



**OSL - OFICINA DE SOFTWARE LIBRE**  
**2023**  
**UNIVERSIDAD DE GRANADA**

---

# Manual del operador de Clonezilla Server con Ansible.

---

Oficina de Software libre - Pedro A. Mayorgas Parejo

22 de marzo de 2023

# Índice

<b>1</b>	<b>Prefacio</b>	<b>3</b>
<b>2</b>	<b>Software Utilizado</b>	<b>3</b>
2.1	Información de los servicios utilizados: DRBL . . . . .	3
<b>3</b>	<b>Ansible</b>	<b>5</b>
3.1	playbook: deploy_drbl . . . . .	7
3.1.1	Proceso ilustrativo del despliegue con ansible . . . . .	9
3.2	playbook: provisioning_ocs . . . . .	11
3.2.1	Proceso ilustrativo de la subida de la imagen y la selección del modo	13
3.2.2	Los modos no se ejecutan correctamente indicando que hay un problema con el disco/partición . . . . .	17
3.3	playbook: update_clients . . . . .	18
<b>4</b>	<b>Anexo: SSH</b>	<b>22</b>

## 1. Prefacio

Este proyecto surge de la necesidad de reducir los tiempos de trabajo físico de clonación de imágenes preparadas para decenas de máquinas en un tiempo corto. La Oficina de Software Libre, en concreto su personal, ha tenido que responder a una demanda creciente de equipos donados, lo cual constituía un consumo de tiempo elevado por equipo. Siguiendo la **Ley de Amdahl**, este proyecto centra su esfuerzo en mejorar los tiempos de trabajo en la parte de clonación y provisionamiento de software. **¿Cuánto de mejora?** Las

mejoras han sido de pasar de tener preparados 1 equipo por día hasta un máximo de 16 equipos por día. Un S de 15 o en porcentaje un 150 % de mejora, que se puede conseguir más pero el personal se satura si tiene que revisar, ampliar, limpiar, reparar, preparar el equipo desde el almacén.

## 2. Software Utilizado

El software utilizado para tal propósito ha sido:

- **Distribución de GNU/Linux:** Debian 11.6
- **Ansible como provisionamiento.**
- **DRBL Stable 2.5.1-16**

### 2.1. Información de los servicios utilizados: DRBL

**DRBL - Diskless Remote Boot Linux**, utiliza un servicio basado en TFTP para enviar las imágenes de clonación a través de sistemas que lo solicitan por PXE. Primero ofrece una dirección IPv4 privada por DHCPv4 para identificar al cliente que solicita una clonación, luego ofrece un "GRUB con distintos campos, basado el protocolo TFTP, que permite la selección de imágenes NetInstall (No cubiertas por este Manual del Operador), o la imagen precargada para clonación y dicha imagen se indica en el modo en el que va a funcionar Unicast o Multicast. La preferencia de uso de Unicast o Multicast dependerá

del operador, pero se deben tener en cuenta los siguientes factores:

- Selección de Multicast debe ser en base a los siguientes factores:
  - El Switch Físico o la tarjetas de red es un cuello de botella probable, es decir que tiene menos de 1 Gigabit/s.
  - El HDD utilizado es mecánico y no está en RAID.
  - Poca memoria RAM para caché de la imagen.
- Selección de Unicast debe ser en base a los siguientes factores:

- Los equipos no están disponibles en una cantidad determinada para el Multicast. Se quiere sacar en serie de 1 en 1, pero sin pausas en la clonación.
- Uso de SSD o un RAID, esto importa ya que el acceso aleatorio a la imagen a clonar no causa esperas o latencias elevadas. Teniendo en cuenta que cuando el n<sup>o</sup> de equipos por unicast conectados crece y están en momentos distintos de la clonación, estos demandan regiones de datos de la imagen distintos provocando un impacto en los acceso E/S ralentizando los progresos.
- Gran cantidad de RAM disponible como caché.
- Uso de switch y tarjetas de red de gran capacidad 1Gbit/s.

El uso de tarjetas de red de Gigabit es suficiente, ya que los equipos objetivos de la clonación son equipos de hace 10 años o más, por lo que su interfaz SATA es la versión I o II (1.5 Gigabits/s o 3 Gigabits/s) que junto a un disco duro mecánico de 5200 RPM, nunca van a poder alcanzar el máximo rendimiento, la escritura no pasa de los 130 MBytes/s siempre está limitada por los discos de E/S. Tampoco se ha considerado necesario que el servidor necesite mas capacidad de red, por los motivos referidos arriba, el servidor es del orden de 4-5 veces más rápido que la demanda que junto con la caché RAM, apenas le supone una carga real, porque las imágenes son servidas como volúmenes de cintas. Además se debe tener en cuenta la máquina sobre la que se va a instalar el servicio, ya que si esta es UEFI o BIOS, puede generar ciertas incompatibilidades a la hora de generar una imagen de kernel preparada para las máquinas que lo soliciten por PXE, siendo la mayoría a fecha de 2023 de BIOS legacy.

### 3. Ansible

Ansible es una herramienta de provisionamiento que, tiene el objetivo de automatizar tareas que son repetitivas o que requieren de un tiempo pero que ya se han realizado en otros servidores. Ansible usa el servicio SSH para la comunicación segura de las directivas de los playbook, por lo que es necesario tener instalado OpenSSH-server en el servidor. Se complementa con la herramienta DRBL, debido a que la OSL tiene la necesidad de

tener una automatización del despliegue del servicio DRBL, ya que los equipos usados para ofrecer dicho servicio varían a lo largo de las campañas de donación debido al espacio o bien por reutilizar los equipos que anteriormente se usaban como servidor, para otros propósitos que se consideren mas oportunos. Además se intenta ofrecer una automatización atemporal, donde un futuro operador que no conozca la herramienta sea capaz de desplegar este servicio sin tener un conocimiento previo o necesidad de una investigación que pueda consumir un tiempo. La automatización abarca y resuelve 4

problemas:

- **Problema 1:** Despliegue automatizado del servicio DRBL en un equipo nuevo.
  - **playbook:** deploy\_drbl.yml
- **Problema 2:** Provisionamiento o backup de imágenes de clonación hacia/desde el servidor usando Rsync.
  - **playbook:** provisioning\_ocs.yml
- **Problema 3:** Selección de imágenes de clonación y selección del modo del servidor, esto es Multicast o Unicast.
  - **playbook:** provisioning\_ocs.yml
- **Problema 4:** Redimensión de particiones, sistemas de ficheros, provisionamiento y actualización del software de las imágenes ya clonadas en el equipo de destino.
  - **playbook:** client\_ssh\_proxy/playbook/update\_clients.yml

Cada problema tiene un playbook propio menos el 2 y 3, que van en el mismo playbook ya que el provisionamiento de la imagen a veces implica la necesidad de querer indicarle al servicio que ofrezca esa imagen para clonar ya. La ejecución de los playbooks se indicará

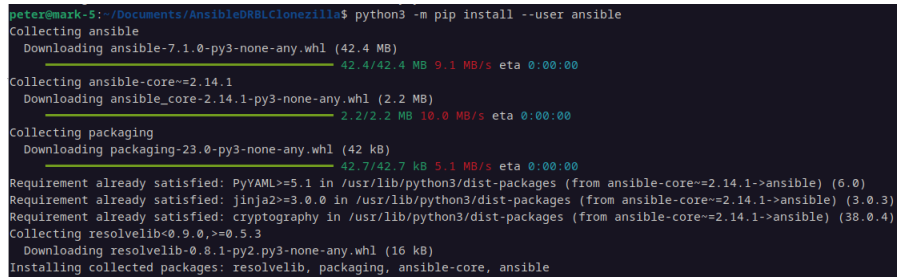
en cada sección que explique cada uno de ellos, ya que se incluyen variables que deben ser editadas para poder desplegar el servicio. **Importante:** Cuando creamos una imagen

para la clonación debemos tener en cuenta de incluir el servicio OpenSSH Server u otro que nos permita realizar conexiones de SSH, así como inyectar una clave pública de la workstation desde la que controlamos el proceso para poder utilizar el playbook del problema nº4. También en el servidor DRBL debemos tener el servicio por supuesto con

su clave pública inyectada para que Ansible pueda enviar los comandos de despliegue o provisionamiento a través de una conexión SSH al servidor, así como permitir usar el servidor de DRBL como un SSH bastión que permita dar un salto a todas las direcciones IPv4 privadas que tengan un lease de DHCP.

Para instalar Ansible debemos ejecutar el siguiente comando:

```
1 python3 -m pip install --user ansible
```



```
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ python3 -m pip install --user ansible
Collecting ansible
  Downloading ansible-7.1.0-py3-none-any.whl (42.4 MB)
    42.4/42.4 MB 9.1 MB/s eta 0:00:00
Collecting ansible-core==2.14.1
  Downloading ansible_core-2.14.1-py3-none-any.whl (2.2 MB)
    2.2/2.2 MB 10.0 MB/s eta 0:00:00
Collecting packaging
  Downloading packaging-23.0-py3-none-any.whl (42 kB)
    42.7/42.7 kB 5.1 MB/s eta 0:00:00
Requirement already satisfied: PyYAML>=5.1 in /usr/lib/python3/dist-packages (from ansible-core==2.14.1->ansible) (6.0)
Requirement already satisfied: Jinja2>=3.0.0 in /usr/lib/python3/dist-packages (from ansible-core==2.14.1->ansible) (3.0.3)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from ansible-core==2.14.1->ansible) (38.0.4)
Collecting resolvelib<0.9.0,>=0.5.3
  Downloading resolvelib-0.8.1-py2.py3-none-any.whl (16 kB)
Installing collected packages: resolvelib, packaging, ansible-core, ansible
```

Figura 1: Proceso de instalación de Ansible.

### 3.1. playbook: deploy\_drbl

#### Requisitos mínimos para el despliegue del servicio:

- **CPU:** 2 a 4 cores.
- **RAM:** 8GiB o más. Si se incorpora mas memoria RAM, el sistema la puede utilizar como caché para evitar el acceso a disco, útil cuando queremos el modo UNICAST.
- **NET:** 2 tarjetas de red 1Gbit Ethernet.

El servicio DRBL se despliega como el siguiente diagrama, donde podemos visualizar por qué necesitamos 2 tarjetas de red asociadas a la máquina, el motivo principal, es que necesita una red privada para poder mandar las solicitudes de DHCPv4 a los clientes que quieran utilizar el protocolo PXE para poder arrancar sin disco. Por lo tanto no podemos utilizar solamente una tarjeta de red. Además esta configuración que se despliega permite a los clientes cuando hayan terminado de realizar el clonado conectarse a internet por medio de una NAT preconfigurada por el servicio DRBL que le permita obtener actualizaciones.

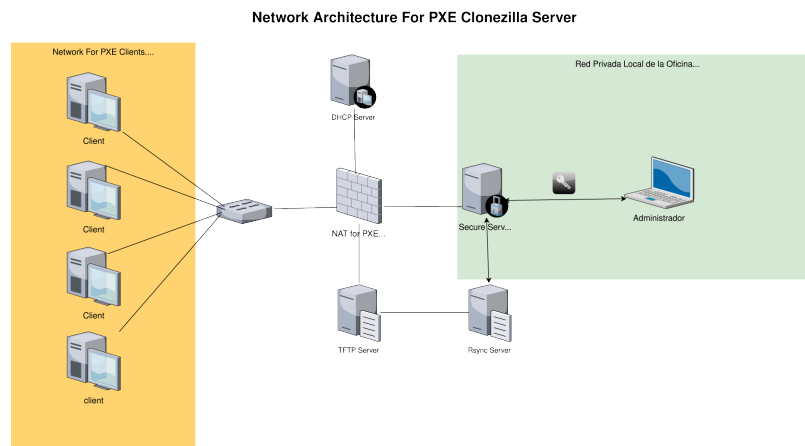


Figura 2: Podemos ver los diversos servicios que necesita DRBL para poder trabajar así como los tipos de conexión establecida y la NAT que tiene.

Ahora explicamos las partes del playbook que debemos modificar para poder desplegar el servicio:

La primera parte consiste en que debemos poner la dirección IPv4 de la red de la oficina del servicio correcta, tiene que ser la misma que la que tiene asignada por DHCPv4 por el router o manualmente si lo hemos hecho así. También debemos incluir la información de la puerta de enlace, el motivo por el que hacemos esto, es que el playbook desde las líneas 16 a 26 genera a partir del fichero **templates/interfaces.j2**, un fichero en **/etc/interfaces**, con las variables arriba referidas, incluyendo la variable LAN que también puede ser tocada pero necesitaría de una modificación adicional en el siguiente fichero

**templates/drblpush.conf**, el motivo es porque dicha variable, al variar necesitas vigilar si el rango de direcciones ip es el bueno para tu subred nueva, entonces la sección que debemos mirar son las líneas 40 a 47 ya que esas líneas son esenciales para un despliegue correcto del servicio DHCPv4 automatizado.

```
1 vars:
2     # DANGEROUS: Your WAN IP need to be the same if already as assigned to
3     # DHCP or Ansible did not auto reconnect
4     iface_wan:
5         - 192.168.122.68
6     iface_wan_gateway:
7         - 192.168.122.1
8     iface_wan_nameservers:
9         - 1.1.1.1
10    iface_lan:
11        - 192.168.111.1
```

Si cambiamos algún parámetro anterior, debemos de ir además al fichero de inventory.yml de la carpeta raíz ya que debemos ajustar las direcciones IPv4 y los hosts de manera acorde. En concreto en ese fichero debemos poner el **ansible\_host: 192.168.122.68** igual que la interface\_wan. Si modificamos algo de iface\_wan debemos ir además al fichero de drblpush.conf.j2 y también al inventory.yml de la subcarpeta client\_ssh\_proxy ya que tenemos que poner la dirección ssh del servidor en la siguiente variable de ssh y en la del host igual que en el inventory.yml de provisionamiento.

```
1 ansible_ssh_common_args: "-o ProxyCommand='ssh clonezilla@192.168.122.68 -
  o Port=22 -W %h:%p'"
```

La ejecución es totalmente automática con el siguiente comando, ask-become-pass te pide la contraseña para usar sudo o elevación de privilegios:

```
1 ansible-playbook --ask-become-pass playbooks/deploy_drbl.yml
```

Notas adicionales: En el momento de la edición del documento, el downgrade de grub aparentemente ya no hace falta, por lo que esa sección (líneas 80 85) están comentadas, si se usa una versión antigua de Debian 11, descomentar dichas líneas para evitar la rotura de dependencias.



### 3.1.1. Proceso ilustrativo del despliegue con ansible

Una vez ejecutado el playbook y puesta la contraseña de sudo (become). La primera parte del playbook, indica si encuentra al servidor, rehará la configuración de red y reiniciará el servicio de red para poder aplicar los cambios. Si la tarea de reconnect no sigue adelante, es que ha fallado algo relacionado con la configuración de red, puede ser normalmente por las configuraciones de variables de ip\_wan, que difiere con la ip real de reconexión del inventario.

```
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ ansible-playbook --ask-become-pass playbooks/deploy_drbl.yml
BECOME password:

PLAY [clonezillaservers] *****
TASK [Gathering Facts] *****
ok: [c11]

TASK [Defining dinamyc interfaces] *****
ok: [c11]

TASK [Restarting network services] *****
changed: [c11]

TASK [reconnect] *****
ok: [c11]
```

Figura 3: En esta sección el playbook indicara problemas de conectividad.

Luego a continuación, se puede observar el proceso de instalación de dependencias, que no es necesario reflejarlo en el libro ya que si ocurre algún fallo Ansible automáticamente lo indicará, pero lo que si es clave, es que la siguiente tarea se complete bien observando el log antes de continuar, lo pondrá claramente gracias a la salida que se ha generado con el playbook. El proceso es muy largo, ya que la configuración es decargar las dependencias necesarias de todos los subservicios dependientes y además las imágenes de instalación por red (netinstall)

```
TASK [Installing DRBL service with parameters - Long process] *****
```

Figura 4: Tarea crucial para el correcto funcionamiento futuro del servicio.

```
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ ssh-copy-id clonezilla@192.168.122.9
The authenticity of host '192.168.122.9 (192.168.122.9)' can't be established.
ED25519 key fingerprint is SHA256:mjfhWh3NwFzGwPVhf3g06BT6gfnX9kQBHOYi5sx0rlc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
clonezilla@192.168.122.9's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'clonezilla@192.168.122.9'"
and check to make sure that only the key(s) you wanted were added.
```

Figura 5: Si se ve la salida como en la ilustración es que todo ha ido bien y procede a realizar las configuraciones.

Las configuraciones de DRBL, son de manera resumida, los servicios DHCPv4, TFTP, de imagen, preparación de un kernel de arranque a partir del que tenemos, entre otros...

```

"u001b[0;39mCreating the uEFI network booting bootable image /tftpboot/nbi_img/bootx64.efi...
"Preparing the grub modules in /tftpboot/nbi_img/grub...",
"The uEFI network booting is ready.",
"*****",
"Creating the image files for PXE and Etherboot client computer(s), this will take a few minutes",
"The latest kernel for the DRBL clients is 5.10.0-21-amd64",
"Running mkinitrd --kernel 5.10.0-21-amd64 --all --no-modules",
"Will client check DHCP server name is \"drbl\" or not: yes",
"The maximum times to try to get IP address for a client: 5",
"The pause time after network card is up: 0",
"The timeout to wait for network card linked (Unit: 0.1 secs): 70",
"The NFS protocol for DRBL system: nfs3",
"Setting port for udhcpd request to default...",
"Using the kernel modules from /tftpboot/node_root/lib/modules...",
"The selected kernel for DRBL clients is: 5.10.0-21-amd64",
"Creating the network boot initrd for PXE clients by: mkpxeinitrd-net -k 5.10.0-21-amd64 -n
"Use kernel modules from /tftpboot/node_root/lib/modules/5.10.0-21-amd64",
"u001b[1;33m/tftpboot/node_root/lib/modules/.../boot/config-5.10.0-21-amd64 is found but e
"u001b[0;39mTrying to include network card firmwares if they exist in /tftpboot/node_root/li
"Calling hook udev...",
"Creating the initramfs image...",
"Initrd, assign ramdisk_size (80929) and ramdisk_block (1024) in /tftpboot/nbi_img/pxelinux.cf
"Finished!",
"Done!",
"*****",
"Done!"
}
)
TASK [Press enter to continue or cancel if a error is encountered.] *****
[Press enter to continue or cancel if a error is encountered.]
Press enter to continue, Ctrl+C to interrupt:

```

Figura 6: Si se ve la salida como en la ilustración es que todo ha ido bien y ya tenemos el servidor casi desplegado.

Ahora generará la configuración y nos indicará que ha ido todo bien tal y como sigue.

```

"Setting drbl_mode=\"none\" in /etc/drbl/drbl_deploy.conf and /etc/drbl/drblpush.conf... done!",
"Full clonezilla mode. Remove clientdir opt for label clonezilla in pxelinux config... ",
"Setting clonezilla_mode=\"full_clonezilla_mode\" in /etc/drbl/drbl_deploy.conf and /etc/drbl/drblpush.conf... done!",
"u001b[1;33mYou have to use \"dcs\" -> clonezilla-start to start Clonezilla service, so that there will be a Clonezilla menu when client boots",
"u001b[0;39mNo hidden label Clonezilla-live was found! Skip it.",
"u001b[0;39mNo hidden menuentry with ID clonezilla-live-client found! Skip it.",
"u001b[0;39mSome files are found in /usr/share/drbl/posrun/drbl/. Run them now...",
"*****",
"Enjoy DRBL!!!",
"http://drbl.org; http://drbl.nchc.org.tw",
"NCHC Free Software Labs, Taiwan. http://free.nchc.org.tw",
"*****",
"If you like, you can reboot the DRBL server now to make sure everything is ready...(This is not necessary, just an option)",
"*****",
"u001b[1;33mThe DRBL server is ready! Now set the client machines to boot from PXE. (refer to http://drbl.org for more details)",
"P.S. The config file is saved as /etc/drbl/drblpush.conf. Therefore if you want to run drblpush with the same config again, you may run it as: drblpush -c /
"u001b[0;39m"
}
)
TASK [Press any key to continue or cancel if a error is encountered.] *****
[Press any key to continue or cancel if a error is encountered.]
Press enter to continue, Ctrl+C to interrupt:
Msg: [c11]

PLAY RECAP *****
c11 : ok=22 changed=7 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ ;3

```

Figura 7: Si se ve la salida como en la ilustración es que todo ha ido bien y ya tenemos el servidor desplegado junto al resumen de Ansible.

### 3.2. `playbook: provisioning_ocs`

**Nota antes de instalar nada:** Debemos instalar rsync en nuestra workstation.

Este playbook se encarga de automatizar el proceso de subir la imagen en el directorio `/home/partimag`, que es donde el servicio DRBL lee para poder preparar la imagen para servirla mediante PXE. La subida de la imagen se realiza mediante Rsync con varias comprobaciones de SHA (Hash) para comprobar que se han subido de manera correcta al servidor. El playbook también incluye variables de control (yes/no) que permite controlar

qué queremos realizar con él, en concreto activa y desactiva las siguientes secciones:

- Subida de imagen nueva \*toda la carpeta.
- **Modo UNICAST:** Útil para cuando no queremos esperar a que haya un nº de máquinas prefijado o no las tengamos disponibles en estos momentos si no que va lanzando el programa a cada máquina que lo solicite por PXE sin importar como vayan las demás máquinas. Este modo quizás sobrecargue más el servidor, ya que la demanda de los PCs en distintos tiempos en los que se encuentra puede generar una latencia de E/S importante junto al ancho de banda. Pero debido a que el servidor normalmente es más potente.
- **Modo MULTICAST:** Al contrario que en el modo anterior, espera dos tipos de variables, el tiempo de espera desde que el primer equipo se conecta y/o el nº de equipos que Clonezilla debe esperar para ejecutarse. Si lo ponemos de manera combinada, prevalece el tiempo de espera normalmente, si no te da tiempo a conectar la cantidad de PCs indicada en este modo.
  - Espera de un Nº de equipos: Ponemos un nº acorde de equipos sobre los cuales se va a lanzar el proceso de copia en MULTICAST. Máximo 18 por el límite de DHCPv4.
  - Espera de segundos: Ponemos un número predeterminado de segundos que indique cuánto tiempo espera el servicio desde la primera solicitud de PXE.
  - Modo combinado anterior: Se indican las dos variables anteriores, prevalece la que se cumpla antes, es decir si conectamos todos los equipos a tiempo antes de que cumpla dicho tiempo arranca el programa y en caso contrario se inicia el Multicast con los que haya, siempre que se haya conectado primeramente un equipo por PXE. Este es el modo en el que está configurado el Ansible, ya que es el más eficiente.

La parte de la imagen en concreto debemos tener obligatoriamente `imagen_name` correctamente, ya que aunque no subamos una imagen, el playbook debe conocer cuál es su nombre para poder obtenerlo desde `/home/partimag`, en caso contrario se termina la ejecución del playbook.

La parte de `imagen_origin` es necesaria, ya que es una ruta local, absoluta o relativa, de nuestra workstation, donde tenemos la carpeta con los ficheros necesarios para realizar

el proceso de clonado. La transferencia se realiza por rsync, el playbooks se encarga de resolver dependencias de rsync si este no está disponible en el servidor (líneas 46 a 59) y en la máquina local debemos instalarlo por nuestra cuenta. El proceso de copia se cancela automáticamente junto con todo el playbook, si la imagen indicada no existe en el equipo local, o se haya copiado mal en el servidor remoto. Indicando cuál es el paso que ha fallado mediante assertions.

image\_name se puede poner de dos formas, debemos comentar una o otra según lo que vayamos a utilizar, si usamos la segunda forma con split debemos asegurarnos de que ha obtenido bien el nombre.

Para ejecutar el playbook debemos hacerlo como en el de despliegue del servicio con (ask-become-pass te pide la contraseña para usar sudo o elevación de privilegios):

La ejecución es totalmente automática con el siguiente comando:

```
1 ansible-playbook --ask-become-pass playbooks/provisioning_ocs.yml
```

### 3.2.1. Proceso ilustrativo de la subida de la imagen y la selección del modo

Primero tenemos que tener en cuenta la ruta de la imagen, en el caso ilustrativo está en:

```
peter@mark-5: /media/peter/CLONEZILLA/A-IMAGENES/guinea-ecuatorial-2022-ok$ pwd
/media/peter/CLONEZILLA/A-IMAGENES/guinea-ecuatorial-2022-ok
```

Figura 8: Ruta de la imagen absoluta que debemos poner en nuestra imagen.

Las variables que debemos tener en cuenta al principio del plabook, son:

```
vars:
  push_image: no
  image_origin: "/home/partimag/guinea-ecuatorial-2022"
  #image_name: "imagenname"
  image_name: "{{ image_origin.split('/') [3] }}"
  workstation_username: username
  clonezilla_mode_unicast: yes
  clonezilla_mode_multicast: "{{ not clonezilla_mode_unicast }}"
  clonezilla_mode_multicast_wait_client: 19
  clonezilla_mode_multicast_wait_time: 60
  clonezilla_disk_restore: "sdb"
```

Explicación de las variables fijadas:

- **push\_image** yes or no, activa el modo de rsync, util si no tienes la imagen instalada en el servidor.
- **image\_origin** Indicamos la ruta absoluta o relativa de la ubicación de la carpeta de la imagen.
- **image\_name** Aquí podemos usar las dos formas, una con split (con comprobador de si lo has realizado bien) o poniendo el nombre de la imagen a mano.
- **workstation\_username** Nombre de usuario que usas actualmente en tu workstation local, necesario para comprobar que la ruta que has indicado como parámetro en push\_image existe.
- **clonezilla\_mode\_unicast** Se debe poner yes o no, es mutuamente excluyente con el modo multicast, si ponemos que sí en este modo el multicast se fija en no.
- **clonezilla\_mode\_multicast** Al ser mutuamente excluyente, no debemos modificar nada, solo que si ponemos no en el modo unnicast este modo se activa.
- **clonezilla\_mode\_multicast\_wait\_client** Variable que en la cual indicamos el n de clientes máximo que debemos esperar en este modo. Debemos tener en cuenta que DHCPv4 tiene un máximo de leases de 14
- **clonezilla\_mode\_multicast\_wait\_time** Variable en la cual indicamos la espera máxima (en segundos), desde que el primer cliente se conecta.

- **clonezilla\_disk\_restore** Variable a indicar dependiendo del nombre de la unidad clonada, puede ser sda, sdb,... Ver el final de esta sección para poder saber cuál es la correcta.

Proceso:

```
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ ansible-playbook --ask-become-pass playbooks/provisioning_ocs.yml
BECOME password:

PLAY [clonezillaservers] *****
TASK [Gathering Facts] *****
ok: [cl1]

TASK [Comprobar si has hecho el split bien! Cancela en el caso de que no lo hayas hecho bien] *****
ok: [cl1] => {
  "image_name": "guinea-ecuatorial-2022"
}

TASK [Check that the guinea-ecuatorial-2022 exists on local computer.] *****
ok: [cl1 -> 127.0.0.1]

TASK [Assertion of guinea-ecuatorial-2022 exists] *****
ok: [cl1] => {
  "changed": false,
  "msg": "OK image found on /media/peter/internaldisk/guinea-ecuatorial-2022"
}
```

Figura 9: Proceso de copiado de la imagen y comprobación de que existe en el directorio de la workstation.

```
TASK [AutoInstalling rsync on server] *****
ok: [cl1]

TASK [Check and changing the folder owner] *****
ok: [cl1]

TASK [Copy a new image file file into place, backing up the original if it differs from the copied version] *****
ok: [cl1]

TASK [Showing Output rsync] *****
ok: [cl1] => {
  "rsync_output": {
    "changed": false,
    "cmd": "/usr/bin/rsync --delay-updates -F --compress --checksum --archive --rsh='/usr/bin/ssh' -S none -i /home/
KeyChecking=no -o UserKnownHostsFile=/dev/null' --rsync-path='sudo -u clonezilla rsync' --out-format='<<CHANGED>>%i %n'
inea-ecuatorial-2022 clonezilla@192.168.122.68:/home/clonezilla",
    "failed": false,
    "msg": "",
    "rc": 0,
    "stdout_lines": []
  }
}
```

Figura 10: Proceso de instalación de rsync en el servidor y copiado de la carpeta con rsync.

```

ln<f+***** guinea-ecuatorial-2022/sdb-mbr\n<f+***** guinea-ecuatorial-2022/sdb-pt.parted\n<f+***** guinea-ecuatorial-2022/sdb-pt.sf\n<f+***** guinea-ecuatorial-2022/sdb2.ext4-ptcl-img.bz2\n<f+***** sdb1_info\n",
    "rc": 0,
    "stdout_lines": [
        "cd+***** guinea-ecuatorial-2022/",
        "<f+***** guinea-ecuatorial-2022/Info-05-prober.txt",
        "<f+***** guinea-ecuatorial-2022/Info-dm1.txt",
        "<f+***** guinea-ecuatorial-2022/Info-img-id.txt",
        "<f+***** guinea-ecuatorial-2022/Info-img-size.txt",
        "<f+***** guinea-ecuatorial-2022/Info-lshw.txt",
        "<f+***** guinea-ecuatorial-2022/Info-lspci.txt",
        "<f+***** guinea-ecuatorial-2022/Info-packages.txt",
        "<f+***** guinea-ecuatorial-2022/Info-saved-by-cmd.txt",
        "<f+***** guinea-ecuatorial-2022/Info-smart.txt",
        "<f+***** guinea-ecuatorial-2022/MD5SUMS",
        "<f+***** guinea-ecuatorial-2022/SHA1SUMS",
        "<f+***** guinea-ecuatorial-2022/blkdev.list",
        "<f+***** guinea-ecuatorial-2022/blkid.list",
        "<f+***** guinea-ecuatorial-2022/clonezilla-img",
        "<f+***** guinea-ecuatorial-2022/dev-fs.list",
        "<f+***** guinea-ecuatorial-2022/disk",
        "<f+***** guinea-ecuatorial-2022/efi-nvram.dat",
    ]
}

```

```
TASK [Remove old image on destination folder] *****
changed: [c11]

TASK [Moving new guinea-ecuatorial-2022 to /home/partimag] *****
changed: [c11]
```

[illegible]

```

skipping: [cli]

TASK [Assertion of guinea-ecuatorial-2022 exists] *****
skipping: [cli]

TASK [Creating OCS * Clonezilla server session in unicast mode] *****
changed: [cli]

TASK [Showing Output unicast] *****
ok: [cli] => {
  "ocs_output": {
    "changed": true,
    "cmd": "sudo drbl-ocs -g auto -e1 auto -e2 -r -x -j2 -sc0 -p reboot -l es_ES.UTF-8 startdisk restore guinea-ecu",
    "delta": "0:00:10.701856",
    "end": "2023-01-24 19:19:14.000571",
    "failed": false,
    "rc": 0,
    "start": "2023-01-24 19:19:03.298715",
    "stdout": "Setting the TERM as linux\r\n*****\r\n\r\nClean all the previous saved config file if they exist...Clean all the previous s",
    "stderr": "*****\r\n\r\nClean all the previous saved config file if they exist...Clean all the previous s",
    "stdout_lines": [
      "*****\r\n\r\nClean all the previous saved config file if they exist...Clean all the previous s",
      "hey exist...done!\r\n\r\nClean all the previous saved GRUB EFI NB config files if they exist...done!\r\n\r\n\r\n33mClient",
      "ob is: nfsroot\r\n\r\n\r\n33mstart_ocs_service -t restoredisk -o guinea-ecuatorial-2022 sda\r\n\r\n\r\n33mlock dir: /v",
      "33mAtenci3n! la opci3n \"range\" se encuentra en dhcpd.conf, esto no es bueno para Clonezilla... Es mejor dejar a su d",
      "P mediante la configuraci3n de la MAC en dhcpd.conf por lo que NO sobrescribir3 el S.O. de m3quinas desconocidas.\r\n\r\n\r\n33m"
    ]
  }
}

```

Figura 14: Modo UNICAST ejecutado correctamente.

Si alguna vez alguno de los dos modos no funciona correctamente o quiere realizar alguna otra operaci3n como ejecutar un Memtest86+ o una imagen netinstall ejecute el siguiente comando:

```
1 sudo dcs
```



### 3.2.2. Los modos no se ejecutan correctamente indicando que hay un problema con el disco/partición

Si falla alguno de los dos modos indicando que el nombre del disco no es el correcto porque sea distinto a sda (como por ejemplos sdb o cualquier otro). Debemos ir a la ruta del directorio en /home/partimag/carpetaimagen, para ejecutar el siguiente comando y visualizar los binarios comprimidos que tengan la imagen, como en la imagen, se puede ver que es sdb en vez de sda, esto ocurre cuando hay más de un disco usado en el sistema clonado, debemos estar pendiente de esto.

```
-rwxr-x-- 1 clonezilla clonezilla 667 Dec 16 12:31 MD5SUMS
-rwxr-x-- 1 clonezilla clonezilla 5 Dec 16 12:26 parts
-rwxr-x-- 1 clonezilla clonezilla 11054389942 Dec 16 12:26 sdb2.ext4-ptcl-img.bz2
-rwxr-x-- 1 clonezilla clonezilla 36 Dec 16 12:26 sdb-chs.sf
-rwxr-x-- 1 clonezilla clonezilla 1048064 Dec 16 12:26 sdb-hidden-data-after-mbr
-rwxr-x-- 1 clonezilla clonezilla 512 Dec 16 12:26 sdb-mbr
-rwxr-x-- 1 clonezilla clonezilla 352 Dec 16 12:26 sdb-pt.partitioned
-rwxr-x-- 1 clonezilla clonezilla 318 Dec 16 12:26 sdb-pt.partitioned.compact
-rwxr-x-- 1 clonezilla clonezilla 209 Dec 16 12:26 sdb-pt.sf
-rwxr-x-- 1 clonezilla clonezilla 829 Dec 16 12:35 SHA1SUMS
-rwxr-x-- 1 clonezilla clonezilla 53 Dec 16 12:26 swappt-sdb1.info
clonezilla@clonezilla:~$ ls -la /home/partimag/guinea-ecuatorial-2022/
```

Figura 15: Buscando el nombre del disco que queremos recuperar correcto.

```
mult... ", "Make \"clonezilla\" as default label in /tftpboot/nc1_img/pxelinux.cfg/default", "The MENU LABEL is \"Clonezilla: multicast restore guinea-ecuatorial-2022 to disk sda\"", "Setting the GRUB EFI clients to DRBL mode with label \"Clonezilla: multicast restore guinea-ecuatorial-2022 to disk sda\"", "The MENUENTRY LABEL is \"Clonezilla: multicast restore guinea-ecuatorial-2022 to disk sda\"", "\u001b[1;33mThe status received by ocsgrd will be saved to /var/log/clonezilla/ocsgrd-notify.log.", "\u001b[0;39mnohup: appending output to 'nohup.out'", "Los trabajos del cliente se guardan en este archivo: /var/log/clonezilla/clonezilla-jobs.log.", "El log de sfdisk cuando Clonezilla est\u00e1 ejecut\u00e1ndose est\u00e1 en el cliente /var/log/clonezilla.log", "\u001b[1;31mNo target partitions were found from image guinea-ecuatorial-2022!", "Failed to start multicast clone service for image guinea-ecuatorial-2022!", "\u001b[0;39mPrograma terminado!"]
PLAY RECAP *****
cli : ok=11 changed=4 unreachable=0 failed=1 skipped=2 rescued=0 ignored=0
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$
```

Figura 16: Identificación del error de los discos.

```
DRBL (http://drbl.org)
Clonezilla: unicast restore guinea-ecuatorial-2022 to disk sdb
Local operating system (if available)

Press (Tab) to edit options

* Clonezilla version: 4.5.16-drbl1. (C) 2003-2023, NCHC, Taiwan
* Disclaimer: Clonezilla comes with ABSOLUTELY NO WARRANTY
```

Figura 17: Cliente que ha recibido respuesta a la solicitud PXE del servidor.

### 3.3. playbook: update\_clients

Esta necesidad es diferente a lo exigido antes, se parte de la necesidad de tener un método de actualización eficaz que no implique tener que ir al equipo físico ya que implica una pérdida de tiempo en el que lo dedicamos a conectar el equipo a un monitor y teclado para poder realizar las actualizaciones. La solución ha sido configurar Ansible,

en otra subcarpeta llamada **client\_ssh\_proxy**, otro inventario totalmente distinto que contiene la dirección IPv4 de cada host cedida por lease con DHCPv4. Para poder usar un ssh bastion o jump, que implica usar al servidor donde está DRBL, como intermediario para saltar a los otros equipos clonados ya que están detrás de una NAT, por motivos de aislar las peticiones PXE. Como pre-requisito clave, debemos tener la clave pública

de nuestra workstation copiada dentro de la imagen y el servidor ssh. Así como cambiar el nombre de usuario de los hosts clonados al que corresponda en su inventory.yml de la subcarpeta.

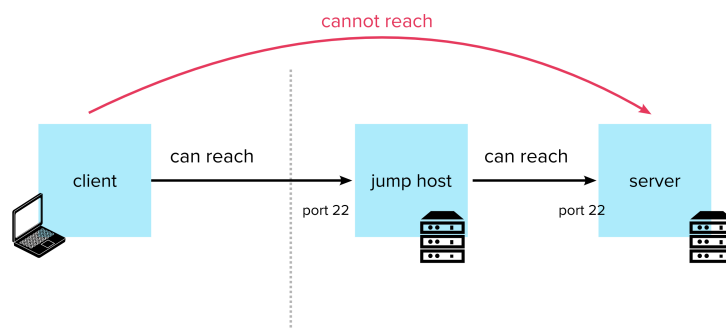


Figura 18: Funcionamiento de un servidor ssh bastión que se proporciona como punto de salto hacia otros hosts de ssh.

El playbook es sencillo implica la ejecución de un script previamente preparado para ser lanzado en contra de todos los clientes que haya detrás de la NAT de Clonezilla.

```
—
- hosts: nodes
  become: yes
  gather_facts: yes

  tasks:
    - name: Set authorized key taken from file
      ansible.posix.authorized_key:
        user: usuario
        state: present
        key: "{{ lookup('file', '~/ssh/id_rsa.pub') }}"

    - name: Execute script
      ansible.builtin.script: ../scripts/actualizacion.sh
```

```
register: script
```

El script deberá estar correctamente escrito o comprobado, o en otro caso usar las utilidades de Ansible para provisionar a los clientes. Los scripts deben estar ubicados con un nombre único en la subcarpeta scripts.

Como ejemplo de sustitución de los scripts de bash mejor controlados ponemos un ejemplo bastante ampliado en la V2.0, donde tenemos que tener en cuenta que, las particiones de los discos donde si son en UEFI, la partición EXT-4 que queremos extender está en sda3, por lo que la V1.0 nos daría fallo, al estar solamente preparada para los sistemas BIOS.

Esto surge debido a que los sistemas UEFI (EFI), tienen una partición FAT32 EFI que tienen las variables y los archivos de arranque. Antes, en los sistemas MBR, el grub se solía guardar en el primer sector del disco.

Tenemos que tener en cuenta las variables de los discos, pero en el caso de no poder visualizarlas en los clientes distribuidos, se ha incluido un controlador que muestra al operador la tabla de particiones de sda, antes de continuar la ejecución con el playbook. Pudiendose cancelar y cambiar.

```
—
- hosts: nodes
  become: yes
  gather_facts: yes
  vars:
    disk: "/dev/sda"
    partition: 3

  tasks:
    # Execute with: ansible-playbook --ask-become-pass --inventory-file ../host.ini update_
    - name: "Showing partition table of one of the system for check if it's correct run_once"
      ansible.builtin.shell: "fdisk -l | grep {{ disk }}"
      register: fdisk_register
      run_once: true

    - name: "Make sure if your partition and disk is correct {{ disk }} with {{ partition }}"
      ansible.builtin.debug:
        var: fdisk_register
      run_once: true

    - name: "Pause for checking"
      ansible.builtin.pause:
        minutes: 1

    - name: Set authorized key taken from file
      ansible.posix.authorized_key:
        user: usuario
        state: present
        key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"

    - name: apt update
      ansible.builtin.apt:
        upgrade: yes
        update_cache: yes
        cache_valid_time: 3600

    - name: Remove useless packages from the cache
      ansible.builtin.apt:
        autoclean: yes
```

- name: Remove dependencies that are no longer required  
`ansible.builtin.apt:`  
`autoremove: yes`
- name: snap update  
`ansible.builtin.command: "sudo snap refresh"`
- name: Instalando utilidades para incrementar el tamaño de las particiones.  
`ansible.builtin.apt:`  
`name: cloud-guest-utils`  
`state: present`  
`update_cache: yes`
- name: Incrementando las particiones  
`ansible.builtin.command: "sudo growpart {{ disk }} {{ partition }}"`  
`register: growpart`
- name: Showing growpart.  
`ansible.builtin.debug:`  
`var: growpart`
- name: Incrementando el sistema de ficheros  
`ansible.builtin.command: "sudo resize2fs {{ disk }}{{ partition }}"`  
`register: resize2fs`
- name: Showing resize2fs  
`ansible.builtin.debug:`  
`var: resize2fs`
- name: desactivar servidor SSH  
`ansible.builtin.systemd:`  
`enabled: false`  
`name: ssh`
- name: apagado del sistema  
`community.general.shutdown:`

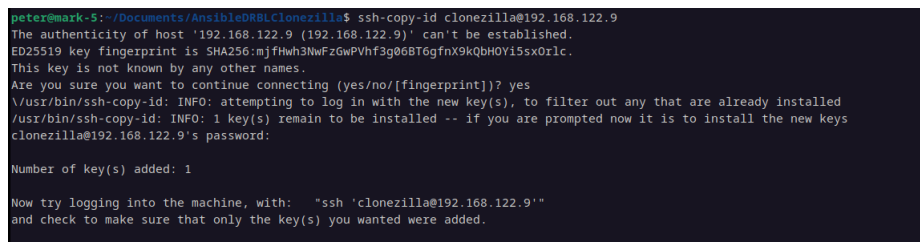
## 4. Anexo: SSH

Generar nuestro `id_rsa` básico, para poder utilizar Ansible, ya que depende del uso de las claves públicas para poder ejecutar todos los playbooks.

```
1 ssh-keygen
```

Para copiar nuestro `id_rsa.pub` al servidor nuevo antes de la ejecución de los playbook y si queremos usar el módulo de actualizaciones de software también tenemos que hacerlo en la imagen que queramos clonar, pero este proceso será manual o distinto ya que la imagen se crea en un entorno aislado normalmente.

```
1 ssh-copy-id clonezilla@192.168.1.101
```



```
peter@mark-5:~/Documents/AnsibleDRBLClonezilla$ ssh-copy-id clonezilla@192.168.122.9
The authenticity of host '192.168.122.9 (192.168.122.9)' can't be established.
ED25519 key fingerprint is SHA256:mjfhWh3NwFzGwPVhf3g06BT6gfnX9kQbHOYi5sxOrlc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
clonezilla@192.168.122.9's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'clonezilla@192.168.122.9'"
and check to make sure that only the key(s) you wanted were added.
```

Figura 19: Proceso de copia de clave pública RSA al servidor..

Por ultimo para poder ejecutar el playbook, que nos permite actualizar los clientes tenemos que ejecutar el siguiente comando.

```
1 ansible-playbook --ask-become-pass --inventory-file ../host.ini
  update_clients_ansible.yml
```