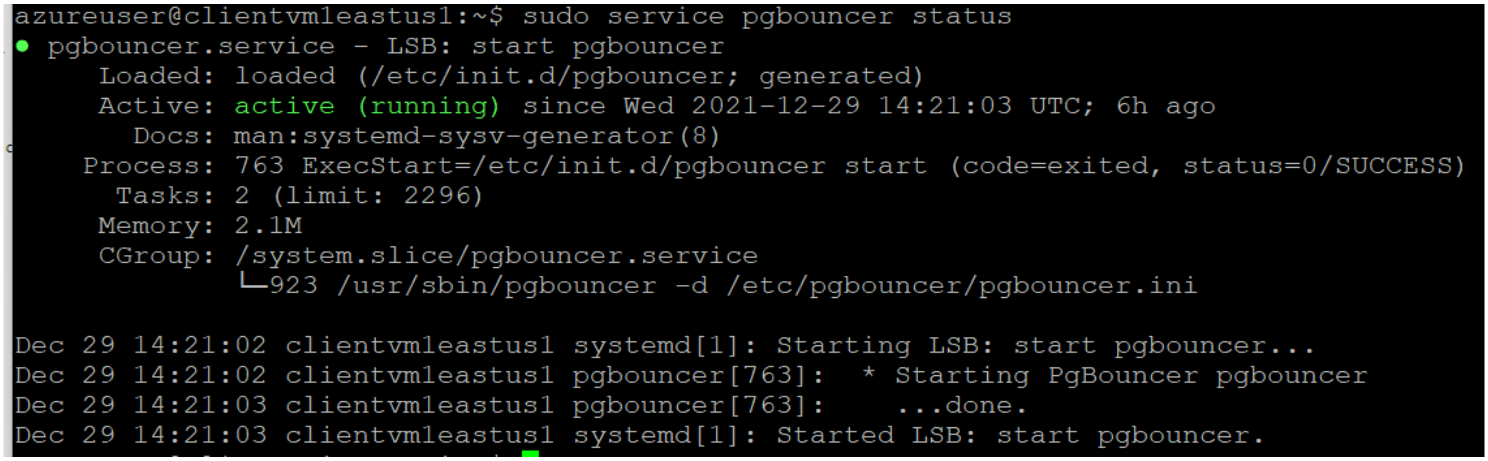
```
"username@servername" "password"
```

5. Start the service

```
sudo service pgbouncer start
```

6. Check that the service is running correctly:

```
sudo service pgbouncer status
```

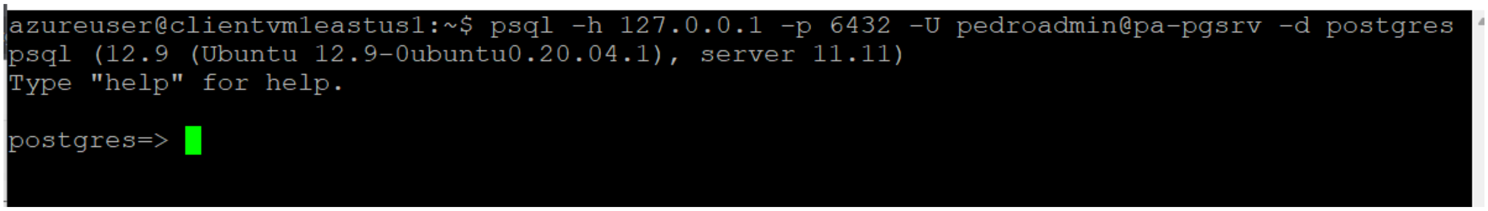


```
azureuser@clientvm1eastus1:~$ sudo service pgbouncer status
● pgbouncer.service - LSB: start pgbouncer
     Loaded: loaded (/etc/init.d/pgbouncer; generated)
     Active: active (running) since Wed 2021-12-29 14:21:03 UTC; 6h ago
       Docs: man:systemd-sysv-generator(8)
    Process: 763 ExecStart=/etc/init.d/pgbouncer start (code=exited, status=0/SUCCESS)
      Tasks: 2 (limit: 2296)
     Memory: 2.1M
     CGroup: /system.slice/pgbouncer.service
             └─923 /usr/sbin/pgbouncer -d /etc/pgbouncer/pgbouncer.ini

Dec 29 14:21:02 clientvm1eastus1 systemd[1]: Starting LSB: start pgbouncer...
Dec 29 14:21:02 clientvm1eastus1 pgbouncer[763]:  * Starting PgBouncer pgbouncer
Dec 29 14:21:03 clientvm1eastus1 pgbouncer[763]:    ...done.
Dec 29 14:21:03 clientvm1eastus1 systemd[1]: Started LSB: start pgbouncer.
```

7. Connect to pgbouncer:

```
psql -h 127.0.0.1 -p 6432 -U pedroadmin@pa-pgsrv -d postgres
```



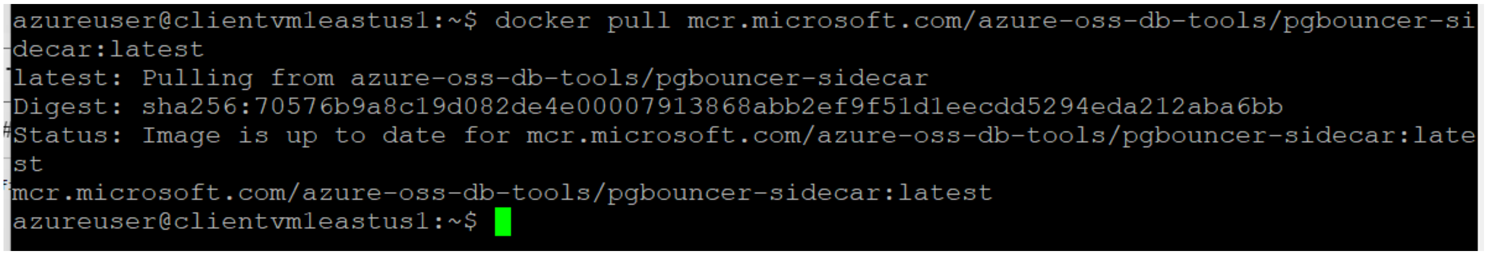
```
azureuser@clientvm1eastus1:~$ psql -h 127.0.0.1 -p 6432 -U pedroadmin@pa-pgsrv -d postgres
psql (12.9 (Ubuntu 12.9-0ubuntu0.20.04.1), server 11.11)
Type "help" for help.

postgres=>
```

Install PGbouncer as an pod sidecar in AKS:

1 Download the docker container:

```
docker pull mcr.microsoft.com/azure-oss-db-tools/pgbouncer-sidecar:latest
```



```
azureuser@clientvm1eastus1:~$ docker pull mcr.microsoft.com/azure-oss-db-tools/pgbouncer-si
decar:latest
latest: Pulling from azure-oss-db-tools/pgbouncer-sidecar
Digest: sha256:70576b9a8c19d082de4e00007913868abb2ef9f51d1eecdd5294eda212aba6bb
Status: Image is up to date for mcr.microsoft.com/azure-oss-db-tools/pgbouncer-sidecar:late
st
mcr.microsoft.com/azure-oss-db-tools/pgbouncer-sidecar:latest
azureuser@clientvm1eastus1:~$
```

2 Create files pgbouncer.ini and userlist.txt:

```
touch pgbouncer.ini userlist.txt
```

3 Copy this information to the pgbouncer.ini file:

```
  [databases]

    \* = host=<servername>.postgres.database.azure.com port=5432

  [pgbouncer]

    \# Do not change these settings:

    listen_addr = 0.0.0.0

    auth_file = /etc/pgbouncer/userlist.txt

    auth_type = trust

    server_tls_sslmode = verify-ca

    server_tls_ca_file = /etc/root.crt.pem

    \# These are defaults and can be configured. Leave them as defaults if you are uncertain.

    listen_port = 5432

    unix_socket_dir =

    user = postgres

    pool_mode = transaction

    max_client_conn = 100

    ignore_startup_parameters = extra_float_digits

    admin_users = postgres
```

4 Enter the username and password in the users.txt file:

```
"username@servername" "password"
```

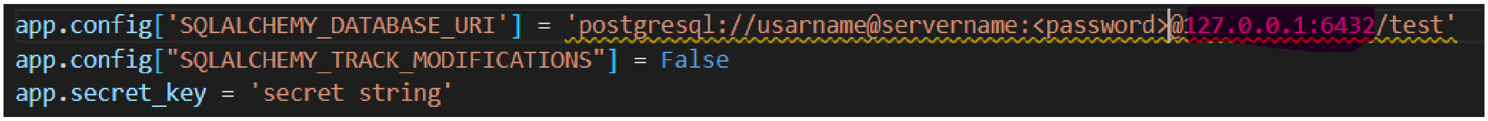
5 Create a secrets file from the information above:

```
kubectl create secret generic azure-pgbouncer-config --from-file=pgbouncer.ini --from-file=userlist.txt
```

6 Add this information to the application YAML file in the containers section:

```
- name: azure-pgbouncer-sidecar
    image: mcr.microsoft.com/azure-oss-db-tools/pgbouncer-sidecar:latest
    # Uncomment below if you always want the latest version
    # imagePullPolicy: Always
    ports:
      - containerPort: 5432
    volumeMounts:
      - name: configfiles
        mountPath: "/etc/pgbouncer"
        # writes update the secret, we do not want to do this
        readOnly: true
    livenessProbe:
      tcpSocket:
        # This must match the port your applications use to talk to postgres
        port: 6432
      periodSeconds: 60
    lifecycle:
      preStop:
        exec:
          command: ["/bin/sh", "-c", "killall -INT pgbouncer && sleep 120"]
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop: ['all']
  volumes:
    - name: configfiles
      secret:
          # This must match the name of your secret above
          secretName: azure-pgbouncer-config
```

7 In the application file, change the connection string to localhost and the listening port indicated in the pgbouncer.ini file.



```
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://usarname@servername:<password>@127.0.0.1:6432/test'
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.secret_key = 'secret string'
```

8 Deploy the application in kubernetes:

```
kubectl create -f <path to yaml file>
```

TSG for common issues

Error	Cause	Mitigation
psql: error: ERROR: pgbouncer cannot connect to server	Incorrect credentials in file /etc/pgbouncer/userlist.txt	Ensure credentials are correct in the file
psql: error: ERROR: no such user: user org.postgresql.util.PSQLException: FATAL: "trust" authentication failed	Incorrect username in file /etc/pgbouncer/userlist.txt	Ensure username is correct and with format user@servernam
could not connect to server: Connection refused (0x0000274D/10061) Is the server running on host "20.83.152.142" and accepting TCP/IP connections on port 6432?	Pgbouncer is only listening for incoming connections in localhost	Edit file /etc/pgbouncer/pgbouncer.ini with: listen_addr to 0.0.0.0
No response, freezes or timeout. psql: error: ERROR: client_login_timeout (server down)	Firewall or incorrect host name	Ensure the host name is correct in the /etc/pgbouncer/pgbouncer.ini file. Ensure there is no NSG blocking port 5432 or 6432

Public Doc Reference

Install PGbouncer on a VM: https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/steps-to-install-and-setup-pgbouncer-connection-pooling-proxy/ba-p/730555

Install PGbouncer as an AKS sidecar: https://hub.docker.com/_/microsoft-azure-oss-db-tools-pgbouncer-sidecar

PGbouncer config: https://www.pgbouncer.org/config.html