

Apêndice D – Simulação com o exemplo de Replicação da Aplicação Crud php

NOTA: As figuras a seguir estão apresentadas de acordo com a ordem dos passos que foram seguidos.

Figura D 1 - Criação de Container e Base de dados

Criar Replica de uma Aplicação web
Configure a Replica da aplicação em um Container Docker

Configurações Do Container

Nome do Container: * crudphp

Usuario SSH: * narciso

Senha SSH: * ****

HostName: * crudphp

Imagem: * ubuntursync

Criar **Cancelar**

Figura D 2 - Sucesso na Criação do Container

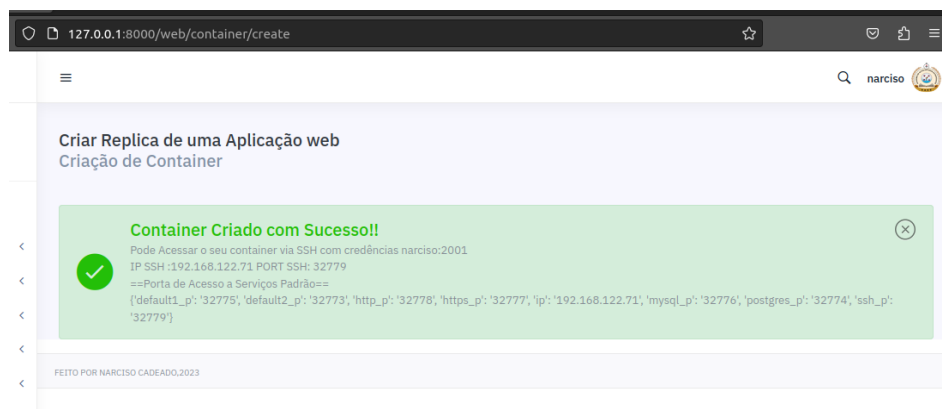


Figura D 3 - Acedendo o Container via SSH

```
narciso@magiccode-X270: $ ssh narciso@192.168.122.71 -p 32779
The authenticity of host '[192.168.122.71]:32779 ([192.168.122.71]:32779)' can't be established.
ED25519 key fingerprint is SHA256:cfaCKMvGVVf8zhpRzyuAuI0C8PKl7r0GBgHGoZzBITY.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:48: [hashed name]
~/.ssh/known_hosts:51: [hashed name]
~/.ssh/known_hosts:52: [hashed name]
~/.ssh/known_hosts:53: [hashed name]
~/.ssh/known_hosts:55: [hashed name]
~/.ssh/known_hosts:56: [hashed name]
~/.ssh/known_hosts:57: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

Figura D 4 - Detalhes do container no sistema

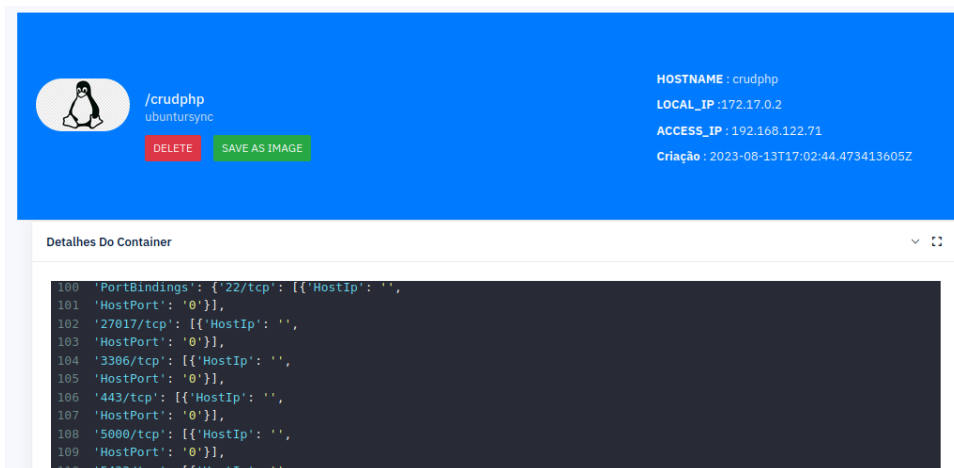


Figura D 5 - Executando a aplicação portátil no servidor da AAEE para copiar o código fonte para o container.

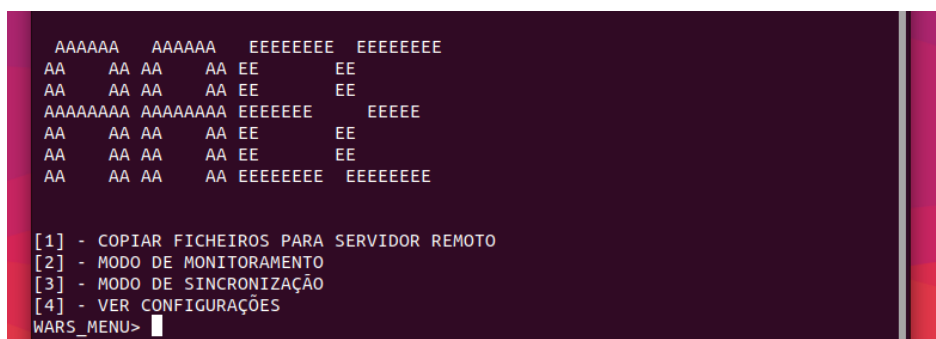


Figura D 6 - Copiando a aplicação para o container a partir da aplicação da aplicação portátil

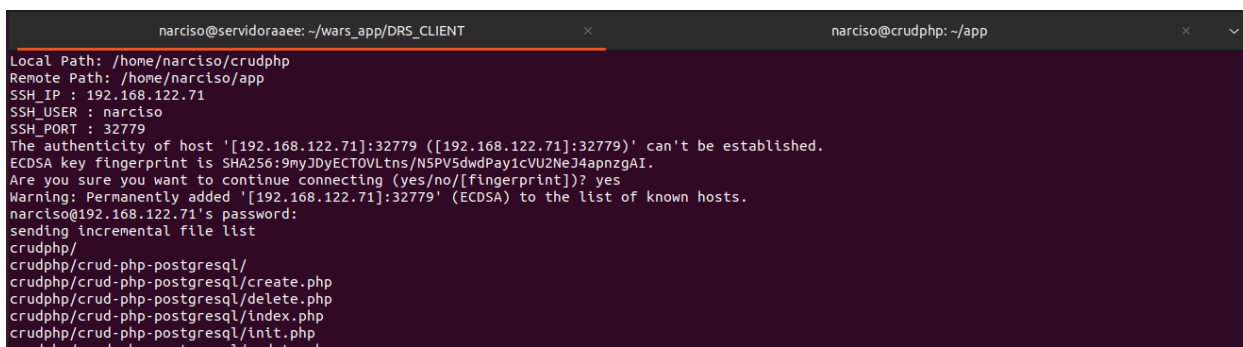


Figura D 7 - Aplicação Crudphp do servidor, replicada em um container

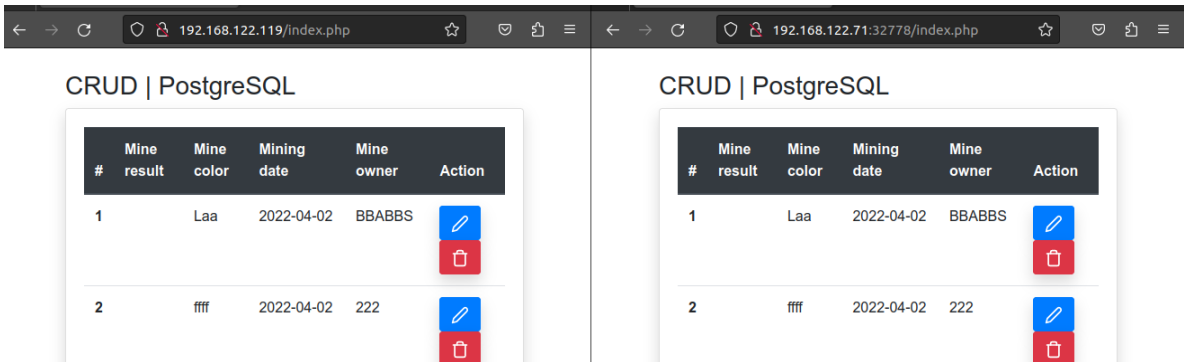


Figura D 8 - Criação do container de base de dados para *Postgres*.



Figura D 9 - Detalhes do Container de base de dados

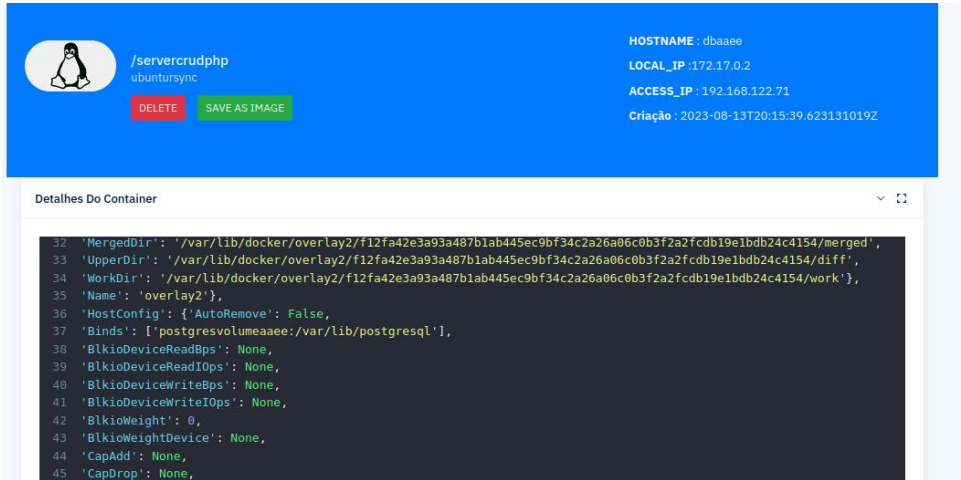


Figura D 10 - Instalação do *postgres*

```
narciso@servidoraeee:~$ sudo apt install postgresql-14 postgresql-client-14 postgresql-contrib-14
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'postgresql-14' instead of 'postgresql-contrib-14'
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  postgresql-client-common postgresql-common
Suggested packages:
  postgresql-doc-14
The following NEW packages will be installed:
  postgresql-14 postgresql-client-14 postgresql-client-common postgresql-common
```

Figura D 11 - Configurações do ficheiro /etc/postgresql/14/main/postgresql.conf

```
GNU nano 4.8 /etc/postgresql/14/main/postgresql.conf
external_pid_file = '/var/run/postgresql/14-main.pid' # write an extra PID file
# (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = 'localhost,192.168.122.119' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
# (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour = off # advertise server via Bonjour
# (change requires restart)
```

Figura D 12 - Configurações do ficheiro /etc/postgresql/14/main/postgresql.conf part 2

```
#parallel_leader_participation = on # can be used in parallel operations
#old_snapshot_threshold = -1 # 1min-60d; -1 disables; 0 is immediate
# (change requires restart)

#-----
# WRITE-AHEAD LOG
#-----

# - Settings -

wal_level = logical # minimal, replica, or logical
# (change requires restart)
#fsync = on # flush data to disk for crash safety
# (turning this off can cause
# unrecoverable data corruption)
#synchronous_commit = on # synchronization level;
```

Figura D 13 - Configurações do arquivo /etc/postgresql/14/main/pg_hba.conf

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local    all             postgres                                peer

# TYPE      DATABASE      USER      ADDRESS              METHOD

# "local" is for Unix domain socket connections only
local    all             all                                peer
# IPv4 local connections:
host     all             all        0.0.0.0/0             scram-sha-256
# IPv6 local connections:
host     all             all        ::1/128               scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local    replication     all                                peer
host     replication     all        0.0.0.0/0             scram-sha-256
host     all              all        192.168.122.71/24     scram-sha-256
host     replication     all        ::1/128               scram-sha-256
```

Figura D 14 - Habilitando o firewall

```
narciso@servidoraeee:~$ sudo nano /etc/postgresql/10/main/postgresql.conf
narciso@servidoraeee:~$ sudo nano /etc/postgresql/14/main/postgresql.conf
narciso@servidoraeee:~$ sudo nano /etc/postgresql/14/main/postgresql.conf
narciso@servidoraeee:~$ sudo nano /etc/postgresql/10/main/pg_hba.conf
narciso@servidoraeee:~$ sudo nano /etc/postgresql/14/main/pg_hba.conf
narciso@servidoraeee:~$ sudo ufw allow from 192.168.122.71 to any port 5432
Skipping adding existing rule
narciso@servidoraeee:~$ sudo systemctl restart postgresql
narciso@servidoraeee:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2023-08-14 12:20:28 UTC; 7s ago
     Process: 13218 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 13218 (code=exited, status=0/SUCCESS)

Aug 14 12:20:28 servidoraeee systemd[1]: Starting PostgreSQL RDBMS...
```

Figura D 15 - Criação da base de dados

```
psql (14.9 (Ubuntu 14.9-1.pgdg20.04+1))
Type "help" for help.

postgres=# CREATE DATABASE crudphp;
CREATE DATABASE
postgres=# \c crudphp;
You are now connected to database "crudphp" as user "postgres".
crudphp=# exit
narciso@servidoraeee:~$ ls
```

Figura D 16 - Configurações da base de dados para replicação

```
narciso@servidoraeee:~/crudphp$ sudo -u postgres psql
psql (14.9 (Ubuntu 14.9-1.pgdg20.04+1))
Type "help" for help.

postgres=# \c crudphp
You are now connected to database "crudphp" as user "postgres".
crudphp=# CREATE TABLE mining_tb(
crudphp(#      mine_id serial PRIMARY KEY,
crudphp(#      mine_color varchar(15) NOT NULL,
crudphp(#      install_date date,
crudphp(#      mine_owner varchar(20)
crudphp(# );
CREATE TABLE
crudphp=# CREATE ROLE narciso WITH REPLICATION LOGIN PASSWORD '2001';
CREATE ROLE
crudphp=# GRANT ALL PRIVILEGES ON DATABASE example TO narciso;
ERROR:  database "example" does not exist
crudphp=# GRANT ALL PRIVILEGES ON DATABASE crudphp TO narciso;
GRANT
crudphp=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO narciso;
GRANT
crudphp=# CREATE PUBLICATION crudphpsub;
CREATE PUBLICATION
crudphp=# ALTER PUBLICATION crudphpsub ADD TABLE mining_tb;
ALTER PUBLICATION
```

Figura D 17 - Configurações da base de dados para replicação

```
narciso@dbaaee:~$ sudo -u postgres psql
could not change directory to "/home/narciso": Permission denied
psql (14.8 (Ubuntu 14.8-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE crudphp;
CREATE DATABASE
postgres=# \c crudphp;
You are now connected to database "crudphp" as user "postgres".
crudphp=# CREATE TABLE mining_tb(
      mine_id serial PRIMARY KEY,
      mine_color varchar(15) NOT NULL,
      install_date date,
      mine_owner varchar(20)
);
CREATE TABLE
crudphp=# CREATE SUBSCRIPTION crudphpsub CONNECTION 'host=192.168.122.119 port=5432 password=2001 user=narciso dbname=crudphp' PUBLICATION crudphpsub;
NOTICE:  created replication slot "crudphpsub" on publisher
CREATE SUBSCRIPTION
```

Figura D 18 - Testando funcionamento

The screenshot shows a terminal window on the left and a web browser window on the right. The terminal window displays the following queries and results:

```
crudphp=# SELECT * FROM mining_tb;
 mine_id | mine_color | install_date | mine_owner
-----+-----+-----+-----
 2 | vvv | 2022-04-02 | hhhh
 3 | ACADEMIA | 2022-04-02 | nARCISO
(2 rows)

crudphp=# SELECT * FROM mining_tb;
 mine_id | mine_color | install_date | mine_owner
-----+-----+-----+-----
 2 | vvv | 2022-04-02 | hhhh
 3 | ACADEMIA | 2022-04-02 | nARCISO
(2 rows)

crudphp=# SELECT * FROM mining_tb;
 mine_id | mine_color | install_date | mine_owner
-----+-----+-----+-----
 2 | vvv | 2022-04-02 | hhhh
 3 | ACADEMIA | 2022-04-02 | nARCISO
(2 rows)

crudphp=# SELECT * FROM mining_tb;
 mine_id | mine_color | install_date | mine_owner
-----+-----+-----+-----
 3 | ACADEMIA | 2022-04-02 | nARCISO
(1 row)

crudphp=#
```

The web browser window displays a CRUD application for PostgreSQL. The title is "CRUD | PostgreSQL". It shows a table with the following columns: Mine #, result, Mine color, Mining date, and Mine owner. The table contains one row with the following data:

Mine #	result	Mine color	Mining date	Mine owner
1		ACADEMIA	2022-04-02	nARCISO

Below the table, there is a blue button labeled "Add new".

Apêndice E – Configuração do monitoramento

NOTA: As figuras a seguir estão apresentadas de acordo com a ordem dos passos que foram seguidos.

Figura E 1 - Criando um servidor de monitoramento para monitorar o servidor

Adicionar Servidor
Configure aqui os dados básicos do Servidor Externo que Deseja Monitorar

Dados Do Servidor

192.168.122.119 crudphp server

Guardar Cancel

Figura E 2 - Servidor Adicionado com sucesso

Gerir Servidores
Aqui tem uma visão geral dos Servidores Monitorados.

✓ Servidor Adicionado com Sucesso

NOME	ENDEREÇO	Actions
SIGCA	sigca.aaee.ac.mz	
crudphp server	192.168.122.119	

Figura E 3 - Iniciando Monitoramento

```
narciso@servidoraee: ~/wars_app/DR5_CLIENT
narciso@servidoraee: ~/wars_app/DR5_CLIENT
[1] - MONITORAR INSTANCIA
[2] - INFO
WARS_MENU>MONITOR> 1
[#]STATUS ENVIADO
{'response': 200}
[#]STATUS ENVIADO
{'response': 200}
```

Figura E 4 - Servidor sendo monitorado

127.0.0.1:8000/web/server/details?id=P8nS6j-1dd2pSoVsKA18e7_YXzdM2Drz4zhLrfUSzPc

crudphp server
MONITORED SERVER
DELETE

TOKEN
Pclj71rC88STOW_Yvw5LTGmsWOTHVGJHHxLVszRdrId
bb8dc11098045379a2d5aafedc8ad2

SERVER ID P8nS6j-
1dd2pSoVsKA18e7_YXzdM2Drz4zhLrfUSzPc

● CPU: 0
● RAM: 40.1
● STORAGE: 29.8

Apêndice F – Configuração de *Failover* da aplicação Crud php

NOTA: As figuras a seguir estão apresentadas de acordo com a ordem dos passos que foram seguidos.

Figura F 1 - Containers de replicação de base de dados e de aplicação em execução

Gerir Replicas
Aqui tem uma visão geral dos Containers.

ID	NOME	IMAGEM	LOCAL_IP	ESTADO	Actions
92c170c3ff28	[/aeeeapp]	ubuntursync	172.17.0.3	running	
f7b0e30b0f2f	[/dbaaee]	ubuntursync	172.17.0.2	running	

Figura F 2 - Guardando o container como uma imagem

Guardar o Container 92c170c3ff28 Como Imagem
Ao guardar o container como imagem, ele estará Acessível para o Cluster Swarm

Salvar Container Como Imagem

Registrar

Figura F 3 - Confirmar o envio para o registrador privado

Confirmar Envio de 192.168.122.10/aeeeaplicacao
Confirme o envio da imagem para o Registry

Confirmar Push

Figura F 4 - Criação de um serviço usando a imagem previamente criada

Criar Serviço

Os Serviços Possibilitam Executar a Aplicação no Cluster

Antes de Criar um Serviço, Certifique-se de ter a Imagem da Aplicação em um Registry

O Serviço é criado usando um objecto Json.

Configurações Do Serviço

```

1 {
2   "Name": "servicoapp",
3   "TaskTemplate": {
4     "ContainerSpec": {
5       "Image": "192.168.122.10/aeeeaplicacaoweb",
6       "Command": [
7         "bash",
8         "-c",
9         "sudo service nginx start && sudo service php7.4-fpm start && /usr/sbin/sshd -D"
10      ]
11     },
12     "Networks": [
13       {
14         "Target": "narcisonet"
15       }
16     ]
17   }
18 }

```


Figura F 5 - Criação do serviço

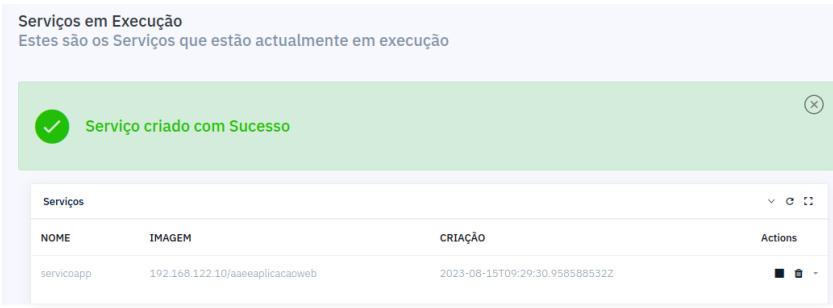


Figura F 6 - Detalhes do serviço criado



Figura F 7 - Verificar o serviço em funcionamento



CRUD | PostgreSQL



Figura F 8 - Criação de um *Failover* para um servidor

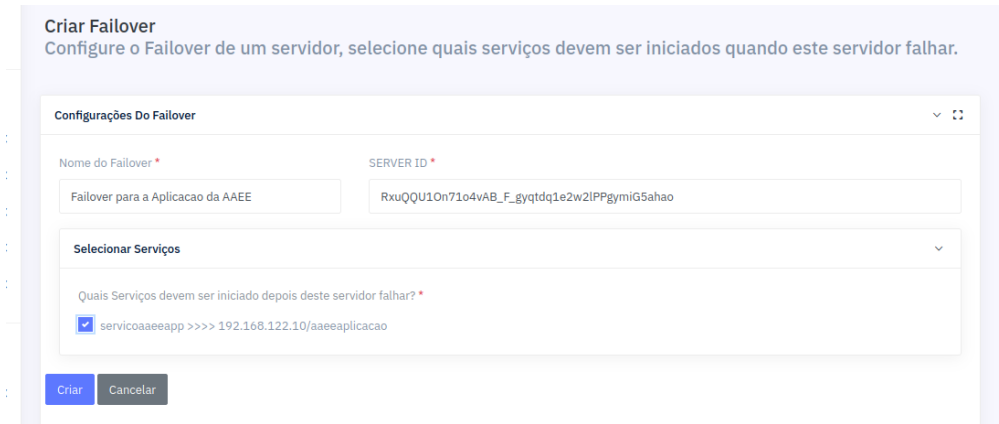


Figura F 9 - Serviço offline porque o servidor continua a enviar o status (esta online)

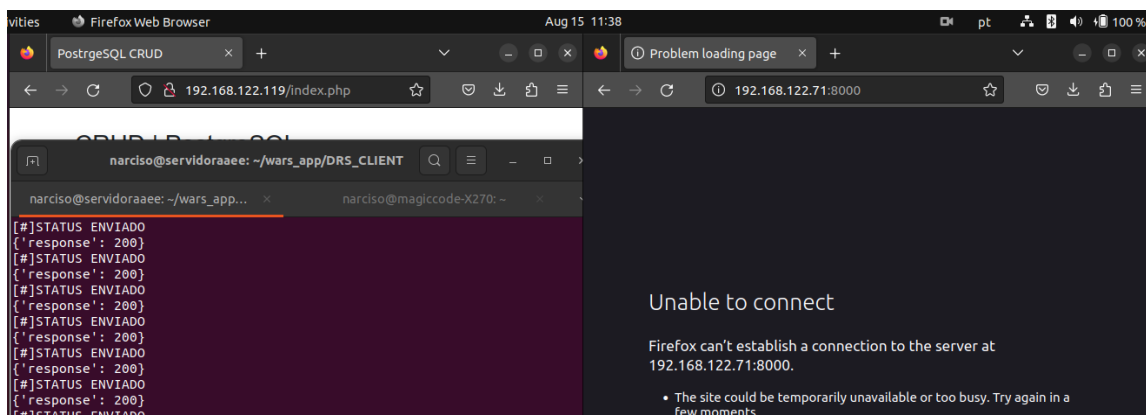


Figura F 10 - O serviço é criado pelo *failover* configurado logo após do servidor parar de enviar os dados para o sistema (o servidor está *offline*)

