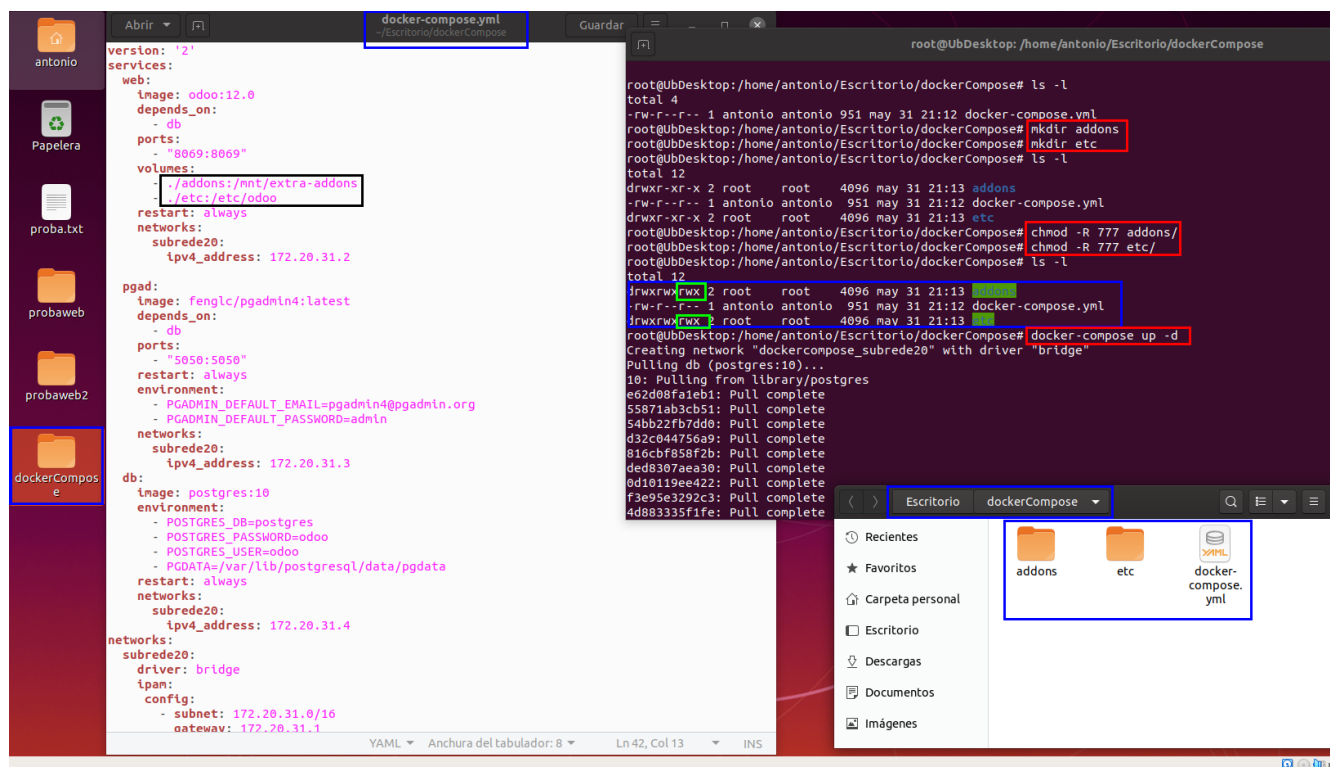


Instalacións

c) Exemplo Docker Compose

odoo + postgresql + pgadmin

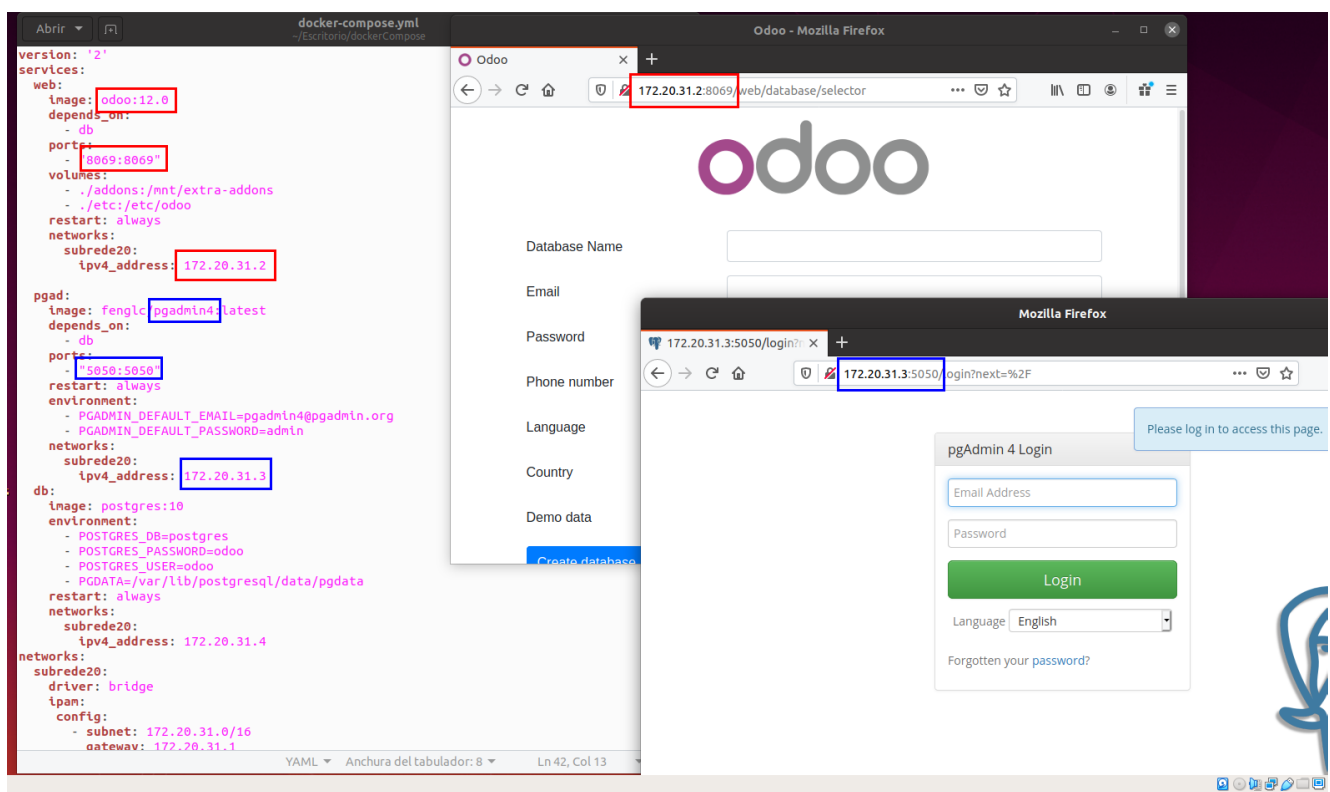
Imos executar un docker-compose que nos prepara un entorno de traballo con odoo, postgresql e pgadmin:



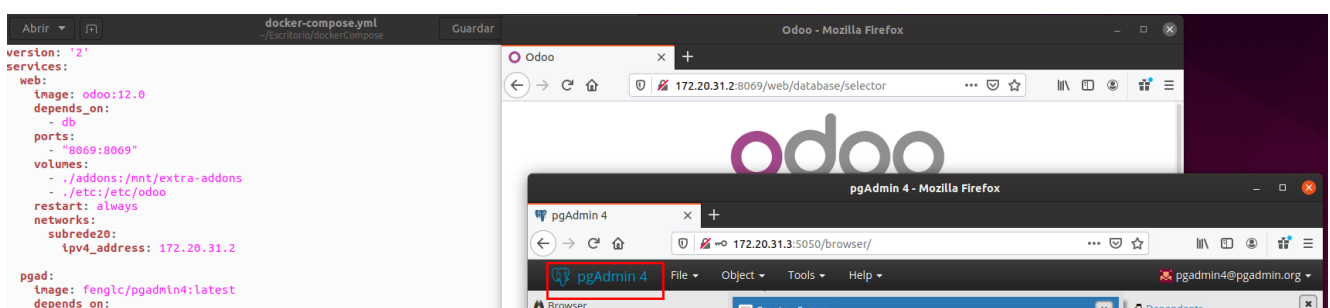
- Necesitamos no noso cartafol de traballo (dockerCompose):
 - O arquivo **docker-compose.yml**
 - Os cartafóis **addons** e **etc** que estamos mapeando na configuración de Odoo.
 - A estos cartafóis cambiamoslle os permisos para 777 como vemos na captura de pantalla
- Executamos no cartafol de traballo (dockerCompose) o comando **docker-compose up -d**
 - Có que se descargarán as imaxes de odoo (versión 12), de pgadmin4 (a última, xa que poñemos **latest**) e postgresql (versión 10)

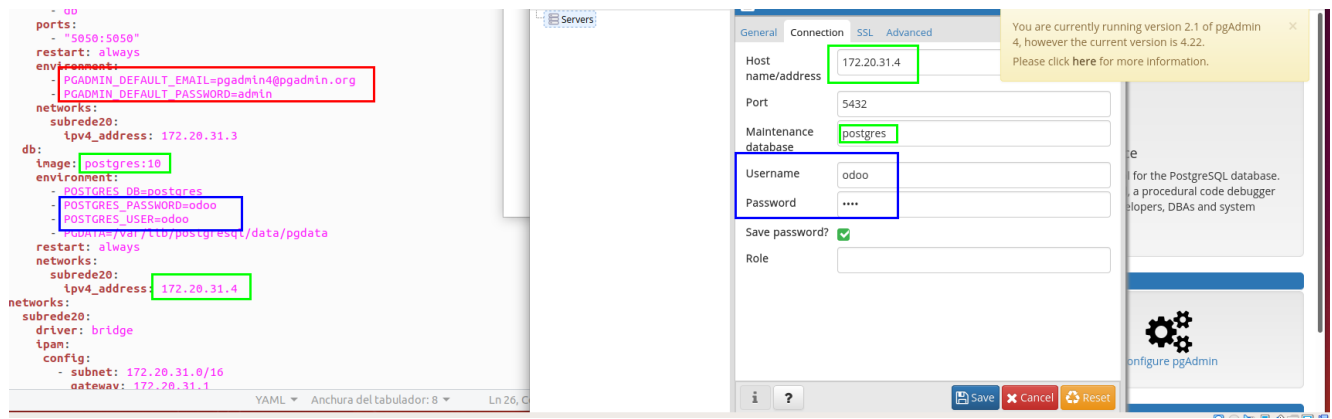
- Tamén vemos no arquivo docker-compose.yml que definimos unha rede interna 172.20.31.0/16 na que asignamos 3 IPs unha para cada servizo.
 - 172.20.31.2 para odoo
 - 172.20.31.3 para pgadmin
 - 172.20.31.4 para postgresql

Podemos ver os servizos funcionando nas rutas determinadas:



Iniciamos sesión no pgadmin:

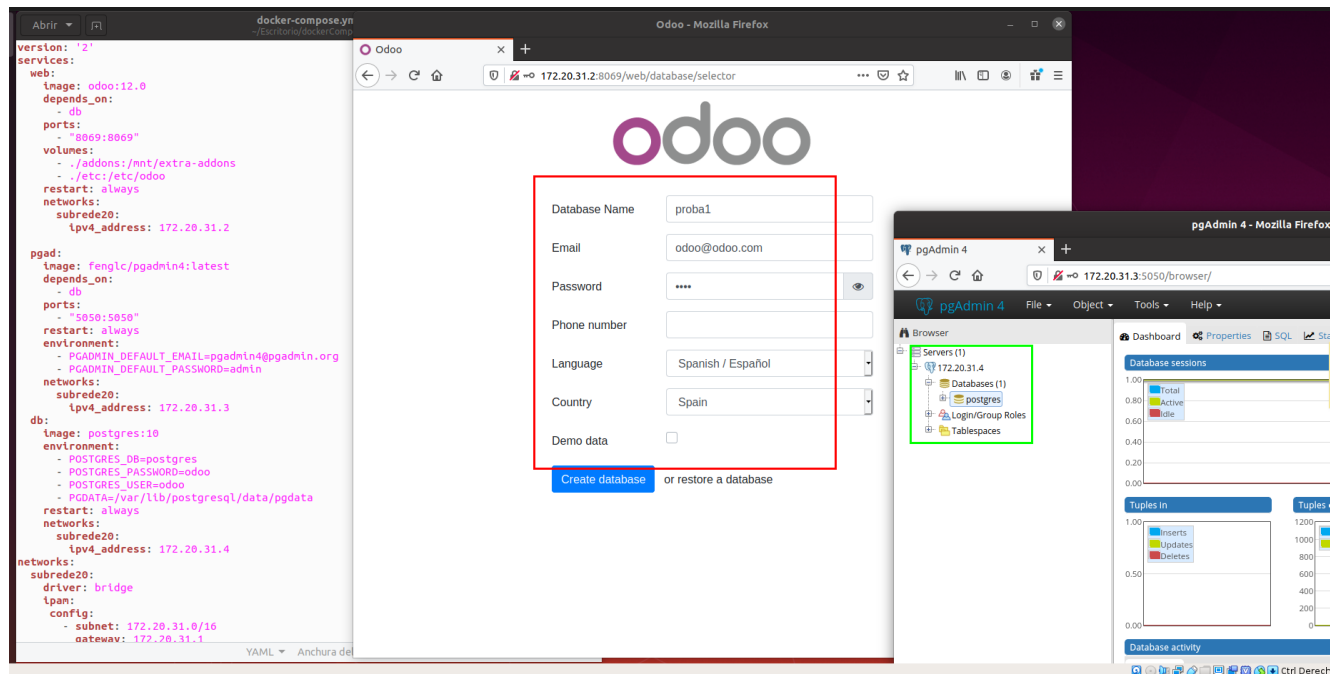




Iniciamos sesión en pgadmin e conectámonos a postgres.

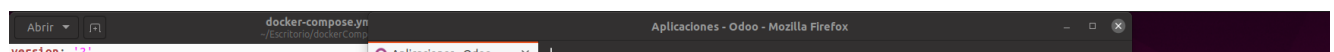
- Para elo usamos a información enmarcada en vermello.
- Despois creamos unha conexión en **Servers** para a máquina onde temos postgres (172.20.31.4)
- Usamos a información enmarcada en azul.

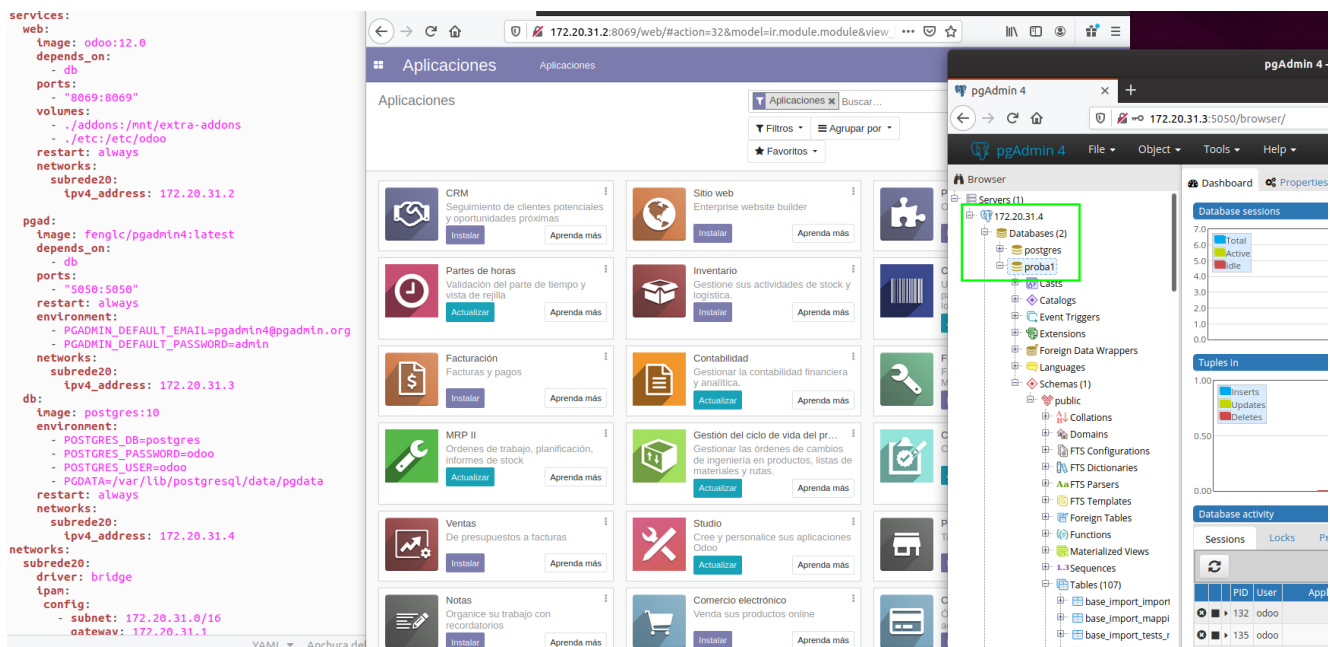
Creamos unha base de datos en Odoo:



Introducimos os valores para crear unha BD en Odoo. Podemos ver que polo de agora só existe unha BD (postgres)

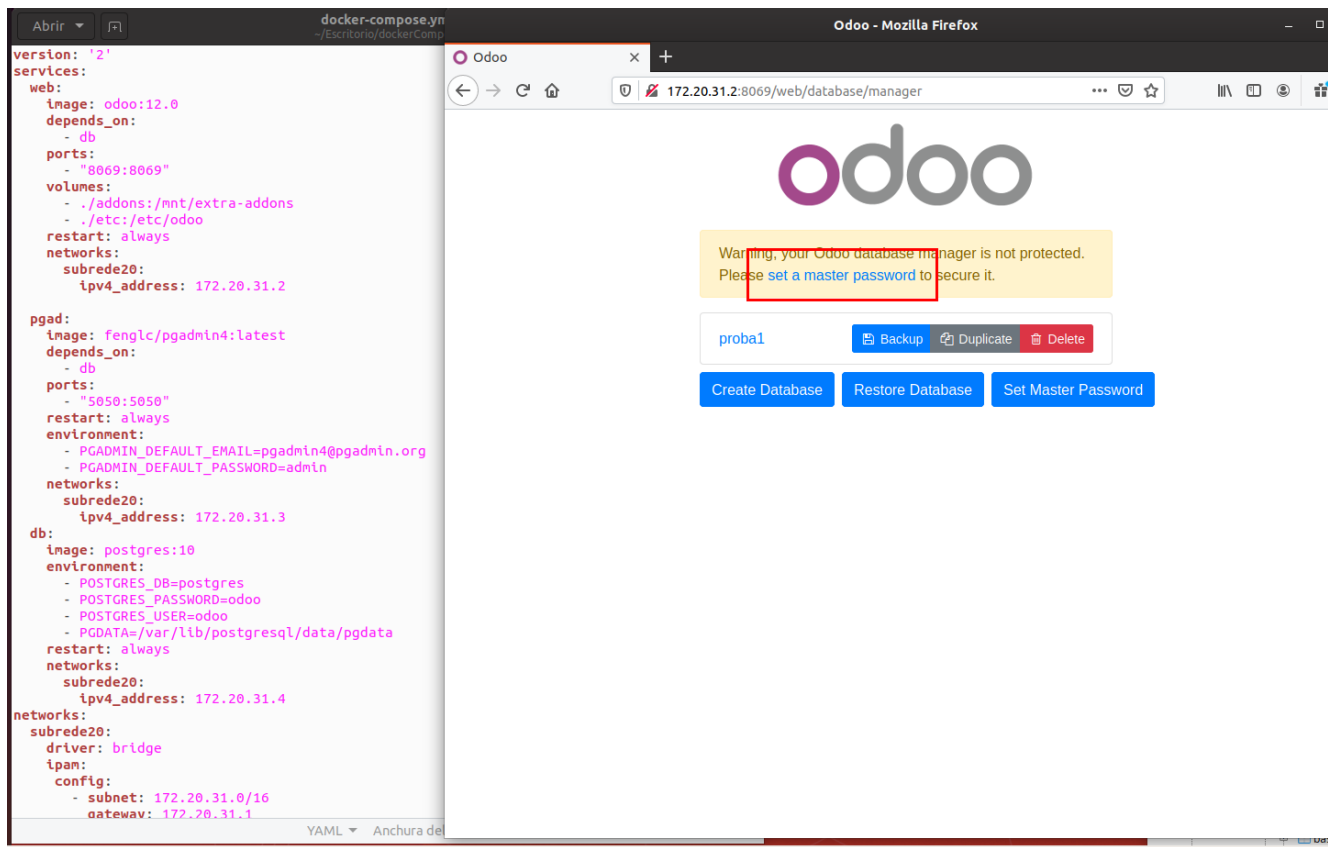
Unha vez creada:



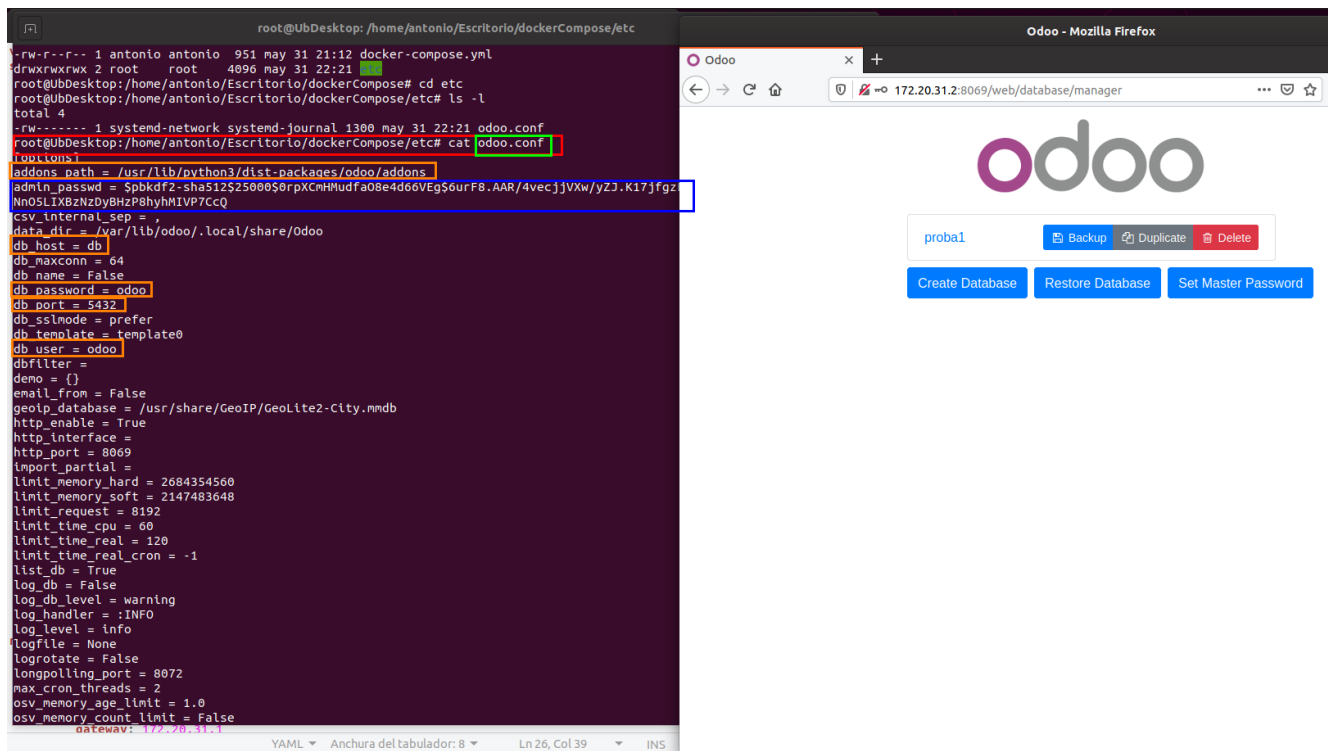


Podemos ver que xa temos 2 BD (postgres e proba1)

Configuramos a "master password" en Odoo



Unha vez que creamos a master password podemos ver que na nosa ruta etc creouse o arquivo odoo.conf onde entre outros está a password que vimos de configurar:

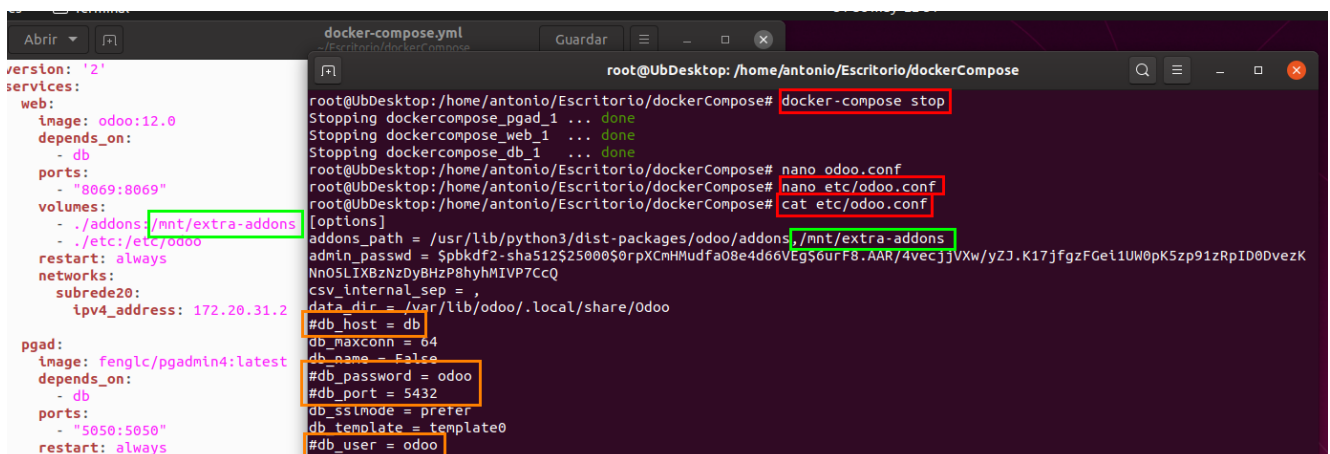


Podemos fixarnos nos campos enmarcados en marrón, xa que imos a modificalos no seguinte apartado.

Engadimos a ruta para os addons e deshabilitamos certos parámetros

Podemos engadir en **addons_path** a ruta para poñer os nosos módulos para que Odoo os atope cando queiramos instalalos.

E temos que comentar **db_host**, **db_password**, **db_port**, **db_user**. Se non cando reiniciemos os containers **NON Funcionará**.



```

environment:
  - PGADMIN_DEFAULT_EMAIL=pgadmin4@localhost
  - PGADMIN_DEFAULT_PASSWORD=admin
networks:
  subrede20:
    ipv4_address: 172.20.31.3
db:
  image: postgres:10
  environment:
    - POSTGRES_DB=postgres
    - POSTGRES_PASSWORD=odoo
    - POSTGRES_USER=odoo
    - PGDATA=/var/lib/postgresql/data
  restart: always
  networks:
    subrede20:
      ipv4_address: 172.20.31.4
networks:
  subrede20:
    driver: bridge
  ipam:
    config:
      - subnet: 172.20.31.0/16
        gateway: 172.20.31.1
demo: {}
email_from = False
geolite_database = /usr/share/GeoIP/GeoLite2-City.mmdb
http_enable = True
http_interface =
http_port = 8069
import_partial =
limit_memory_hard = 2684354560
limit_memory_soft = 2147483648
limit_request = 8192
limit_time_cpu = 60
limit_time_real = 120
limit_time_real_cron = -1
list_db = True
log_db = False
log_db_level = warning
log_handler = :INFO
log_level = info
logfile = None
logrotate = False
longpolling_port = 8072
max_cron_threads = 2
osv_memory_age_limit = 1.0

```

Para ello temos que:

1. Detener os servizos do "docker compose" mediante **docker-compose stop**
2. Editar o arquivo `odoo.conf` mediante **nano etc/odoo.conf** (xa que estamos no cartafol dockerCompose)
3. Visualizamos o modificado mediante **cat etc/odoo.conf**
 - Vemos en verde a ruta que engadimos para os nosos módulos, separada mediante unha coma (,) da ruta onde están os módulos de Odoo (`/usr/lib/python3/dist-packages/odoo/addons`)
 - Dentro do container será **/mnt/extra-addons**
 - Fora do container será **dockerCompose/addons**
 - E comentadas con # as liñas de **db_host**, **db_password**, **db_port**, **db_user**

Reiniciamos o compose e vemos que todo funciona

The screenshot shows a terminal window with the following commands and output:

```

root@UbDesktop: /home/antonio/Escritorio/dockerCompose# docker-compose start
Starting db ... done
Starting pgad ... done
Starting web ... done
root@UbDesktop: /home/antonio/Escritorio/dockerCompose# docker container ls

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
915bc017ddb4	fenglc/pgadmin4:latest	"docker-entrypoint.s..."	2 hours ago	Up 19 seconds	0.0.0.0:5050->5050/tcp	dockercompose_pgad_1
562034606d99	odoo:12.0	"/entrypoint.sh odoo"	2 hours ago	Up 19 seconds	0.0.0.0:8069->8069/tcp, 8071-8072/tcp	dockercompose_web_1
f1f0963dc566	postgres:10	"docker-entrypoint.s..."	2 hours ago	Up 22 seconds	5432/tcp	dockercompose_db_1
f21c2272bd92	nginx	"nginx -g 'daemon of..."	2 hours ago	Up 2 hours	0.0.0.0:2080->80/tcp	outro-nginx
262a06f7aa7	nginx	"nginx -g 'daemon of..."	3 hours ago	Up 3 hours	0.0.0.0:80->80/tcp	sone-nginx

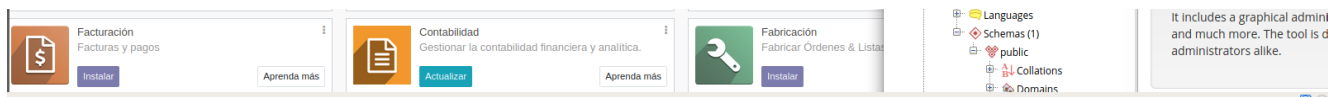
```

root@UbDesktop: /home/antonio/Escritorio/dockerCompose# docker system df

```

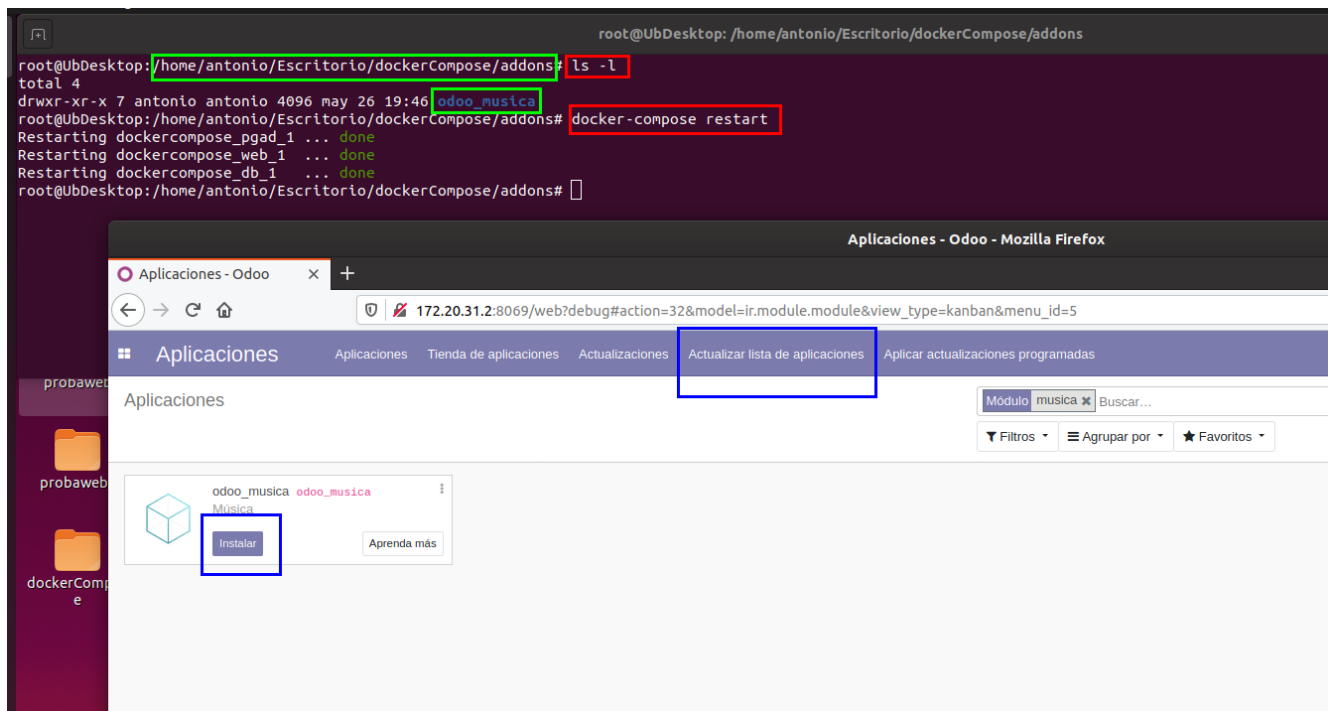
TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	5	5	2.272GB	55.32MB (2%)
Containers	6	5	238.1kB	0B (0%)
Local Volumes	3	3	86.67MB	0B (0%)
Build Cache	0	0	0B	0B

The web browser shows the Odoo application interface with the URL `172.20.31.2:8069/web#action=32&model=ir.module.module&view_type=kanban&menu_id=5`. The pgAdmin 4 interface is also visible, showing the database connection details.

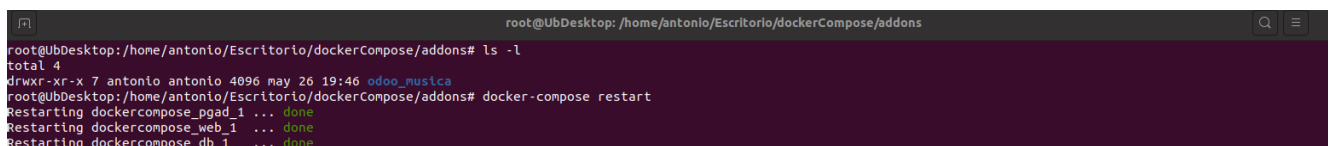


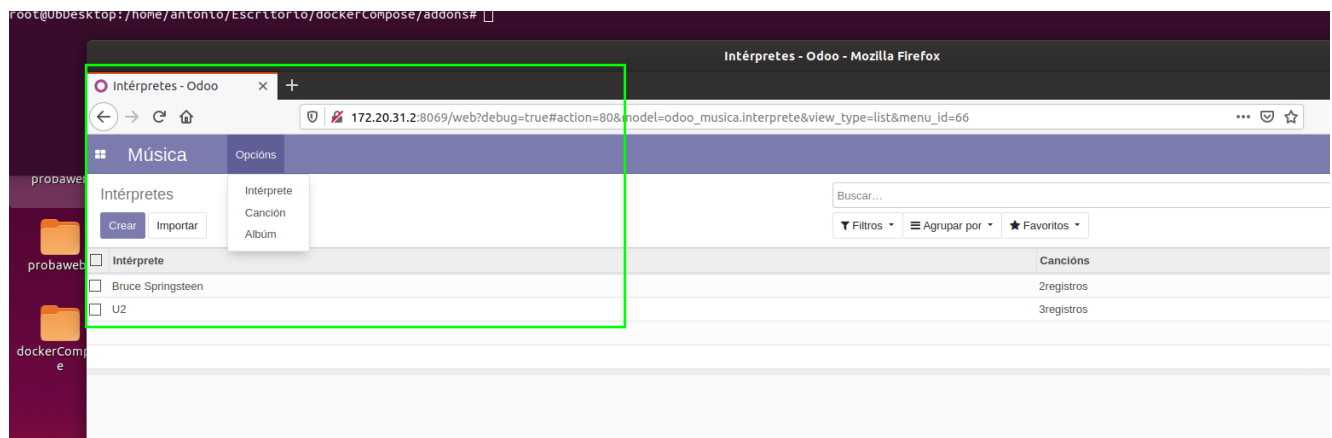
- Iniciamos de novo o compose mediante **docker-compose start**
- Vemos como de novo volven a funcionar os 3 servizos: odoo, postgres e pgadmin
- Mediante **docker container ls** vemos os 5 containers que temos funcionando, os 3 anteriores e os 2 nginx
- E mediante **docker system df** vemos o resumo do sistema.

Instalamos un módulo en Odoo



1. Copiamos o noso módulo (odoo_musica) na ruta dockerCompose/addons que tiñamos definida no docker-compose.yml
2. Reiniciamos o compose (odoo,postgres,pgadmin) con docker-compose restart
3. No menú de Ajustes de Odoo "Activar modo Desarrollador"
4. E despois no menú de Aplicaciones de Odoo clickamos en "Actualizar lista de aplicaciones" como se ve na captura de pantalla
5. E finalmente click en "Instalar"
6. E comprobamos que funciona:

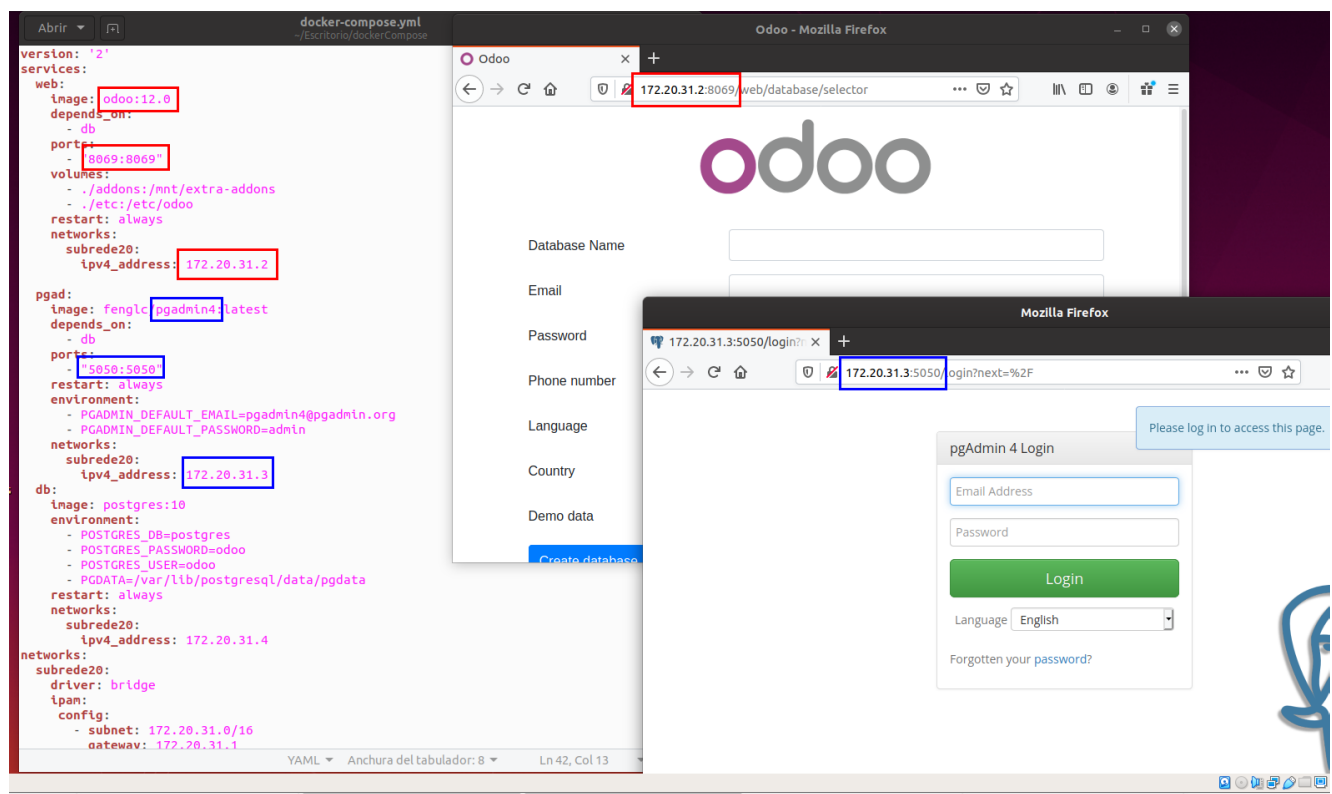




Mellora se queremos ter varios compose

Como vemos na configuración do "docker compose" definimos un redireccionamento de portos entre localhost e a ip do servizo docker:

- Neste caso 8069:8069 para **Odoo**, así podemos acceder mediante localhost:8069 ou 172.20.31.2:8069
- E 5050:5050 para **Pgadmin**, así podemos acceder mediante localhost:5050 ou 172.20.31.3:5050

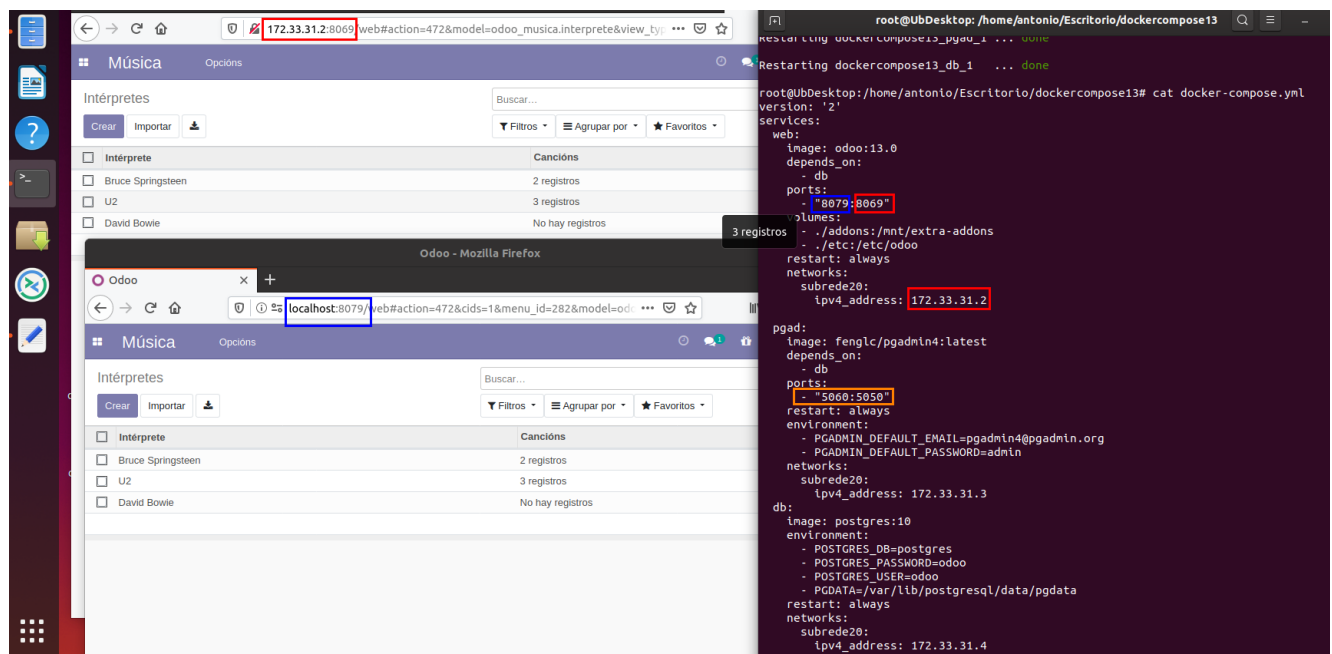


Se quixesemos facer outro docker compose con por exemplo outras versións (Odoo 13) poderíamos facelo usando outra rede por exemplo a 172.33.31.0/16

podemos facelo usando outra rede, por exemplo a 172.33.31.2/16

Pero teríamos que cambiar os portos que fan referencia a localhost, xa que 8069 e 5050 xa están en uso polo outro "docker compose":

- Neste caso 8079:8069 para **Odoo**, así podemos acceder mediante localhost:8079 ou 172.20.31.2:8069
- E 5060:5050 para **Pgadmin**, así podemos acceder mediante localhost:5060 ou 172.20.31.3:5050



Esta configuración obrigaríanos a estar determinando novos portos para localhost cada vez que engadisemos novos "docker compose".

Chegados a este punto a mellor solución é **non facer referencia aos portos** na definición do "docker compose" e acceder sempre mediante a **ip e o porto por defecto** de cada servizo (odoo 8069, pgadmin 5050) como no exemplo seguinte:

- Para acceder a Odoo sería 172.20.31.2:8069
- Para acceder a Pgadmin sería 172.20.31.3:5050

```
version: '2'
```

```
services:
```

```
  web:
```

```
    image: odoo:12.0
```

```
    depends_on:
```

```
      - db
```

```
    volumes:
```

```
      - ./addons:/mnt/extra-addons
```

```
- ./etc:/etc/odoo
restart: always
networks:
  subrede:
    ipv4_address: 172.20.31.2
pgad:
  image: fenglc/pgadmin4:latest
  depends_on:
    - db
  restart: always
  environment:
    - PGADMIN_DEFAULT_EMAIL=pgadmin4@pgadmin.org
    - PGADMIN_DEFAULT_PASSWORD=admin
  networks:
    subrede:
      ipv4_address: 172.20.31.3
db:
  image: postgres:10
  environment:
    - POSTGRES_DB=postgres
    - POSTGRES_PASSWORD=odoo
    - POSTGRES_USER=odoo
    - PGDATA=/var/lib/postgresql/data/pgdata
  restart: always
  networks:
    subrede:
      ipv4_address: 172.20.31.4
networks:
  subrede:
    driver: bridge
  ipam:
    config:
      - subnet: 172.20.31.0/16
        gateway: 172.20.31.1
```