AS2 : Práctica Nº 2

Objetivo : Puesta en marcha de servicios distribuidos básicos, NTP y DNS, con la configuración de red y VMs necesarias.

Entrega: La entrega de la Práctica 2 se realizará a través de la tarea habilitada para tal efecto en Moodle, siendo la **fecha límite** para la **entrega el jueves 2 de Marzo** y la **evaluación el viernes 3 de marzo**.

Se deberá **entregar** una **memoria** en la que, al menos, se incluya:

1.- Resumen.
2.- Introducción y objetivos.
3.- Arquitectura de elementos relevantes.
4.- Comprehensión de elementos significativos de la práctica (máquinas, subredes, componentes e información de servicios distribuidos, recursos sistema, configuraciones).
5.- Problemas encontrados y su solución.

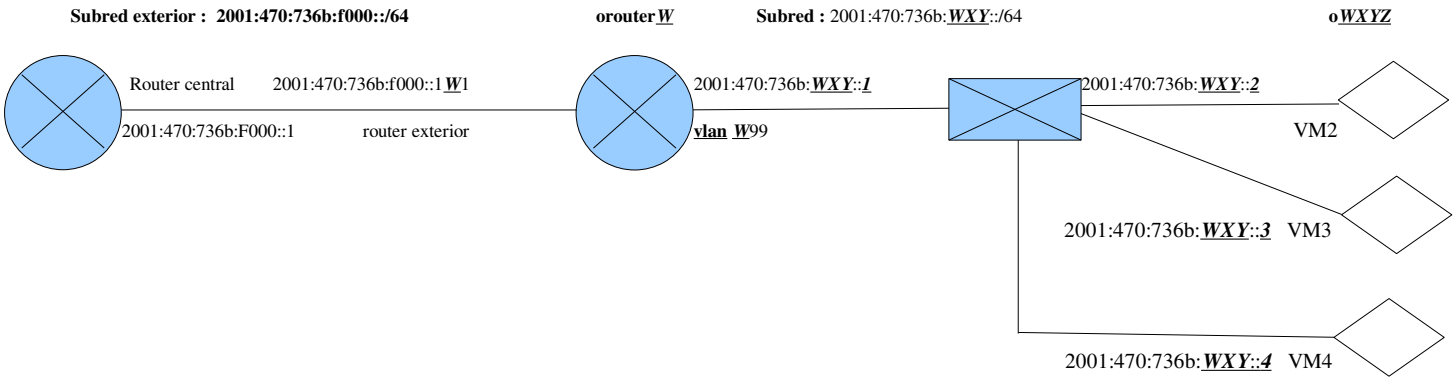
Adicionalmente, se deberá responder a la siguiente cuestión:

¿ Cuáles son los valores numéricos definidos en el registro SOA , qué significan y cuál es la utilidad de cada uno ?

Terminos a utilizar: vlans, subredes IP (y su prefijo correspondiente), servicios distribuidos, recursos sistema, configuraciones.

Enunciado : <u>La gran mayoría de la operativa debería hacerse por ssh.</u>

Configuracion de red y servidores



Variables :

W **:** <u>nº grupo seleccionado en moodle en Hexadecimal</u> (para MAC e IPv6). ***<u>Pasar a decimal</u>*** para VLAN (máximo 15)

<u>XY</u> : Designación hexadecimal de subred para cada práctica. Para **esta práctica <u>XY = FF</u>**

<u>Z</u> : Designación de máquinas específicas en subred :

* Z=1 para -router- ,
* Z=2 para VM2 : servidor ntp y DNS recursivo con cache (máquina pruebas práctica nº 1)
* Z=3 para VM3 : servidor DNS maestro (primario)
* Z=4 para VM4 : servidor DNS esclavo (secundario)

@ *MAC* VMs qemu : 52:54:00:0***W:XY:0Z***

Clientes DNS y NTP

Configurar, como clientes DNS (resolv.conf), todas las máquinas (router, servidor ntp, servidores DNS maestro y esclavo, y las de prácticas posteriores) para que las consultas DNS se envien al servidor DNS recursivo y con caché que vais a poner en marcha en la siguiente sección y opere en vuestro subdominio.

De forma similar, se debería configurar todas las máquinas como cliente ntp utilizando el servidor ntp que vais a poner a continuación.

Servicio de tiempo

Configurar el servicio de tiempo **ntp** de la siguiente forma :

- La máquina cliente de la 1ª práctica, VM2, la configurais como **servidor ntp** para todas vuestras VMs (salvo para el router exterior que se quedará con un servidor ntp externo). Su nombre DNS es **ntp1**, el cual deberá ser incluido en vuestra zona DNS (y por lo tanto en vuestros servidores DNS)**.**
  - Utilizar, como referencia los servidores de stratum 1: **2001:470:0:50::2** y **2001:470:0:2c8::2** y no os olvideis de activarle la escucha y aceptación de peticiones a los clientes (*listen on*).

- El resto de las máquinas, que utiliceis en todas las prácticas, se pondrán en funcionamiento como clientes ntp con las herramients que habeis puesto en marcha en la primera sección de esta práctica.

Comprobar que funciona el servidor comparando con otros relojes (lista en https://linuxreviews.org/IPv6-listening_NTP_servers) desde la máquina central de la siguiente forma :

```
central:~/ ntpdate -q ntp.time.nl
server 2a00:d78:0:712:94:198:159:10, stratum 1, offset -0.000483, delay 0.06357
server 2a00:d78:0:712:94:198:159:14, stratum 1, offset -0.000352, delay 0.06381
server 94.198.159.10, stratum 1, offset -0.001626, delay 0.07544
server 94.198.159.14, stratum 1, offset -0.001589, delay 0.07559
26 Feb 12:32:48 ntpdate[62348]: adjust time server 2a00:d78:0:712:94:198:159:10 offset -0.000483 sec
```

Y después lo mismo desde central, pero a vuestro servidor ntp1.

Comprobar al de un tiempo si se ha enviado a syslog algun mensaje de desviación de tiempo grande. Normalmente en daemon.log [.gz] (por si acaso "grep ntp /var/log/*").

Hacer que el script de arranque de ntp fije la hora de forma inmediata (mirar opción -s paginas man ntp en OpenBSD).

Servicio DNS

Crear 2 nuevas VMs diferenciales, VM3 y VM4, para poner en marcha un servidor DNS maestro y uno esclavo en cada uno de ellos. Se va a utilizar una configuración IPv6 estática (no automática) para ambas, con las @IP definidas en el esquema anterior. Es importante configurar correctamente estas @IP, ya que permitiran funcionar, correctamente, los "glue records" que estan definidos en el dominio "ff.es.eu.org".

Creareis un nuevo subdominio debajo del dominio "ff.es.eu.org". El nombre de vuestro subdominio es **vuestro nº de grupo**. Vuestro subdominio inverso es <nº grupo>.0.b.6.3.7.0.7.4.0.1.0.0.2, para los nº de

grupo de una cifra, y <cifra menor peso>.<cifra mayor peso>.b.6.3.7.0.7.4.0.1.0.0.2, para nº grupo de 2 cifras. Los 2 servidores DNS autorizados tendrán los nombres **ns1** (@IP con Z=3) y **ns2** (@ IP con Z=4), tanto en DNS como el hostname de cada máquina. El router **orouter*W*** tendrá nombre DNS **router1** que será incluido en vuestra base de datos DNS como ordenador en vuestra zona DNS. Podeis utilizar entradas CNAME para poner, además, los nombres de VM de cada fichero xml. Debereis incluir también los nombres DNS de vuestras máquinas clientes en vuestra zona DNS a lo largo de las prácticas.

Para ello tendreis que configurar el arranque de **nsd** (en rc.conf.local con nsd_flags), generar clave **nsd-control** con "**nsd-control-setup**" y configurar los ficheros : /var/nsd/etc/**nsd.conf** y los **2 ficheros** (de *resolución directa e inversa*) **de información de la zona.**

Además, configurareis un servidor DNS recursivo y con cache mediante **unbound**. Añadir la opción "forwarders", en unbound.conf, con la IPv6 "2001:470:20::2" del servidor público DNS de Hurricane Electric para que os pueda resolver las peticiones recursivas a servidores DNS que solo funcionen con IPv4. Recordar que todas nuestras VMs solo tienen direcciones Ipv6. También podeis utilizar la opción "stub-zone" para obtener una resolución local de vuestra zona directamente de vuestros servidores DNS con autoridad (que deberían tener nsd funcionando correctamente), asi no dependeis de todo el recorrido de internet para la resolución de vuestras zonas.

Incluir todos los aspectos de seguridad que contemplamos en clase para estos servidores DNS (restricciones de acceso, de tranferencia, etc), fundamentalmente a quien dejais solicitar peticiones recursivas DNS. Comprobar que dichos aspectos de seguridad funcionan correctamente.

El resto de máquinas, que utiliceis en todas las prácticas, se pondrán en funcionamiento como clientes DNS mediante el procedimiento realizado en la primera sección de esta práctica.

Ejemplos de configuración DNS para IPv6 :

* https://wikispaces.psu.edu/display/ipv6/IPv6+DNS
* https://www.nlnetlabs.nl/projects/nsd/
* https://calomel.org/nsd_dns.html
* https://www.sixxs.net/wiki/DNS_Configuration
* https://www.sixxs.net/faq/dns/?faq=reverse

Gestionar el servicio nsd con **nsd-control**.

Verificar configuraciones de servidor con **nsd-checkconf** y **nsd-checkzone** y como clientes con **dig** (o **host** en casos sencillos). también podeis utilizar **nds-control write**.

Para comprobar que, tanto la zona directa como la inversa, funcionan bien, solicitar la resolución al servidor DNS público de google (@ IP = 2001:4860:4860::8888) con comandos "host -6 ..." y "dig -6..." :

$ dig -6 @2001:4860:4860::8888 **AAAA** <nombre_máquina>.<vuestro_subdominio>.ff.es.eu.org
$ dig -6 @2001:4860:4860::8888 **-x** 2001:470:736b:Los64bitsDeVuestraMáquina

Tener en cuenta que hay servidores DNS en Internet que solo tienen IPv4 (como unizar), y por lo tanto no tendreis respuesta de algunos servidores ni de algunos nombres. Utilizar los servidores publicos de Hurricane electric o Google para resolver en el exterior de vuestro subdominio.

Tutorial sobre dig : http://www.thegeekstuff.com/2012/02/dig-command-examples/

Una vez que tengais configurados tanto los servidores DNS como los clientes, y hayais comprobado su funcionamiento, añadir las entradas de registro necesarios para resolver el nombre DNS de una **nueva**

máquina tanto en directo como en inverso, y comprobar su correcta resolución desde un cliente. El nombre a añadir es "otro_servidor" y su IP es 2001:470:736b:***WXY***::f

Referencias

https://www.nlnetlabs.nl/documentation/unbound/

https://calomel.org/unbound_dns.html

https://www.nlnetlabs.nl/documentation/nsd/

https://calomel.org/nsd_dns.html

Anexo A : Encendido remoto de máquinas

La gestión remota del estado hardware de un servidor es un requisito básico hoy en dia. Tecnologías como el estándar IPMI (y sus vriante propietarias, HP ILO, Dell IDRAC,...) para servidores, o Intel AMT para desktop, permiten controlar remotamente el estado del hardware de un ordenador. Conocer su estado y ejecutar acciones sobre él, siendo las más básicas el encendido y apagado eléctrico de la máquina.

Nosotros no vamos a tener acceso a esas tecnologías, sino que vamos a utilizar un método más básico para el encendido y apagado como es el *wakeonlan* y el comando *shutdown*.

Wakeonlan es un mecanismo disponible en tarjetas de red ethernet que permite, mediante una secuencia de tramas que reciban, dar la orden del encendido eléctrico a una máquina apagada, pero que esta conectada eléctricamente. Para ello, el envio de dichas tramas debe ser efectuado en el mismo segmento ethernet (por cuanto que lo que se envían son tramas, no paquetes IP).

La disponibilidad wakeonlan en una tarjeta ethernet es posible identificarla en linux mediante el programa *ethertool*.

En linux, programas que permiten enviar dichas tramas de encendido son powerwake, wakeonlan, etc :

https://help.ubuntu.com/community/WakeOnLan

Para vuestras prácticas, existe la opción de utilizar el servidor centos del laboratorio 1.02, *central.cps.unizar.es*, al cual os podeis conectar por ssh, para encender máquinas de dicho laboratorio mediante el comando :

$ wakeonlan <@MAC>

Posteriormente el apagado lo podréis hacer, en la máquina que habeis encendido, con el comando **sudo shutdown** para el que teneis privilegios de ejecución.

Pero previamente a la utilización de cualquiera de los 2 comandos debeis primero verificar que no molestais a nadie que use la máquina. En el caso del encendido, primero comprobas con un **ping** (quizas esten en windows) y posteriormente con un **ping ssh** de la herramienta **u** que habeis desarrolado en la 1ª práctica. Si no responde ninguno de los 2 podeis asumir que la máquina está apagada.

En el caso del apagado, primero teneis que comprobar, con el comando Unix **who**, que no hay otro usuario conectado en sesión a la misma máquina, y que la este utilizando. Si es fuera de horas lectivas, el último que salga de sesión es responsable del apagado de la máquina.

La lista de @ MAC, utilizable con wakeonlan, de los ordenadores del laboratorio 1.02 de prácticas es la siguiente (subred 155.210.154/24) :

191: 00:10:18:80:6b:36

192: 00:10:18:80:6e:27
193: 00:10:18:80:6e:1f
194: 00:10:18:80:67:f3
195: 00:10:18:80:73:46
196: 00:10:18:80:6c:72
197: 00:10:18:80:6e:33
198: 00:10:18:80:67:f0
199: 00:10:18:80:72:ed
200: 00:10:18:80:70:46
201: 00:10:18:80:6b:b3
202: 00:10:18:80:67:80
203: 00:10:18:80:67:83
204: 00:10:18:80:6c:6e
205: 00:10:18:80:67:94
206: 00:10:18:80:67:f4
207: 00:10:18:80:73:38
208: 00:10:18:80:67:84
209: 00:10:18:80:74:15
210: 00:10:18:80:74:13

Anexo B : Gestión remota de VMs con libvirt, virsh, virt-manager

Como recordatorio de lo que ya comentamos en el tema 1, libvirt permite la gestión remota de VMs, tanto a través de *virsh* como de *virt-manager* mediante la opción *connect*. Ambos funcionan bien, incluso con latencias de 50 a 150 ms. Los podeis utilizar para gestionar la VMs del laboratorio de forma remota trás el encendido de la máquina con *wakeonlan*. Recordar apagar correctamente las VMs, antes de apagar la máquina física con *shutdown -h*.

Ejemplo : *$ virsh -c qemu+ssh://a252525@155.210.154.197/system list --all*