

Despliegue CMS

GESTIÓN Y ADMINISTRACIÓN DE REDES

ADRIÁN CARRASCO ESPINOSA, HÉCTOR ALBERCA S-Q, RUBÉN
CRESPO CALCERRADA Y JAVIER DE LA CONCEPCIÓN DORADO
PROFESOR: TOBÍAS DÍAZ DÍAZ-CHIRÓN

Contenido

0. Tabla de variables por defecto	2
1. Configuración del Nodo 0	2
1.1 Servicio DHCP	2
1.2 PXE	4
1.3 Servicio TFTP	4
1.4 Servicio HTTP	4
2. Cuando se crea un nuevo nodo	4
3. Nodos que creamos en VirtualBox:	5
4. Adaptadores de red del nodo Kubernetes en VirtualBox:	5
5. Configuración del sistema del nodo kubernetes en VirtualBox:	6
6. Esquema en Packet Tracer	6
7. Despliegue	7
8. Problemas encontrados y aprendizaje	8
9. Bibliografía	8

0. Tabla de variables por defecto

Servicio	Variable
HTTP	Default por instalación (IP: 192.168.10.12)
TFTP	Default por instalación (IP: 192.168.10.12)
DHCP	Cambios en el archivo de configuración, vistos después (IP: 192.168.10.12)
UBUNTU	Versión 20.04 (Tanto en el nodo0, como en los demás nodos)

1. Configuración del Nodo 0

A partir del nodo 0 se deben instalar el resto de los nodos. Para ello, se utilizarán 3 servicios en este nodo:

- HTTP
- TFTP
- DHCP

1.1 Servicio DHCP

Este servicio se encarga de asignar direcciones IP a los nuevos nodos. Para ello, dentro del archivo dhcpd.conf, en el nodo 0, definimos dos redes distintas:

- La red interna -> 192.168.10.0.
- La red main -> 192.168.11.0

Los rangos de IPs que asigna cada red van de x.10 a x.200.

```
GNU nano 4.8 /etc/dhcp/dhcpd.conf
# }
# pool {
#     deny members of "foo";
#     range 10.0.29.10 10.0.29.230;
# }
#}
# config red interna dhcp

subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.10 192.168.10.200;
    default-lease-time 3600;
    max-lease-time 86400;
    option routers 192.168.10.1;
    option domain-name-servers 8.8.8.8;
}

subnet 192.168.11.0 netmask 255.255.255.0 {
    range 192.168.11.10 192.168.11.200;
    default-lease-time 3600;
    max-lease-time 86400;
    option routers 192.168.11.1;
    option domain-name-servers 8.8.8.8;
}
}
```

Además, cada nodo tiene su respectiva configuración:

```
GNU nano 4.8 /etc/dhcp/dhcpd.conf

host kubernetes {

    hardware ethernet 00:00:00:00:00:01;
    fixed-address 192.168.10.10;
    next-server 192.168.10.12;
    filename "ubuntu2004/pxelinux.0";
}

host nginx {

    hardware ethernet 00:00:00:00:00:02;
    fixed-address 192.168.11.10;
    next-server 192.168.10.12;
    filename "ubuntu2004/pxelinux.0";
}

host server1 {

    hardware ethernet 00:00:00:00:00:03;
    fixed-address 192.168.11.11;
    next-server 192.168.10.12;
    filename "ubuntu2004/pxelinux.0";
}

host server2 {
    hardware ethernet 00:00:00:00:00:04;
    fixed-address 192.168.11.13;
}

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste Text ^T To Spell   ^G Go To Line M-E Redo
```

Lo que hacemos aquí es asignar la configuración que tendrá un dispositivo que solicite una IP, filtrando según su MAC.

Por ejemplo, el host “kubernetes” indica que al nodo con la dirección MAC 00:00...01:

- Se le asignará la IP 192.168.10.10
- Se le asignará el servidor FTP 102.168.10.12
- Se le asignará el archivo pxelinux.0, y dentro de este, el específico de su mac.

1.2 PXE

Aunque en todos los hosts el archivo de PXE es el mismo (pxelinux.0) realmente a cada host se le asigna un archivo distinto. Esta instrucción lo que hace es buscar en el directorio ubuntu2004/pxelinux.cfg un archivo de configuración de PXE cuyo nombre es la MAC del respectivo host (o en su defecto, un archivo con nombre default)

Por ejemplo, el host de un nodo con MAC 00:00:00:00:00:03 será el server1. Su archivo de configuración PXE será ubuntu2004/pxelinux.cfg/**01**-00-00-00-00-03 (el 01 va primero siempre por formato, y el 03 como método de identificación para la mac).

Si ese archivo no existe, el archivo de configuración será ubuntu2004/pxelinux.cfg/default

Un archivo de configuración PXE tendrá este contenido:

```
tryhardkillua@nodo0:~$ cat /srv/tftp/ubuntu2004/pxelinux.cfg/01-00-00-00-00-01
DEFAULT install
LABEL install
  KERNEL vmlinuz
  INITRD initrd
APPEND root=/dev/ram0 ramdisk-size=3000000 ip=dhcp url=http://192.168.10.12/ubuntu2004/ubuntu-20.04.6-live-server-amd64.iso autoinstall ds=nocloud-net;s=http://192.168.10.12/ubuntu2004/autoinstall/tes
t/
tryhardkillua@nodo0:~$
```

1.3 Servicio TFTP

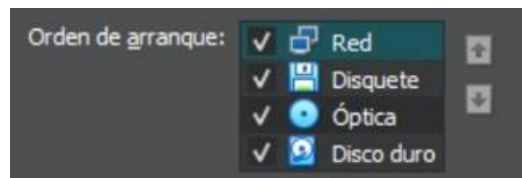
Este servicio se encarga de transferir archivos a los nodos. Con esto lo que hacemos es configurar PXE, que es el sistema de arranque por red. Mediante TFTP montamos la ISO de Ubuntu, en este caso, en el nuevo nodo, para obtener los archivos initrd y vmlinuz (archivos necesarios para tema de kernel y permisos). También creamos el archivo de configuración de PXE (pxelinux.cfg)

1.4 Servicio HTTP

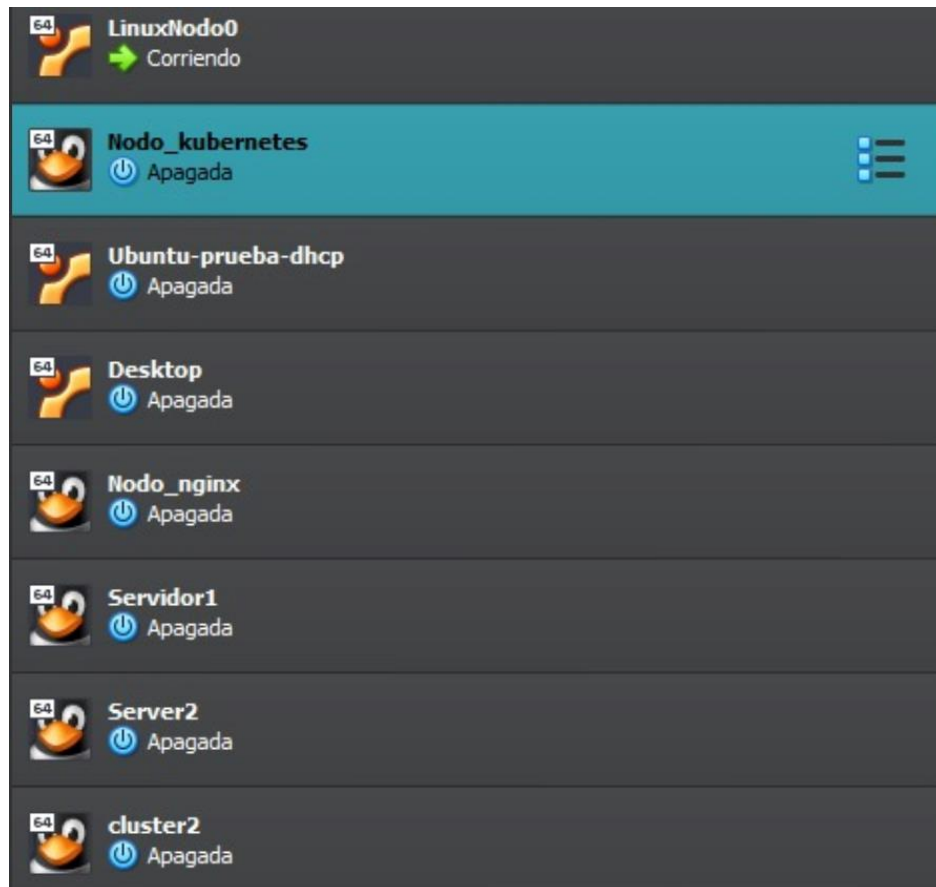
En el servidor HTTP es donde va a estar todo el contenido, para que otros servicios puedan acceder a estos. Por ejemplo, el servidor TFTP obtiene la ISO de Ubuntu mediante HTTP. También es donde se encuentra el archivo de configuración de cloud-init.

2. Cuando se crea un nuevo nodo

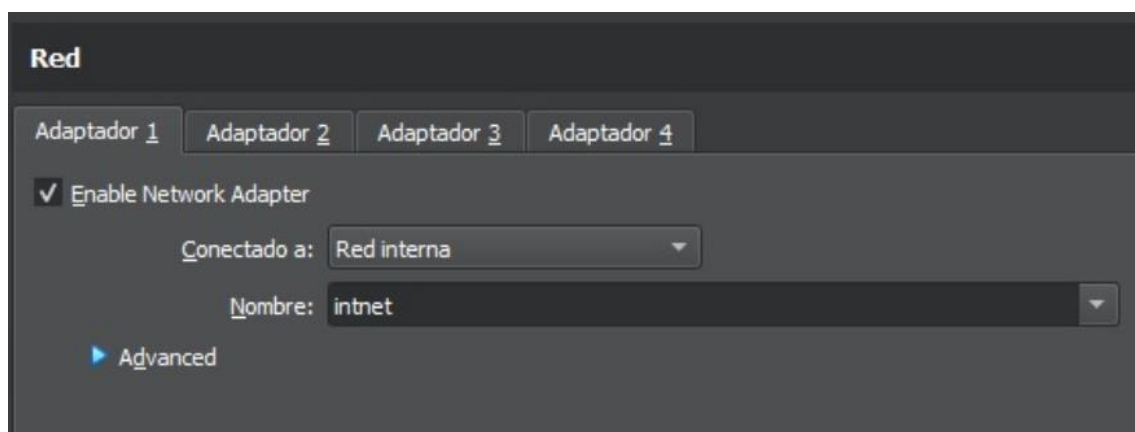
Para que un nuevo nodo se inicie y se configure correctamente (se le asigne una IP por medio de DHCP, obtenga la ISO de Ubuntu y los archivos de PXE...) su modo de arranque debe ser por red. Para ello cambiamos el orden de arranque en VirtualBox y ponemos la de red con mayor prioridad, dentro de la configuración de sistema.

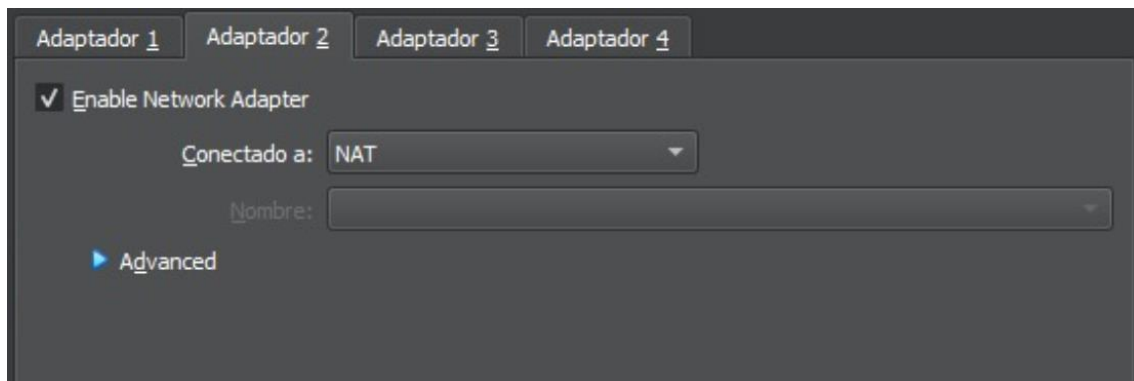


3. Nodos que creamos en VirtualBox:

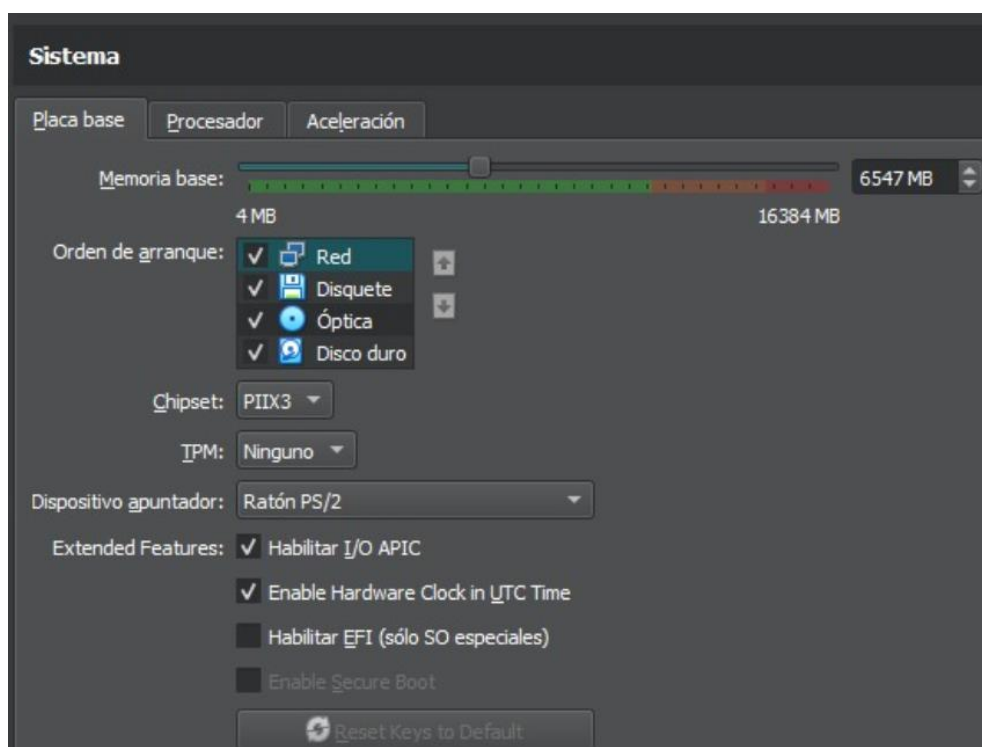


4. Adaptadores de red del nodo Kubernetes en VirtualBox:



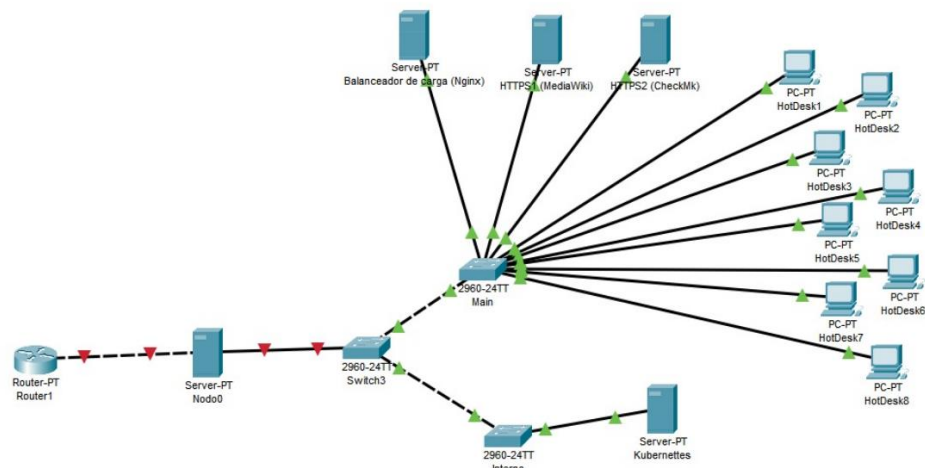


5. Configuración del sistema del nodo kubernetes en VirtualBox:



6. Esquema en Packet Tracer

Partimos de dos subredes, red interna y red main. La red main tendrá acceso a Internet y la red interna no. Del nodo 0 salen dos switches, cada uno con cada red distintas. Encontramos los servidores http y el balanceador de carga (Nginx) en la main, así como los hot-desks. En la red interna nos encontramos con el nodo de kubernetes.



7. Despliegue

Aquí solo hablaremos de la parte que nosotros hemos conseguido, dando pinceladas de lo que podríamos haber conseguido.

Lo primero que debemos hacer es tener un Ubuntu 20.04 con un servidor dhcp, tftp y http configurados de base que modificaremos con lo que hemos mencionado anteriormente. Antes hemos mostrado la configuración del nodo de kubernetes, ya que como solo hemos conseguido la instalación de pxe y la instalación de Ubuntu de forma atendida, todos los nodos tendrán la misma configuración. Lo único que cambia es la red interna, que puede ser la “main” o la “interna”. También necesitaremos una imagen iso de Ubuntu 20.04.

Destacar que nuestros servidores NO tienen por qué estar en un Ubuntu 20, pero en nuestro caso sí los tenemos así.

Primeramente, debemos descargar pxelinux y syslinux-common. Al servidor http moveremos la iso, a una carpeta que se encuentre en la ruta. Tras esto, creamos dos archivos, uno de meta-data (vacío) y otro de user-data (donde irá nuestra configuración de cloud-config)

```
tryhardkillua@nodo0:~$ cat /var/www/html/ubuntu2004/autoinstall/test/user-data
# cloud-config
autoinstall:
  version: 1
  storage:
    layout:
      name: direct
  locale: es_ES.UTF-8
  keyboard:
    layout: es
  identity:
    hostname: ubuntu2004
    password: "$6$exDY1mhS4KUYCE/2$zmn9To2wTKLhCw.b4/b.ZRTI2M30JZ4Qr0Q2a0XJ8yk96xpcCof0K
xKwuX1kqLG/ygbJ1f8wxED22bTL4F46P0"
    username: ubuntu
  ssh:
    allow-pw: true
    install-server: true
    package_update: false
    package_upgrade: false
tryhardkillua@nodo0:~$
```


Aquí estaría la instalación desatendida de Ubuntu, dejándolo perfectamente funcional. En nuestro caso, tras muchas horas de investigación perdidas, no hemos logrado que se realice la instalación automática, aunque sí una instalación normal.

Tras esto, moveremos los archivos `ldlinux.c32` y `pxelinux.0` a nuestro servidor TFTP. Montamos la iso y copiamos tanto el archivo `initrd` y `vmlinuz` en el servidor.

También crearemos aquí nuestro archivo de configuración de pxe. No entraremos más en detalle porque está explicado arriba. Del servidor DHCP se ha dicho todo arriba.

Una vez tengamos todos estos pasos, sería tan simple como encender el nodo destino con un arranque por red; este recibirá una dirección mediante el servicio DHCP y de esta manera podrá coger su archivo de configuración dependiendo de su mac. Tras esto, comenzará la instalación. Si la instalación desatendida hubiera proseguido, podríamos haber instalado los distintos servicios simplemente configurando el archivo de `cloud-init` para instalar el servicio que quisiéramos, para más tarde configurarlo con `ansible`. Esto sería todo lo que habríamos conseguido.

8. Problemas encontrados y aprendizaje

En este trabajo hemos encontrado un problema principal; la poca información (o al menos la que nosotros hemos encontrado). Todas las informaciones eran difusas y no llegamos a encontrar exactamente lo que queríamos hacer. Eso sí, nos ha ayudado a aprender mucho sobre pxe y su funcionamiento, así como sobre el despliegue automático. Esperamos que en un futuro próximo podamos lograr el objetivo de una instalación completamente desatendida.

9. Bibliografía

Pese a que hemos investigado en muchísimas páginas, no valdría la pena ponerlas ya que nos han sido de poca ayuda. Sin embargo, hay una página que nos ayudó mucho a saber cómo funcionaba todo el tema:

<https://butwt.wordpress.com/2020/06/29/pxe-booting-ubuntu-20-04-installer/>

También, para la configuración de servidores dhcp, tftp y http:

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>

<https://www.solvetic.com/tutoriales/article/9417-instalar-y-configurar-servidor-dhcp-en-ubuntu-21-04-y-20-04/>

<https://www.networkreverse.com/2020/05/how-to-install-tftp-server-on-ubuntu-20.04.html>