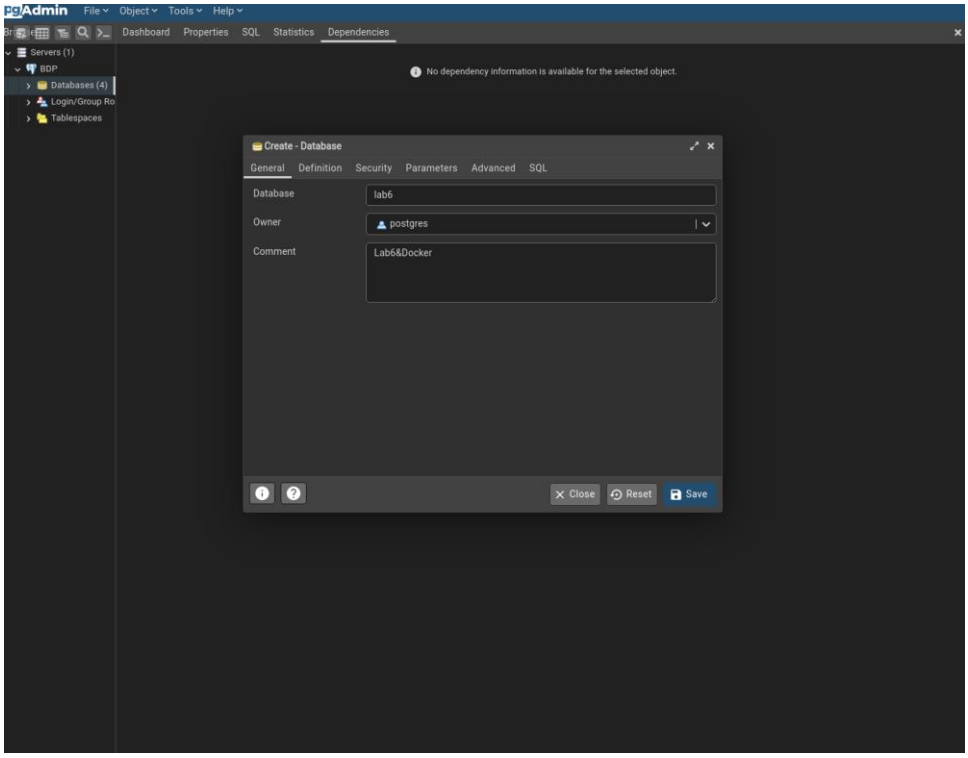


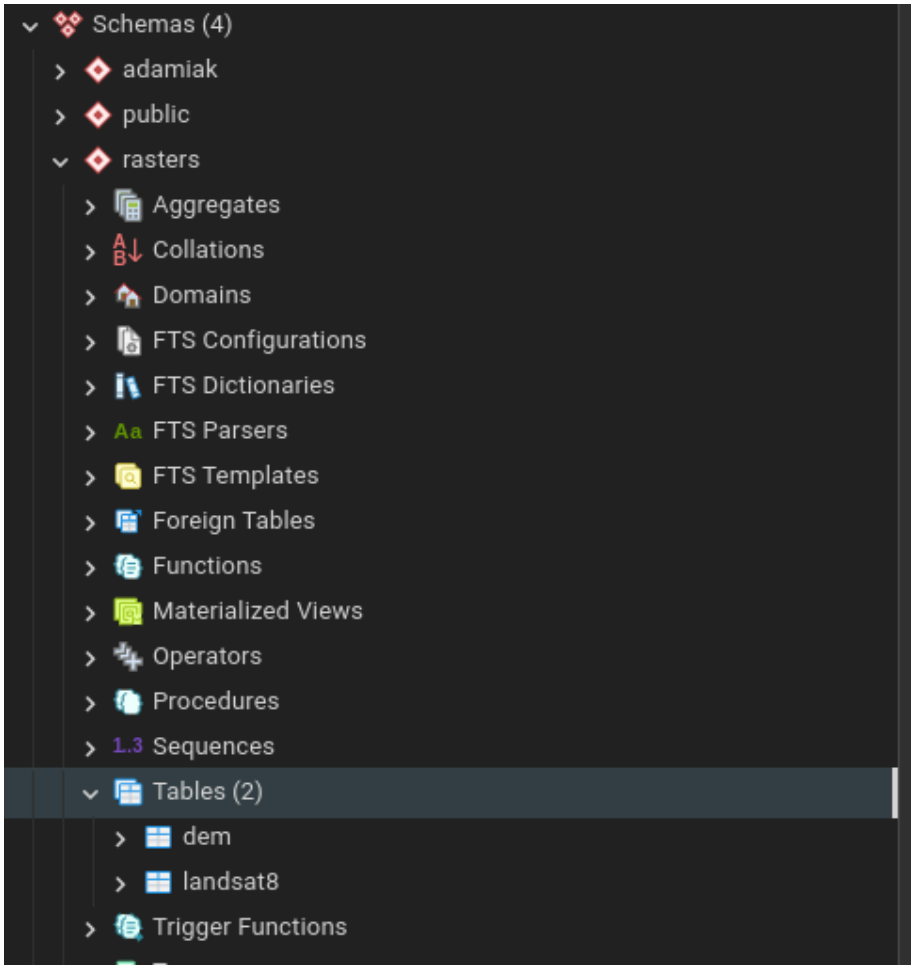
/usr/bin/pg_restore --host "localhost" --port "5432" --username "postgres" --no-password --dbname "lab6" --section=pre-data --verbose "/home/wildfire/Downloads/postgis_raster_backup

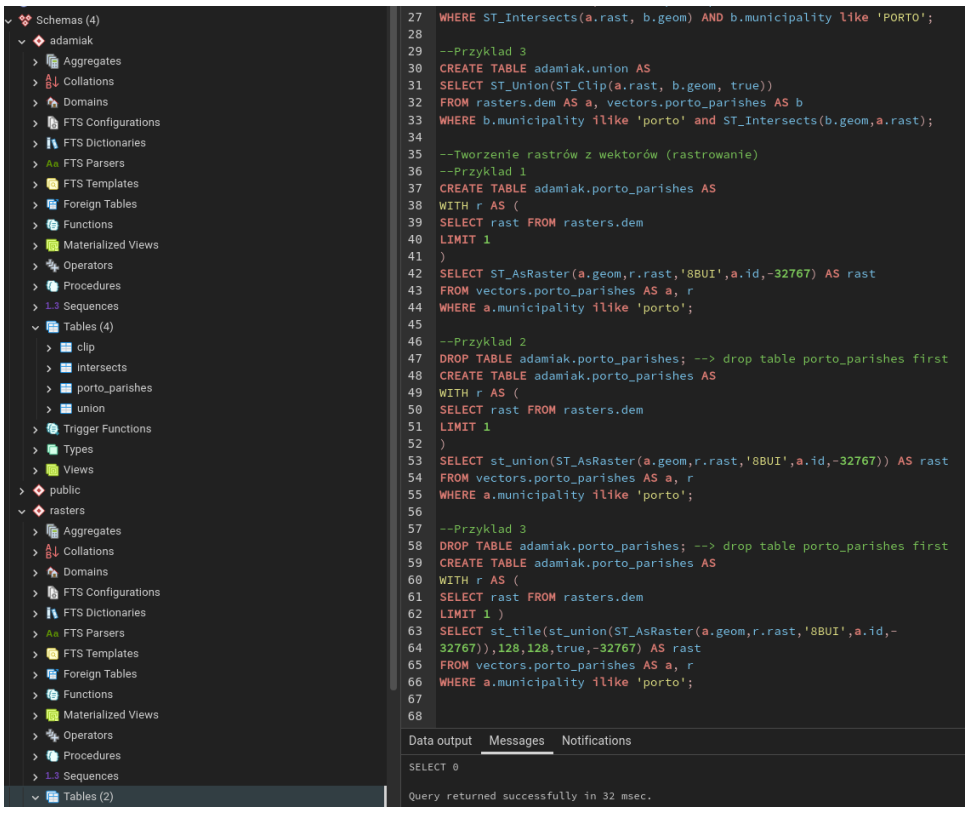
raster2pgsql -s 32767 -t 100x100 -I -C -M -d srtm_1arc_v3.tif rasters.dem | psql -d lab6 -h localhost -U postgres -p 5432

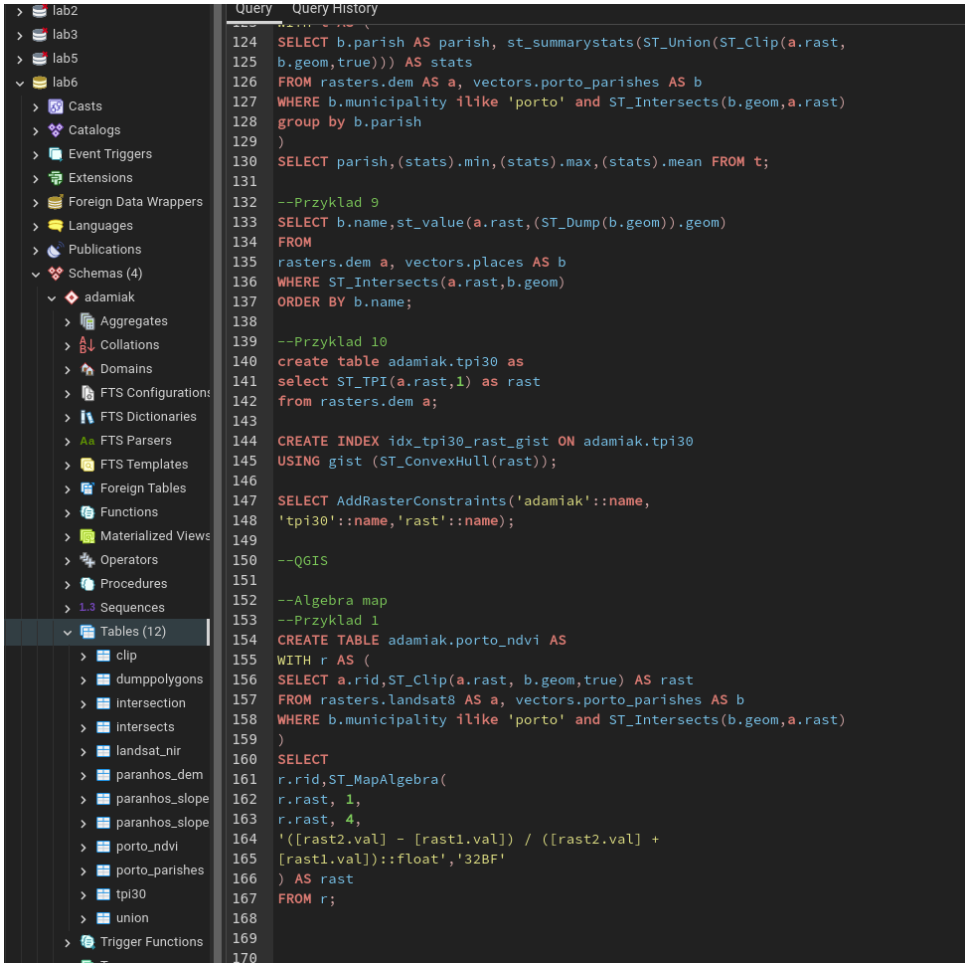
```
(base) wildfire@pop-os:~/Downloads$ raster2pgsql -s 32767 -t 100x100 -I -C -M -d srtm_1arc_v3.tif rasters.dem | psql -d lab
6 -h localhost -U postgres -p 5432
Processing 1/1: srtm_1arc_v3.tif
Password for user postgres:
BEGIN
NOTICE:  table "dem" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

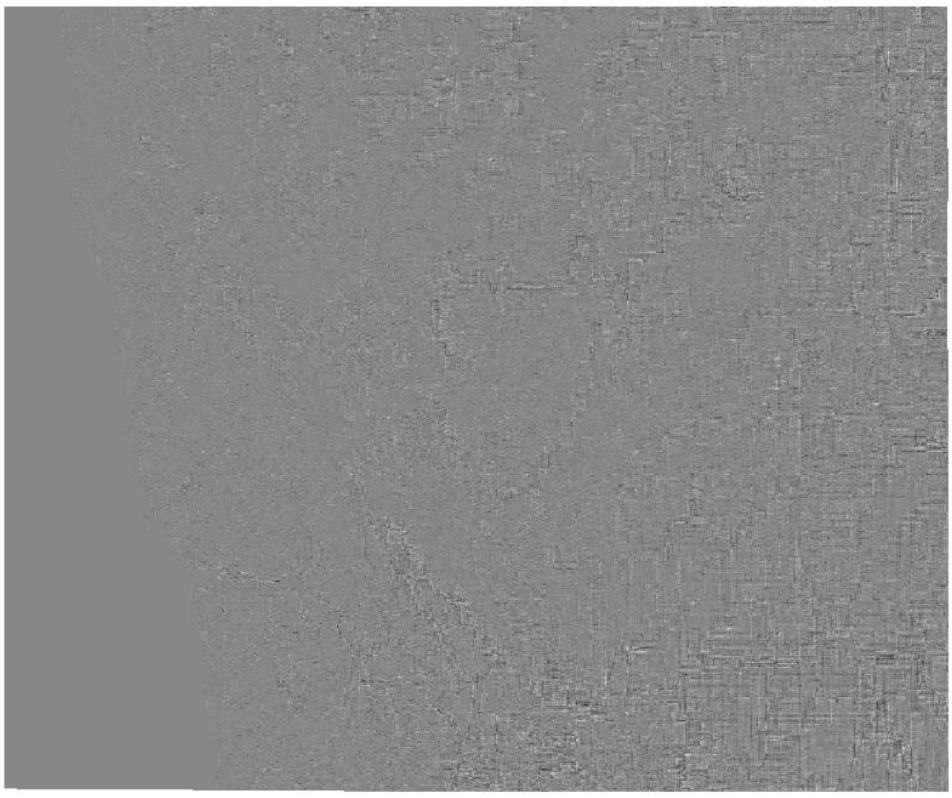
raster2pgsql -s 32767 -t 100x100 -I -C -M -d Landsat8_L1TP_RGBN.tif rasters.landsat8 | psql -d lab6 -h localhost -U postgres -p 5432

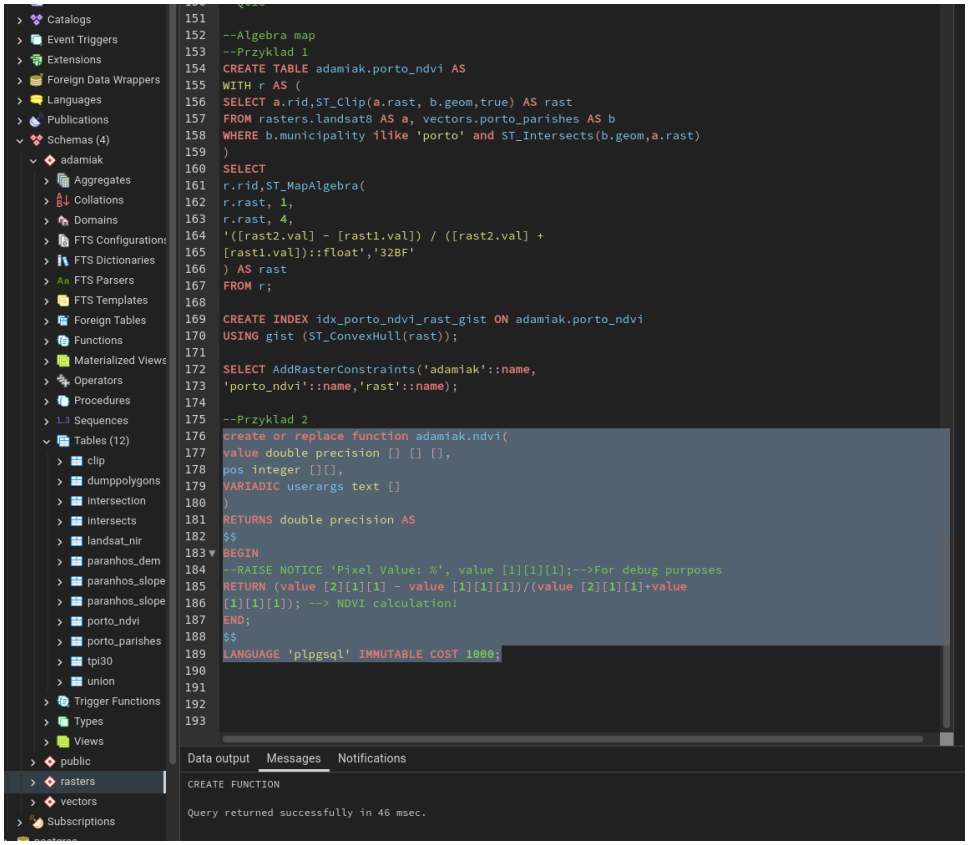
```
(base) wildfire@pop-os:~/Downloads$ raster2pgsql -s 32767 -t 100x100 -I -C -M -d Landsat8_L1TP_RGBN.tif rasters.landsat8 |
psql -d lab6 -h localhost -U postgres -p 5432
Processing 1/1: Landsat8_L1TP_RGBN.tif
Password for user postgres:
BEGIN
NOTICE:  table "landsat8" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 1
```

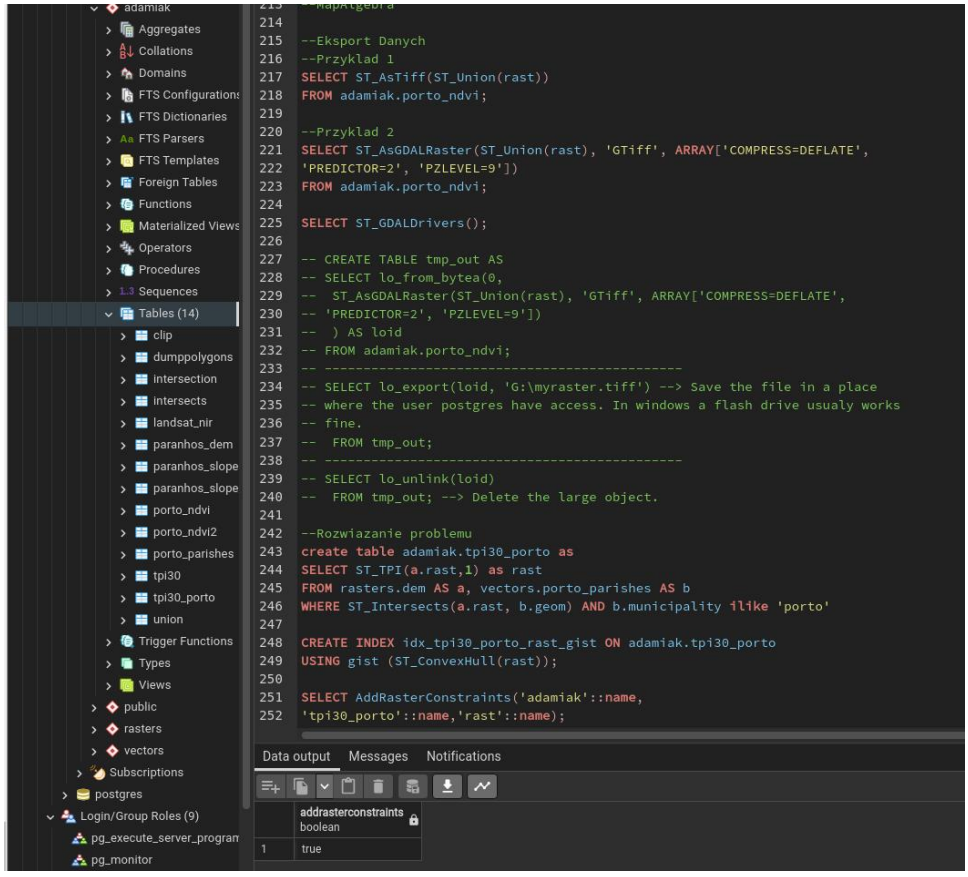

Schemas (4)
adamiak
public
rasters
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (2)
dem
landsat8
Trigger Functions


Schemas (4)
adamiak
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
clip
intersects
porto_parishes
union
Trigger Functions
Types
Views
public
rasters
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (2)
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
--Przyklad 3
CREATE TABLE adamiak.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);
--Tworzenie rastrów z wektorów (rastrowanie)
--Przyklad 1
CREATE TABLE adamiak.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)
SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
--Przyklad 2
DROP TABLE adamiak.porto_parishes; --> drop table porto_parishes first
CREATE TABLE adamiak.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)
SELECT st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
--Przyklad 3
DROP TABLE adamiak.porto_parishes; --> drop table porto_parishes first
CREATE TABLE adamiak.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1)
SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
32767)),128,128,true,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
Data output
Messages
Notifications
SELECT 0
Query returned successfully in 32 msec.


lab2
lab3
lab5
lab6
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (4)
adamiak
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (12)
clip
dumppolygons
intersection
intersects
landsat_nir
paranhos_dem
paranhos_slope
paranhos_slope
porto_ndvi
porto_parishes
tpi30
union
Trigger Functions
Query
Query History
SELECT b.parish AS parish, st_summarystats(ST_Union(ST_Clip(a.rast,
b.geom,true))) AS stats
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
group by b.parish
)
SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;
--Przyklad 9
SELECT b.name,st_value(a.rast,(ST_Dump(b.geom)).geom)
FROM
rasters.dem a, vectors.places AS b
WHERE ST_Intersects(a.rast,b.geom)
ORDER BY b.name;
--Przyklad 10
create table adamiak.tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;
CREATE INDEX idx_tpi30_rast_gist ON adamiak.tpi30
USING gist (ST_ConvexHull(rast));
SELECT AddRasterConstraints('adamiak'::name,
'tpi30'::name,'rast'::name);
--QGIS
--Algebra map
--Przyklad 1
CREATE TABLE adamiak.porto_ndvi AS
WITH r AS (
SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT
r.rid,ST_MapAlgebra(
r.rast, 1,
r.rast, 4,
'([rast2.val] - [rast1.val]) / ([rast2.val] +
[rast1.val])::float','32BF'
) AS rast
FROM r;








```
Run a command in a new container
(base) wildfire@pop-os:~/docker-mapserver$ sudo docker run -d -p 8182:80 --name mapserver2 -v `pwd`/map:/map kartoza/mapserver_kartoza
59ecd9e3038907c133db18ac52c5ca0cce052f30984bacb0c244b86ef0612a80
(base) wildfire@pop-os:~/docker-mapserver$ sudo docker exec -it mapserver2 /bin/bash
root@59ecd9e30389:/# apt-get install -y postgresql-client
```

For linux systems, you can – starting from major version `20.04` of the docker engine – now also communicate with the host via `host.docker.internal`. This won't work *automatically*, but you need to provide the following run flag:

```
--add-host=host.docker.internal:host-gateway
```

See the answer here: https://stackoverflow.com/a/61424570/3757139

See also this answer below to add to a docker-compose file - https://stackoverflow.com/a/67158212/243392