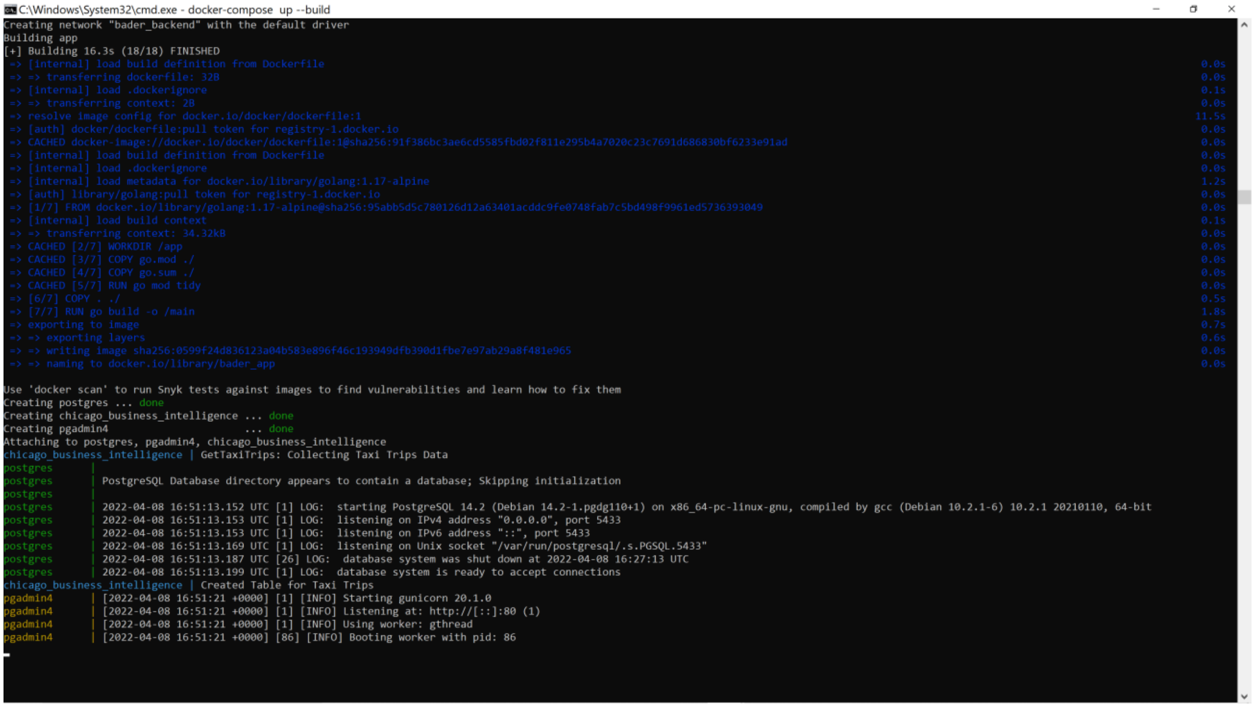
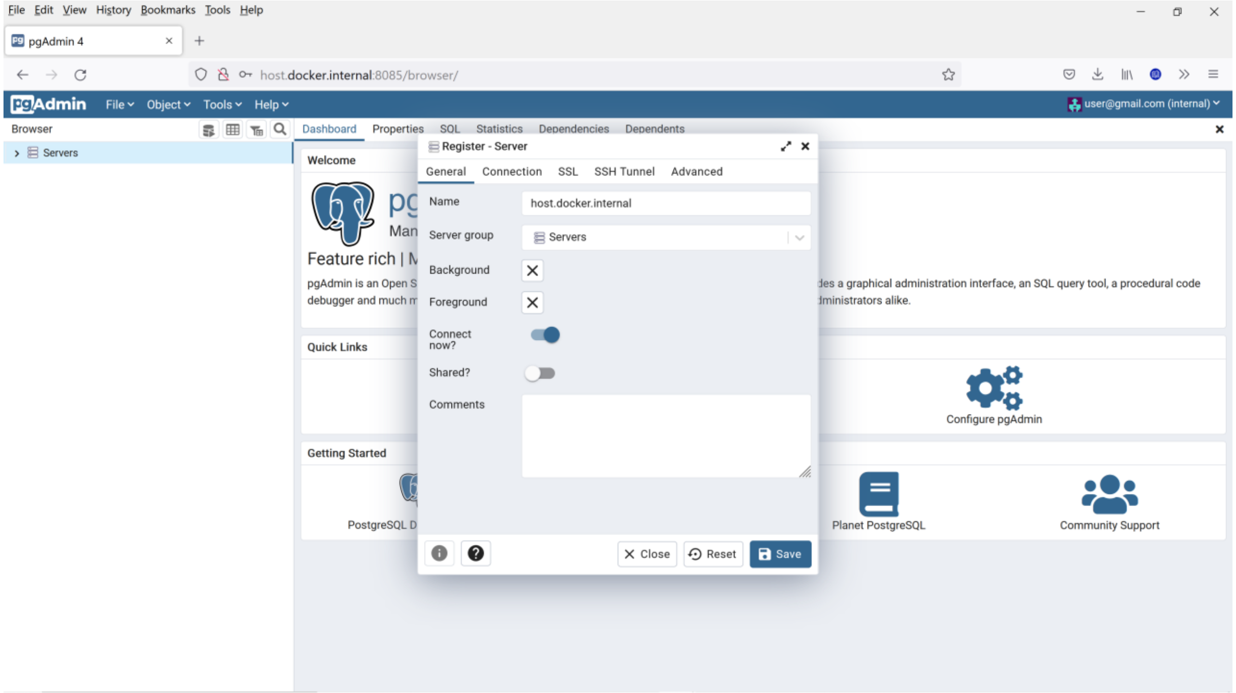
Your docker-compose up –build will look similar to this :



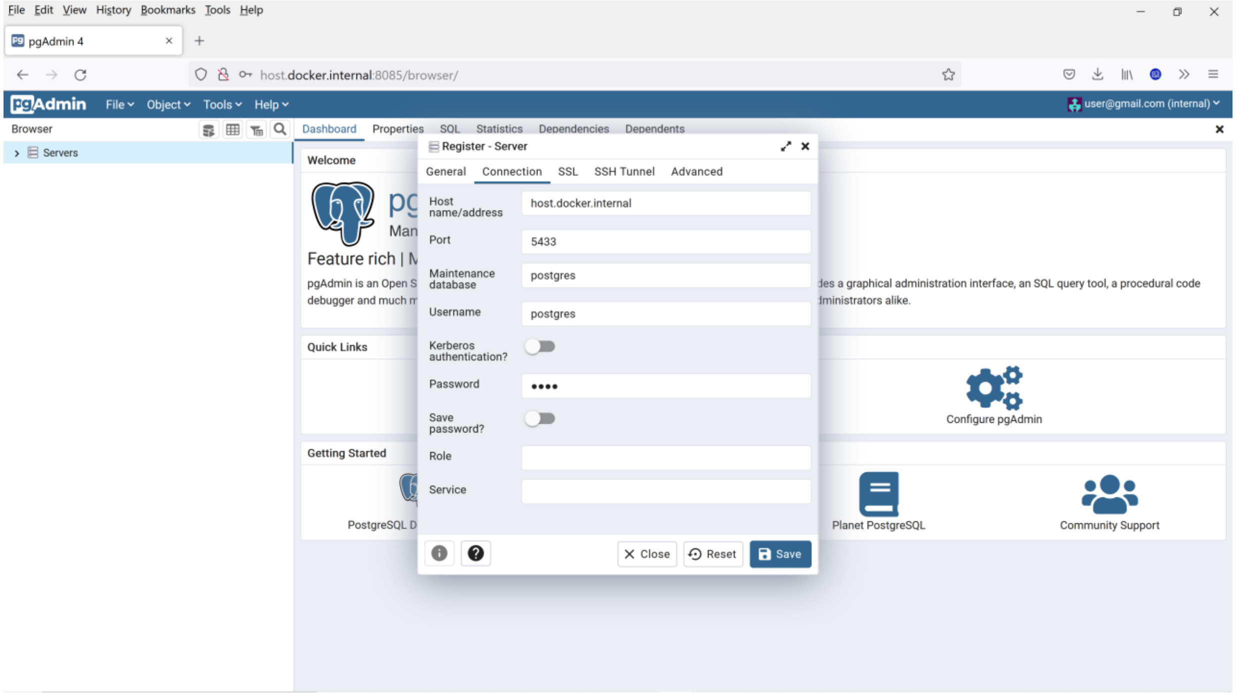
```
Creating network "bader_backend" with the default driver
Building app
[+] Building 16.3s (18/18) FINISHED
 => [internal] load build definition from Dockerfile                                                                                0.0s
 => => transferring dockerfile: 32B                                                                                                 0.0s
 => [internal] load .dockerignore                                                                                                   0.1s
 => => transferring context: 2B                                                                                                     0.0s
 => resolve image config for docker.io/docker/dockerfile:1                                                                         11.5s
 => [auth] docker/dockerfile:pull token for registry-1.docker.io                                                                    0.0s
 => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:91f386bc3ae6cd5585fbd02f811e295b4a7020c23c7691d686830bf6233e91ad     0.0s
 => [internal] load build definition from Dockerfile                                                                                0.0s
 => [internal] load .dockerignore                                                                                                   0.0s
 => [internal] load metadata for docker.io/library/golang:1.17-alpine                                                               1.2s
 => [auth] library/golang:pull token for registry-1.docker.io                                                                       0.0s
 => [1/7] FROM docker.io/library/golang:1.17-alpine@sha256:95abb5d5c780126d12a63401acddc9fe0748fab7c5bd498f9961ed5736393049         0.0s
 => [internal] load build context                                                                                                   0.1s
 => => transferring context: 34.32kB                                                                                                0.0s
 => CACHED [2/7] WORKDIR /app                                                                                                       0.0s
 => CACHED [3/7] COPY go.mod ./                                                                                                     0.0s
 => CACHED [4/7] COPY go.sum ./                                                                                                     0.0s
 => CACHED [5/7] RUN go mod tidy                                                                                                    0.0s
 => [6/7] COPY . ./                                                                                                                 0.5s
 => [7/7] RUN go build -o /main                                                                                                     1.8s
 => exporting to image                                                                                                              0.7s
 => => exporting layers                                                                                                             0.6s
 => => writing image sha256:0599f24d836123a04b583e896f46c193949dfb390d1fbe7e97ab29a8f481e965                                        0.0s
 => => naming to docker.io/library/bader_app                                                                                        0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
Creating postgres ... done
Creating chicago_business_intelligence ... done
Creating pgadmin4                      ... done
Attaching to postgres, pgadmin4, chicago_business_intelligence
chicago_business_intelligence | GetTaxiTrips: Collecting Taxi Trips Data
postgres                      |
postgres                      | PostgreSQL Database directory appears to contain a database; Skipping initialization
postgres                      |
postgres                      | 2022-04-08 16:51:13.152 UTC [1] LOG:  starting PostgreSQL 14.2 (Debian 14.2-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit
postgres                      | 2022-04-08 16:51:13.153 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5433
postgres                      | 2022-04-08 16:51:13.153 UTC [1] LOG:  listening on IPv6 address "::", port 5433
postgres                      | 2022-04-08 16:51:13.169 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5433"
postgres                      | 2022-04-08 16:51:13.187 UTC [26] LOG:  database system was shut down at 2022-04-08 16:27:13 UTC
postgres                      | 2022-04-08 16:51:13.199 UTC [1] LOG:  database system is ready to accept connections
chicago_business_intelligence | Created Table for Taxi Trips
pgadmin4                      | [2022-04-08 16:51:21 +0000] [1] [INFO] Starting gunicorn 20.1.0
pgadmin4                      | [2022-04-08 16:51:21 +0000] [1] [INFO] Listening at: http://[::]:80 (1)
pgadmin4                      | [2022-04-08 16:51:21 +0000] [1] [INFO] Using worker: gthread
pgadmin4                      | [2022-04-08 16:51:21 +0000] [86] [INFO] Booting worker with pid: 86
```

Open "**http://host.docker.internal:8085/**" to check if pgadmin is running. **Right Click on Server> Register>Server.** Add server with hostname as **host.docker.internal** and Port as **5433** user as **postgres** and password as **root**.


File Edit View History Bookmarks Tools Help
pgAdmin 4
host.docker.internal:8085/browser/
pgAdmin File Object Tools Help
user@gmail.com (internal)
Browser
Servers
Dashboard Properties SQL Statistics Dependencies Dependents
Welcome
Feature rich | M
Quick Links
Getting Started
PostgreSQL D
Configure pgAdmin
Planet PostgreSQL
Community Support
Register - Server
General Connection SSL SSH Tunnel Advanced
Close Reset Save

Name: host.docker.internal
Server group: Servers
Background: ☒
Foreground: ☒
Connect now? ●
Shared? ○
Comments:

pgAdmin is an Open S... debugger and much m... ...des a graphical administration interface, an SQL query tool, a procedural code ...dministrators alike.

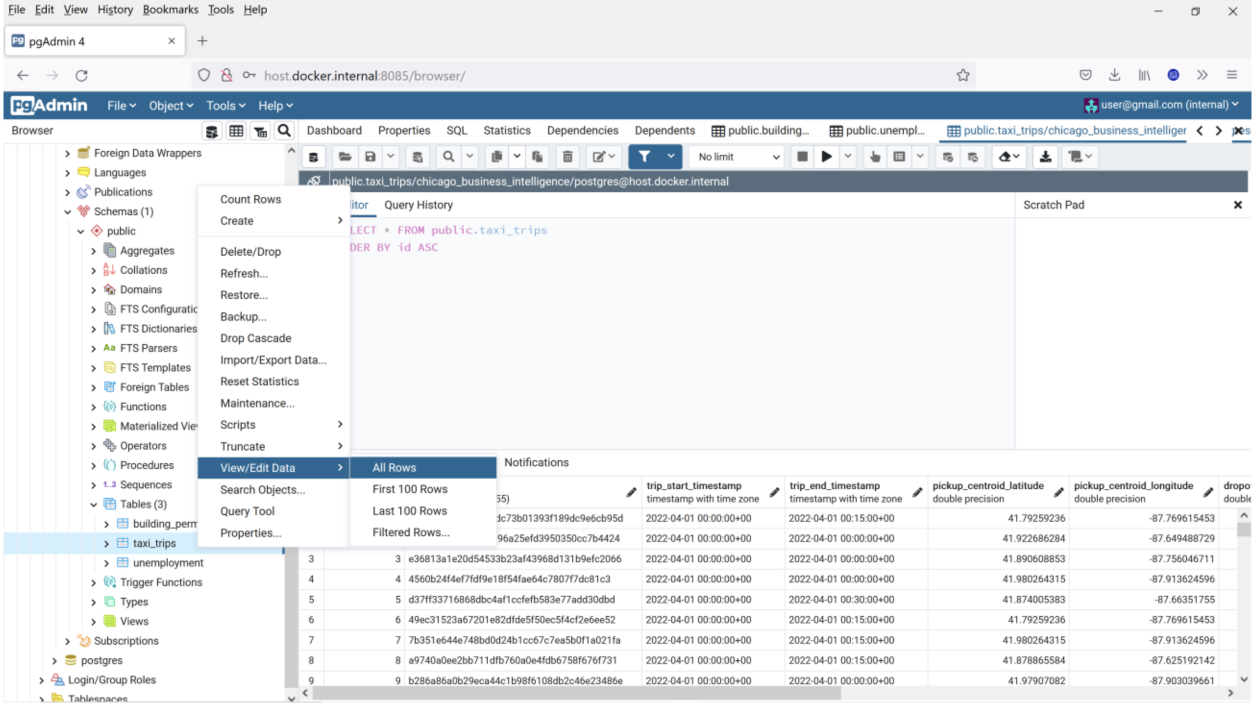

File Edit View History Bookmarks Tools Help
pgAdmin 4
host.docker.internal:8085/browser/
pgAdmin File Object Tools Help
user@gmail.com (internal)
Browser
Servers
Dashboard Properties SQL Statistics Dependencies Dependents
Welcome
Feature rich | M
Quick Links
Getting Started
PostgreSQL D
Configure pgAdmin
Planet PostgreSQL
Community Support
Register - Server
General Connection SSL SSH Tunnel Advanced
Close Reset Save

Host name/address: host.docker.internal
Port: 5433
Maintenance database: postgres
Username: postgres
Kerberos authentication? ○
Password: ••••
Save password? ○
Role:
Service:

pgAdmin is an Open S... debugger and much m... ...des a graphical administration interface, an SQL query tool, a procedural code ...dministrators alike.

Use **PGAdmin** Verify you got **CBI DB** constructed and its **Tables** populated:

Your **PGAdmin** will look similar to this :



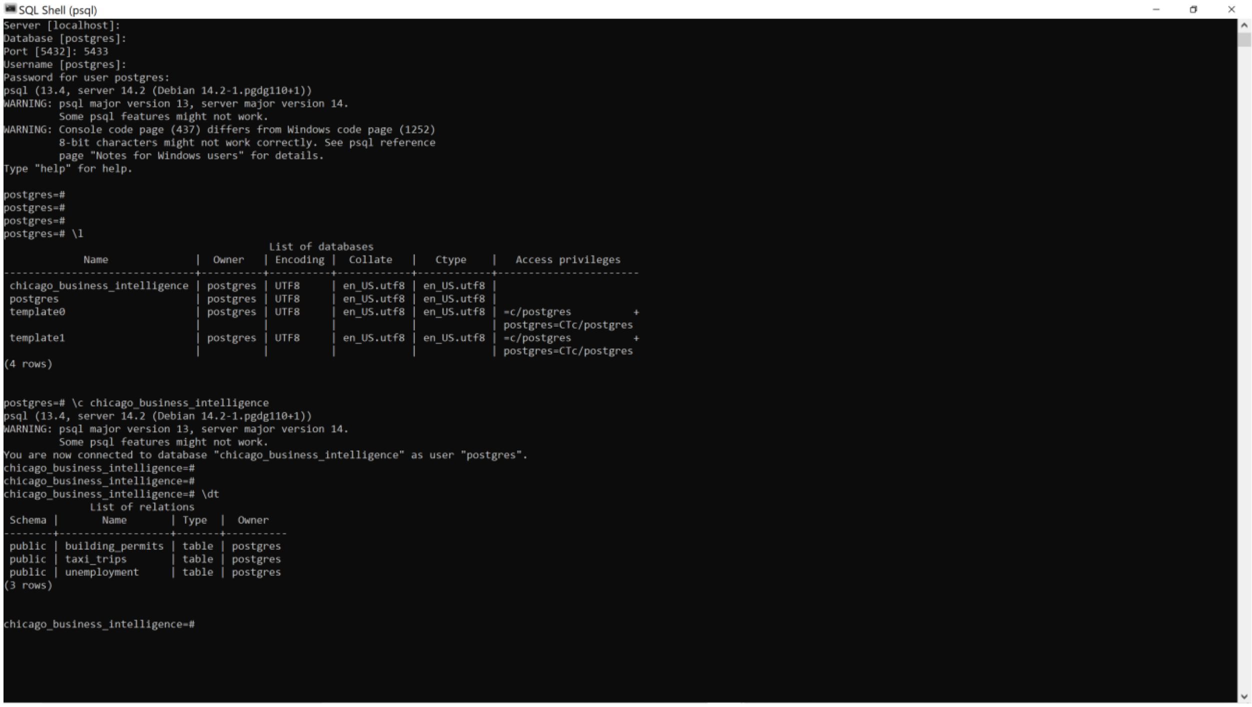
public.taxi_trips/chicago_business_intelligence/postgres@host.docker.internal

```
LECT * FROM public.taxi_trips
DER BY id ASC
```

		55)	trip_start_timestamp timestamp with time zone	trip_end_timestamp timestamp with time zone	pickup_centroid_latitude double precision	pickup_centroid_longitude double precision	dropo double
		dc73b01393f189dc9e6cb95d	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.79259236	-87.769615453	
		96a25efd3950350cc7b4424	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.922686284	-87.649488729	
3	3	e36813a1e20d54533b23af43968d131b9efc2066	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.890608853	-87.756046711	
4	4	4560b24f4ef7fdf9e18f54fae64c7807f7dc81c3	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.980264315	-87.913624596	
5	5	d37ff33716868dbc4af1ccfefb583e77add30dbd	2022-04-01 00:00:00+00	2022-04-01 00:30:00+00	41.874005383	-87.66351755	
6	6	49ec31523a67201e82dfde5f50ec5f4cf2e6ee52	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.79259236	-87.769615453	
7	7	7b351e644e748bd0d24b1cc67c7ea5b0f1a021fa	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.980264315	-87.913624596	
8	8	a9740a0ee2bb711dfb760a0e4fdb6758f676f731	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.878865584	-87.625192142	
9	9	b286a86a0b29eca44c1b98f6108db2c46e23486e	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.97907082	-87.903039661	

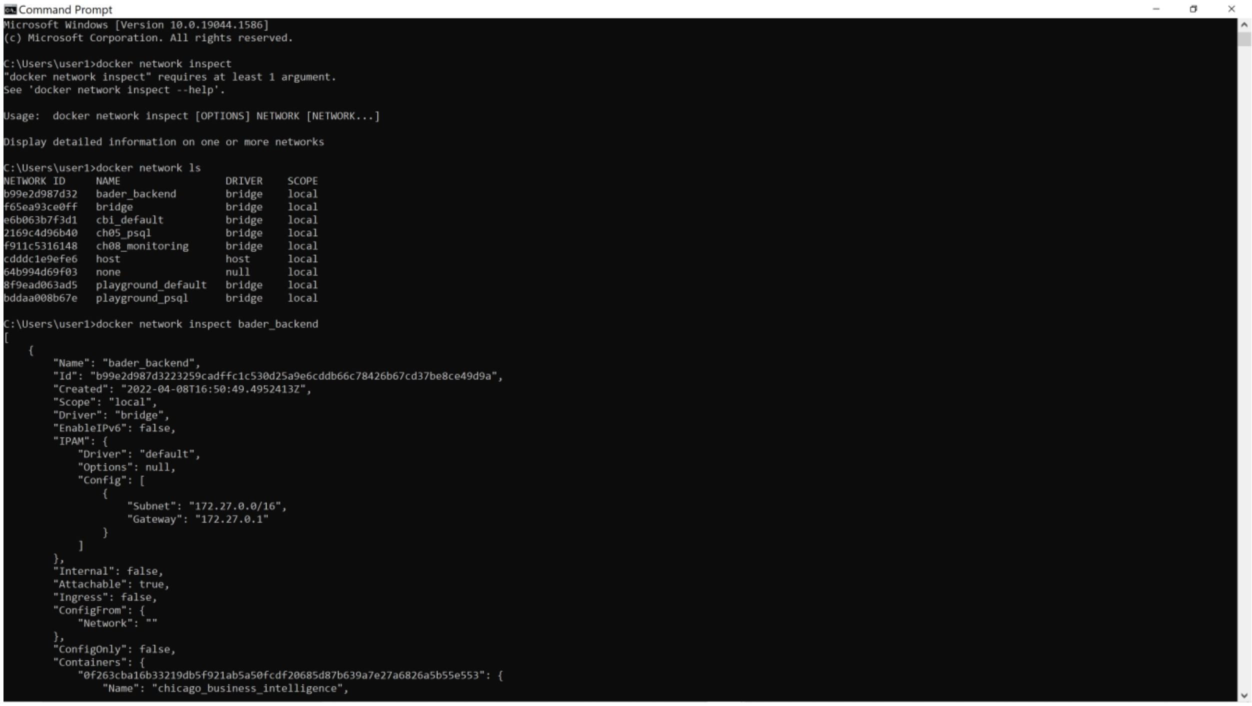
You can also connect to your **postgres** server container from **localhost** using **psql** shell.

Your **psql** will look similar to this :



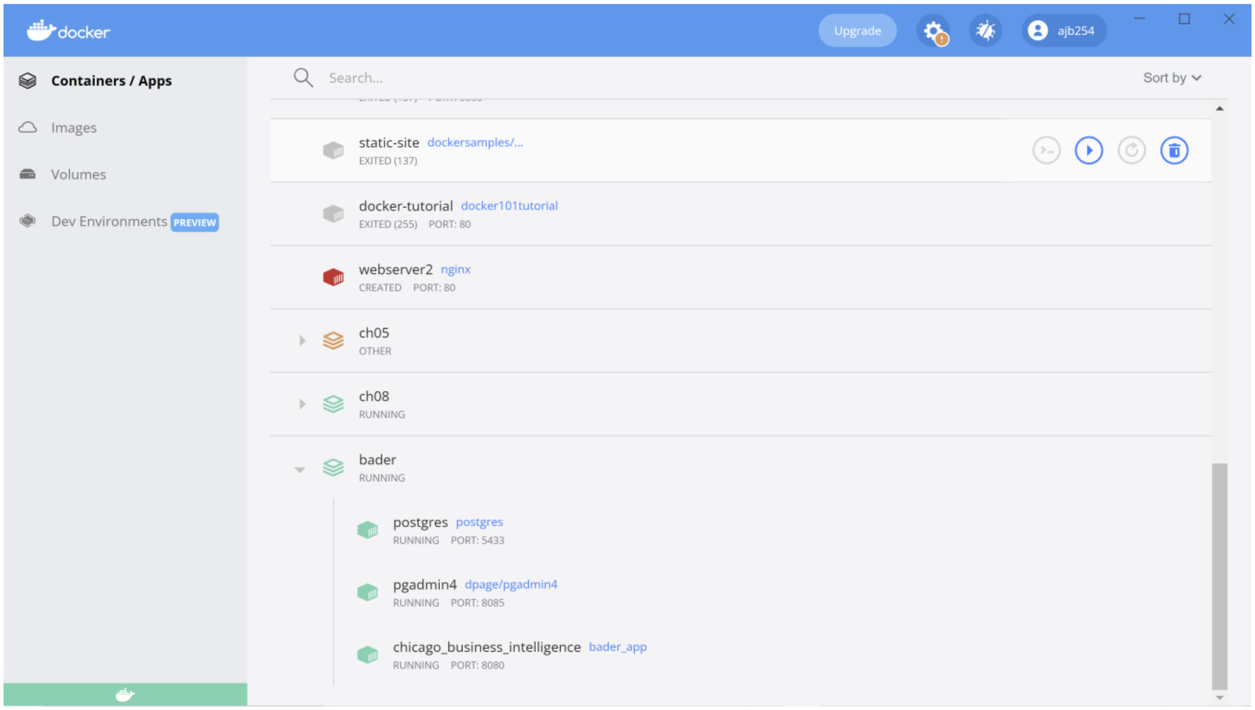
```
Server [localhost]:
Database [postgres]:
Port [5432]: 5433
Username [postgres]:
Password for user postgres:
psql (13.4, server 14.2 (Debian 14.2-1.pgdg110+1))
WARNING: psql major version 13, server major version 14.
         Some psql features might not work.
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
postgres=#
postgres=#
postgres=# \l
                                            List of databases
             Name              |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-------------------------------+----------+----------+------------+------------+-----------------------
 chicago_business_intelligence | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres                      | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0                     | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                               |          |          |            |            | postgres=CTc/postgres
 template1                     | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                               |          |          |            |            | postgres=CTc/postgres
(4 rows)


postgres=# \c chicago_business_intelligence
psql (13.4, server 14.2 (Debian 14.2-1.pgdg110+1))
WARNING: psql major version 13, server major version 14.
         Some psql features might not work.
You are now connected to database "chicago_business_intelligence" as user "postgres".
chicago_business_intelligence=#
chicago_business_intelligence=#
chicago_business_intelligence=# \dt
              List of relations
 Schema |       Name       | Type  |  Owner
--------+------------------+-------+----------
 public | building_permits | table | postgres
 public | taxi_trips       | table | postgres
 public | unemployment     | table | postgres
(3 rows)


chicago_business_intelligence=#
```

Your **docker network ls** will look similar to this:



```
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user1>docker network inspect
"docker network inspect" requires at least 1 argument.
See 'docker network inspect --help'.

Usage:  docker network inspect [OPTIONS] NETWORK [NETWORK...]

Display detailed information on one or more networks

C:\Users\user1>docker network ls
NETWORK ID     NAME                 DRIVER    SCOPE
b99e2d987d32   bader_backend        bridge    local
f65ea93ce0ff   bridge               bridge    local
e6b063b7f3d1   cbi_default          bridge    local
2169c4d96b40   ch05_psql            bridge    local
f911c5316148   ch08_monitoring      bridge    local
cdddc1e9efe6   host                 host      local
64b994d69f03   none                 null      local
8f9ead063ad5   playground_default   bridge    local
bddaa008b67e   playground_psql      bridge    local

C:\Users\user1>docker network inspect bader_backend
[
    {
        "Name": "bader_backend",
        "Id": "b99e2d987d3223259cadffc1c530d25a9e6cddb66c78426b67cd37be8ce49d9a",
        "Created": "2022-04-08T16:50:49.4952413Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.27.0.0/16",
                    "Gateway": "172.27.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": true,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "0f263cba16b33219db5f921ab5a50fcdf20685d87b639a7e27a6826a5b55e553": {
                "Name": "chicago_business_intelligence",
```

Your Docker Desktop should look similar to this:



static-site dockersamples/...
EXITED (137)

docker-tutorial docker101tutorial
EXITED (255) PORT: 80

webserver2 nginx
CREATED PORT: 80

ch05
OTHER

ch08
RUNNING

bader
RUNNING

postgres postgres
RUNNING PORT: 5433

pgadmin4 dpage/pgadmin4
RUNNING PORT: 8085

chicago_business_intelligence bader_app
RUNNING PORT: 8080