

# Install PostgreSQL in Docker

```
WD=/opt/dcontainer
mkdir -p $WD/{dcontainer_sql,dcontainer_conf,setup}
cd $WD/setup
cat << 'EOF' >docker-compose.yaml
version: '3.2'

services:
  dcontainer:
    container_name: dcontainer
    image: dcontainer
    hostname: dcontainer
    networks:
      - dcontainernet
    ports:
      - '6565:80'
    volumes:
      - type: bind
        source: /opt/dcontainer/dcontainer_conf/
        target: /etc/dcontainer

    environment:
      - 'TZ=America/Whitehorse'

  dcontainerdb:
    container_name: dcontainer_sql
    hostname: dcontainer_sql
    networks:
      - dcontainernet
    image: postgres
    volumes:
      - type: bind
        source: /opt/dcontainer/dcontainer_sql/
        target: /var/lib/postgresql/data

    environment:
      - 'POSTGRES_USER=${DB_USER}'
      - 'POSTGRES_PASSWORD=${DB_PWD}'
      - 'POSTGRES_DB=${DB_NAME}'
      - 'POSTGRES_ENCODING=UTF8'
      - 'POSTGRES_COLLATE=C'
      - 'POSTGRES_COLLATE_TYPE=C'

networks:
  dcontainernet:
    driver: bridge
EOF
chmod +x docker-compose.yaml

cd $WD/setup
cat << 'EOF'>.env
DB_NAME=dcontainer
DB_USER=dcontainer
DB_PWD=dcontainer
EOF
chmod +x .env
```

```

mkdir /opt/postgresql
cat << 'EOF' >/opt/postgresql/postgresql_docker.sh
docker run -d \
  --name postgresql \
  -v /opt/postgresql/data:/var/lib/postgresql/data \
  -v /opt/postgresql/mnt:/mnt \
  -e POSTGRES_PASSWORD=mysecretpassword \
  -e POSTGRES_USER=postgres \
  -p 5432:5432 \
  postgres
EOF
chmod +x /opt/postgresql/postgresql_docker.sh
/opt/postgresql/postgresql_docker.sh

```

Note: The above creates a superuser postgres. Personally, I debated not creating the user this way, and using the cli inside the machine to create the user, but sometimes it's good to leave setup files to other techs to see what you have done. You could always go in and change after, I used this for setup /testing.

#### CLI:

```

echo "alias pgcli='docker exec -it postgresql /bin/bash'" >> ~/.bashrc
source ~/.bashrc

```

#### Network:

```

-p 5432:5432 # to expose to "localhost"

```

#### Security (pg\_hba.conf):

```

sed -i '/host all all all md5/s/^/#/g' /opt/postgresql/data/pg_hba.conf
cat << 'EOF' >>/opt/postgresql/data/pg_hba.conf
host    all         all             172.17.0.0/16      md5
EOF
cat /opt/postgresql/data/pg_hba.conf
HOSTIP="$(ip -4 addr show eth0 | grep -oP '(?<=inet\s)\d+(\.\d+){3}')"
cat << EOF >>/opt/postgresql/data/pg_hba.conf
host    all         all             $HOSTIP/32        md5
EOF
cat /opt/postgresql/data/pg_hba.conf

```

```

host all all all md5 #default
host  all         all             172.17.0.0/16     md5 #Docker Subnet
host  all         all             $HOSTIP/32        md5 #Local IPv4

```

#### Disable "All IP Access":

```

sed -i '/host all all all md5/s/^/#/g' /opt/postgresql/data/pg_hba.conf

```

#### Docker Only Access:

```

cat << 'EOF' >>/opt/postgresql/data/pg_hba.conf
host  all         all             172.17.0.0/16     md5
EOF
cat /opt/postgresql/data/pg_hba.conf

```

#### Add Access from "Local IP Only":

```
HOSTIP="$(ip -4 addr show eth0 | grep -oP '(?<=inet\s)\d+(\.\d+){3}')"
cat << EOF >>/opt/postgresql/data/pg_hba.conf
host    all             all             $HOSTIP/32      md5
EOF
cat /opt/postgresql/data/pg_hba.conf
```

#### SystemD:

```
cat << 'EOL' >>/lib/systemd/system/postgresql.service
[Unit]
Description=PostgreSQL DOCKER Container
Requires=docker.service network-online.target

[Service]
Restart=on-abnormal
ExecStart=/usr/bin/docker start -a postgresql
ExecStop=/usr/bin/docker stop -t 2 postgresql

[Install]
WantedBy=multi-user.target
EOL
systemctl enable postgresql
systemctl restart postgresql
systemctl status postgresql
```