

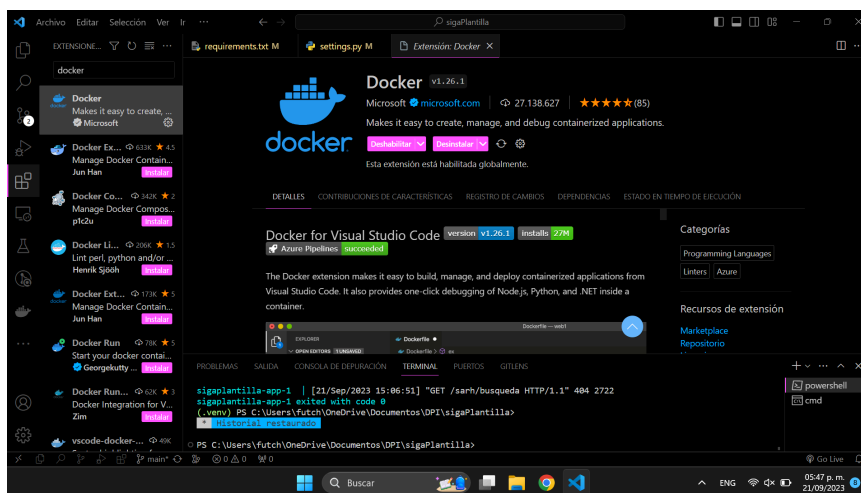
Configuración de Docker para el proyecto de la DEP

Para la configuración necesitamos tener instalado:

Docker desktop:



Y la extensión de vs llamada docker.



1. Clonar el proyecto del git.

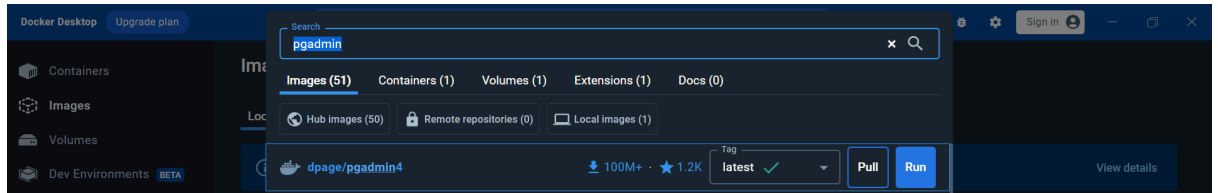
Primero que nada clonar el repositorio

2. Creación de imágenes y contenedores en docker.

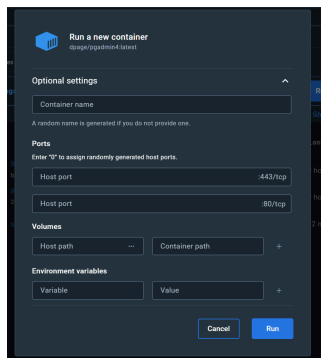
al abrir la aplicación de docker no tendrás nada instalado para ello vamos a buscar:

2.1 pgadmin

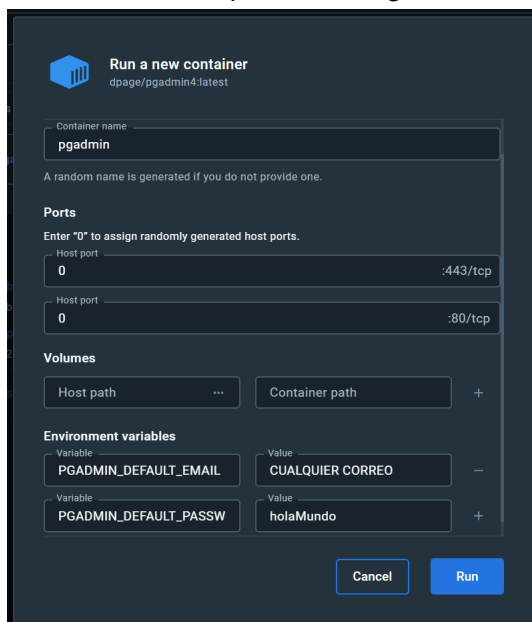
Nos la primera imagen dpape/pgadmin4 es el que vamos a utilizar y damos click en run.



al darle click en run te aparecerá la siguiente pantalla



vas a llenar los inputs de la siguiente manera:



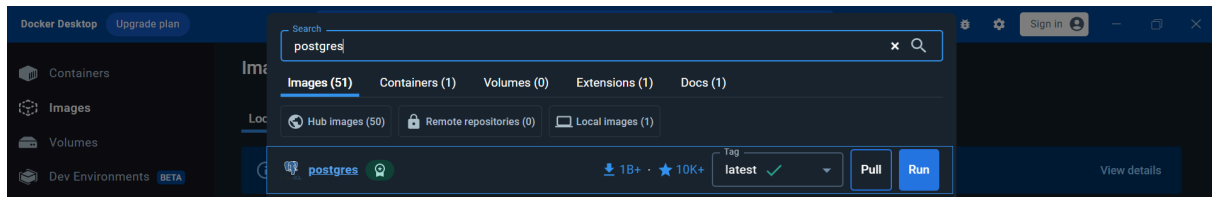
VARIABLE 1 -> PGADMIN_DEFAULT_EMAIL : cualquier correo.

VARIABLE 2 -> PGADMIN_DEFAULT_PASSWORD: cualquier contraseña; a recomendada holaMundo.

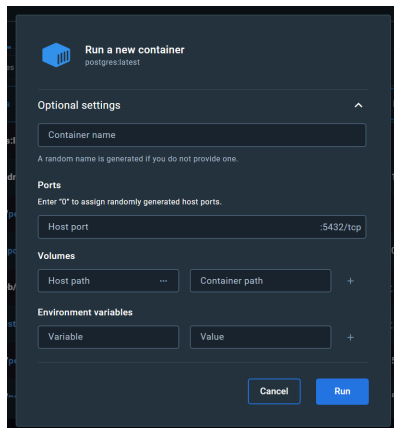
ahora si le das click en run de nuevo esto les creará un contenedor.

postgres

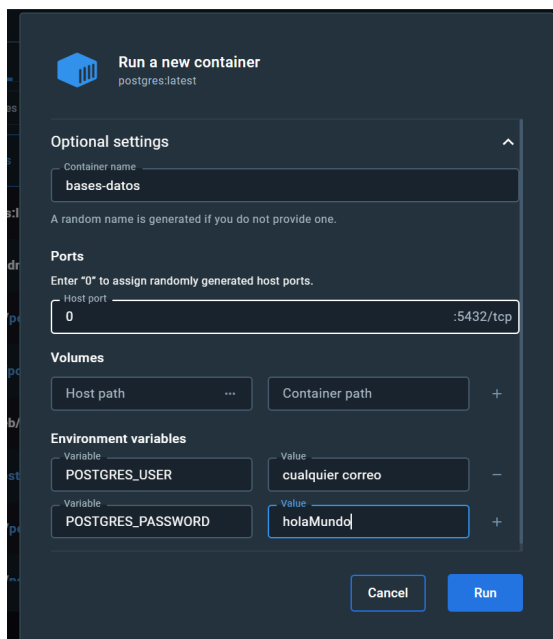
Igual que la imagen anterior usamos la primera y damos click en run.



igual nos desplegará una ventana como esta.



y vamos agregar la siguiente información



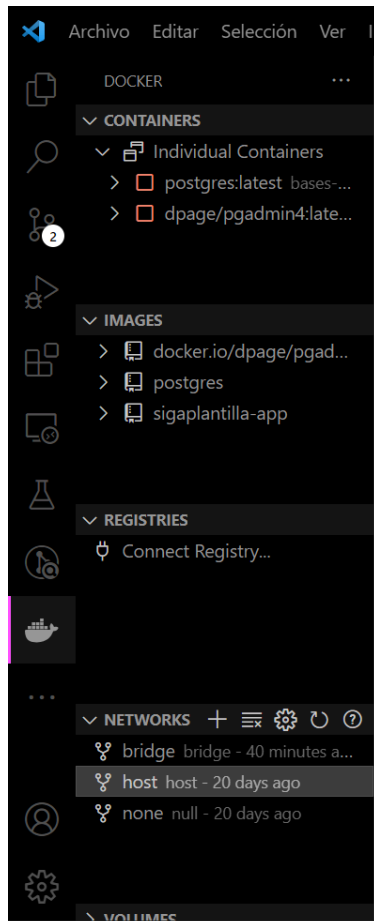
VARIABLE 1 -> POSTGRES_USER: cualquier usuario recomendado administrador.
VARIABLE 2 -> POSTGRES_PASSWORD: cualquier contraseña recomendado
holaMundo.

y le damos click en run.

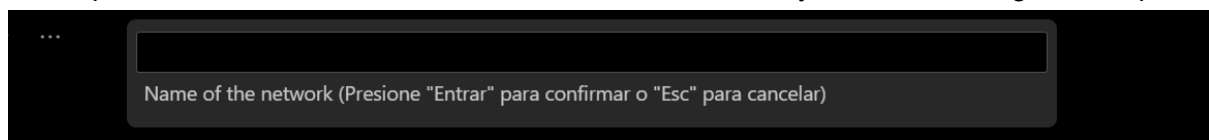
3. Creación del siga-bridge.

Vamos a crear la red siga-bridge que nos servirá como un puente para conectar los contenedores que acabamos de crear para ello nos vamos al vs en la carpeta del proyecto.

cuando instalamos la extensión del docker del lado izquierdo nos aparecerá un icono de la ballena del docker y le damos click nos aparecera lo siguiente

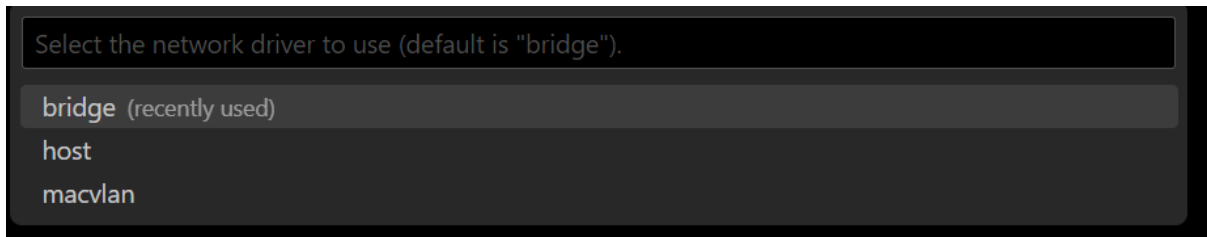


en el apartado de network le damos click en el icono de mas y nos abrirá el siguiente input



aquí vamos a ingresar el nombre de siga_bridge y le damos enter,

nos dejará elegir lo siguiente:



nosotros le daremos en bridge y ya tendremos una red para conectar lo anterior y continuamos con el siguiente paso

4. Conexión de las plantillas.

Primero vamos a comprobar si ya tenemos la red que creamos para la conexión, abrimos un power shell de windows y vamos a colocar el siguiente comando:

docker network list

```
Windows PowerShell
PS C:\Users\futch> docker network list
NETWORK ID        NAME          DRIVER        SCOPE
b6c5b6154c9b     bridge       bridge        local
991438c50d01     host        host          local
d0ea07997e42     none        null          local
db5369b26dc7     siga_bridge  bridge        local
PS C:\Users\futch> |
```

vemos que ya tenemos la network siga_bridge

ahora vamos a conectar nuestros contenedores y para ello vamos a necesitar el siguiente comando:

docker network connect (la red que creamos) (contenedor).

docker network connect siga_bridge bases-datos
docker network connect siga_bridge pgadmin

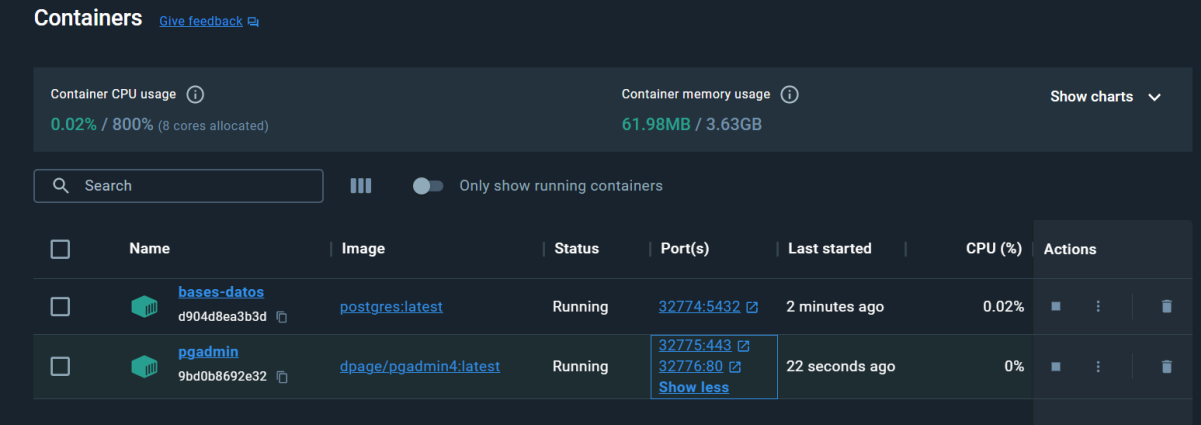
```
PS C:\Users\futch> docker network connect siga_bridge bases-datos
PS C:\Users\futch> docker network connect siga_bridge pgadmin
PS C:\Users\futch> |
```

No debería de dar ningún error

por ahorita ya terminamos la conexión de estos dos contenedores.

5. Añadir un servidor y creación de las bases de datos, roles.

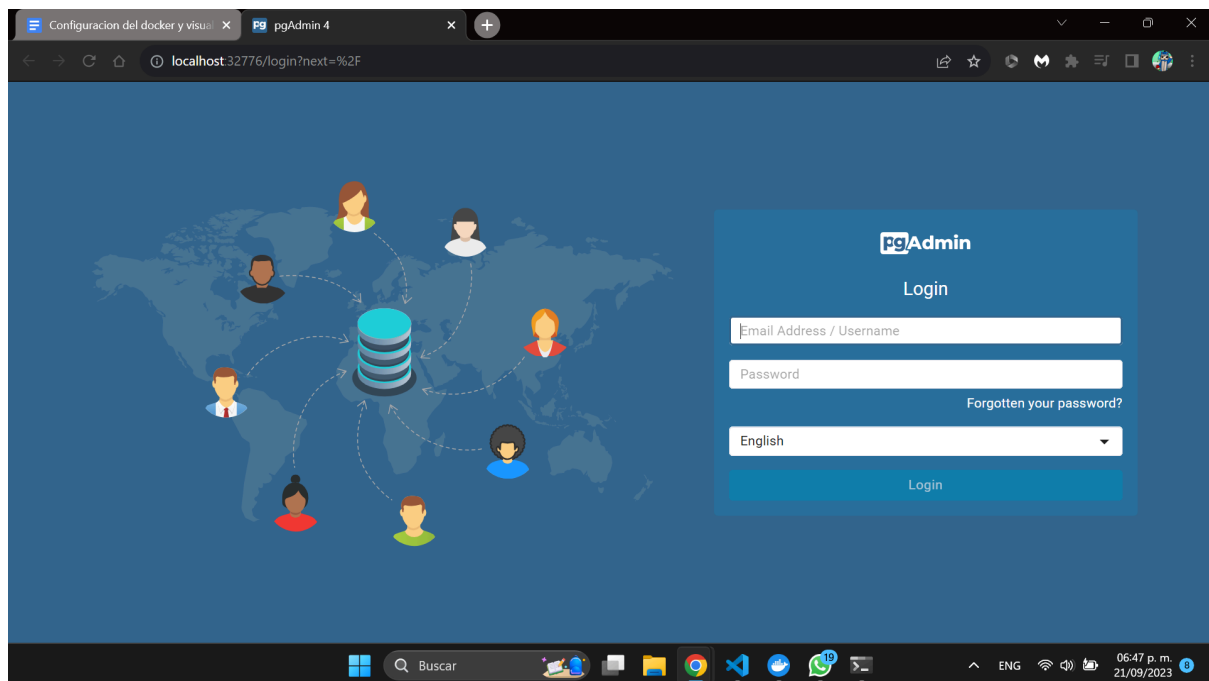
En la sección de contenedores deberán de tener corriendo o en play sus contenedores



The screenshot shows the Docker Desktop interface. At the top, it displays 'Containers' with a 'Give feedback' link. Below this, there are two summary cards: 'Container CPU usage' showing 0.02% / 800% (8 cores allocated) and 'Container memory usage' showing 61.98MB / 3.63GB. A 'Show charts' dropdown is also present. Below the summary cards is a search bar and a toggle switch for 'Only show running containers'. The main part of the interface is a table listing the containers.

	Name	Image	Status	Port(s)	Last started	CPU (%)	Actions
<input type="checkbox"/>	bases-datos d904d8ea3b3d	postgres:latest	Running	32774:5432	2 minutes ago	0.02%	[Stop] [Refresh] [Delete]
<input type="checkbox"/>	pgadmin 9bd0b8692e32	dpage/pgadmin4:latest	Running	32775:443 32776:80 Show less	22 seconds ago	0%	[Stop] [Refresh] [Delete]

en la parte del pgadmin vamos a darle click en el link que tiene como puerto el número 80 y nos abra una pagina local de pgadmin:



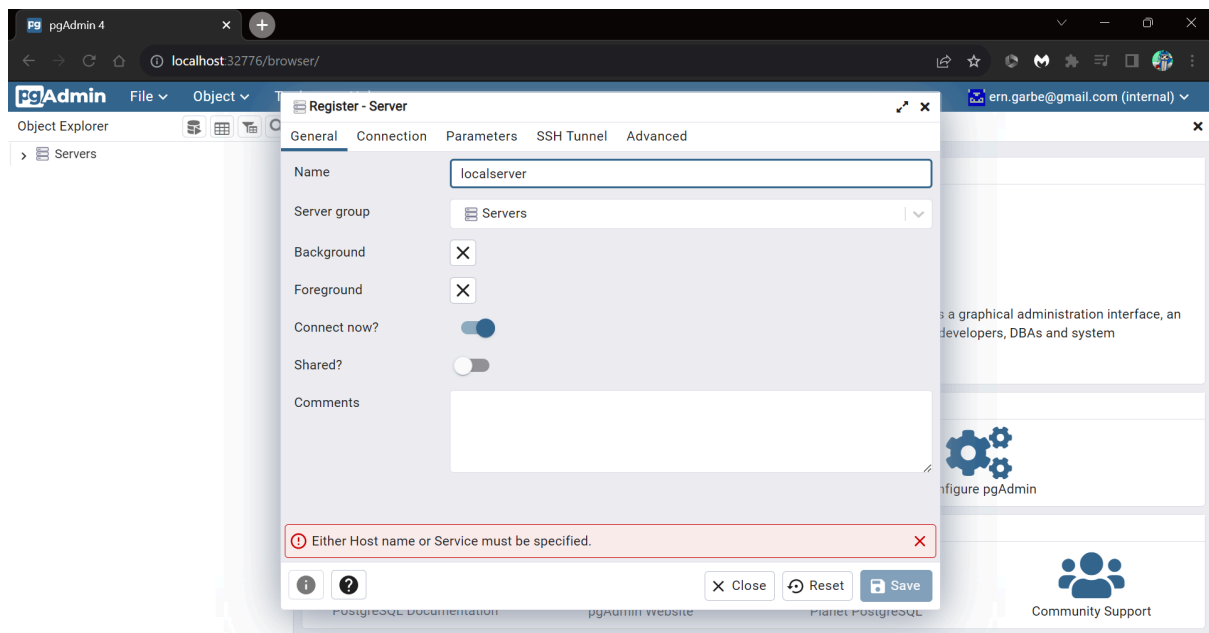
Aquí vamos a poner el correo y la contraseña que creamos anteriormente con el contenedor pgadmin

5.1 Añadir servidor

veremos la siguiente pantalla y le damos click en añadir un nuevo server



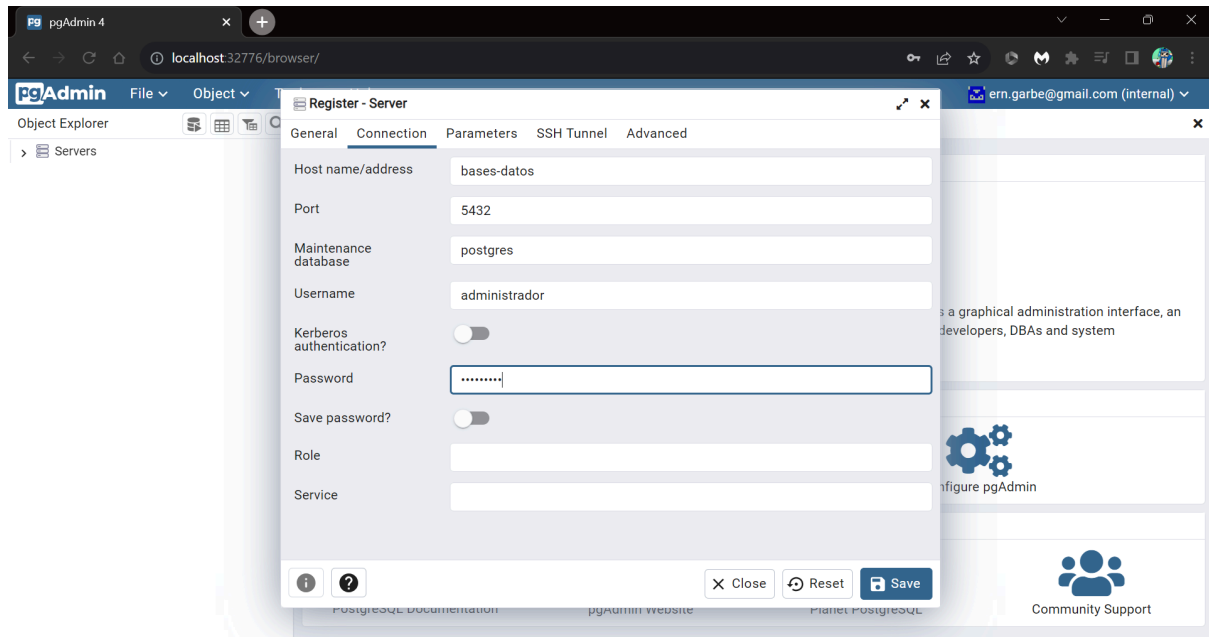
nos aparece la siguiente ventana y le ponemos como nombre localserver y nos vamos a la parte de connection



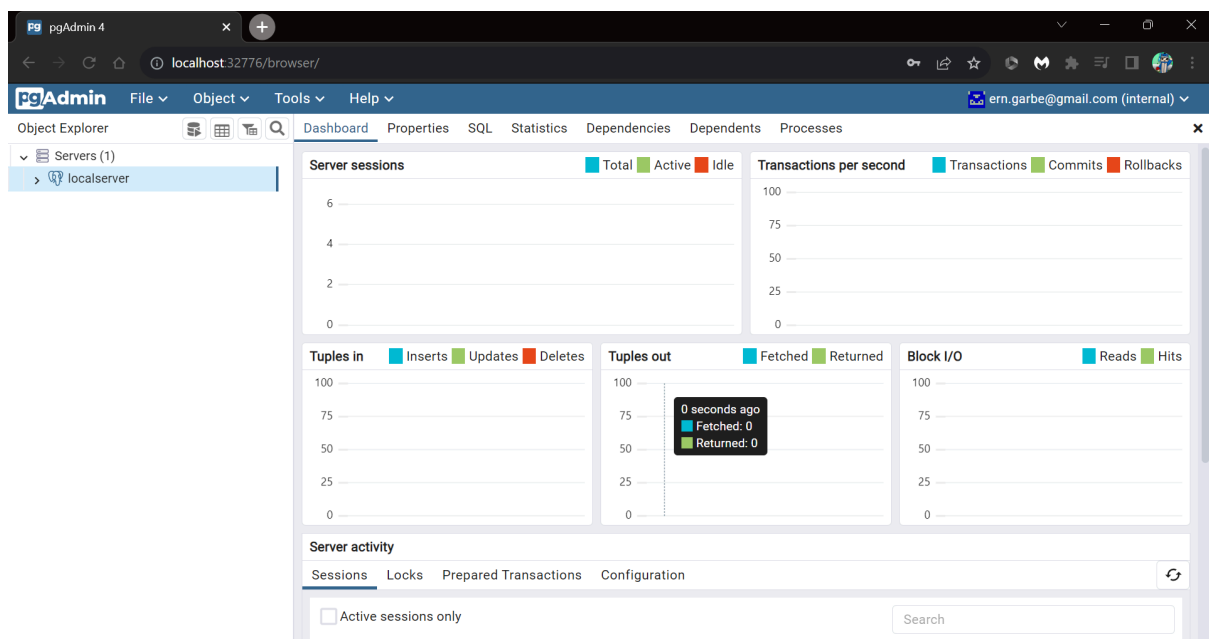
y aquí vamos añadir los siguientes datos, el host/addr es a donde nos vamos a conectar en este caso es al contenedor de bases-datos

Username es el user que le pusimos al contenedor de bases-datos y al igual que su contraseña, cuando llenemos los inputs le daremos click al boton de save y si toda la

conexion esta bien nos dejara crear el servidor de la base de datos

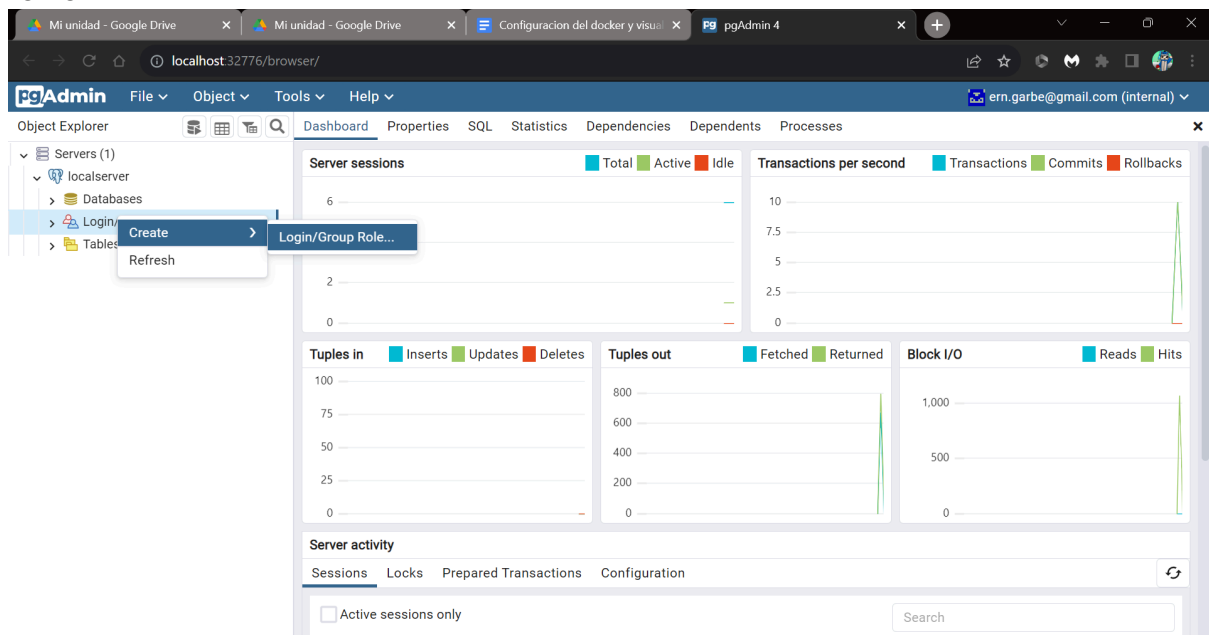


y nos mostrará la siguiente pantalla

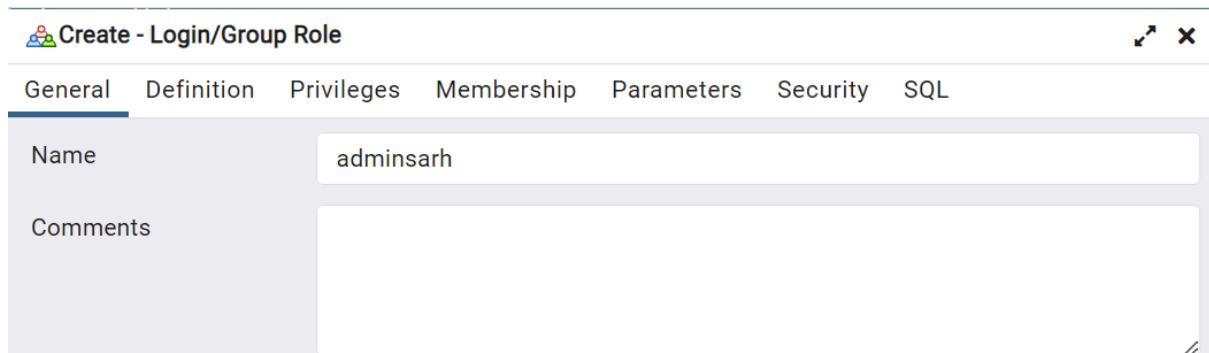


5.2 Creación de los roles.

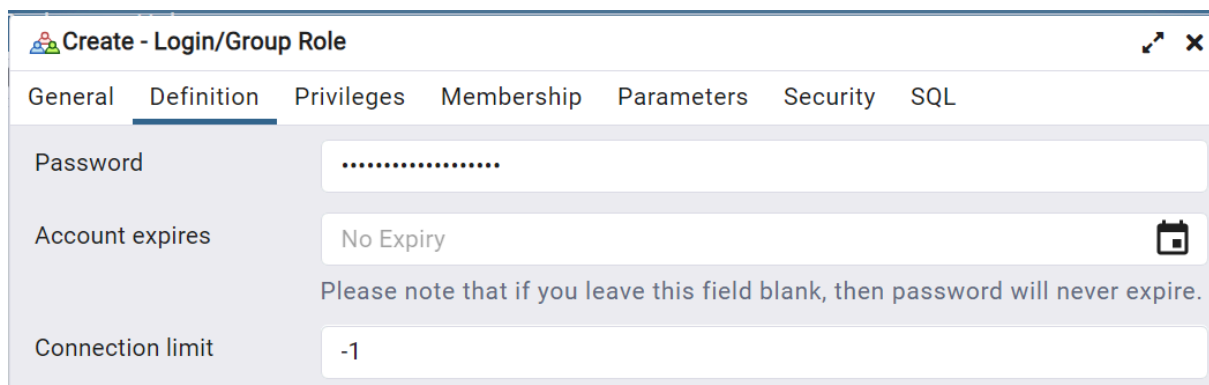
Vamos a crear los roles primero y para ello le damos click a lo que se ve en la imagen y agregaremos el primer rol que es: adminsarh



nos abrirá lo siguiente y ponemos los datos que están en las siguientes imágenes, cabe aclarar que estos usuarios y contraseñas se agregan en el settings de mysite en el proyecto django.



la password es la siguiente: `adminsarhpassword.$`



y por último, activamos el login

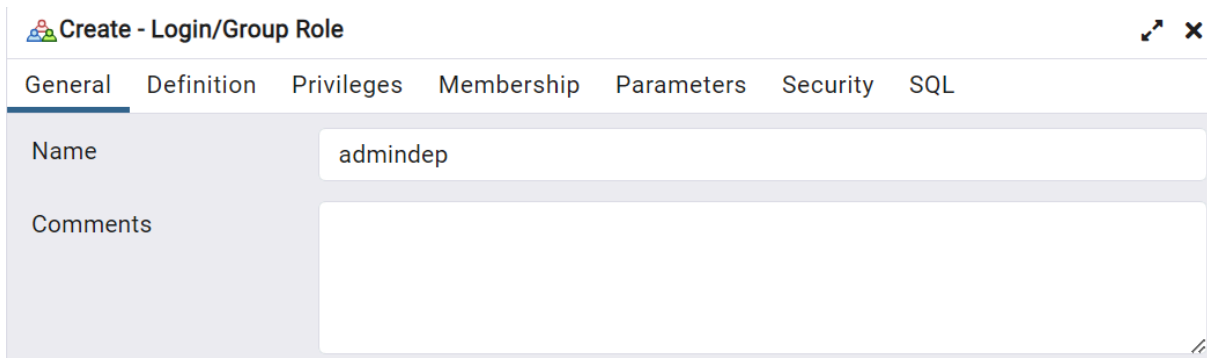


The screenshot shows the 'Create - Login/Group Role' window with the 'Privileges' tab selected. The window has a title bar with a maximize icon and a close button. Below the title bar are tabs: General, Definition, Privileges (active), Membership, Parameters, Security, and SQL. The main area contains a list of privileges with toggle switches:

Privilege	Toggle State
Can login?	On (Blue)
Superuser?	Off (Grey)
Create roles?	Off (Grey)
Create databases?	Off (Grey)
Inherit rights from the parent roles?	On (Blue)
Can initiate streaming replication and backups?	Off (Grey)

y le damos click en guardar.

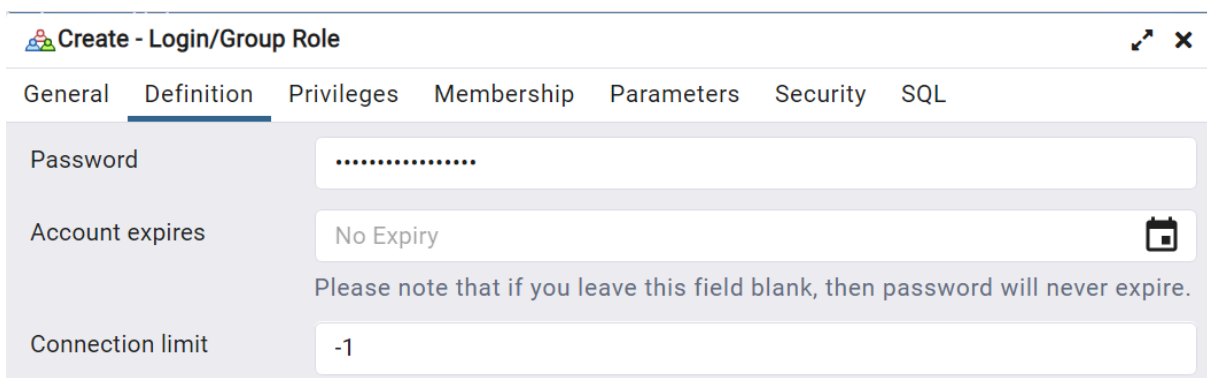
Ahora vamos a crear al segundo rol que es: admindep
y será igual que en las imágenes anteriores



The screenshot shows the 'Create - Login/Group Role' window with the 'General' tab selected. The window has a title bar with a maximize icon and a close button. Below the title bar are tabs: General (active), Definition, Privileges, Membership, Parameters, Security, and SQL. The main area contains two fields:

Field	Value
Name	admindep
Comments	

contraseña es : `admindeppassword.$`

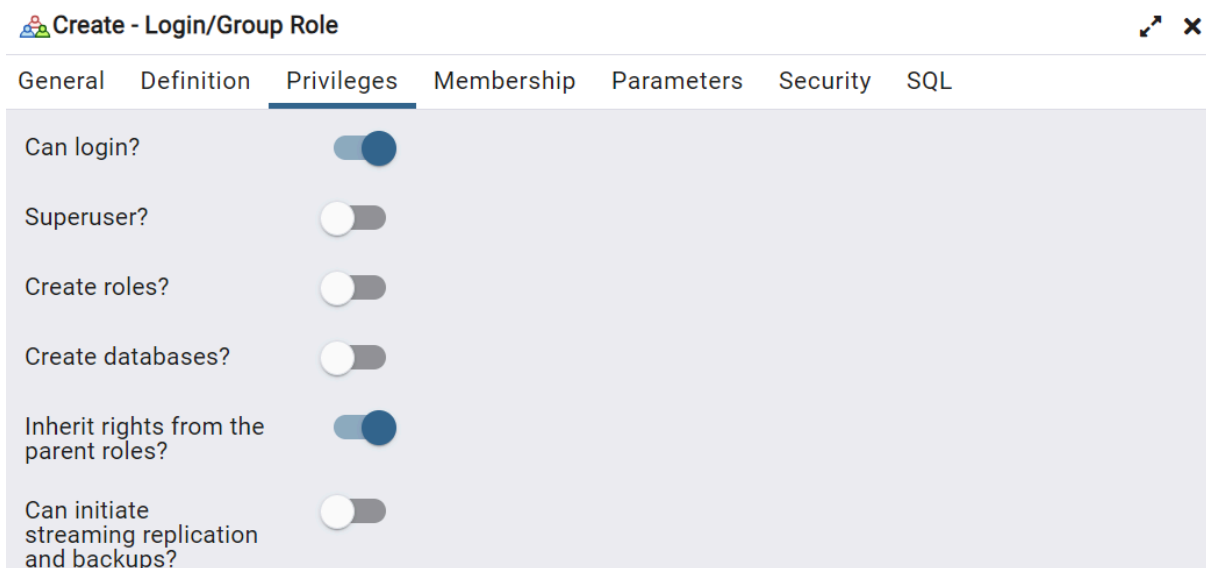


The screenshot shows the 'Create - Login/Group Role' window with the 'Definition' tab selected. The window has a title bar with a maximize icon and a close button. Below the title bar are tabs: General, Definition (active), Privileges, Membership, Parameters, Security, and SQL. The main area contains three fields:

Field	Value
Password
Account expires	No Expiry
Connection limit	-1

Below the 'Account expires' field, there is a note: "Please note that if you leave this field blank, then password will never expire."

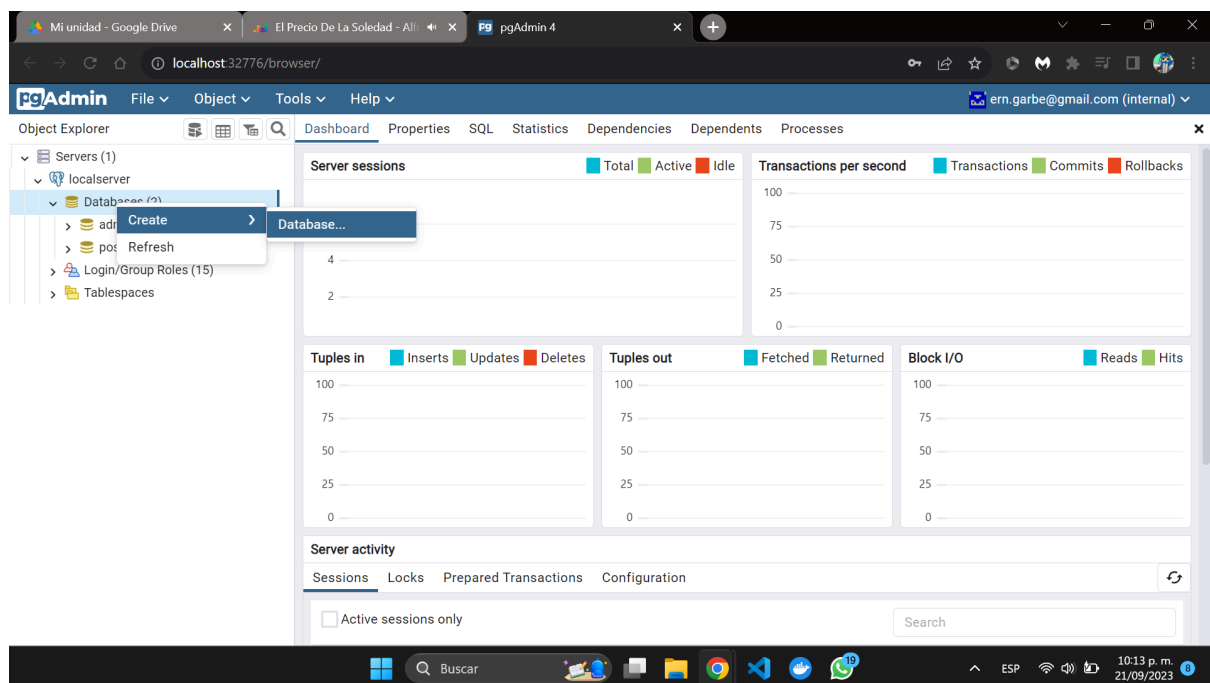
y activamos el login



y lo guardamos, ya estarían los dos roles que necesitamos

5.3 Creación de las bases de datos

Vamos a crear las bases de datos y asociarlas con nuestros roles creados para crear una base de datos hacemos lo que en la siguiente imagen:






Nos abrirá una ventana para colocar la información de la base de datos que queremos crear y aquí solamente vamos a modificar el nombre de la base de datos y la asociamos con un rol.

Nombre de la base de datos: auth_db

rol asociado: adminsarh

y guardamos

 **Create - Database**  

General

Definition

Security

Parameters



Advanced

SQL

Database

auth_db

Owner

 adminsarh 




Comment

Creamos otra base de datos con la siguiente información:

Nombre de la base de datos: dep_db

rol asociado: admindep

y guardamos

 **Create - Database**  

General

Definition

Security

Parameters



Advanced

SQL

Database

dep_db

Owner

 admindep 

Comment

6.- últimas configuraciones

Abrimos el vs en la carpeta donde está el proyecto, una vez dentro abrimos una terminal y prendemos el entorno virtual con el siguiente comando

```
..\venv\bin\activate
```

```
PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla>
```

```
* Historial restaurado
```

```
● PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla> .\.venv\bin\activate
```

```
○ (.venv) PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla>
```

una vez hecho esto instalamos los requerimientos si ya lo hiciste omite esta parte, para instalar los requerimientos usamos el siguiente comando:

pip install -r .\app\requirements.txt

```
● PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla> .\.venv\bin\activate
```

```
● (.venv) PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla> pip install -r .\app\requirements.txt
```

```
○ (.venv) PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla>
```

6.1 Creación del contenedor de sigaPlantilla

En la misma terminal del vs pondremos el siguiente comando

docker compose up

```
(.venv) PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla> docker compose up
```

```
[+] Running 2/2
```

```
✓ Network backend Created 0.1s
```

```
✓ Container sigaplantilla-app-1 Created 0.2s
```

```
Attaching to sigaplantilla-app-1
```

```
sigaplantilla-app-1 | Watching for file changes with StatReloader
```

```
sigaplantilla-app-1 | Performing system checks...
```

```
sigaplantilla-app-1 | Exception in thread django-main-thread:
```

```
sigaplantilla-app-1 | Traceback (most recent call last):
```

```
sigaplantilla-app-1 |   File "/usr/local/lib/python3.8/threading.py", line 932, in _bootstrap_inner
```

```
sigaplantilla-app-1 |     self.run()
```

```
sigaplantilla-app-1 |   File "/usr/local/lib/python3.8/threading.py", line 870, in run
```

```
sigaplantilla-app-1 |     self._target(*self._args, **self._kwargs)
```

```
sigaplantilla-app-1 |   File "/usr/local/lib/python3.8/site-packages/django/utils/autoreload.py", line 64, in wrap  
per
```

```
sigaplantilla-app-1 |     fn(*args, **kwargs)
```

```
sigaplantilla-app-1 |   File "/usr/local/lib/python3.8/site-packages/django/core/management/commands/runserver.py"  
, line 133, in inner_run
```

```
sigaplantilla-app-1 |     self.check(display_num_errors=True)
```

```
sigaplantilla-app-1 |   File "/usr/local/lib/python3.8/site-packages/django/core/management/base.py", line 556, in  
check
```

```
sigaplantilla-app-1 |     raise SystemCheckError(msg)
```

```
sigaplantilla-app-1 | django.core.management.base.SystemCheckError: SystemCheckError: System check identified some  
issues:
```

```
sigaplantilla-app-1 | ERRORS:
```













```
sigaplantilla-app-1 | dep.Dep.imagen: (fields.E210) Cannot use ImageField because Pillow is not installed.
```

```
sigaplantilla-app-1 | HINT: Get Pillow at https://pypi.org/project/Pillow/ or run command "python -m pip install  
Pillow".
```

```
sigaplantilla-app-1 | System check identified 1 issue (0 silenced).
```

si te sale eso ya chingaste jajaja

Esto crear una contenedor en el docker

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	CPU (%)	Actions
<input type="checkbox"/>	 pgadmin 9bd0b8692e32	dpage/pgadmin4:lat	Running	32775:443 ↗ 32776:80 ↗ Show less	4 hours ago	0.1%	  
<input type="checkbox"/>	 bases-datos d904d8ea3b3d	postgres:latest	Running	32774:5432 ↗	4 hours ago	0%	  
<input type="checkbox"/>	>  sigaplantilla		Running (1/1)		46 seconds ago	5.95%	  

6.2 Añadir el nuevo contenedor a siga_bridge

Vamos a añadir este contenedor a nuestra network que habíamos creado llamada `sigu_bridge`

ahora es necesario instalar los requerimientos en el contenedor que hemos creado
desplegamos el menú del contenedor del sigaplantilla y veremos la siguiente información

Container ID	Name	Status	Restart Count	Created	Usage	Actions
d7d907d621	sigaplantilla	Running (1/1)	0	8 minutes ago	4.27%	Stop, Details, Delete
d7d907d621	app-1 sigaplantilla-app	Running	0	8 minutes ago	4.27%	Stop, Details, Delete

vamos a clicar en donde dice app-1 y nos abrirá lo siguiente:

sigaplantilla-app

sigaplantilla-app

d7d907d62137

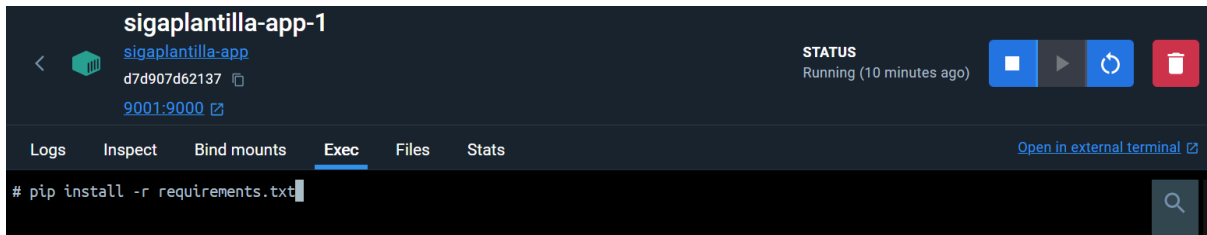
9001:9000 [🔗](#)

STATUS
 Running (9 minutes ago)

Logs	Inspect	Bind mounts	Exec	Files	Stats
2023-09-21 22:37:07 00 90457	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/DataTables-1.12.1/js/jquery.dataTables.min.js HTTP/1.1"	200		
2023-09-21 22:37:07 " 200 17208	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/Responsive-2.3.0/js/dataTables.responsive.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/sarh/img/ITMORELIA-Logo.png HTTP/1.1"	200	274761	
2023-09-21 22:37:07 1450	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/Buttons-2.2.3/js/buttons.bootstrap5.min.js HTTP/1.1"	200		
2023-09-21 22:37:07 25631	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/Buttons-2.2.3/js/dataTables.buttons.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/JZip-2.5.0/js/jzip.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/Buttons-2.2.3/js/buttons.bootstrap5.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/plugins/DataTables/Buttons-2.2.3/js/dataTables.buttons.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/plugins/summernote/summernote-bs4.min.js HTTP/1.1"	200		
2023-09-21 22:37:07 5	[22/Sep/2023 04:37:07]	"GET /static/plugins/bootstrap-multiselect.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/plugins/tinypicker-1.3.5/jquery.tinypicker.min.js HTTP/1.1"	200		
2023-09-21 22:37:07	[22/Sep/2023 04:37:07]	"GET /static/sarh/is/search.is HTTP/1.1"	200		

nos vamos a ir al apartado de la consola en mi caso se llama exec y escribimos el siguiente cuando para instalar los requerimientos

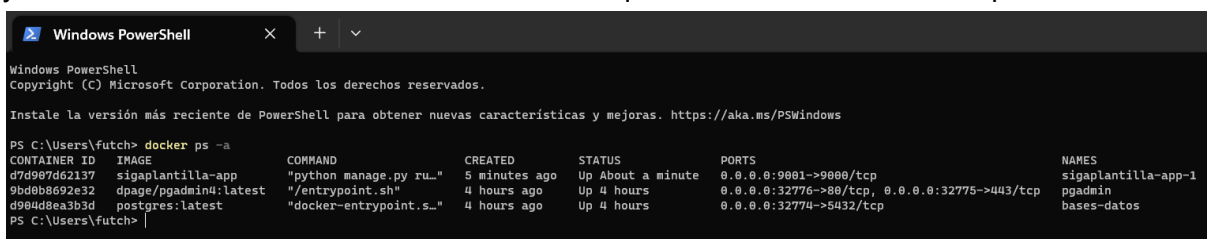
```
pip install -r requirements.txt
```



Después abrimos el powershell de windows y vamos a ver como se llama: con el siguiente comando

docker ps -a

y nos mostrará el nombre de los contenedores que tenemos nos interesa el primero



ahora lo añadimos a la red con el comando anterior que es:

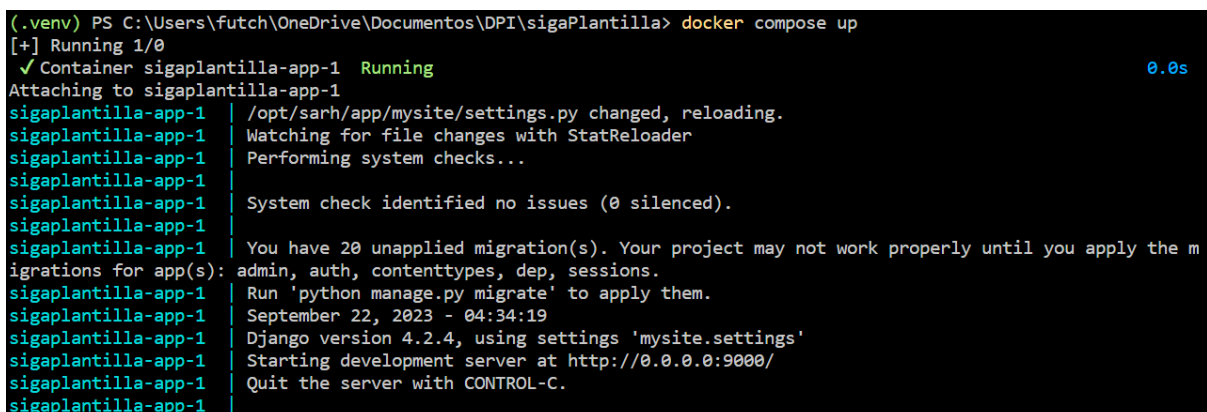
docker network connect siga_bridge sigaplantilla-app-1

```
PS C:\Users\futch> docker network connect siga_bridge sigaplantilla-app-1
PS C:\Users\futch> |
```

6.3 Corriendo a hora si el proyecto

y no deberíamos de tener problemas al agregarla a hora si volvemos a correr el proyecto con el comando:

docker compose up



**** nota

si cuando pones el comando anterior se queda de esta manera por un tiempo largo

```
(.venv) PS C:\Users\futch\OneDrive\Documentos\DPI\sigaPlantilla> docker compose up
[+] Running 1/0
✓ Container sigaplantilla-app-1 Running
Attaching to sigaplantilla-app-1
```

te vas al archivo de setting de la carpeta mysite y le quitas las comillas simple en el puerto de la concesión para la base de datos del auth_db

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'auth_db',
        'HOST': 'bases-datos',
        'PORT': 5432,
        'USER': 'adminsarh',
        'PASSWORD': 'adminsarhpassword.$',
    },
}
```

puede que se arregle, si no tiene las comas se las pones y si las tiene se las quitas para que se actualice el contenedor

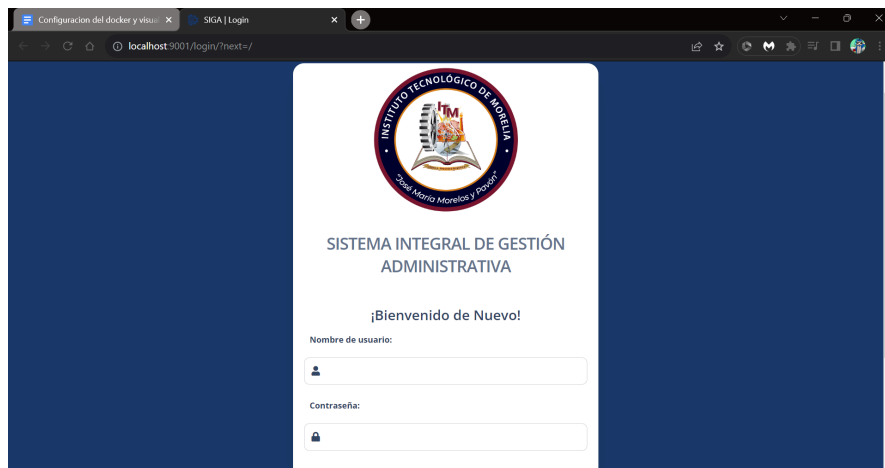
6.4 Ingresar al sistema web

para poder ver la página en el navegador de tu computadora es necesario ir al docker y desplegar el contenedor de siga plantilla y en este caso vamos a clicar la ip que nos da que es la siguiente en mi caso

9001:9000

<input type="checkbox"/>	<div><div></div><div></div><div></div></div> <div>sigaplantilla</div>	Running (1/1)	12 minutes ago	1.28%	<div><div></div><div></div><div></div></div>	
<input type="checkbox"/>	<div><div></div><div></div><div></div></div> <div>app-1</div> <div>d7d907d621</div> <div>sigaplantilla-app</div>	Running	9001:9000	12 minutes ago	1.28%	<div><div></div><div></div><div></div></div>

y nos abrirá la siguiente página



aún no tenemos acceso para ello vamos realizar las migraciones y crear un superusuario para poder ingresar chido, para esto vamos a necesitar abrir la consola de comando de la plantilla

6.4.1 Creación de migraciones

crear las migraciones usando el comando

python3 manage.py makemigrations

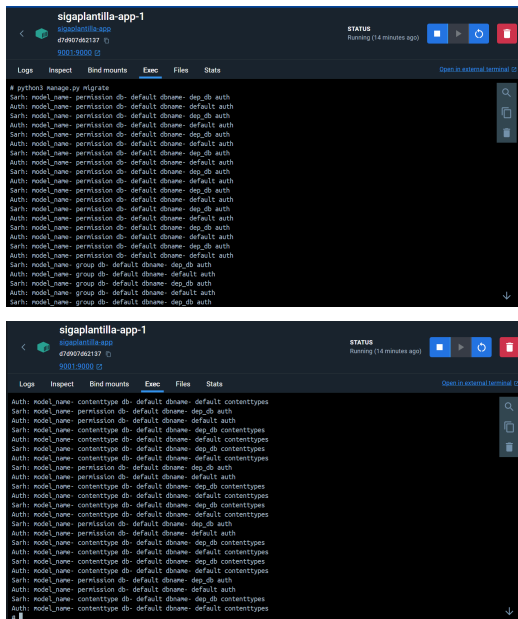
```
# python3 manage.py makemigrations
Sarh: model_name- Session db- default dbname- dep_db sessions
Auth: model_name- Session db- default dbname- default sessions
Sarh: model_name- Session db- dep_db dbname- dep_db sessions
Auth: model_name- Session db- dep_db dbname- default sessions
Sarh: model_name- ContentType db- dep_db dbname- dep_db contenttypes
Auth: model_name- ContentType db- dep_db dbname- default contenttypes
Sarh: model_name- LogEntry db- dep_db dbname- dep_db admin
Auth: model_name- LogEntry db- dep_db dbname- default admin
Sarh: model_name- Permission db- dep_db dbname- dep_db auth
Auth: model_name- Permission db- dep_db dbname- default auth
Sarh: model_name- Group db- dep_db dbname- dep_db auth
Auth: model_name- Group db- dep_db dbname- default auth
Sarh: model_name- User db- dep_db dbname- dep_db auth
Auth: model_name- User db- dep_db dbname- default auth
Sarh: model_name- Dep db- dep_db dbname- dep_db dep
/usr/local/lib/python3.8/site-packages/django/core/management/commands/makemigrations.py:158: RuntimeWarning: Got an error checking
a consistent migration history performed for database connection 'dep_db': connection to server at "bases-datos" (172.19.0.4), port
5432 failed: FATAL: password authentication failed for user "admindep"

  warnings.warn(
No changes detected
#
```

6.4.2 aplicar las migraciones

usamos el siguiente comando

python3 manage.py migrate



ya quedarían las migraciones

6.4.3 Creacion del superusuario

ingresamos el siguiente comando:

python3 manage.py createsuperuser

nos pedirá la siguiente info la que mas nos interesa es el usuario y el password

puedes poner el usuario y contrase;a que tu quiera pero pon uno facil de recordar en mi caso use admin como usuario y como password holaMundo

```
# python3 manage.py createsuperuser
Nombre de usuario (leave blank to use 'root'): admin
Dirección de correo electrónico:
Password:
Password (again):
Superuser created successfully.
#
```

Una vez creado el super usuario, con esta info ya la podemos poner en el login de la página web y



Notas:

Error al ejecutar scripts.

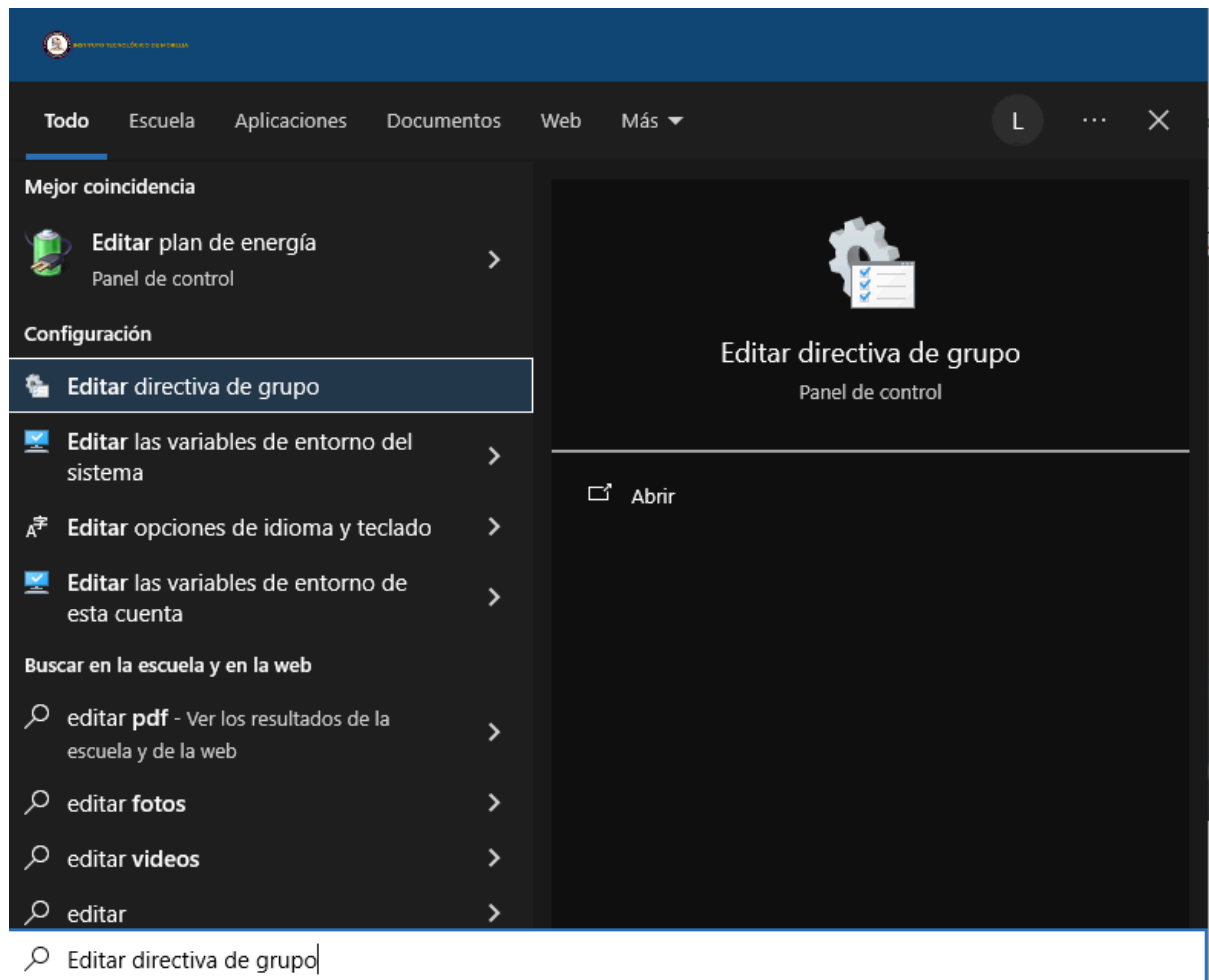
```
+ CategoryInfo          : ObjectNotFound: (get-executionpolicy)
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Windows\system32> get-executionpolicy -list

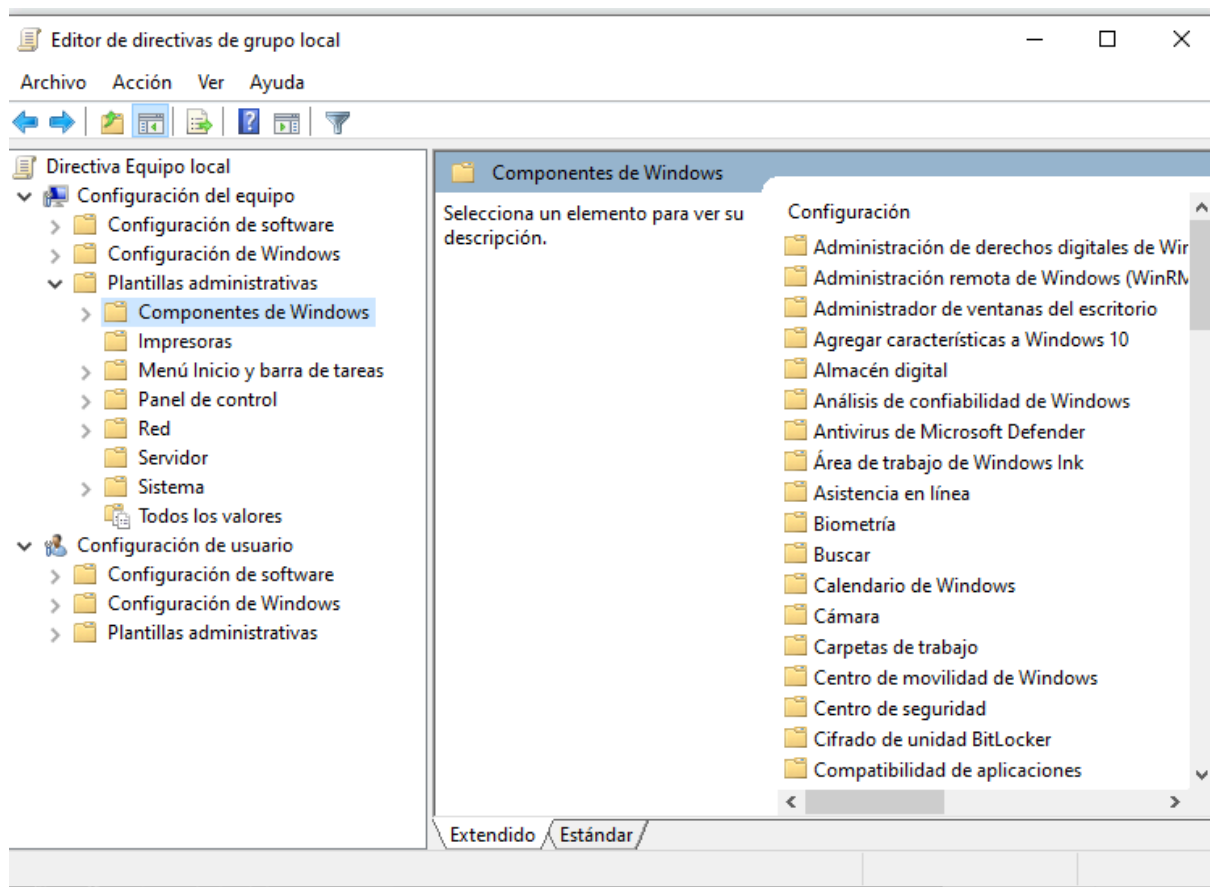
Scope ExecutionPolicy
-----
MachinePolicy AllSigned
UserPolicy    Undefined
Process       Undefined
CurrentUser   RemoteSigned
LocalMachine  Unrestricted
```

Cambiar la directiva del scope "MachinePolicy".

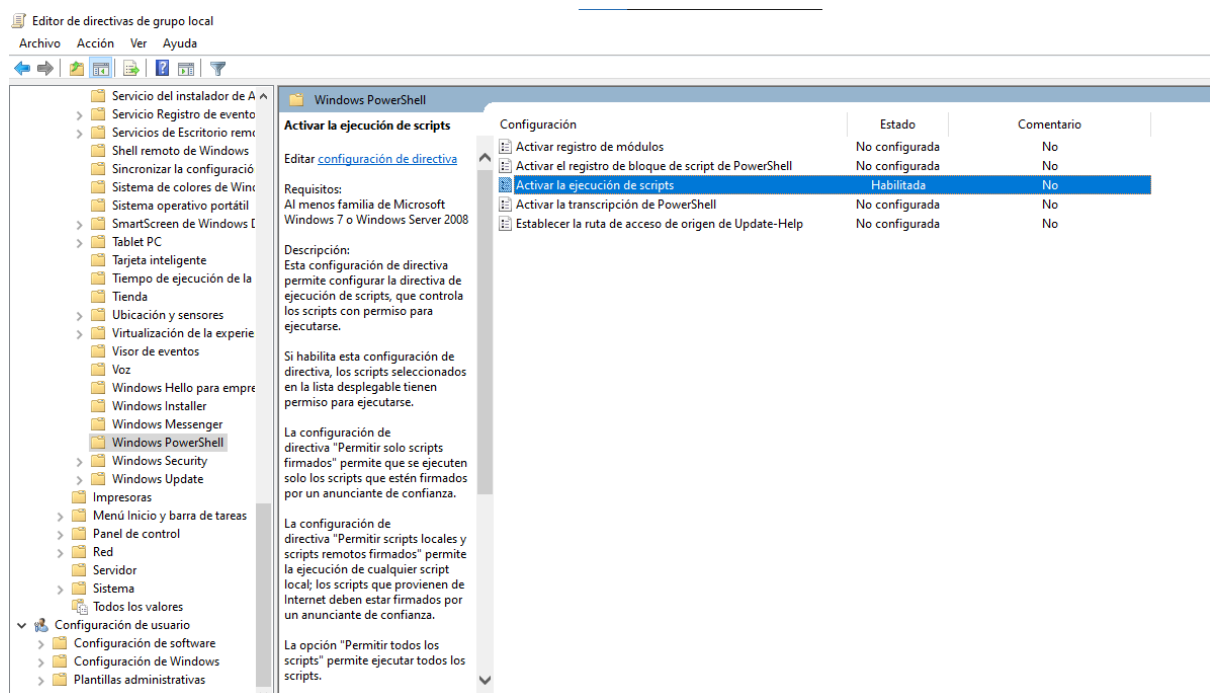
1. Editar directiva de grupo



2. Componentes de Windows > Windows powershell



3. Activar la ejecución de scripts



4. Modificar la directiva de ejecución

Activar la ejecución de scripts

Activar la ejecución de scripts

Valor anterior Valor siguiente

☐ No configurada Comentario:

☒ Habilitada

☐ Deshabilitada

Compatible con: Al menos familia de Microsoft Windows 7 o Windows Server 2008

Opciones:

Ayuda:

Directiva de ejecución

Permitir scripts locales y scripts remotos firmados

Permitir solo scripts firmados

Permitir scripts locales y scripts remotos firmados

Permitir todos los scripts

Esta configuración de directiva permite configurar la directiva de ejecución de scripts, que controla los scripts con permiso para ejecutarse.

Si habilita esta configuración de directiva, los scripts seleccionados en la lista desplegable tienen permiso para ejecutarse.

La configuración de directiva "Permitir solo scripts firmados" permite que se ejecuten solo los scripts que estén firmados por un anunciante de confianza.

La configuración de directiva "Permitir scripts locales y scripts remotos firmados" permite la ejecución de cualquier script local; los scripts que provienen de Internet deben estar firmados por un anunciante de confianza.

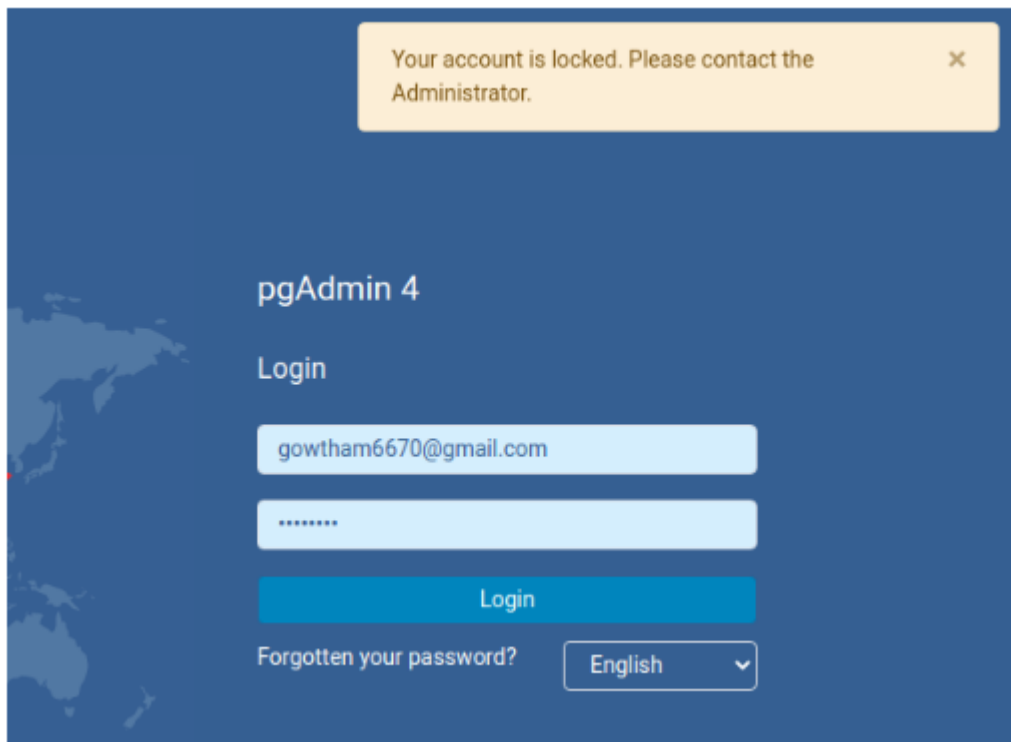
La opción "Permitir todos los scripts" permite ejecutar todos los scripts.

Si deshabilita esta configuración de directiva, no se permitirá la

Aceptar Cancelar Aplicar


Listo 😊

Your Account is locked.




Modificar esta linea en el archivo **config.py** de pgadmin4



<



pgadmin

[dpage/pgadmin4:latest](#)

6493d6e66428 

[32768:443](#)  [32769:80](#) 

STATUS

Running (2 minutes ago)

Logs
















Inspect

Bind mounts

Terminal

Files

Stats

Name 	Note	Size	Last
>  opt			2 minutes ago
▼  pgadmin4	MODIFIED		2 minutes ago
 config_distro.py	MODIFIED	398 Bytes	1 minute ago
 config.py	MODIFIED	34.1 kB	2 minutes ago
 DEPENDENCIES		89.2 kB	2 minutes ago
>  docs			2 minutes ago
 gunicorn_config.py		52 Bytes	2 minutes ago
 LICENSE		1.1 kB	2 minutes ago
>  migrations			2 minutes ago
>  pgacloud			2 minutes ago
>  pgadmin			2 minutes ago
 pgadmin.themes.json		143 Bytes	2 minutes ago
 pgAdmin4.py		8.2 kB	2 minutes ago
 pgAdmin4.wsgi		949 Bytes	2 minutes ago

MAX_LOGIN_ATTEMPTS = 5

