

Documentación: Ocore y Oggui

Escenario

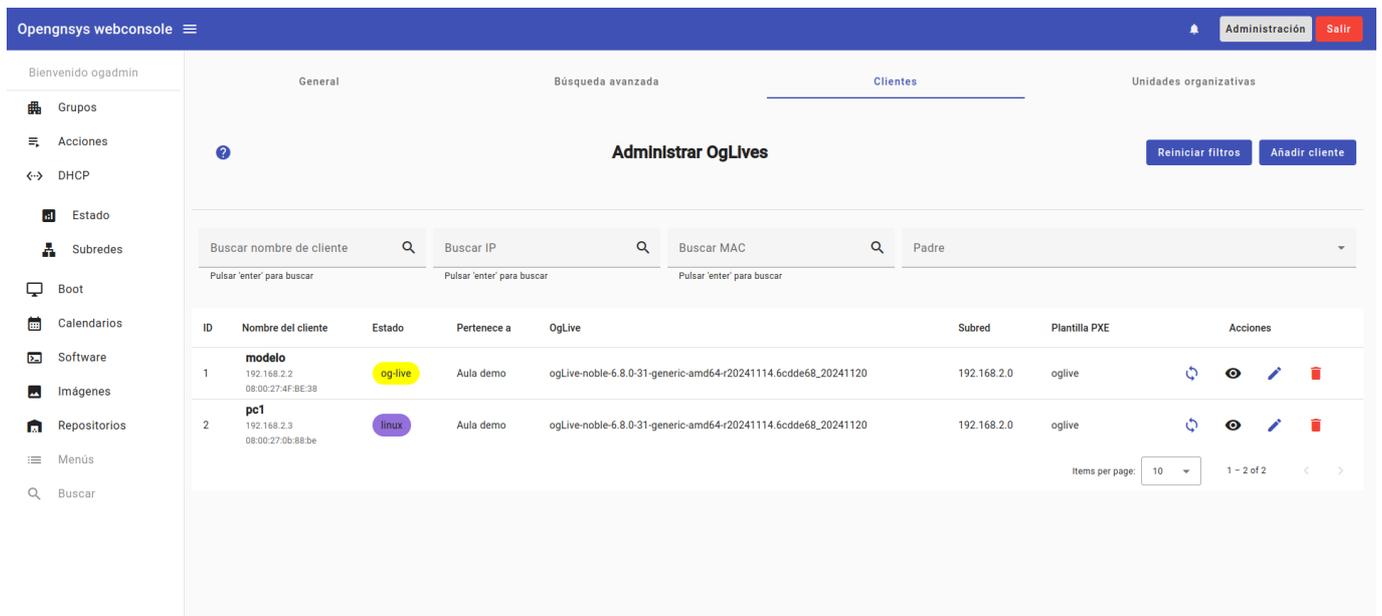
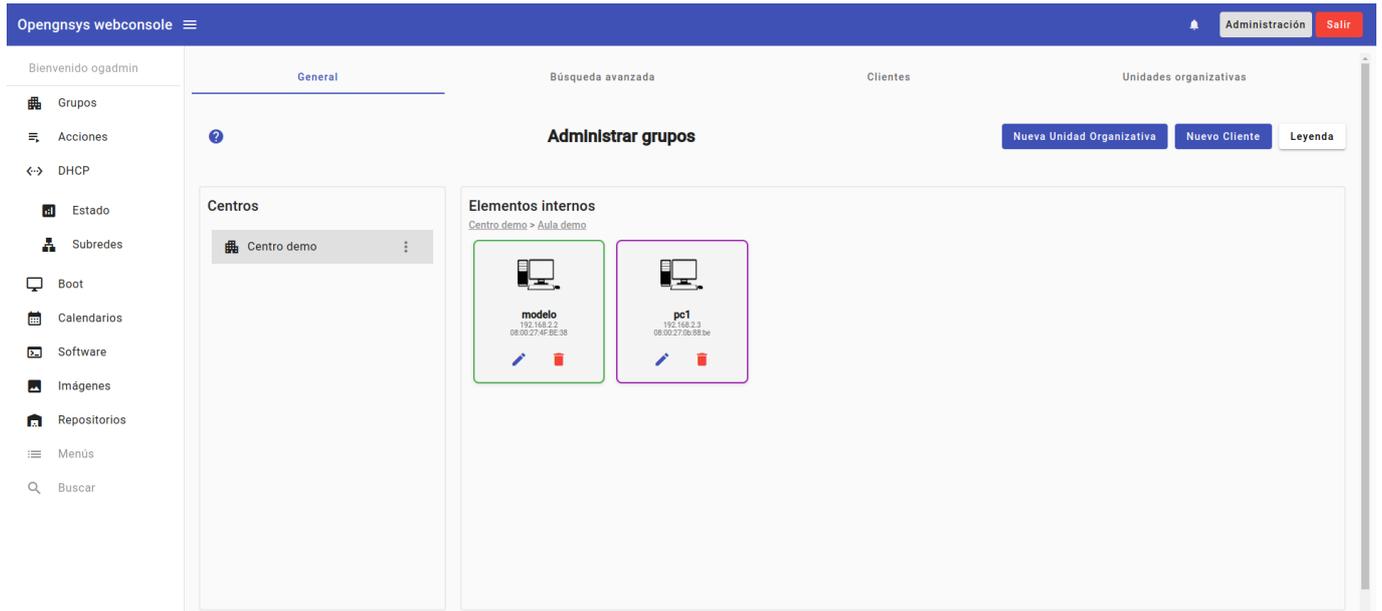
Surge la necesidad de crear una imagen a través de un ordenador al que llamaremos modelo. Para ello necesitamos tener un ordenador creado en nuestra base de datos con el que tengamos conexión y con un sistema operativo.

Este es el escenario que se han utilizado para esta demostración:

- **Centro:**
 - Nombre: `Centro Demo`
- **Aula:**
 - Nombre: `Aula Demo`
 - Router IP: `192.168.2.1`
 - Máscara de red: `255.255.255.0`
- **Ordenador:**
 - MAC Address: `08:00:27:4F:BE:38`
 - Nombre: `modelo`
 - Dirección IP: `192.168.2.2`
- **Subred:**
 - Dirección: `192.168.2.0`
 - Máscara de red: `255.255.255.0`
- **Imagen:**
 - Nombre: `image-test`
 - Descripción: `imagen de referencia`
 - Tipo: `Linux`
 - Versión: `Ubuntu 24.04`
 - Arquitectura: `x86_64`
- **Repositorio:**
 - Ruta: `/opt/opengnsys/ogrepository/images`
 - IP: `192.168.68.55`
- **ogAgent:**
 - versión: `1.4.5~pre7-1 all`

El objetivo es crear una imagen a partir del ordenador `modelo` con las características especificadas, para más adelante replicar/clonar en el resto de clientes.

En las siguientes capturas de pantalla, se muestra la disposición de los grupos y clientes en la interfaz grafica de ogGui.



La herramienta ogGui/ogCore nos proporciona una interfaz grafica mediante la cual podremos crear una imagen. En la versión actual, lo haremos a traves de la pestaña "Clientes" dentro del panel de administración de grupos.

Crear una imagen

1. **Seleccionar cliente modelo.** Mediante la pestaña de **Clientes** en el panel de administración de grupos, accedemos a las opciones del cliente pulsando en el boton "Visualizar" (el que tiene como icono un ojo). Esto nos da acceso al panel del cliente donde aparece su informacion general y el estado de las particiones. Arriba a la derecha, aparece un boton con todas las acciones disponibles.

Opengsys webconsole

Bienvenido ogadmin

Administración Salir

Grupos

Acciones

DHCP

Boot

Calendarios

Software

Imágenes

Repositorios

Menús

Buscar

Datos de cliente

Editar Comandos

Nombre	modelo	IP	192.168.1.10
MAC	08:00:27:0e:65:ff	Nº de serie	
Netiface	eth0	Perfil hardware	
Pxe	aaaa	Remote Pc	
Subred	192.168.1.100	OGlive	
Autoexec		Repositorio	Test

Discos/Particiones

Disco	Particion	Sistema de ficheros	Tamaño	Uso	SO
1	0		100.00 GB	0 %	undefined
1	1	EXT4	100.00 GB	3 %	Debian GNU/Linux 12 (bookworm)

Legend

Partició...

Administración Salir

Discos/Particiones

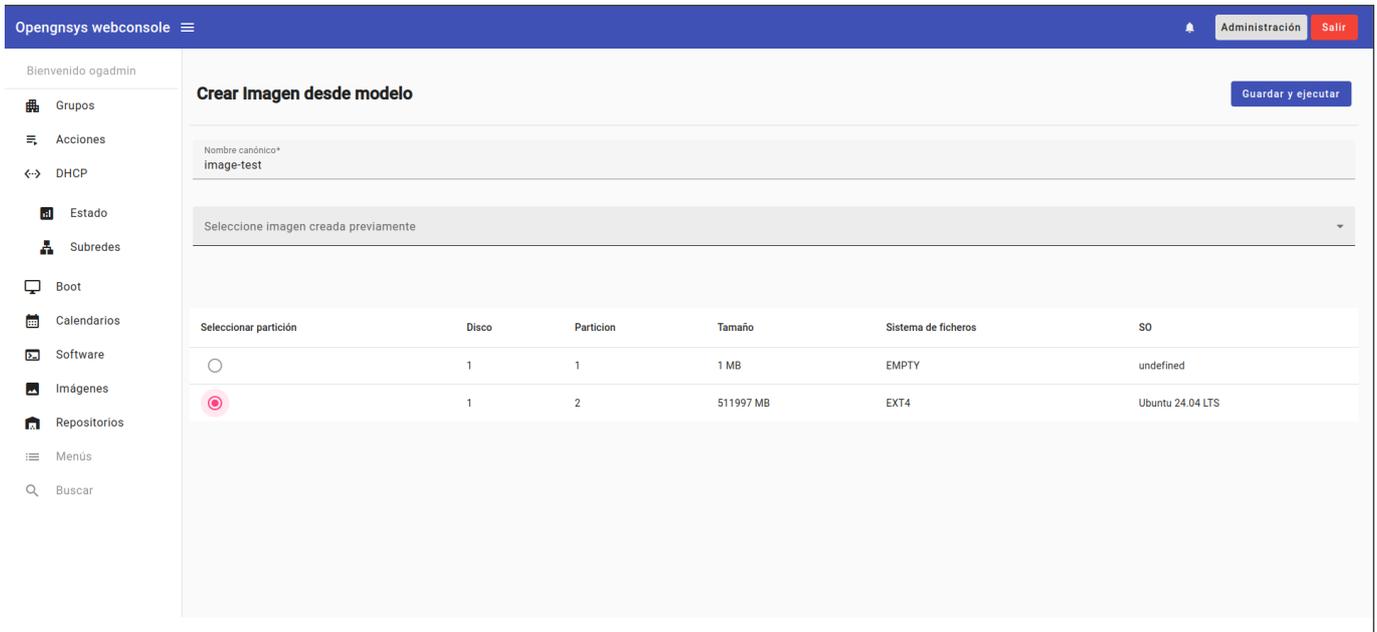
Editar Comandos

- Enceder
- Apagar
- Reiniciar
- Iniciar Sesión
- Crear Image
- Deploy Image
- Eliminar Imagen Cache
- Particionar y Formatear
- Inventario Software
- Inventario Hardware
- Ejecutar script

modelo	Uuid	0193684e-2ae4
192.168.2.2	MAC	
	Netiface	
	Menú	
	Subred	
ve-noble-6.8.0-31-generic-amd64-r20241114.6cdde68_20241120	Autoexec	
Repo 1	Validación	
oglive	Creado por	

2. **Crear imagen.** Pulsamos en el botón "Crear imagen" y rellenamos el formulario con los datos de la imagen que queremos crear. En este caso, los datos son los siguientes:

- **Nombre canónico:** `image-test`
- **Imagen creada previamente:** en este caso, lo dejamos en blanco.
- **Seleccionar partición:** Seleccionamos la particion la cual tiene instalada Ubuntu 24.04.



Si todo ha ido bien, la interfaz gráfica nos devuelve un mensaje de confirmación verde, en la esquina inferior derecha. Esto quiere decir que el ogAgent ha recibido correctamente la llamada mediante API y está trabajando en la creación de la imagen.

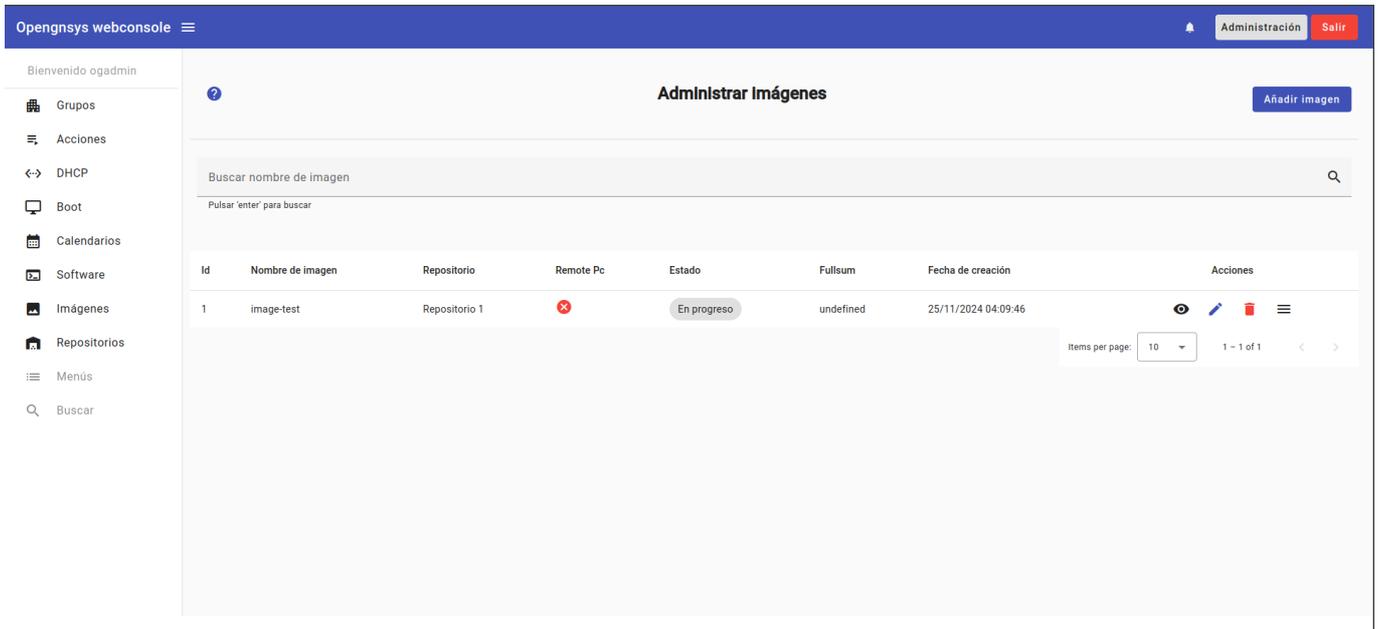
Comprobar la creación de la imagen

En el panel de administración de imágenes, podremos ver el estado de la imagen que hemos creado.

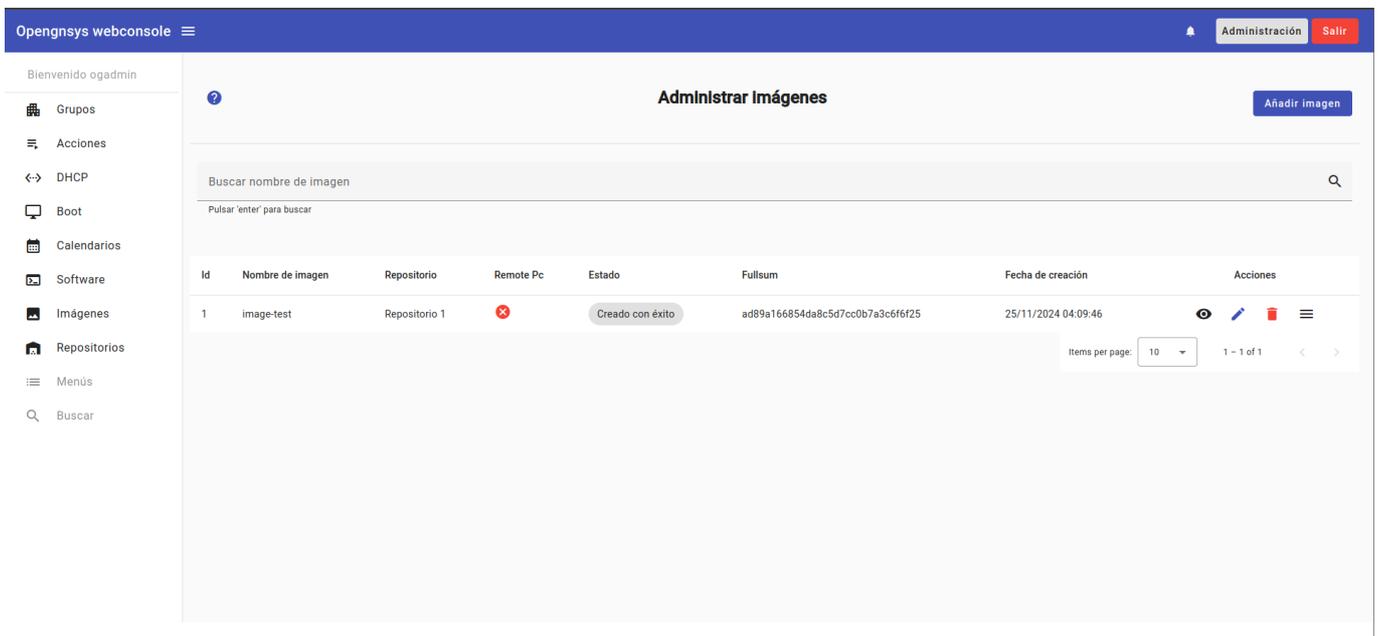
La herramienta almacena las imágenes mediante diferentes estados, los cuales son:

- **Estados:**

- En progreso: **in-progress**. La imagen está en proceso de creación, el ogAgent está trabajando en ello.
- Pendiente de ficheros auxiliares: **aux-files-pending**. La imagen ha sido creada en el repositorio, pero aún no se han generado los archivos auxiliares, ni devuelvo el fullsum (a partir de ahora será el identificador único asociado)
- Completada: **success**. La imagen ha sido creada correctamente. Se encuentra en el repositorio de imágenes, y con sus archivos asociados generados satisfactoriamente.
- Error: **failed**. La imagen no ha podido ser creada, por algún error en el proceso.



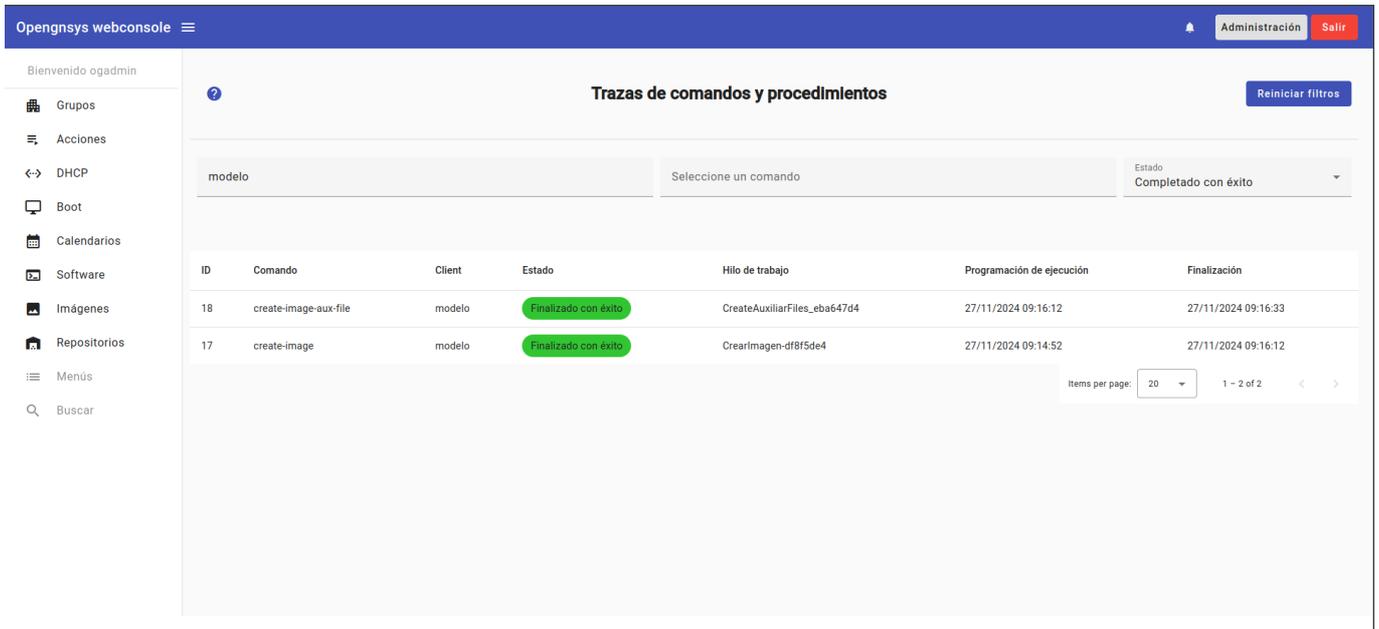
Una vez el proceso haya terminado, deberíamos poder ver algo similar a la siguiente imagen:



Trazas

Para el correcto funcionamiento del sistema, la comunicación entre los componentes **ogCore**, **ogAgent** y **ogRepository** debe ser asíncrona. Para ello, se utilizan trazas que nos permiten ver el estado de las peticiones y las respuestas.

Podremos acceder a las trazas del sistema, en el botón con un icono de una **campana** en el menú superior de la interfaz gráfica. Se nos ofrece un listado con todas las trazas del sistema. Podremos utilizar los filtros proporcionados si es necesario.



Para el ejemplo demostrativo, vemos 2 trazas:

- **Creación de imagen:** la traza nos informa del estado del agente respecto a la creacion de la imagen.
- **Creación de ficheros auxiliares:** la traza nos informa de la creación de los ficheros auxiliares asociados a la imagen.

Repositorio

Dentro del repositorio de imágenes, podremos ver la imagen que hemos creado en `/opt/opengsys/ogrepository/images`

```
-rw-r--r-- 1 root    root      32 Nov 27 03:16 image-test.img.sum
-rw-r--r-- 1 root    root      10 Nov 27 03:16 image-test.img.size
-rw-r--r-- 1 root    root     32 Nov 27 03:16 image-test.img.full.sum
-rw-r--r-- 1 root    root    16470 Nov 27 03:16 image-test.img.torrent
```

Llamadas API a los distintos componentes

1. **ogAgent:** Esta es la llamada que ogCore realiza al ogAgent para que este cree la imagen. Nos devuelve como resultado un **job_id** que será el que nos muestre en la interfaz la traza relacionada con la creación esta imagen.

```
curl -X POST --data '{"dsk":"1", "par":"2", "cpt":"83", "idi":"01936caf-facb-72ad-9446-97e7dbf2b96a", "nci":"image-test", "ipr":"192.168.68.55", "nfn":"CrearImagen", "ids":""}'
https://192.168.2.2:8000/CloningEngine/CrearImagen
{"job_id": "CrearImagen-93992a4b"}
```

2. **ogRepository:** Esta es la llamada que ogCore realiza al ogRepository para que este cree los ficheros auxiliares asociados a la imagen. Nos devuelve como resultado un **job_id** que será el que nos muestre

en la interfaz la traza relacionada con la los mismos.

```
curl -X POST --data '{"image":"image-test.img"}'  
https://192.168.68.55:8006/ogrepository/v1/images/torrentsum  
{"job_id": "CreateAuxiliarFiles_eba647d4"}
```

Estas llamadas comunican sus estados a ogCore mediante **webhooks**, y este se encarga de procesar la respuesta y cambiar el estado cuando sea necesario.