

02:04:35

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

GIVE FEEDBACK

d0v6m329_d0v6m5q91nsg00fpjs6g

IP: 192.168.0.13 OPEN PORT

Memory: 1.20% (47.85MiB / 3.906GiB) CPU: 0.13%

SSH: ssh ip172-18-0-9-d0v6m3291nsg00fpjs60@direct.labs.play

DELETE EDITOR

```
#####
# WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The FWD team.
#####
[node1] (local) root@192.168.0.13 ~
$
```

We'd love to hear about your usage of Play with Docker. Please take a moment to fill out our survey. TAKE SURVEY

1. Crear un volumen para almacenar la información de la base de datos

docker COMANDO CREAR NOMBRE-VOLUMEN =(Javier Rolando Sea Huanca=> jrsh-vol)

Creamos el volumen

```
docker volume create jrsh-vol
```

01:59:46

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

GIVE FEEDBACK

d0v6m329_d0v6m5q91nsg00fpjs6g

IP: 192.168.0.13 OPEN PORT

Memory: 1.19% (47.6MiB / 3.906GiB) CPU: 0.23%

SSH: ssh ip172-18-0-9-d0v6m3291nsg00fpjs60@direct.labs.play

DELETE EDITOR

```
#####
# WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The FWD team.
#####
[node1] (local) root@192.168.0.13 ~
$ docker volume create jrsh-vol
jrsh-vol
[node1] (local) root@192.168.0.13 ~
$ docker volume ls
DRIVER      VOLUME NAME
local      jrsh-vol
[node1] (local) root@192.168.0.13 ~
$
```

Creamos la red

```
docker network create jrsh-red
```

The screenshot shows the Docker Labs interface for a specific instance. The instance name is `d0v6m329_d0v6m5q91nsg00fpjs6g`. The IP address is `192.168.0.13`. The memory usage is `1.21% (48.31MiB / 3.906GiB)` and the CPU usage is `0.17%`. The terminal output shows the following commands and results:

```
jrsh-vol
[node1] (local) root@192.168.0.13 ~
$ docker volume ls
DRIVER    VOLUME NAME
local     jrsh-vol
[node1] (local) root@192.168.0.13 ~
$ docker network create jrsh-red
0ea7aef12c1d271c1ecff69c06f8ed2a17a0d72da19e7d76b2297d9dcb2474da
[node1] (local) root@192.168.0.13 ~
$ docker network ls
NETWORK ID        NAME                DRIVER    SCOPE
24abd3658c2f     bridge              bridge    local
59638dd70014     docker_gwbridge     bridge    local
b7de70416e2e     host                host      local
0ea7aef12c1d     jrsh-red            bridge    local
e6bc771616fa     none               null      local
```

2. Montar la imagen de postgres `docker container run -d -p 5432:5432 --name=postgresjrsh -e POSTGRES_PASSWORD=PassDocker -v jrsh-vol:/var/lib/postgresql/data postgres`

The screenshot shows the Docker Labs interface for the same instance. The IP address is `192.168.0.13`. The memory usage is `14.61% (584.4MiB / 3.906GiB)` and the CPU usage is `0.35%`. The terminal output shows the execution of `docker ps` and the output of the command:

```
410cd7ec9a40: Pull complete
475b0e32b814: Pull complete
e7aba16d6a5e: Pull complete
89ba8b615fa9: Pull complete
82697a7976df: Pull complete
7e11eb1421f3: Pull complete
2bb588ce4e67: Pull complete
Digest: sha256:6efd0df010dc3cb40d5e33e3ef84acecc5e73161bd3df06029ee8698e5e12c60
Status: Downloaded newer image for postgres:latest
cf9e3ea36ac86a0ee46b3eelbe70c8371c2e7d7b82cadf631306996e545e3bfb
[node1] (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
cf9e3ea36ac8   postgres "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:5432->5432/tcp
5432/tcp      postgresjrsh
[node1] (local) root@192.168.0.13 ~
```

The top screenshot shows a Docker container management interface for a PostgreSQL container named `d0v6m329_d0v6m5q91nsg00fjps6g`. The container's IP is `192.168.0.13`, and its port `5432` is open. The container's memory usage is `14.64% (585.7MiB / 3.906GiB)` and its CPU usage is `0.23%`. The SSH command is `ssh ip172-18-0-9-d0v6m3291nsg00fjps60@direct.labs.play`. The container's status is `done` and the server is `stopped`. The PostgreSQL init process is complete, and the server is ready for start up. The logs show the following messages:

```
2025-06-03 04:34:00.413 UTC [1] LOG: starting PostgreSQL 17.5 (Debian 17.5-1.pgdg120+1) on x86_64-pc-linu
x-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
2025-06-03 04:34:00.414 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
2025-06-03 04:34:00.414 UTC [1] LOG: listening on IPv6 address "::", port 5432
2025-06-03 04:34:00.415 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2025-06-03 04:34:00.418 UTC [62] LOG: database system was shut down at 2025-06-03 04:34:00 UTC
2025-06-03 04:34:00.424 UTC [1] LOG: database system is ready to accept connections
2025-06-03 04:39:00.457 UTC [60] LOG: checkpoint starting: time
2025-06-03 04:39:04.772 UTC [60] LOG: checkpoint complete: wrote 46 buffers (0.3%); 0 WAL file(s) added,
0 removed, 0 recycled; write=4.312 s, sync=0.002 s, total=4.316 s; sync files=11, longest=0.001 s, average
=0.001 s; distance=269 kB, estimate=269 kB; lsn=0/1528680, redo lsn=0/1528628
[node1] (local) root@192.168.0.13 ~
```

The bottom screenshot shows the pgAdmin interface. The table `tablatja` is selected, and its columns are displayed in the table structure view:

column name	data type	is nullable	column default
id	serial	NO	NO

3. Crear contenedor pgAdmin

docker container run -d -p 8080:80 --name=pgadminjrsh -e PGADMIN_DEFAULT_PASSWORD=pass-pgAdmin -e PGADMIN_DEFAULT_EMAIL=correo@google.com dpage/pgadmin4

Docker Playground

labs.play-with-docker.com/p/d0v8ihc69qi000997fp0#d0v8ihc6_d0v8ijs69qi000997fr0

02:38:28

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

d0v8ihc6_d0v8ijs69qi000997fr0

IP: 192.168.0.12 OPEN PORT: 8080

Memory: 21.61% (864.4MiB / 3.906GiB) CPU: 0.25%

SSH: ssh ip172-18-0-30-d0v8ihc69qi000997fp0@direct.labs.play

DELETE EDITOR

```
#####
# WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The PWD team.
#####
[node2] (local) root@192.168.0.12 ~
$ docker container run -d -p 8080:80 --name=pgadminjresh -e PGADMIN_DEFAULT_PASSWORD=pass-pgAdmin -e PGADMIN_DEFAULT_EMAIL=correo@google.com dpage/pgadmin4
Unable to find image 'dpage/pgadmin4:latest' locally
latest: Pulling from dpage/pgadmin4
f18232174bc9: Pull complete
20690bbb262d: Pull complete
7c48c5894605: Pull complete
35d02455d858: Pull complete
19c31ea47cfb: Pull complete
9b4f03ce919b: Pull complete
f3888f615432: Pull complete
```

Docker Playground

labs.play-with-docker.com/p/d0v8ihc69qi000997fp0#d0v8ihc6_d0v8ijs69qi000997fr0

02:37:38

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

d0v8ihc6_d0v8ijs69qi000997fr0

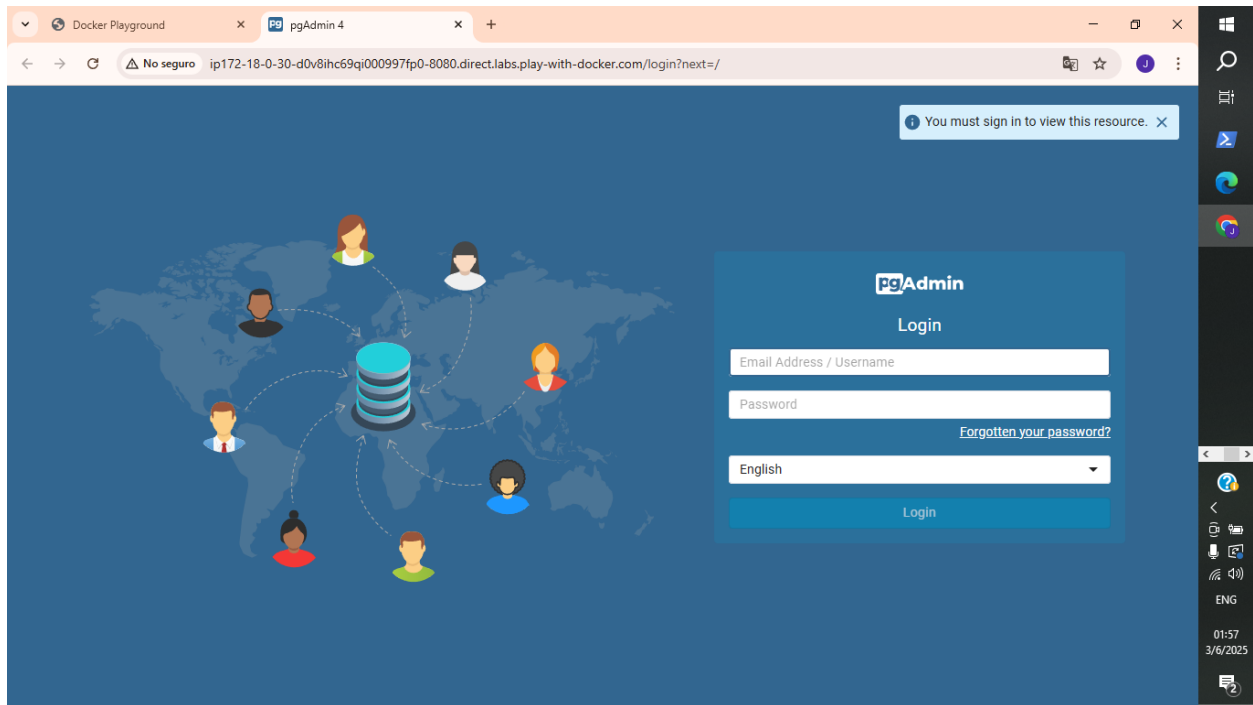
IP: 192.168.0.12 OPEN PORT: 8080

Memory: 21.61% (864.5MiB / 3.906GiB) CPU: 0.24%

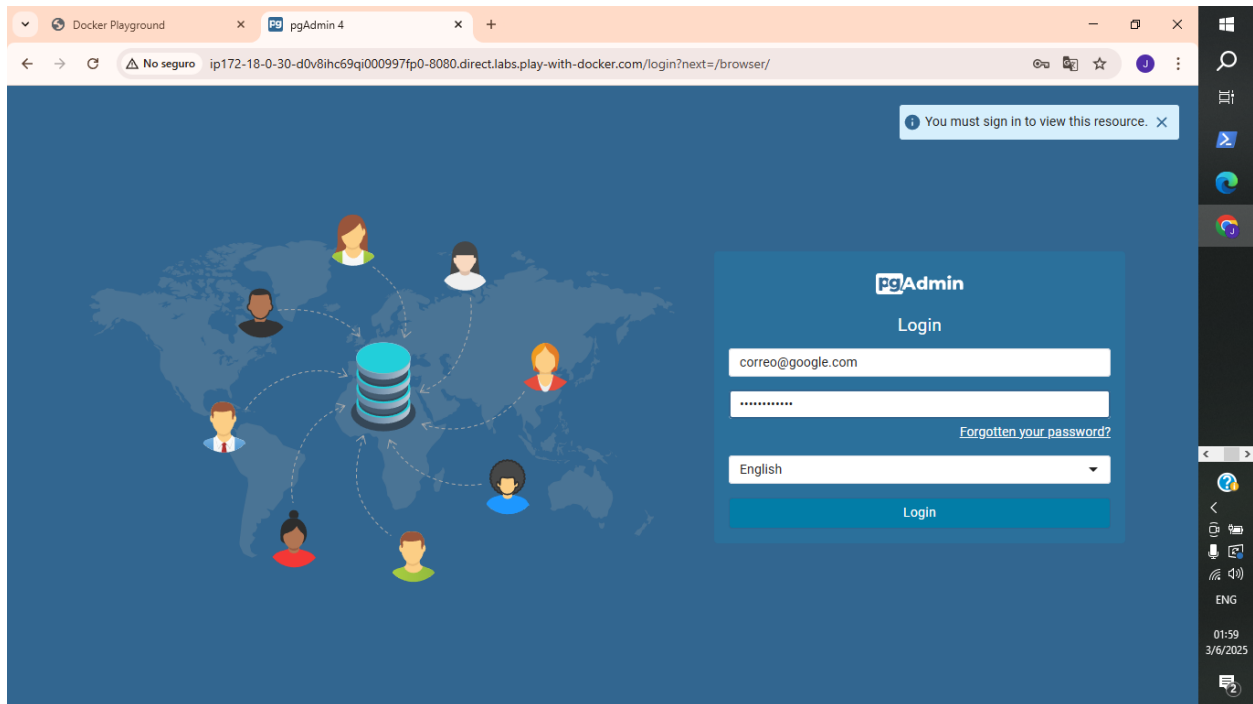
SSH: ssh ip172-18-0-30-d0v8ihc69qi000997fp0@direct.labs.play

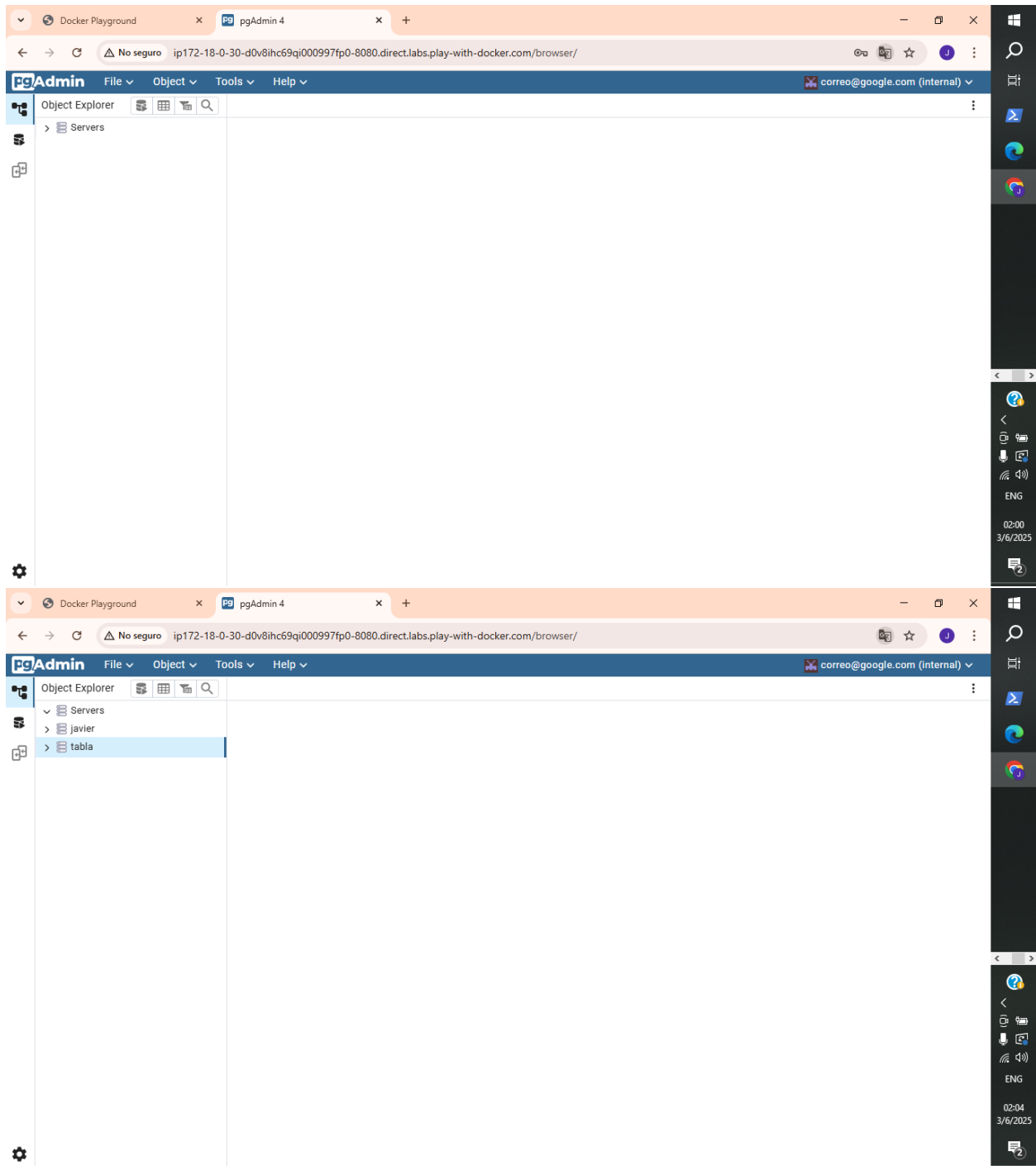
DELETE EDITOR

```
02f79b30c1ff: Pull complete
77da47c991ce: Pull complete
5c5e4eb233fb: Pull complete
8f4e00ca8549: Pull complete
218ba7b7b321: Pull complete
a32d37330f9b: Pull complete
3c2f6c5c12fe: Pull complete
494696b06e5e: Pull complete
abdb8f7da7a0: Pull complete
Digest: sha256:6b1c0db09695758c72abcd5160b7c4cc6d0847ab0c90a2403bdf951f0defb5a
Status: Downloaded newer image for dpage/pgadmin4:latest
ae600f5e52434d9114326ebd66b9270420badab0ee256a4a85434e70de3ce88a
[node2] (local) root@192.168.0.12 ~
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
ae600f5e5243   dpage/pgadmin4   "/entrypoint.sh"        About a minute ago   Up About a minute   443/tcp, 0.0.0
.0:8080->80/tcp   pgadminjresh
[node2] (local) root@192.168.0.12 ~
$
```



4. Ingresar a pgAdmin con las credenciales





Creamos la red

```
docker network create jrsh-red
```

01:55:12

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

d0v8ihc6_d0v8ijs69qi000997fr0

IP: 192.168.0.12

OPEN PORT: 8080

Memory: 21.88% (875.1MiB / 3.906GiB)

CPU: 0.21%

SSH: ssh ip172-18-0-30-d0v8ihc69qi000997fp0@direct.labs.play

DELETE EDITOR

```
[node2] (local) root@192.168.0.12 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
f8f1bce021df        bridge              bridge              local
06d024658f09        docker_gwbridge     bridge              local
517099ab2cff        host                host                local
0ccd1ac4f319        none                null                local
[node2] (local) root@192.168.0.12 ~
$ docker network create jrsh-red
925393ccd1fafa6732f9b43a1dd29969f8a13b6eab50d18a83abfbfcf737a9
[node2] (local) root@192.168.0.12 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
f8f1bce021df        bridge              bridge              local
06d024658f09        docker_gwbridge     bridge              local
517099ab2cff        host                host                local
925393ccd1fa        jrsh-red            bridge              local
0ccd1ac4f319        none                null                local
[node2] (local) root@192.168.0.12 ~
$
```

5. Conectar ambos contenedores a la red jrsh-red

`docker network connect jrsh-red pgadminjrsh`

01:51:21

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

192.168.0.12
node2

d0v8ihc6_d0v8ijs69qi000997fr0

IP: 192.168.0.12

OPEN PORT: 8080

Memory: 21.87% (874.9MiB / 3.906GiB)

CPU: 0.23%

SSH: ssh ip172-18-0-30-d0v8ihc69qi000997fp0@direct.labs.play

DELETE EDITOR

```
[node2] (local) root@192.168.0.12 ~
$ docker network connect jrsh-red pgadminjrsh
[node2] (local) root@192.168.0.12 ~
$ docker inspect jrsh-red
[
  {
    "Name": "jrsh-red",
    "Id": "925393ccd1fafa6732f9b43a1dd29969f8a13b6eab50d18a83abfbfcf737a9",
    "Created": "2025-06-03T06:36:21.537961993Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
    }
  }
]
```

`docker network connect jrsh-red postgresjrsh`

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 01:19:36, a 'CLOSE SESSION' button, and a list of instances: '192.168.0.13 node1' and '192.168.0.12 node2'. The main panel displays details for container 'd0v8ihc6_d0v8iis69qi000997fqg'. It shows the IP '192.168.0.13', an 'OPEN PORT' button with '5432', memory usage '14.65% (585.9MiB / 3.906GiB)', and CPU usage '0.11%'. There's an SSH command: 'ssh ip172-18-0-29-d0v8ihc69qi000997fp0@direct.labs.play'. Below this are 'DELETE' and 'EDITOR' buttons. The bottom section shows a terminal with the following commands and output:

```
Usage: docker network connect [OPTIONS] NETWORK CONTAINER
Connect a container to a network
[node1] (local) root@192.168.0.13 ~
$ docker network connect jrsh-red postgresjrsh
[node1] (local) root@192.168.0.13 ~
$ docker inspect jrsh-red
[
  {
    "Name": "jrsh-red",
    "Id": "95812020c26fa339ec1411a065b2d974038df503cfe59962b5c8f1dc7726bec",
    "Created": "2025-06-03T05:36:10.099899941Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [

```

Verificamos en ambos contenedores

`docker inspect jrsh-red`

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 01:50:38, a 'CLOSE SESSION' button, and a list of instances: '192.168.0.13 node1' and '192.168.0.12 node2'. The main panel displays details for container 'd0v8ihc6_d0v8ijs69qi000997fr0'. It shows the IP '192.168.0.12', an 'OPEN PORT' button with '8080', memory usage '21.88% (875MiB / 3.906GiB)', and CPU usage '0.21%'. There's an SSH command: 'ssh ip172-18-0-30-d0v8ihc69qi000997fp0@direct.labs.play'. Below this are 'DELETE' and 'EDITOR' buttons. The bottom section shows a terminal with the following commands and output:

```
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {
  "ae600f5e52434d9114326ebd66b9270420badeb0ee256a4a85434e70de3ce88a": {
    "Name": "pgadminjrsh",
    "EndpointID": "f6b1f5816cf34b72916386d6f6cb3d9e401c0a777258d60c565f4df8f4011374",
    "MacAddress": "02:42:ac:14:00:02",
    "IPv4Address": "172.20.0.2/16",
    "IPv6Address": ""
  }
},
"Options": {},
"Labels": {}
}
]
[node2] (local) root@192.168.0.12 ~
$
```


The screenshot displays the Docker Playground web interface. The top navigation bar shows the current session and tabs for 'ip172-18-0-29-d0v8ihc69qi000...' and 'pgAdmin 4'. The main content area is for a container named 'd0v8ihc6_d0v8iis69qi000997fqg'. It shows the IP address '192.168.0.13' and a button to 'OPEN PORT' at '5432'. Below this, it displays memory usage at '14.61% (584.5MiB / 3.906GiB)' and CPU usage at '0.49%'. An SSH command is provided: 'ssh ip172-18-0-29-d0v8ihc69qi000997fp0@direct.labs.play'. On the left sidebar, there is a 'CLOSE SESSION' button, an 'Instances' section with a '+ ADD NEW INSTANCE' button, and a list of instances: '192.168.0.13 node1' and '192.168.0.12 node2'. The bottom section of the main area shows a terminal window with a PostgreSQL configuration snippet:

```
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {
  "3e64507ad99df91dc89b4b2e1bb249237a93a78b465b3263d4e7a9ef4386f649": {
    "Name": "postgresjrah",
    "EndpointID": "12c974608c7d02dbaaba5013818de9e7fdfce3c739aa219b1723bc46ba59027c",
    "MacAddress": "02:42:ac:14:00:02",
    "IPv4Address": "172.20.0.2/16",
    "IPv6Address": ""
  }
},
"Options": {},
"Labels": {}
}
]
(node1) (local) root@192.168.0.13 ~
$
```