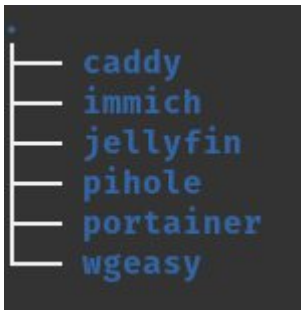
Portainer

1. Crear la estructura de directorios

- Vamos a ser ordenados. Crea una carpeta específica para este servicio para que los volúmenes y configuraciones no se mezclen.



2. Crear el archivo

- `nano docker-compose.yml`

```yaml
version: "3.8"

services:
  portainer:
    image: portainer/portainer-ce:latest
    container_name: portainer
    restart: always
    ports:
      - "9443:9443"
      - "9000:9000"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - portainer_data:/data

volumes:
  portainer_data:
```

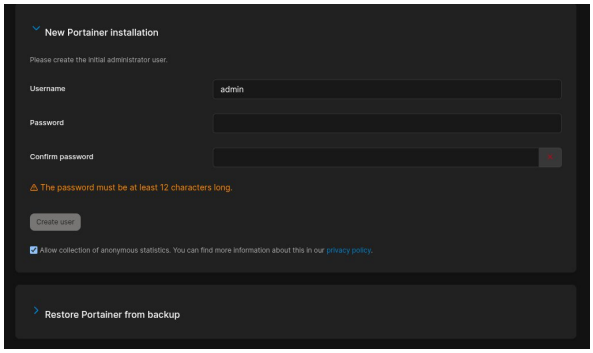
3. Levantar el servicio

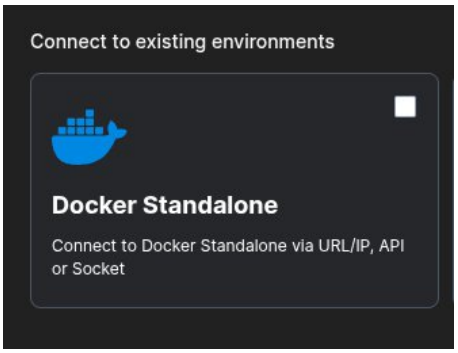
- Ahora, ejecuta el contenedor en segundo plano (detached):
  - `docker compose up -d`

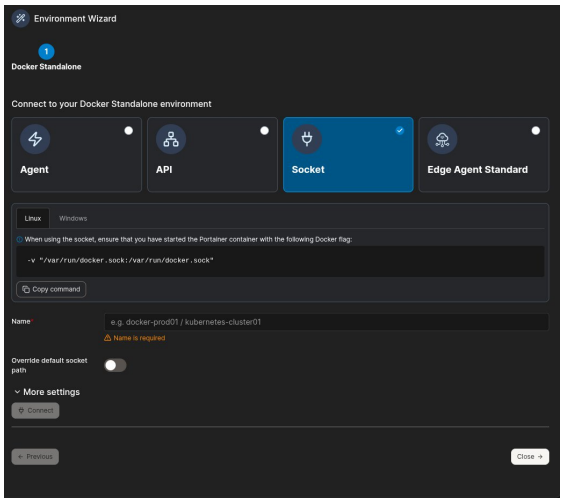
4. Verificación y Primer Acceso

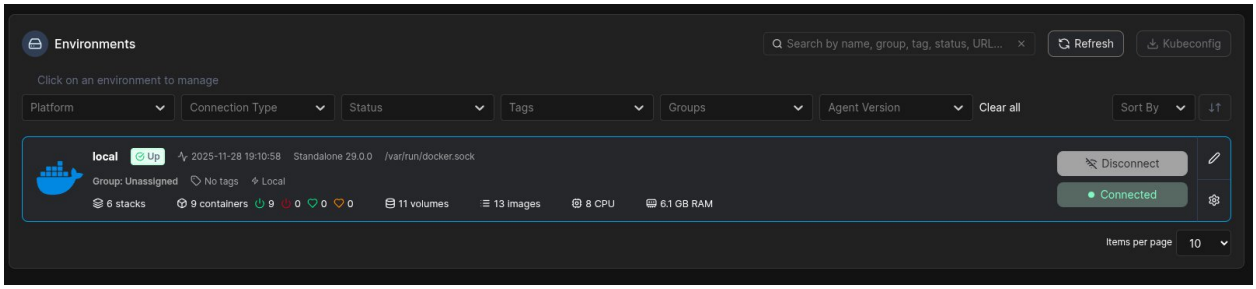
Portainer necesita que configures la contraseña de administrador **inmediatamente** después de encenderlo (por seguridad, si tardas mucho, se bloquea y tienes que reiniciar el contenedor).

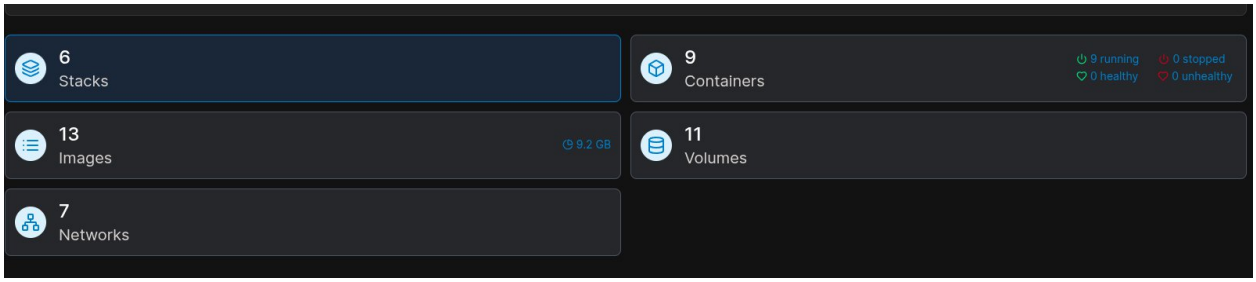
1. Abre tu navegador en tu PC.
2. Entra a: `http://192.168.1.110:9000`
3. `Tener en cuenta que si habeis hecho mi otro docker de caddy pueden a acceder por la url que habeis configurado.`
4. Crea tu usuario `admin` y ponle una contraseña segura.
5. Selecciona **"Get Started"** (entorno Local).

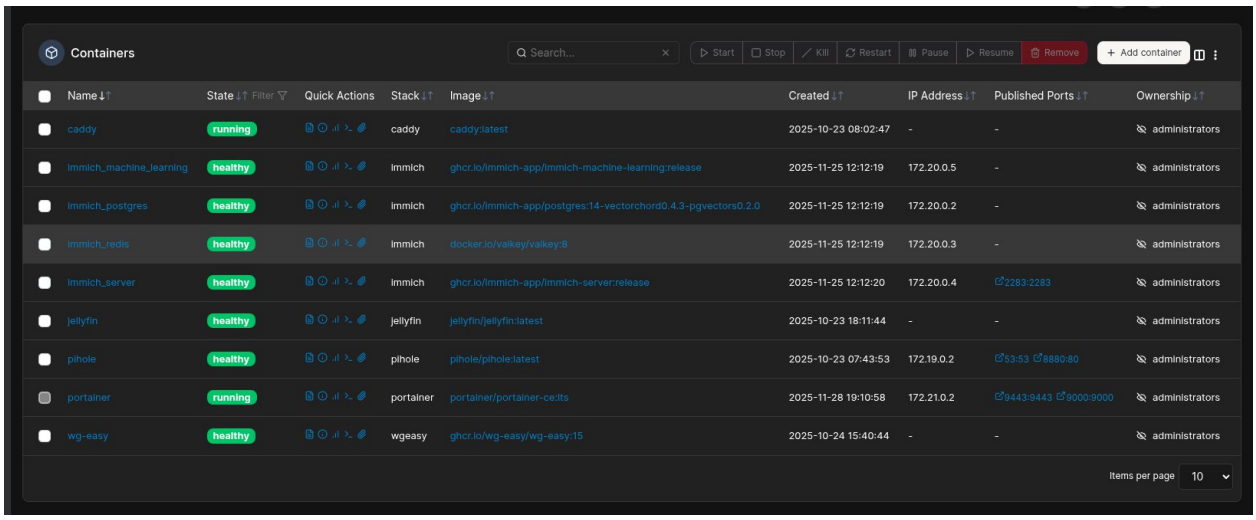

New Portainer installation
Please create the initial administrator user.
Username
admin
Password
Confirm password
The password must be at least 12 characters long.
Create user
Allow collection of anonymous statistics. You can find more information about this in our privacy policy.
Restore Portainer from backup


Connect to existing environments
Docker Standalone
Connect to Docker Standalone via URL/IP, API or Socket


Environment Wizard
1
Docker Standalone
Connect to your Docker Standalone environment
Agent
API
Socket
Edge Agent Standard
Linux
Windows
When using the socket, ensure that you have started the Portainer container with the following Docker flag:
-v "/var/run/docker.sock:/var/run/docker.sock"
Copy command
Name*
e.g. docker-prod01 / kubernetes-cluster01
Name is required
Override default socket path
More settings
Connect
Previous
Close


Environments
Search by name, group, tag, status, URL...
Refresh
Kubeconfig
Click on an environment to manage
Platform
Connection Type
Status
Tags
Groups
Agent Version
Clear all
Sort By
local
Up
2025-11-28 19:10:58
Standalone 29.0.0
/var/run/docker.sock
Group: Unassigned
No tags
Local
6 stacks
9 containers
11 volumes
13 images
8 CPU
6.1 GB RAM
Disconnect
Connected
Items per page
10


6
Stacks
9
Containers
9 running
0 stopped
0 healthy
0 unhealthy
13
Images
9.2 GB
11
Volumes
7
Networks


Containers
Search...
Start
Stop
Kill
Restart
Pause
Resume
Remove
Add container
Name
State
Filter
Quick Actions
Stack
Image
Created
IP Address
Published Ports
Ownership
caddy running caddy caddy:latest 2025-10-23 08:02:47 - - administrators
immich_machine_learning healthy immich ghcr.io/immich-app/immich-machine-learning:release 2025-11-25 12:12:19 172.20.0.5 - administrators
immich_postgres healthy immich ghcr.io/immich-app/postgres:14-vectorchord0.4.3-pgvectors0.2.0 2025-11-25 12:12:19 172.20.0.2 - administrators
immich_redis healthy immich docker.io/valkey/valkey:8 2025-11-25 12:12:19 172.20.0.3 - administrators
immich_server healthy immich ghcr.io/immich-app/immich-server:release 2025-11-25 12:12:20 172.20.0.4 2283:2283 administrators
jellyfin healthy jellyfin jellyfin/jellyfin:latest 2025-10-23 18:11:44 - - administrators
pihole healthy pihole pihole/pihole:latest 2025-10-23 07:43:53 172.19.0.2 53:53 8880:80 administrators
portainer running portainer portainer/portainer-ce:lts 2025-11-28 19:10:58 172.21.0.2 9443:9443 9000:9000 administrators
wg-easy healthy wgeasy ghcr.io/wg-easy/wg-easy:15 2025-10-24 15:40:44 - - administrators
Items per page 10