



Portainer

Alumnos:

Troche Pablo

Nicolas Tancredi

Laboratorio de Sistemas Operativos y Redes.





Índice:

1. Introducción del proyecto.....	3
a. Descripción del proyecto.....	3
b. Funcionalidades.....	4
2. Especificaciones técnicas.....	4
a. Relevamiento Hardware.....	5
b. Relevamiento de Software.....	6
3. Instructivo de instalación.....	7
a. Instalación en Código Binario.....	7
b. Instalación de en Imagen Docker.....	7
c. Instalación con Docker Compose.....	8
4. Configuración Portainer y conexiones en diferentes entornos.....	10
a. Local.....	11
b. Remoto.....	11
c. Agent.....	12
5. Comparativa de uso.....	13
6. Limitaciones encontradas	15
7. Complementos.....	15
8. Bibliografía.....	16



1. Introducción del proyecto

a. Descripción del proyecto

En nuestro proyecto vamos a trabajar en la implementación de la herramienta Portainer, que nos permite, bajo una interfaz gráfica y su interacción con el usuario, gestionar la plataforma de contenedores Docker para que así el trabajo de gestar, construir y mantener toda esa administración sea más sencilla e intuitiva. La herramienta reduce la utilización de líneas de comando ahorrando tiempo para la implementación de un entorno Docker.

A lo largo de este documento se desarrollará el proceso elaborado por nuestro grupo para la puesta en marcha, configuración y su funcionalidad de Portainer. También se detallará las características y relevamientos del contexto donde se trabajó esta implementación.

b. Funcionalidades

Una de las funciones fundamentales de la herramienta Portainer es permitir organizar containers, images, networks, services, volumes, network y la mayoría de las particularidades que nos ofrece la plataforma Docker, a través de una aplicación web que persiste los datos ingresados por el usuario, para algunos o cada uno de estos organizadores mencionados, que posteriormente procesa y genera instrucciones para la ejecución de líneas de comando Docker, según la configuración establecida por el usuario. Toda esta información se resume en un tablero que tiene la funcionalidad de ilustrar y representar la gestión de los containers, images, networks, services y stacks que se lleva en un nodo



particular. Cada una de estas gestiones tiene acciones específicas para poder manipular información o realizar acciones según su comportamiento.

También, otra funcionalidad, es la capacidad de poder tener un acceso remoto de toda esa gestión mencionada. Es decir, tener acceso al aplicativo web desde cualquier ordenador mediante una correcta configuración, sea desde la misma red privada o pública, del servicio web, como, así también, la posibilidad de gestionar la plataforma Docker de un ordenador remoto.

2. Especificaciones técnicas

La aplicación web, mencionada en el anterior apartado, es puesta en marcha a través de un sistema operativo Linux basado en debian, desplegando los servicios HTTP para implementar el código binario proporcionado por la fundación Portainer.io. Portainer CE proporciona un software de núcleo abierto bajo la licencia zlib, gratuito y su código fuente está disponible en GitHub. Este software se descarga, descomprime y se ejecuta bajo líneas de comandos. El acceso al aplicativo se realiza mediante un navegador web, especificando el puerto asignado a dicho servicio web especificado en la configuración del código binario.



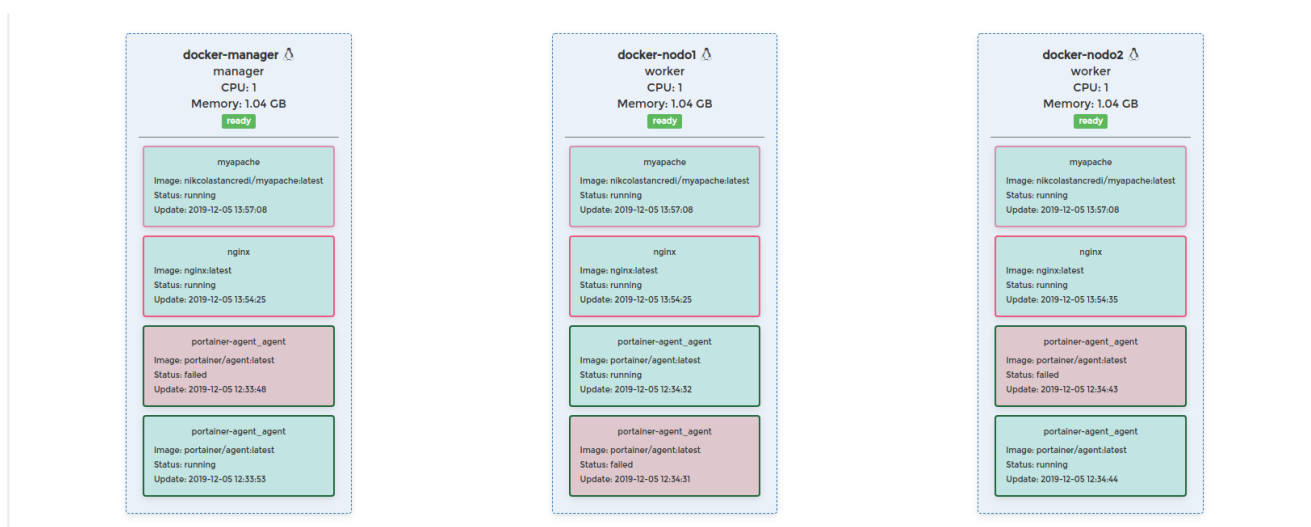
a. Relevamiento Hardware

(Relevamiento de hardware local)

- ✓ CPU: Intel core I5 7ta 5200u
- ✓ Memoria RAM: 8 GB DDR4
- ✓ Disco: 62 GB
- ✓ Tarjeta Gráfica: Intel graphics 620

(Relevamiento de la infraestructura)

- ✓ 2 VM de Docker Machine como Nodo
- ✓ 1 VM de Docker Machine como Manager





b. Relevamiento Software

- ✓ Linux Mint 19.2 Tina KDE PLASMA
- ✓ Portainer 1.23.0
- ✓ Docker Engine Community version 19.03.5, build 633a0ea838
- ✓ Docker CLI
- ✓ Docker Deamon
- ✓ Docker Engine
- ✓ Docker Compose Versión 1.25.0, build 0a186604
- ✓ Docker Machine Versión 0.16.0, build 702c267f
- ✓ Virtual Box Versión 5.2.34_Ubuntu r133883



3 . Instructivo de Instalación

La fundación Portainer ofrece diversos modos de instalación de su producto, en la siguiente sección enumeraremos cada uno de ellos

a. Instalación en Código Binario

(Instalación de portainer deployando un repositorio git)

Descargar el repositorio github de portainer:

Descargue y extraiga el binario a una ubicación en el disco:

```
$ cd /opt
```

```
$ wget
```

```
https://github.com/portainer/portainer/releases/download/1.22.2/portainer-1.22.2-linux-amd64.tar.gz
```

```
$ tar xvpzf portainer-1.22.2-linux-amd64.tar.gz
```

Crear el directorio /data (si ya no existe). Portainer crea intentará escribir de forma predeterminada en ese directorio.

```
$ mkdir /data
```

```
$ cd /opt/portainer
```

```
./portainer --template-file "${PWD}/templates.json"
```

(Se puede agregar al comando anterior la siguiente combinación para cambiar el puerto) `./portainer -p :8080`

(También se puede agregar la siguiente combinación para cambiar el directorio donde escribe) `./portainer --data /opt/portainer-data`

b. Instalación de en Imagen Docker



Es modo de instalación crea un contenedor en base a la imagen de portainer y se lanza su ejecución, es la forma más sencilla de instalación.

```
$ docker volume create portainer_data
```

```
ptroche@ptroche-VirtualBox:~$ sudo docker volume create portainer_data
[sudo] password for ptroche:
portainer_data
```

```
$ docker run -d -p 8000:8000 -p 9000:9000 -v
```

```
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data
```

```
portainer/portainer
```

```
ptroche@ptroche-VirtualBox:~$ sudo docker run -d -p 8000:8000 -p 9000:9000
-v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data port
ainer/portainer
ada02d3d8dde06ec87c80d9aaa34566ad3da5c5153fb07329429360539d779e1
```

c. Instalación via Docker Compose

- Esta instalación consta de armar un archivo Docker compose con extensión yamel para correr luego ejecutarse con los comandos Docker compose.
 - Contenido docker-compose.yml

```
version: '2'
```

```
services:
```

```
  portainer:
```

```
    image: portainer/portainer
```

```
    command: -H unix:///var/run/docker.sock
```

```
    restart: always
```

```
  ports:
```

```
    - 9000:9000
```

```
    - 8000:8000
```




volumes:

- /var/run/docker.sock:/var/run/docker.sock

- portainer_data:/data

volumes:

portainer_data

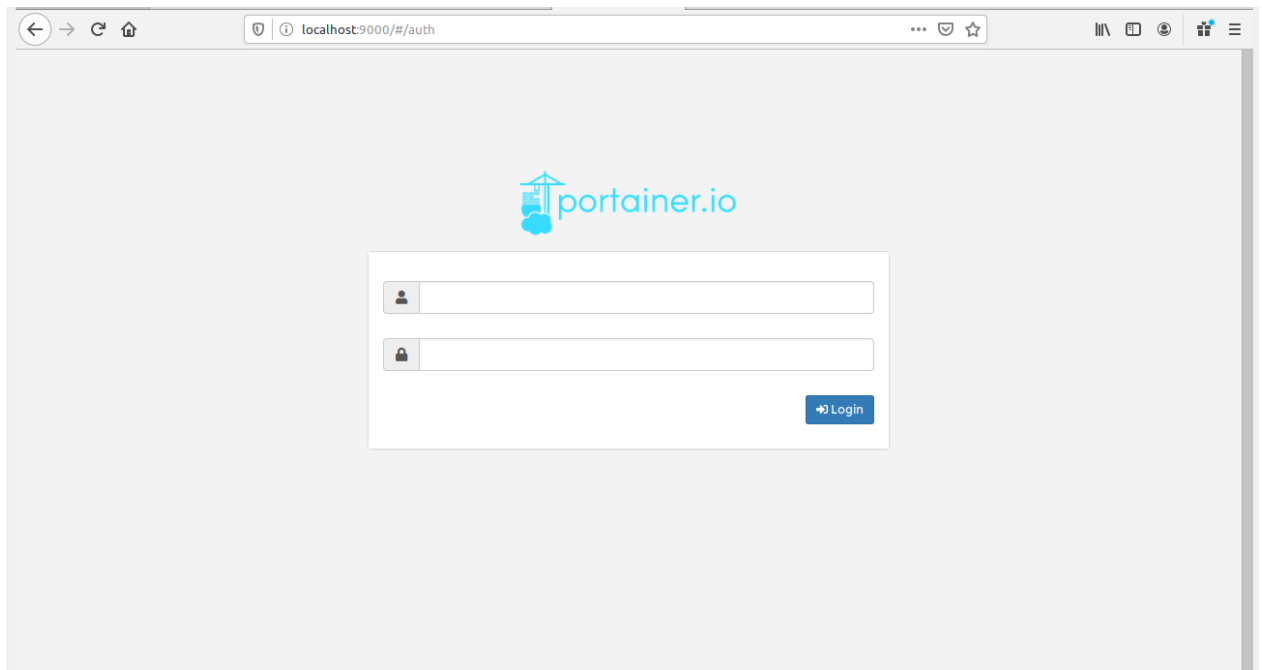
- CLI Docker compose

Docker-compose up (en la ubicación del archivo Docker-compose)

4. Configuración Portainer y conexiones en diferentes entornos.

4.1 Configuración de usuario admin y arranque.

Entrar a la página con el puerto seleccionado. (<http://localhost:9000>)



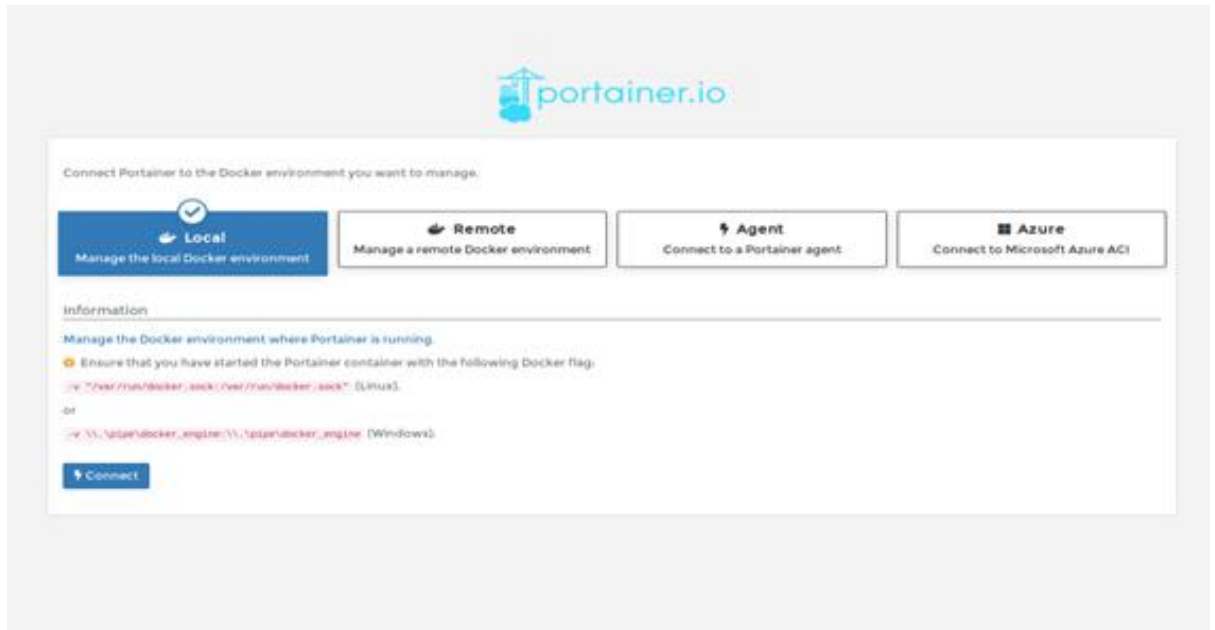
- Solicitará la creación de la contraseña admin.



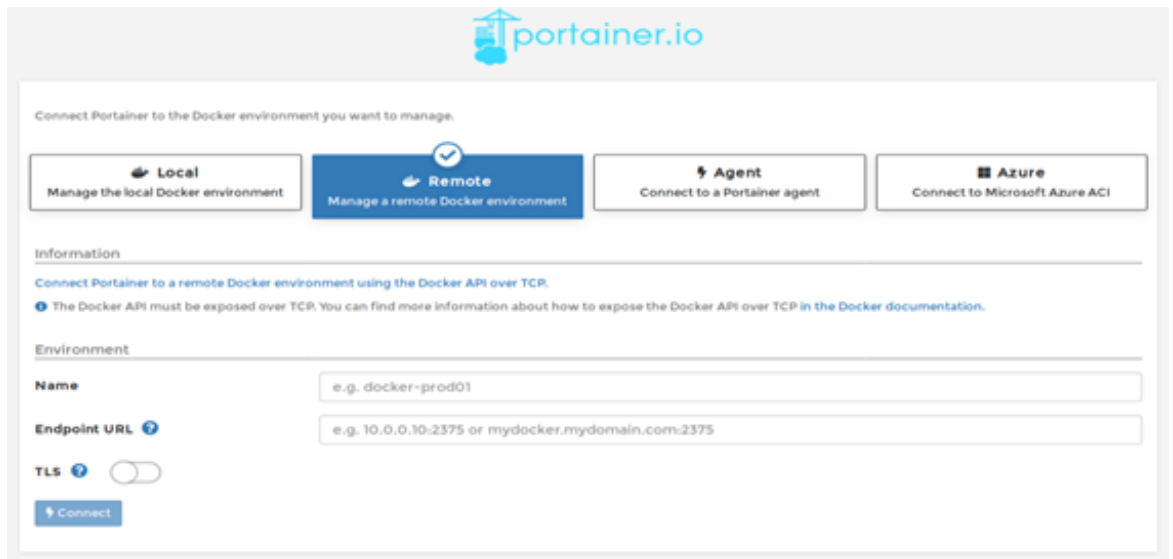
4.2 Conexión a entornos Docker

Portainer nos ofrece un abanico de tipos de conexión para la gestión de Docker.

- Local: La más sencilla y común. Está manipula el socket de Docker en la maquina local donde se instaló la aplicación.



- Remote: la manipulación del socket en un dominio no conocido fue de la propia red.



- Agent: Permite la manipulación de un socket docker dentro de una red conocida



The screenshot shows the Portainer.io web interface. At the top, there is a navigation bar with the Portainer.io logo. Below it, a heading reads "Connect Portainer to the Docker environment you want to manage." There are four buttons: "Local" (Manage the local Docker environment), "Remote" (Manage a remote Docker environment), "Agent" (Connect to a Portainer agent), and "Azure" (Connect to Microsoft Azure ACI). The "Agent" button is selected and highlighted in blue. Below the buttons, there is an "Information" section with a note: "Connect directly to a Portainer agent running inside a Swarm cluster." and a tip: "If you have started Portainer in the same overlay network as the agent, you can use `TASKS_AGENT_SERVICE_NAME/AGENT_SERVICE_PORT` as the endpoint URL format." The "Environment" section contains two input fields: "Name" with the example "e.g. docker-prod01" and "Agent URL" with the example "e.g. 10.0.0.10:9001 or tasks.portainer_agent:9001". A "Connect" button is located at the bottom left of the form.

Pasos para conectar portainer agent:

- Requisitos: se debe tener un cluster swarm para poder desplegar un stack de servicio en el mismo, este stack deploy lo proporciona Portainer.
 - a.
 - b. Descargar y ejecutar stack:
`curl -L https://downloads.portainer.io/agent-stack.yml -o agent-stack.yml && docker stack deploy --compose-file=agent-stack.yml portainer-agent`
 - c.
- Luego, se registra el cluster y se utiliza el puerto 9001 que expuso la ejecución de stack



Environment type

Agent
Portainer agent

Endpoint

Information

Ensure that you have deployed the Portainer agent in your environment. Run the following command in your terminal:

```
curl -L https://downloads.portainer.io/agent-stack.yml
```

[Copy](#)

Environment details

Name cluster

Endpoint URL 192.168.99.1:9001

Public IP 192.168.99

Metadata

Group Unassigned

Tags No tags available.

Actions

[+ Add endpoint](#)

- Dashboard cluster

Cluster overview [Refresh](#) Portainer support [admin](#)

Swarm

Cluster status

Nodes	3
Docker API version	1.40
Total CPU	3
Total memory	3.11 GB

[Go to cluster visualizer](#)

Nodes [Settings](#)

Search...

Name	Role	CPU	Memory	Engine	IP Address	Status	Availability
docker-manager	manager	1	1 GB	19.03.5	192.168.99.100	ready	active
docker-nodo1	worker	1	1 GB	19.03.5	192.168.99.101	ready	active
docker-nodo2	worker	1	1 GB	19.03.5	192.168.99.102	ready	active

Items per page: 10

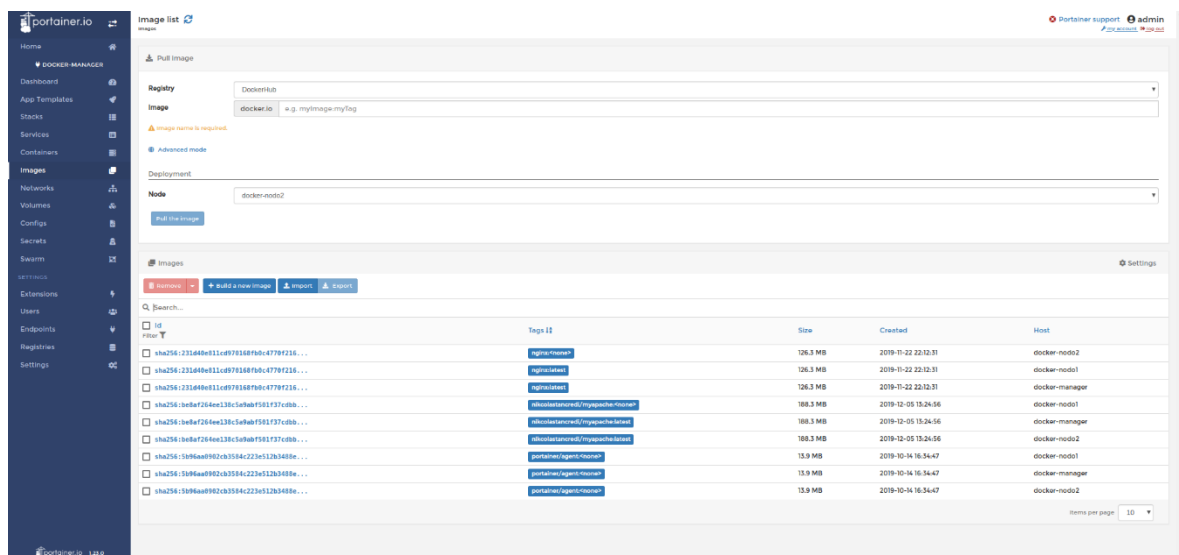
5. Comparativa de uso (contra Docker CLI).



Describimos las pestañas más significativas de Portainer y las comparamos con la consola de comando Docker CLI de más está decir que es muy poderosa teniendo comandos específicos para cada tarea. Pero Portainer nos ofrece una interfaz más amigable y mucha más adaptable a trabajar en múltiples entornos de Docker.

a. Portainer Images:

Listar, buildear, exportar, importar, remover, pull imágenes de docker desde una registry.



Para poder responder todo lo que ofrece portainer en la pestaña Imagen tengo que tener comando como:

```
$ sudo docker images -a
```

```
docker@docker-manager:~$ docker images -a
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
nikolastancredi/myapache  latest      be8af264ee13  2 hours ago  188MB
nginx                latest      231d40e811cd  12 days ago  126MB
portainer/agent      <none>      5b96aa0902cb  7 weeks ago  13.9MB
```

```
$ sudo docker pull "repositorio/imagen:tag"
```

```
$ docker rmi <id_image1> <id_image2> <id_image 3>
```



b. Portainer Container:

Iniciar, parar, matar, reiniciar, pausar, remover contenedores. Obtener información de ellos, nombres, estado, tomar acciones, ver si corren en stack, desde que imagen fue generada, fecha de creación, etc.

The screenshot shows the Portainer web interface. At the top, there are navigation buttons: Start, Stop, Kill, Restart, Pause, Resume, Remove, and Add container. Below is a search bar and a table of containers. The table has columns for Name, State, Quick actions, Stack, Image, Created, IP Address, Host, Published Ports, and Ownership. There are four containers listed, all in a 'running' state. Below the table is a terminal window showing the command 'sudo docker ps -a' and its output, which lists container details like ID, Image, Command, Created, Status, Ports, and Names. At the bottom right of the interface, there is a pagination control showing 'Items per page' set to 10 and page numbers 1 and 2.

6. Limitaciones encontradas.

- a. Docker compose: nos topamos con la limitación de tener proyectos en donde docker-compose resuelve la integración de varios servicios dockerizados y con Portainer no podemos correr el comando \$docker-compose up o \$docker-compose build. No pudimos deployar un docker-compose.yml y encontrar el contenedor integrado de las soluciones.
- b. No pudimos realizar alias de las redes creadas para poder asociarlas a un nombre



7. Complemento.

Estos complementos son características de Docker necesarias para poder trabajar y crear un escenario con la herramienta de portainer.

- **Docker Compose**

Se instalo para poder realizar una de las instalaciones de Portainer

- **Docker Stack:**

- Viene por defecto instalado en Docker Engine y nos permitió realizar la instalacion del agente en los equipos de docker machine remotos.

- **Docker Machine**

- Se hizo uso de esta característica en la creación y manipulación de máquinas virtuales con la finalidad de crear un cluster y posteriormente enlazar sus nodos

- **Docker Swarm**

- Esta característica nos brinda la posibilidad de poder enlazar máquinas virtuales que formando un cluster, de modo que pueda portainer pueda desplegar sus funcionalidades de swarm sobre el cluster.

- **Docker Hub (registry)**

- Se creo una imagen en el repositorio para poder hacer uso de ella desde portainer.

8. Bibliografía.

✓ Instalación de Docker:

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

✓ Instalación de Portainer

<https://portainer.readthedocs.io/en/stable/deployment.html>

✓ Instalación de Docker Compose



<https://docs.docker.com/compose/install/>

- ✓ Instalación de Docker Machine

<https://docs.docker.com/machine/install-machine/>