Proyecto

parte 1

Asignatura: Administración de sistemas 2
Fecha: 19/5/2020
Autor: Diego Marco Beisty, 755232

RESUMEN

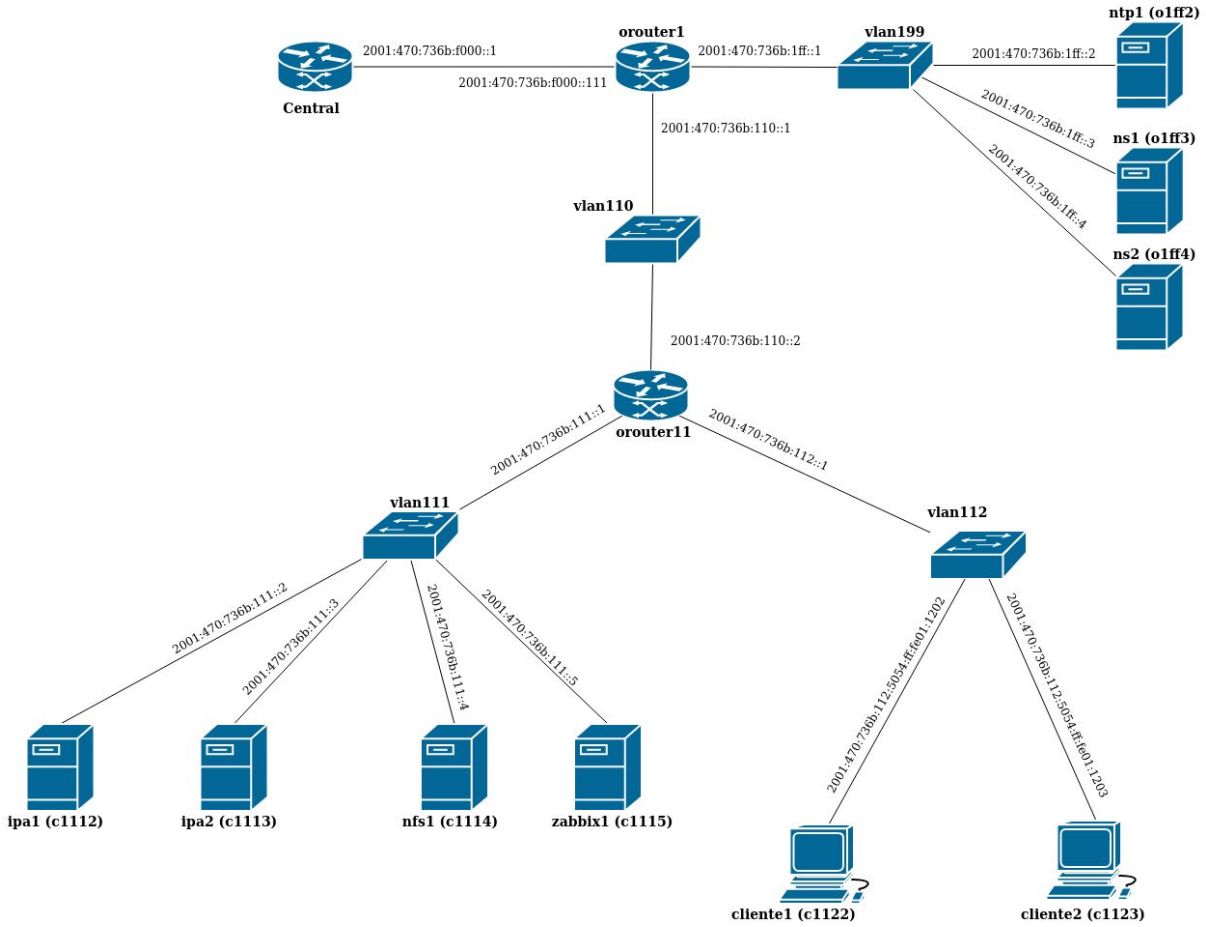
A partir de la arquitectura de la tercera práctica de la asignatura, se elimina la subred vlan 198 y se crea una nueva subred vlan 110 formada por un router interno y otras dos subredes vlan 111 y vlan 112 en la que se configuran cuatro servidores y dos clientes respectivamente.
Dos servidores van a levantar un dominio freeIPA/DNS que cuelga del dominio DNS establecido en prácticas anteriores. Se va a aprovechar la integración de servicios NTP, LDAP, Kerberos y DNS que supone freeIPA para sincronizar los relojes de las máquinas, centralizar cuentas de usuarios, cifrar las comunicaciones del sistema y ofrecer un servicio de resolución de nombres a los integrantes del dominio.
Además un tercer servidor se encargará de centralizar y compartir los directorios de los usuarios en las máquinas clientes mediante NFSv4.
Por último, el cuarto servidor se encargará de monitorizar el estado del sistema (CPU, memoria,...) mediante el servicio zabbix.

INTRODUCCIÓN Y OBJETIVOS

El despliegue del sistema se ha hecho por capas de abstracción.
Primero se han creado las máquinas virtuales a partir de sus respectivas imágenes base. Concretamente los servidores y clientes a partir de centOS y el router interno a partir de openBSD.
Posteriormente se ha configurado la red de todo el sistema, apoyándolo con los servicios básicos de la subred vlan 1FF para la descarga de paquetes como traceroute para verificar la correcta puesta en marcha de la red.
Finalmente se han configurado los glue records y demás servicios en los respectivos servidores(freeIPA,NFSv4,zabbix) y clientes(autoFS,automounter,zabbix-agent).

El principal objetivo de este proyecto ha sido la configuración de un dominio freeIPA y la kerberización de todo el sistema para asegurar comunicaciones cifradas.
Puesto que centOS no se había utilizado antes, también ha sido un objetivo mejorar el manejo con este sistema operativo.

ARQUITECTURA DE ELEMENTOS RELEVANTES



En el diagrama quedan representados de arriba a abajo los siguientes elementos:

- El router Central que conecta las MVs a internet.
- orouter1, que conecta la subred externa 2001:470:736b:f000::/52 con las subredes vlan 199 2001;470:736b:1ff::/64 y vlan 110 2001:470:736b:110::/60.
- o1ff2(nombre DNS ntp1), es el servidor NTP y unbound.
- o1ff3(nombre DNS ns1), es el servidor primario DNS del dominio 1.ff.es.eu.org
- o1ff4(nombre DNS ns2), es el servidor secundario DNS del dominio 1.ff.es.eu.org
- orouter11 conecta la subred vlan 110, 2001:470:736b:1ff::/64 con las subredes vlan 111, 2001:470:736b:111::/64 y vlan 112, 2001:470:736b:112::/64.
- c1112(nombre DNS ipa1), es el servidor primario del dominio freeIPA/DNS 1.1.ff.es.eu.org.
- c1113(nombre DNS ipa2), es el servidor secundario del dominio freeIPA/DNS 1.1.ff.es.eu.org
- c1114(nombre DNS nfs1, es el servidor NFSv4.
- c1115(nombre DNS zabbix1), es el servidor de monitorización zabbix.
- c1122(nombre DNS cliente1), es el cliente 1 del sistema.
- c1123(nombre DNS cliente2), es el cliente 2 del sistema.

Cabe destacar que las 4 máquinas servidores y las 2 máquinas cliente tienen sistema operativo centOS y el router interno “orouter11” tiene S.O openBSD.

COMPREHENSIÓN DE ELEMENTOS RELEVANTES

Configuración de red

orouter1

Se ha partido de la configuración de la práctica 3 en la que la tarjeta vio0 está activa y la subred vlan 1ff está activa. Para añadir la nueva subred vlan 110 se ha creado el fichero /etc/hostname.vlan110:

```
inet6 alias 2001:470:736b:110::1 64 vlan 110 vlandev vio0
!route add -inet6 2001:470:736b:110::/60 2001:470:736b:110::2
-autoconfprivacy
```

En la primera línea se observa la dirección ip de orouter1 en esta nueva subred junto con su máscara de red. Se indica también que la tarjeta vlan se crea sobre la tarjeta “física” vio0.
En la segunda línea se ha añadido una ruta estática para encaminar correctamente los paquetes que van dirigidos a las subredes vlan 111 y vlan 112.
De esta forma, primero se aplicará la máscara 64 y si coincide con la subred 2001:470:736b:110:: se encaminará hacia esta, pero si no coincide, se aplicará la máscara 60. Como resultado si con esta máscara la subred destino coincide con 2001:470:736b:110:: entonces se encaminará el paquete al siguiente salto, es decir, al router orouter11.

orouter11

Inicialmente se deja la tarjeta física vio0 únicamente activa, sin ningún tipo de configuración ip. /etc/hostname.vio0:

```
-inet6
up
-autoconfprivacy
```

Posteriormente se activa la interfaz vlan110 sobre vio0 para conectar con orouter1. /etc/hostname.vlan110:

```
inet6 alias 2001:470:736b:110::2 60 vlan 110 vlandev vio0
  -autoconfprivacy
```

Se ha asignado la dirección estática 2001:470:736b:110::2 para esta subred. Cabe destacar que en este caso no hace falta añadir otra ruta estática puesto que si la ip destino pertenece a otra red distinta a 2001:470:736b:110::, se redirige al encaminador por defecto, que es 2001:470:736b:110::1, (orouter1).
Se ha activado el parámetro ip forwarding en /etc/sysctl.conf para que encamine paquetes.
Después de verificar la conexión con orouter1, se ha procedido a configurar las vlan 111 y vlan 112 para crear la subred de servidores y la de clientes, respectivamente. Se ha segido el mismo procedimiento.
/etc/hostname.vlan111:

```
  inet6 alias 2001:470:736b:111::1 64 vlan 111 vlandev vio0
     -autoconfprivacy
```
/etc/hostname.vlan112:
```
   inet6 alias 2001:470:736b:112::1 64 vlan 112 vlandev vio0
    -autoconfprivacy
```

Donde se observa que le ha asignado las dirección estáticas 2001:470:736b:111::1 para la subred de servidores, y 2001:470:736b:112::1 para la subred de clientes.
Por último se ha activado el servicio rad para servir los prefijos de la subred vlan 112 para poder configurar IPs dinámicas en los clientes.

Servidores ipa1, ipa2, nfs1, zabbix1
El sistema operativo de estas máquinas es centOS y se les va a asignar una ip estática en le subred vlan 111, 2001:470:736b:111::/64.
Primero se ha eliminado el servicio NetworkManager puesto que la configuración de red se va a realizar con los scripts de red por defecto del sistema.
La tarjeta física se ha configurado para estar únicamente activa, sin tener asignada ninguna dirección IP. Para ello, en /etc/sysconfig/network-scripts/ifcg-eth0, se han añadido parámetros relevantes como el tipo de la tarjeta “TYPE=Ethernet”. Se ha indicado que no se quiere añadir ninguna configuración IPv6 “IPV6INIT=no”. Se ha indicado que se active en el arranque con “ONBOOT=yes”.
Además en el fichero /etc/sysctl.conf se han modificado los parámetros del kernel use_tempaddr, autoconf y accept_ra a cero, para asegurar que no se configura ninguna dirección IP en la tarjeta eth0.
Por último se ha configurado la tarjeta vlan 111 sobre vio0 con una IPv6 estática.
En el fichero /etc/sysconfig/network-scripts/ifcg-eth0.111 se han añadido parámetros como “VLAN=yes”, indicando que es una tarjeta virtual. “IPV6INIT=yes” indicando la configuración de una dirección IPv6. “IPV6ADDR=2001:470:736b:111::2” para designar la IP. Además se ha generado un UUID para la tarjeta virtual mediante el comando uuidgen.

Clientes cliente1, cliente2
Se les ha asignado direcciones dinámicas siguiendo casi los mismos pasos que en la configuración de red de los servidores.
Concretamente, en la configuración de la tarjeta vlan 112, en el fichero /etc/sysconfig/network-scripts/ifcg-eth0.112 se ha añadido el parámetro “IPV6_AUTOCONF=yes”. De esta forma los clientes pueden recibir los prefijos de subred del servidor rad en la máquina orouter11 y generar así sus propias IPs.
El cliente1 ha generado la ip: 2001:470:736b:112:5054:ff:fe01:1202
El cliente2 ha generado la ip: 2001:470:736b:112:5054:ff:fe01:1203
Al igual que en los servidores, se han modificado los parámetros del kernel en el fichero sysctl para asegurar que la tarjeta eth0 no recibe ninguna asignación IP.

Glue records DNS

Hasta este proyecto únicamente se ha configurado un dominio DNS “1.ff.es.eu.org”, situado en la subred vlan 1ff.

Para poder posteriormente configurar un nuevo dominio descendiente del ya configurado, es necesario añadir los registros denominados “glue records” en las zonas directas e inversas del dominio “1.ff.es.eu.org” del servidor DNS primario ns1.
Sin estos cambios, cualquier máquina que pregunte por un nombre situado en un subdominio simplemente no obtendrá respuesta. Pero con los glue records, el servidor del dominio podrá responder con la IP del subdominio donde sí se encuentra el nombre DNS solicitado.
Fichero /var/nsd/zones/1.ff.es.eu.org.directo:

```
1       IN    NS    ipa1.1.1.ff.es.eu.org.
1       IN    NS    ipa2.1.1.ff.es.eu.org.
ipa1.1  IN    AAAA  2001:470:736b:111::2
ipa2.1  IN    AAAA  2001:470:736b:111::3
```

Fichero /var/zones/1.ff.es.eu.org.inverso:

```
1       IN    NS    ipa1.1.1.ff.es.eu.org.
1       IN    NS    ipa2.1.1.ff.es.eu.org.
```

Integración de servicios freeIPA

Requisitos
Antes de instalar y configurar freeIPA se han seguido varios pasos en cada máquina.
Se ha añadido en /etc/hostname el nombre de dominio de cada máquina.
Se ha añadido en /etc/hosts la dirección IP y el nombre DNS asociado.
En las máquinas ipa1 e ipa2 se ha configurado como servicio de resolución de nombres recursivos el servidor unbound de la vlan 1ff añadiendo al fichero /etc/sysconfig/network-scripts/ifcg-eth0.111 la línea: “DNS1=2001:470:736b:1ff::2”.
En las demás máquinas de las subredes vlan 111 y vlan 112 se ha configurado como servidores DNS las máquinas ipa1e ipa2. Para ello se ha añadido las siguientes líneas en el fichero /etc/sysconfig/network-scripts/ifcg-eth0.11X:

```
DNS1=2001:470:736b:111::2
DNS2=2001:470:736b:111::2
```

Se ha comprobado que esta configuración actualiza el fichero /etc/resolv.conf y se asignan por lo tanto los servidores freeIPA como servidores DNS.
Respecto al servicio NTP, se ha configurado durante la instalación de freeIPA en las máquina ipa1 que utilice el servidor NTP de la subred 1ff como forwarder.
En las demás máquinas se han asignado como servidores NTP los servidores ipa1 e ipa2 en /etc/ntp.conf

Dominio 1.1.ff.es.eu.org
Tras instalar el servicio freeIPA en ipa1 se han añadido las zonas directas e inversa del nuevo dominio. Además se han añadido todos los registros necesarios para resolver los nombres de dominio de los servidores y clientes. Este paso es necesario para apoyar la operativa de Kerberos.

Cuentas de usuario freeIPA

Despueś de instalar el cliente freeIPA en el resto de máquinas, se ha añadido una cuenta para un usuario llamado “pepe”. Posteriormente se ha accedido a la cuenta desde la máquina cliente1 exitosamente.

Réplica freeIPA

Se ha configurado una réplica de freeIPA en la máquina ipa2 para garantizar tolerancia ante la caída del primario ipa1. Para ello se ha instalado el cliente freeIPA como las demás máquinas pero además se ha añadido al grupo de máquinas “ipaServers”.

Servicio NFS kerberizado

Operativa kerberos

Se han añadido seis principals correspondientes a los tres hosts (nfs1,cliente1,cliente2) y a los tres servicios NFS levantados sobre ellos.

```
ipa host-add nfs1.1.1.ff.es.eu.org
ipa host-add cliente1.1.1.ff.es.eu.org
ipa host-add cliente2.1.1.ff.es.eu.org
ipa service-add nfs/nfs1.1.1.ff.es.eu.org
ipa service-add nfs/cliente1.1.1.ff.es.eu.org
ipa service-add nfs/cliente2.1.1.ff.es.eu.org
```

Posteriormente se han generado las correspondientes claves de sesión asociadas a cada principal y se han guardado en el fichero /etc/krb5.keytab a nivel local de cada máquina:

```
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p host/nfs1.1.1.ff.es.eu.org -k /etc/krb5.keytab
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p nfs/nfs1.1.1.ff.es.eu.org -k /etc/krb5.keytab
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p host/cliente1.1.1.ff.es.eu.org -k /etc/krb5.keytab
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p nfs/cliente1.1.1.ff.es.eu.org -k /etc/krb5.keytab
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p host/cliente2.1.1.ff.es.eu.org -k /etc/krb5.keytab
ipa-getkeytab -s ipa1.1.1.ff.es.eu.org -p nfs/cliente2.1.1.ff.es.eu.org -k /etc/krb5.keytab
```

Exportación NFv4

Las comunicaciones entre el servidor nfs y los clientes ya está cifrada con Kerberos.
A continuación se va a configurar una exportación NFSv4 en el servidor nfs1.
Concretamente se va a exportar el directorio /srv/nfs4/home/ como un pseudo sistema de ficheros en el que se añadirán los directorios de los usuarios freeIPA.
En el fichero /etc/exports se añade la línea:

```
/srv/nfs4/home  2001:470:736b:112::0/64(rw,sync,fsid=0,sec=krb5,no_subtree_check)
```

Donde se indica en primer lugar la ruta del directorio que se exporta.
Después la subred a la que se da acceso al montaje de este pseudo sistema.
Por último las opciones de exportación entre las que se encuentran fsid=0, propia de nfsv4 y que indica que /srv/nfs4/home va a ser la raíz del pseudo sistema de ficheros que se monte en el cliente. Sec=krb5 indica que el montaje se va a realizar mediante Kerberos.

Montaje automático NFSv4

Antes de realizar el montaje automático se ha probado a hacerlo de forma manual con el comando `mount -v -t nfs4 -o sec=krb5 nfs1.1.1.ff.es.eu.org:/ /home`

Posteriormente se ha configurado el fichero /etc/auto.master añadiendo el punto de montaje /home y el fichero mapa /etc/auto.home donde se encuentran las especificaciones del montaje.

```
/home    /etc/auto.home
```

En el fichero /etc/auto.home se ha añadido la especificación de montaje:

```
 *    -rw,fstype=nfs4,sec=krb5          nfs1.1.1.ff.es.eu.org:/&
```

Con el asterisco se captura cualquier directorio que se acceda bajo /home y si coincide con un directorio existente "&" que se exporte en el servidor remoto entonces se montará en el cliente.
La opción fstype=nfs4 obliga a hacer un montaje sobre nfsv4.
La opcón sec=krb5 se incluye para que se monte utilizando kerberos.

Servicio zabbix

Requisitos

Este servicio se ha configurado sobre la máquina zabbix1.1.ff.es.eu.org.
Para instalar este servicio ha sido necesario instalar previamente el gestor de bases de datos mysql, php y el servicio httpd. Esto se debe a que necesita un frontend donde visualizar los datos registrados y una base de datos donde guardarlos.
Ha diferencia del servicio NFS, este servicio no se ha kerberizado por simplicidad.

Configuración

Inicialmente se ha instalado el servidor y el agente zabbix sobre la máquina zabbix1.
Se ha configurado una cuenta mysql de nombre zabbix y con privilegios de administrador sobre la base de datos.
Después se ha poblado la base de datos con las tablas y registros que necesita inicialmente zabbix para poder operar.
Se ha añadido la contraseña de la cuenta de usuario de mysql creada a varios archivos de configuración para que tenga acceso tanto el backend como el frontend.
Finalmente se ha accedido al servicio mediante un navegador con la url: zabbix1.1.1.ff.es.eu.org/zabbix y se han configurado las ips y dominios de las máquinas ip1, ipa2, nfs1, cliente1 y cliente2.
En estas máquinas se ha instalado el agente de zabbix y se les ha configurado la ip de zabbix1 como servidor zabbix.
Finalmente se ha elaborado un dashboard que refleje el CPU consumido, la memoria disponible, el ancho de banda ocupado, el espacio en disco y el estado de cada una de las máquinas.

PROBLEMAS ENCONTRADOS

En cada parte de este proyecto se han ido encontrando diversos problemas que se han solucionado.
En la etapa de configuración de red se detectó que Network Manager estaba activo y para evitar posteriores problemas por incompatibilidades con el sistema de red nativo, se decidió eliminarlo.

La instalación de freeIPA no se ejecutaba correctamente debido a problemas con el demonio certmonger. Se solucionó reiniciando el servicio dbus y posteriormente el servicio certmonger.

Al añadir las claves de sesión de los principals de kerberos, se ejecutaba el comando ipa-getkeytab con sudo pero el ticket (kinit admin) se ejecutaba sin sudo. Esto provocaba que no coincidieran las credenciales de la caché de las claves y no se pudiera ejecutar correctamente el comando ipa-getkeytab. Se solucionó ejecutando ambos comandos con sudo.

ANEXO

VERIFICACIÓN

1) Configuración de red:

- Primero se ha verificado que desde los servidores y clientes hay conectividad con el servidor ntp de la subred vlan 1ff. De esta forma se comprueba que los paquetes de los servidores y clientes salen por la vlan correspondiente (vlan 111 o vlan 112).
Para ello se ha ejecutado el comando traceroute en cada uno de ellos.
Máquina ipa1:

```
traceroute $o1ff2
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte packets
 1  gateway (2001:470:736b:111::1)  0.239 ms  0.226 ms  0.217 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  0.417 ms  0.409 ms  0.399 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  1.244 ms  1.240 ms  1.233 ms
```

Máquina ipa2:

```
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte
packets
 1  gateway (2001:470:736b:111::1)  0.406 ms  0.394 ms  0.383 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  0.499 ms  0.928 ms  0.923 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  1.302 ms  1.321 ms  1.337 ms
```

Máquina nfs1:

```
 traceroute $o1ff2
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte
packets
 1  gateway (2001:470:736b:111::1)  0.361 ms  0.347 ms  0.336 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  0.930 ms  0.924 ms  0.916 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  1.317 ms  1.309 ms  1.311 ms
```

Máquina zabbix1:

```
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte
packets
 1  gateway (2001:470:736b:111::1)  0.350 ms  0.338 ms  0.330 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  0.855 ms  0.848 ms  0.841 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  0.831 ms  1.088 ms  1.080 ms
```

Máquina cliente1:

```
traceroute 2001:470:736b:1ff::2
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte
packets
 1  2001:470:736b:112::1 (2001:470:736b:112::1)  0.370 ms  0.358 ms  0.350 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  0.481 ms  1.246 ms  1.240 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  1.475 ms  1.468 ms  1.463 ms
```

Máquina cliente2:

```
traceroute 2001:470:736b:1ff::2
traceroute to 2001:470:736b:1ff::2 (2001:470:736b:1ff::2), 30 hops max, 80 byte
packets
 1  2001:470:736b:112::1 (2001:470:736b:112::1)  0.301 ms  0.970 ms  0.963 ms
 2  2001:470:736b:110::1 (2001:470:736b:110::1)  1.479 ms  1.471 ms  1.462 ms
 3  ntp1.1.ff.es.eu.org (2001:470:736b:1ff::2)  1.453 ms  1.442 ms  1.434 ms
```

- Se ha verificado la conectividad con las máquinas desde central.
Para ello se ha usado el comando traceroute:
Máquina orouter11:

```
traceroute $orouter11
traceroute to 2001:470:736b:110::2 (2001:470:736b:110::2), 30 hops max, 80 byte
packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.339 ms  0.269 ms  0.213 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.386 ms  0.348 ms  0.484 ms
```

Máquina ipa1:

```
 traceroute $c1112
traceroute to 2001:470:736b:111::2 (2001:470:736b:111::2), 30 hops max, 80 byte
packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.412 ms  0.344 ms  0.290 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  1.211 ms  1.162 ms  1.115 ms
 3  2001:470:736b:111::2 (2001:470:736b:111::2)  0.660 ms  0.618 ms  0.543 ms
```

Máquina ipa2:

```
traceroute to 2001:470:736b:111::3 (2001:470:736b:111::3), 30 hops max, 80 byte
packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.829 ms  0.779 ms  0.730 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.766 ms  0.731 ms  0.685 ms
 3  2001:470:736b:111::3 (2001:470:736b:111::3)  1.113 ms  1.069 ms  1.024 ms
```

Máquina nfs1:

```
traceroute $c1114
traceroute to 2001:470:736b:111::4 (2001:470:736b:111::4), 30 hops max, 80 byte
packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.444 ms  0.378 ms  0.323 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.602 ms  0.546 ms  0.499 ms
 3  2001:470:736b:111::4 (2001:470:736b:111::4)  0.576 ms  0.524 ms  0.756 ms
```

Máquina zabbix1:

```
traceroute $c1115
traceroute to 2001:470:736b:111::5 (2001:470:736b:111::5), 30 hops max, 80 byte
packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.458 ms  0.391 ms  0.335 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.762 ms  0.726 ms  0.678 ms
 3  2001:470:736b:111::5 (2001:470:736b:111::5)  0.665 ms  0.617 ms  0.567 ms
```

Máquina cliente1:

```
traceroute $c1122
traceroute to 2001:470:736b:112:5054:ff:fe01:1202
(2001:470:736b:112:5054:ff:fe01:1202), 30 hops max, 80 byte packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.425 ms  0.367 ms  0.315 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.562 ms  0.520 ms  0.584 ms
 3  2001:470:736b:112:5054:ff:fe01:1202 (2001:470:736b:112:5054:ff:fe01:1202)
0.692 ms  0.648 ms  0.735 ms
```

Máquina cliente2:

```
traceroute $c1123
```

```
çtraceroute to 2001:470:736b:112:5054:ff:fe01:1203
(2001:470:736b:112:5054:ff:fe01:1203), 30 hops max, 80 byte packets
 1  2001:470:736b:f000::111 (2001:470:736b:f000::111)  0.529 ms  0.466 ms  0.410 ms
 2  2001:470:736b:110::2 (2001:470:736b:110::2)  0.804 ms  0.765 ms  0.717 ms
 3  2001:470:736b:112:5054:ff:fe01:1203 (2001:470:736b:112:5054:ff:fe01:1203)
0.736 ms  0.687 ms  0.638 ms
```

2) resolución DNS:

Se ha comprobado con el comando dig +trace que se accede correctamente al dominio 1.1.1.ff.es.eu.org y se resuelve la petición. Se muestra el comando dig +short para cada una de las máquinas.

Máquina ipa1:

Resolución directa

```
dig ipa1.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:111::2
```

Resolución inversa

```
dig -x 2001:470:736b:111::2  @8.8.8.8 +short
ipa1.1.1.ff.es.eu.org.
```

Máquina ipa2:

Resolución directa

```
dig ipa2.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:111::3
```

Resolución inversa

```
dig -x 2001:470:736b:111::3  @8.8.8.8 +short
ipa2.1.1.ff.es.eu.org.
```

Máquina nfs1:

Resolución directa

```
 dig nfs1.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:111::4
```

Resolución inversa

```
dig -x 2001:470:736b:111::4  @8.8.8.8 +short
nfs1.1.1.ff.es.eu.org.
```

Máquina zabbix1:

Resolución directa

```
dig zabbix1.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:111::5
```

Resolución inversa

```
dig -x 2001:470:736b:111::5  @8.8.8.8 +short
zabbix1.1.1.ff.es.eu.org.
```

Máquina cliente1:

Resolución directa

```
dig cliente1.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:112:5054:ff:fe01:1202
```

Resolución inversa

```
dig -x 2001:470:736b:112:5054:ff:fe01:1202  @8.8.8.8 +short
cliente1.1.1.ff.es.eu.org.
```

Máquina cliente2:

Resolución directa

```
dig cliente2.1.1.ff.es.eu.org AAAA @8.8.8.8 +short
2001:470:736b:112:5054:ff:fe01:1203
```

Resolución inversa

```
dig -x 2001:470:736b:112:5054:ff:fe01:1203  @8.8.8.8 +short
cliente2.1.1.ff.es.eu.org.
```

3) freeIPA:

Para comprobar que el servicio funciona correctamente se ha creado la cuenta de usuario “pepe” en la máquina ipa1 y se ha accedido a la cuenta desde la máquina cliente1:

```
ssh pepe@$c1122
Password:
Last login: Wed May 20 00:18:24 2020 from 2001:0:53aa:64c:c2b:493b:a7fc:36f5
-sh-4.2$ pwd
/home/pepe
```

Para comprobar que la kerberización funciona primero se ha obtenido el ticket de autenticación.

```
kinit admin
Password for admin@1.1.FF.ES.EU.ORG:
```

Después se ha procedido a cambiar la contraseña del usuario “pepe”.

```
ipa user-mod pepe --password
Contraseña:
Ingrese Contraseña nuevamente para verificar:
--------------------------------------
Ha sido modificado el usuario "pepe"
--------------------------------------
Ingreso de usuario: pepe
Nombre: Pepe
Apellido: Garcia
Directorio principal: /home/pepe
Shell de ingreso: /bin/sh
Nombre principal: pepe@1.1.FF.ES.EU.ORG
Principal alias: pepe@1.1.FF.ES.EU.ORG
Dirección de correo electrónico: pepe@1.1.ff.es.eu.org
UID: 122000001
GID: 122000001
Cuenta inhabilitada : False
Contraseña: True
Miembros de los grupos: ipausers
Claves Kerberos disponibles: True
```

Para comprobar que la réplica freeIPA opera correctamente se ha añadido otro usuario “antonio” desde la máquina ipa2:

```
[a755232@ipa2 ~]$ ipa user-add antonio  --password
Nombre: antonio
Apellido: garcia
Contraseña:
Ingrese Contraseña nuevamente para verificar:
----------------------------------
Ha sido agregado el usuario "antonio"
----------------------------------
Ingreso de usuario: antonio
Nombre: antonio
Apellido: garcia
Nombre y apellidos: antonio garcia
Mostrar nombre: antonio garcia
Iniciales: ag
Directorio principal: /home/antonio
GECOS: antonio garcia
Shell de ingreso: /bin/sh
Nombre principal: antonio@1.1.FF.ES.EU.ORG
Principal alias: antonio@1.1.FF.ES.EU.ORG
User password expiration: 20200520130822Z
Dirección de correo electrónico: antonio@1.1.ff.es.eu.org
UID: 122000005
GID: 122000005
Contraseña: True
Miembros de los grupos: ipausers
Claves Kerberos disponibles: True
```

4) NFSv4:

Se ha ejecutado exportfs -v en el servidor nfs1.1.1.ff.es.eu.org para comprobar que se exporta el directorio correctamente.

```
/srv/nfs4/home
2001:470:736b:112::0/64(sync,wdelay,hide,no_subtree_check,fsid=0,sec=krb5,rw,secure
,root_squash,no_all_squash)
```

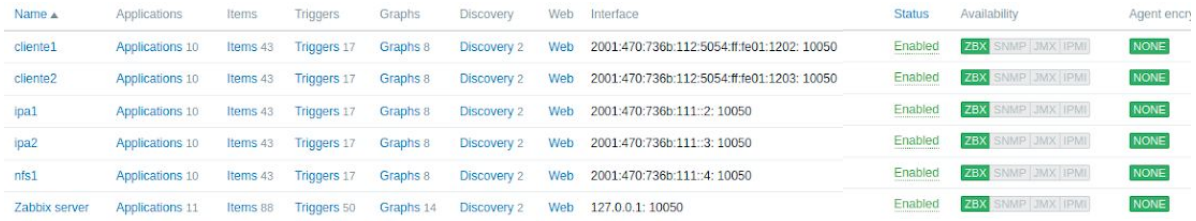
Se ha ejecutado el comando showmount -e nfs1.1.1.ff.es.eu.org desde cliente1 y cliente2 para comprobar que tienen acceso a la exportación:

```
Export list for nfs1.1.1.ff.es.eu.org:
/srv/nfs4/home 2001:470:736b:112::0/64
```

Se ha verificado que el automontaje funciona correctamente haciendo cd en el directorio /home/pepe y comprobando que se montaba tras este comando.
También se han comprobado escrituras y lecturas en archivos incluidos en el directorio exportado para verificar que los permisos 755 son suficiente.

4) Zabbix:

Se ha comprobado que en el frontend de zabbix se verificase que todas las máquinas del sistema están siendo monitorizadas. Para ello se ha observado que el campo “availability” esté en verde para cada máquina.



Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Status	Availability	Agent encr
cliente1	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web	2001:470:736b:112:5054:ff:fe01:1202: 10050	Enabled	ZBX SNMP JMX IPMI	NONE
cliente2	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web	2001:470:736b:112:5054:ff:fe01:1203: 10050	Enabled	ZBX SNMP JMX IPMI	NONE
ipa1	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web	2001:470:736b:111::2: 10050	Enabled	ZBX SNMP JMX IPMI	NONE
ipa2	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web	2001:470:736b:111::3: 10050	Enabled	ZBX SNMP JMX IPMI	NONE
nfs1	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web	2001:470:736b:111::4: 10050	Enabled	ZBX SNMP JMX IPMI	NONE
Zabbix server	Applications 11	Items 88	Triggers 50	Graphs 14	Discovery 2	Web	127.0.0.1: 10050	Enabled	ZBX SNMP JMX IPMI	NONE

SCRIPTS USADOS

Como en prácticas anteriores, se ha automatizado la puesta en marcha y parada de las máquinas virtuales mediante el script vir.sh

```
#!/bin/bash
path=/misc/alumnos/as2/as22019/a755232/
maquinas=$(ls $path | egrep -v '^o.xml' |egrep -v '^u1604.xml' | egrep
-v 'c74.xml' | egrep '*.xml' | tr -d '.xml')

if [ $1 = d ];then
    for vm in $maquinas;do
       sudo virsh define $path${vm}.xml
    done
    sudo virsh list --all
    virt-manager
elif [ $1 = u ];then
    read -p "HAS APAGADO TODAS LAS MAQUINAS?? [yes|no]: " response
    if [ $response = "yes" ];then
       for vm in $maquinas;do
             sudo virsh undefine $vm
       done
      pkill virt-manager
      sudo virsh list --all
    else
      echo "Abortando..."
    fi
else
    echo "Usage: . vir [d|u] define or undefine vms in virsh"
fi
```