



Prefacio

Este proyecto surge de la necesidad de reducir los tiempos de trabajo físico de clonación de imágenes preparadas para clonación.

**¿Cuánto de mejora?** Las mejoras han sido de pasar de tener preparados 1 equipo por día hasta un máximo de 16 equipos por día.

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**

Información de los servicios utilizados: DRBL

**DRBL - Diskless Remote Boot Linux**, utiliza un servicio basado en TFTP para enviar las imágenes de clonación a los equipos.

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 predeterminada para el Multicast. Se quiere sacar en serie de 1 en 1, pero se quiere sacar en paralelo.

Uso de SSD o un RAID, esto importa ya que el acceso aleatorio a la imagen a clonar no causa esperas o latencias elevadas.

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.

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 haber problemas.

Ansible

Ansible es una herramienta de provisionamiento que, tiene el objetivo de automatizar tareas que son repetitivas o que n

Se complementa con la herramienta DRBL, debido a que la OSL tiene la necesidad de tener una automatización del de

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

**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 ejecución de los playbooks se indicará en cada sección que explique cada uno de ellos, ya que se incluyen variables q

**Importante:** Cuando creamos una imagen para la clonación debemos tener en cuenta de incluir el servicio OpenSSH S

Para instalar Ansible debemos ejecutar el siguiente comando:

[style=mybash] python3 -m pip install --user ansible

[H] [scale=0.30]ansible/install-ansible Proceso de instalación de Ansible.

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.

**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.

[H] [scale=0.30]drbl/network<sub>d</sub>DiagramaPodemosverlosdiversosserviciosquenecesitaDRBLparapodertrabajarasícomolos

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 de la oficina.

[style=mybash] vars: DANGEROUS: Your WAN IP need to be the same if already as assigned to DHCP or Ansible de la oficina.

Si cambiamos algún parámetro anterior, debemos de ir además al fichero de inventory.yml de la carpeta raíz ya que debe de tener el mismo valor.

[style=mybash] ansible<sub>s</sub>sh<sub>common</sub>args : " - oProxyCommand = sshclonezilla@192.168.122.68 - oPort = 22 - W" - oPort = 22 - W

La ejecución es totalmente automática con el siguiente comando, ask-become-pass te pide la contraseña para usar sudo.

Notas adicionales: En el momento de la edición del documento, el downgrade de grub aparentemente ya no hace falta, pero se recomienda hacerlo.

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 encuen

[H] [scale=0.30]deploy-drbl/deploy01 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 l

[H] [scale=0.30]deploy-drbl/deploy02 Tarea crucial para el correcto funcionamiento futuro del servicio.

[H] [scale=0.30]deploy-drbl/deploy03 Si se ve la salida como en la ilustración es que todo ha ido bien y procede a realiz

Las configuraciones de DRBL, son de manera resumida, los servicios DHCPv4, TFTP, de imagen, preparación de un ke

[H] [scale=0.30]deploy-drbl/deploy04 Si se ve la salida como en la ilustración es que todo ha ido bien y ya tenemos el se

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

[H] [scale=0.30]deploy-drbl/deploy05 Si se ve la salida como en la ilustración es que todo ha ido bien y ya tenemos el se  
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 e

Subida de imagen nueva \*toda la carpeta.

**Modo UNICAST:** Útil para cuando no queremos esperar a que haya un n<sup>o</sup> de máquinas prefijado o no las tengamos disp

**Modo MULTICAST:** Al contrario que en el modo anterior, espera dos tipos de variables, el tiempo de espera desde que e

Espera de un N<sup>o</sup> de equipos: Ponemos un n<sup>o</sup> acorde de equipos sobre los cuales se va a lanzar el proceso de copia en MULT

Espera de segundos: Ponemos un número predeterminado de segundos que indique cuánto tiempo espera el servicio desde la

Modo combinado anterior: Se indican las dos variables anteriores, prevalece la que se cumpla antes, es decir si conectamos t

La parte de la imagen en concreto debemos tener obligatoriamente imagen\_name correctamente, ya que aunque no suba

La parte de imagen\_origin es necesaria, ya que es una ruta local, absoluta o relativa, de nuestra workstation, donde ten

image\_name se puede poner de dos formas, debemos comentar una o otra según lo que vayamos a utilizar, si usamos la

[style=mybash] push\_image : noimage\_origin : " /Documents/ImagesOSL/reciclaje - 2022 - 3 - ok"image\_name : "gu

Los modos se indican con las siguientes variables, debiendo poner yes en el respectivo modo que no debemos poner y no

[style=mybash] En el ejemplo se puede ver clonezilla\_mmode\_multicast : noclonezilla\_mmode\_multicast\_wait\_client : 6clonezilla

Para ejecutar el playbook debemos hacerlo como en el de despliegue del servicio con (ask-become-pass te pide la cont

La ejecución es totalmente automática con el siguiente comando: [style=mybash] ansible-playbook -ask-become-pass pl

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:

[H] [scale=0.10]provisioning-ocs/provisioning01 Funcionamiento de un servidor ssh bastión que se proporciona como punto de partida  
[firstline=6,lastline=15,style=yaml]../playbooks/provisioning\_ocs.yml

Explicación de las variables fijadas:

**push\_image** yes or no, activa el modo de rsync, útil 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.

**workstation\_username** Nombre de usuario que usas actualmente en tu workstation local, necesario para comprobar que la imagen es la correcta.

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 dependa de la imagen.

La solución ha sido configurar Ansible, en otra subcarpeta llamada **client\_ssh\_proxy**, otro inventario totalmente distinto al anterior.

Como pre-requisito clave, debemos tener la clave pública de nuestra workstation copiada dentro de la imagen y el servicio de ssh habilitado.

[H] [scale=0.10]update-clients/bastion.png Funcionamiento de un servidor ssh bastión que se proporciona como punto de partida

Anexo: SSH

Generar nuestro id\_rsa básico, para poder utilizar Ansible, ya que depende del uso de las claves públicas para poder ejecutar los playbooks.

[style=mybash] ssh-keygen

Para copiar nuestro id\_rsa.pub al servidor nuevo antes de la ejecución de los playbooks y si queremos usar el módulo de copia de archivos.

[style=mybash] ssh-copy-id clonezilla@192.168.1.101

[H] [scale=0.30]anexo/copy-id Proceso de copia de clave pública RSA al servidor..