Administración de sistemas II

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Despliegue de sistemas distribuidos de configuración, seguridad, integración y almacenamiento

Francisco Javier Pizarro Martínez 821259

31/03/2023

ÍNDICE

OBJETIVOS

En esta práctica se buscan 2 objetivos principales:

- Implementar los sistemas distribuidos de configuración, seguridad, integración y almacenamiento.
- Segmentación de red en base a la distribución de la misma de forma esquematica.

Para ello se ha rediseñado la red anterior de forma que esta más organizada, además se han incluido nuevas subredes una para routers, otra para los nuevos servicios que van a dar soporte a los clientes y por último una para los clientes. La red cuenta con un nuevo router, así como con 3 máquinas CentOS que nos van a brindar los nuevos servicios, siendo estas:

- FreeIPA Master
- FreeIPA Réplica
- NFS Kerberizado

Para el correcto funcionamiento de todos estos servicios es necesaria la correcta configuración de la resolución inversa de nombres en el DNS así como en lo relativo a los Glue Records. Dentro de la zona de clientes existen 2 MVs CentOS con IPv6 dinámica.

Dentro de los objetivos definidos FreeIPA nos ofrece en una solución compacta configuración e integración, dentro de este esta Kerberos ofreciendonos seguridad y para aportar el almacenamiento se ha empleado NFS en su formato Kerberizado para una mayor seguridad en la red.

DISEÑO DE ALTO NIVEL

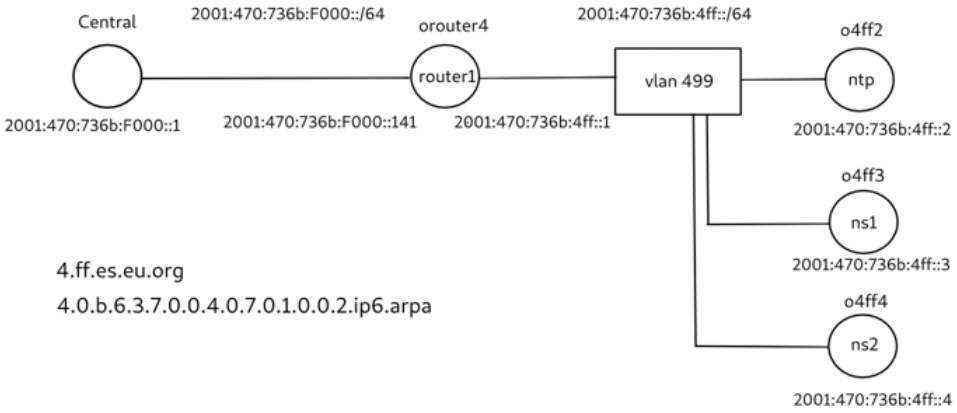
Dada la magnitud de la red a desarrollar en esta práctica se ha tenido que seguir un orden predeterminado de forma estricta para asegurar el correcto funcionamiento. El orden seguido ha sido el siguiente:

1. Realizar las modificaciones de la red previa.
2. Crear y configurar la MV router1 y comprobar su correcto funcionamiento.
3. Crear la imagen base de las MV CentOS.
4. Crear el cliente1 solo con la configuración general.
5. Crear y aplicar la configuración general básica sobre todas las MVs de la VLAN 411.
6. Configurar zona DNS IPA.
7. Configurar el servicio FreeIPA en la MV ipa y comprobar su correcto funcionamiento.
8. Configurar la replica del servicio FreeIPA en la MV ipa1 y comprobar su correcto funcionamiento desde el servidor maestro y desde el cliente.
9. Configurar el servicio NFS kerberizado en la MV nfs y comprobar su correcto funcionamiento desde la propia MV, así como desde las MV ipa y cliente.
10. Configurar el cliente FreeIPA en la MV cliente1 y comprobar el correcto funcionamiento tanto del cliente como del servidor IPA.
11. Crear y configurar el cliente2.
12. Comprobar el funcionamiento conjunto de toda la infraestructura montada.

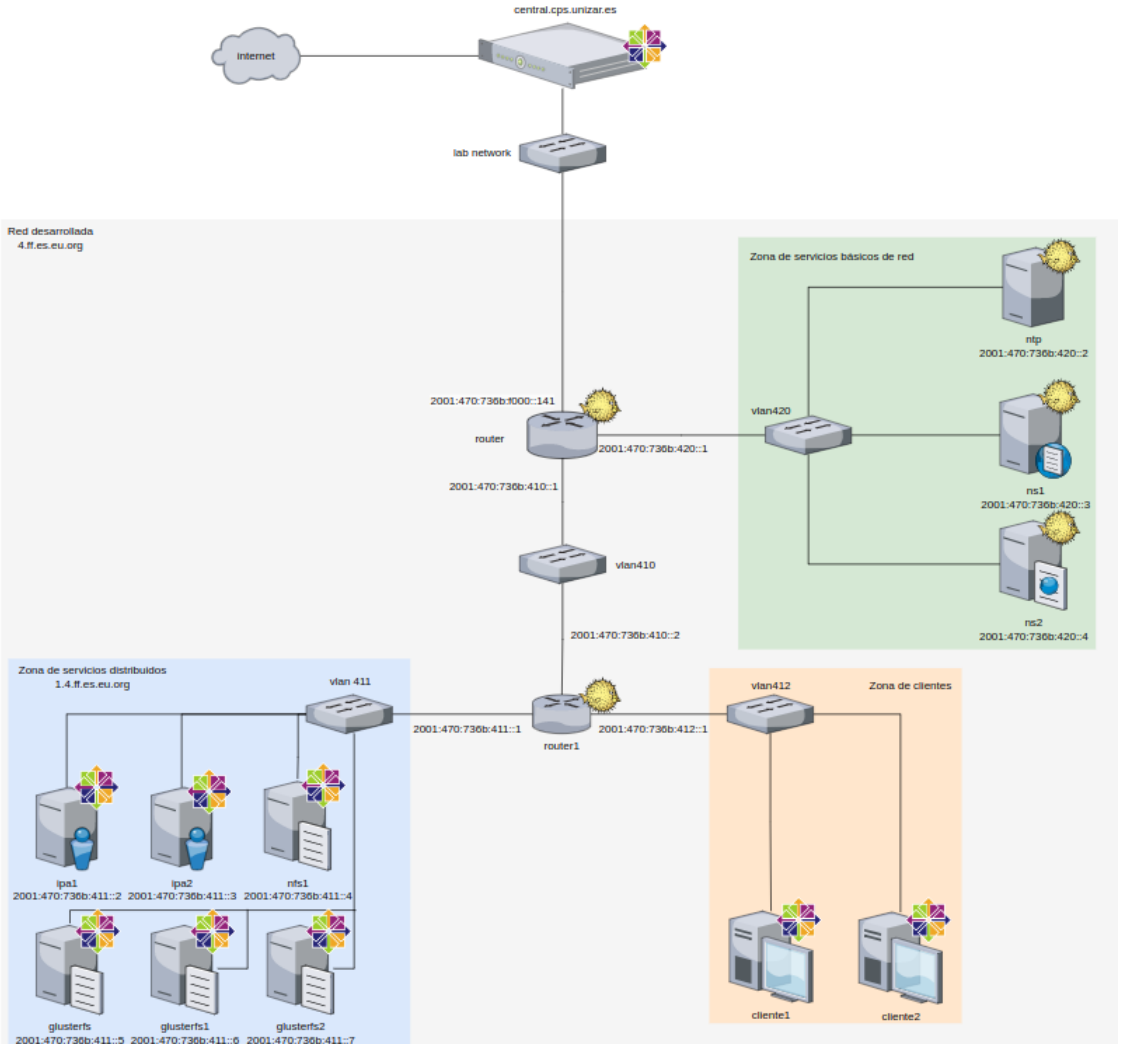
Los pasos problemáticos son a partir del paso 6 incluido este, de forma que además de realizarlos ha sido muy importante el uso de copias de seguridad para poder deshacerlos y rehacerlos de forma correcta.

MAPAS DE RED

Red original:



Red deseada:



IPAM

Dada la magnitud de la red es necesario tener las siguientes tablas para consultar detalles mas concretos sobre las MVs de los que ofrece el mapa de red.

TABLA IPAM ROUTERS

Nombre máquina	Nombre DNS	IPv6
Central	central.cps.unizar.es	2001:470:736b:f000::1
orouter4	router1.4.ff.es.eu.org	2001:470:736b:f000::141
orouter41	router2.4.ff.es.eu.org	2001:470:736b:410::2

TABLA IPAM SOPORTE BÁSICO DE RED

Nombre máquina	Nombre DNS	IPv6
orouter4	router1.4.ff.es.eu.org	2001:470:736b:420::1
o4ff2	ntp.4.ff.es.eu.org	2001:470:736b:420::2
o4ff3	ns1.4.ff.es.eu.org	2001:470:736b:420::3
o4ff4	ns2.4.ff.es.eu.org	2001:470:736b:420::4

TABLA IPAM DE SERVICIOS DISTRIBUIDOS DE CONFIGURACIÓN, INTEGRACIÓN, SEGURIDAD Y ALMACENAMIENTO

Nombre máquina	Nombre DNS	IPv6
orouter41	router2.4.ff.es.eu.org	2001:470:736b:411::1
ipa41	ipa.1.4.ff.es.eu.org	2001:470:736b:411::2
ipa42	ipa1.1.4.ff.es.eu.org	2001:470:736b:411::3
nfs41	nfs.1.4.ff.es.eu.org	2001:470:736b:411::4
glusterfs	glusterfs.1.4.ff.es.eu.org	2001:470:736b:411::5
glusterfs1	glusterfs1.1.4.ff.es.eu.org	2001:470:736b:411::6
glusterfs2	glusterfs2.1.4.ff.es.eu.org	2001:470:736b:411::7

MODIFICACIONES DE LA RED PREVIA

En este apartado se realizan los cambios pertinentes para modificar todo lo relativo a la VLAN 4ff y actualizarla a la 420 de forma que toda la red montada previamente siga funcionando en su totalidad.

MODIFICACIONES GENERALES

Estas se aplican sobre todas las MVs ya definidas mediante ssh de forma simultanea.

/etc/hostname.vlan499
inet6 2001:470:736b:0420::1 64

```
sed -i 's/2001:470:736b:4ff:/2001:470:736b:420:/g' /etc/{hostname.vlan499,ntpd.conf,resolv.conf,mygate}
cat /etc/resolv.conf > ./tmp.txt;echo 'search 4.ff.es.eu.org' >> ./tmp.txt;mv ./tmp.txt /etc/resolv.conf
# el comando de arriba es tan rebuscado porque el sed de la MV no cuenta con el comando \$ a y una simple
redirección ejecutada por ssh da errores de permisos :D
sh /etc/netstart
```

MODIFICACIONES MASTER DNS

Debemos actualizar tanto la configuración general como la configuración de las zonas para mantener el correcto funcionamiento en la nueva VLAN. Para ello ejecutamos los siguientes comandos:

```
sed -i 's/0.0.0.0.0.0.0.0.0.0.f.f/0.0.0.0.0.0.0.0.0.0.0.2/g' /var/nsd/zones/4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.
sed -i 's/2001:470:736b:4ff:/2001:470:736b:420:/g' /var/nsd/zones/4.ff.es.eu.org.directo

nsd-checkconf /var/nsd/etc/nsd.conf
nsd-checkzone 4.ff.es.eu.org /var/nsd/zones/4.ff.es.eu.org.directo
nsd-checkzone 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa. /var/nsd/zones/4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.


nsd-control reconfig
nsd-control reload 4.ff.es.eu.org
nsd-control reload 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.
nsd-control zonestatus 4.ff.es.eu.org
nsd-control zonestatus 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.

dig -6 @2001:470:736b:420::3 ns2.4.ff.es.eu.org
dig -6 @2001:470:736b:420::3 -x 2001:470:736b:420::1
```

MODIFICACIONES UNBOUND DNS

Debemos actualizar la configuración general para mantener el correcto funcionamiento en la nueva VLAN. Para ello ejecutamos los siguientes comandos:

```
sed -i 's/2001:470:736b:4ff:/2001:470:736b:420:/g' /var/unbound/etc/unbound.conf

unbound-checkconf
unbound-control reload

nslookup -type=AAAA router1.4.ff.es.eu.org 2001:470:736b:420::4
dig -6 @2001:470:736b:420::4 AAAA router1.4.ff.es.eu.org
dig -6 @2001:470:736b:420::4 -x 2001:470:736b:420::1
host -t PTR 2001:470:736b:420::1 2001:470:736b:420::4
dig -6 @2001:4860:4860::8888 AAAA router1.4.ff.es.eu.org
dig -6 @2001:4860:4860::8888 -x 2001:470:736b:420::1
```

Comprobamos que todos los ping funcionan en todas las máquinas:

- [x] ~~ping6 router1.4.20.es.eu.org.~~
- [x] ~~ping6 ntp.4.20.es.eu.org.~~
- [x] ~~ping6 ns1.4.20.es.eu.org.~~

- [x] ~~ping6 ns2.4.20.es.eu.org.~~

Con todo funcionando reiniciamos todas las máquinas para asegurarnos de no conservar ni IPs anteriores ni configuraciones obsoletas.

MODIFICACIONES ROUTER

Para crear la VLAN 410:
Creamos el fichero /etc/hostname.vlan410 y su contenido es el siguiente:

> vlan 410 vlandev vio0 up
> inet6 2001:470:736b:410::1
> inet6 -temporary
> inet6 -soii
> !route add 2001:470:736b:411::0/64 2001:470:736b:410::2
> !route add 2001:470:736b:412::0/64 2001:470:736b:410::2

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
sh /etc/netstart
```

Adicionalmente y de forma voluntaria se ha implementado un script en ERB, el cual es ejecutado cada minuto mediante cron y el cual sobreescribe con la información actualizada del estado de la red el archivo /etc/motd, de forma que al entrar por ssh en esta MV siempre vamos a conocer de forma actualizada el estado de la red, el script se encuentra en el ANEXO VI

CREACIÓN Y CONFIGURACIÓN ROUTER1

Debemos realizar los siguientes pasos:

1. Crear la MV.
2. Configurar la VLAN 410.
3. Configurar la MV como cliente de los servicios básicos de red ofrecidos por la VLAN 420, así como añadirla en los scripts elaborados.
4. Configurar la VLAN 411.
5. Configurar la VLAN 412 y su servicio rad.
6. AÑadir la MV al DNS para que sea localizable.

CREACIÓN MV

Ejecutamos los siguientes comandos:

```
qemu-img create -f qcow2 -o backing_file=o4.qcow2 orouter41.qcow2
chmod g+w orouter41.qcow2
cp o4ff2.xml orouter41.xml
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos la MV mediante el siguiente comando:

```
virsh -c qemu+ssh://a821259@155.210.154.207/system define orouter41.xml
```

CONFIGURACIÓN VLAN 410

Editamos el contenido del fichero /etc/hostname.vio0 y escribimos el siguiente:

> up
> -inet6

Creamos el fichero /etc/hostname.vlan410 y su contenido es el siguiente:

> vlan 410 vlandev vio0 up
> inet6 2001:470:736b:410::2
> inet6 -temporary
> inet6 -soii

En el fichero /etc/myname escribimos lo siguiente:

> orouter41

Añadimos en /etc/mygate

> 2001:470:736b:410::1

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
sh /etc/netstart
```

Realizamos las pruebas de ping para verificar que todo funciona que debería:

- [x] ~~ping6 orouter41 -> orouter4~~
- [x] ~~ping6 orouter4 -> orouter41~~

- [x] ~~ping6 orouter41 -> central~~
- [x] ~~ping6 central -> orouter41~~
- [x] ~~ping6 orouter41 -> ntp~~
- [x] ~~ping6 ntp -> orouter41~~

Escribimos en el fichero /etc/sysctl.conf el siguiente contenido:

> net.inet6.ip6.forwarding=1

Para aplicar los cambios y chequear si se han aplicado correctamente ejecutamos los comandos:

```
shutdown -r now
sysctl net.inet6.ip6.forwarding
```

SERVICIOS BÁSICOS DE RED

Configuramos la máquina para que emplee los servicios básicos que nos facilita la subred 420, siendo estos DNS,NTP y scripts creados previamente.

NTP

Comentamos los servidores del fichero /etc/ntpd.conf y escribimos lo siguiente:

> server 2001:470:736b:420::2

Ejecutamos el comando:

```
ntpd
```

Añadimos en el fichero /etc/rc.conf.local el siguiente contenido:

> ntpd_flags=-s

Comprobamos que funciona con el comando:

```
ntpctl -s all
```

DNS

Editar el fichero /etc/resolv.conf y escribir lo siguiente:

> nameserver 2001:470:736b:420::4
> search 4.ff.es.eu.org

Para aplicar los cambios ejecutar el comando:

```
sh /etc/netstart
```

Para comprobar que todo funciona correctamente ejecutar los comandos:

```
ping6 router1.4.20.es.eu.org
ping6 ipv6.google.com
```

SCRIPTS

Añadimos la nueva máquina en:

- Variable de entorno de central
- .u/hosts
- Scripts de arranque y apagado

CONFIGURACIÓN VLAN 411

Creamos el fichero /etc/hostname.vlan411 y su contenido es el siguiente:

> vlan 411 vlandev vio0 up
> inet6 2001:470:736b:411::1
> inet6 -temporary
> inet6 -soii

Ejecutamos el siguiente comando para reiniciar la configuración y aplicar todos los cambios:

```
sh /etc/netstart
```

Realizamos la siguiente prueba de ping para verificar que todo funciona que debería, las demás realizadas anteriormente no es necesario repetirlas dado que por transitividad funcionan:

- [x] ~~ping6 orouter4 -> orouter41(vlan411)~~

CONFIGURACIÓN VLAN 412

Ejecutamos el comando:

```
sed 's/411/412/g' /etc/hostname.vlan411 > /etc/hostname.vlan412
```

Ejecutamos el siguiente comando para reiniciar la configuración y aplicar todos los cambios:

```
sh /etc/netstart
```

Realizamos la siguiente prueba de ping para verificar que todo funciona que debería, las demás realizadas anteriormente no es necesario repetirlas dado que por transitividad funcionan:

- [x] ~~ping6 orouter4 -> orouter41(vlan412)~~

Modificamos el fichero /etc/rad.conf con el siguiente contenido:

> interface vlan412

Guardamos el fichero y ejecutamos el siguiente comando para activar el servicio rad:

```
rcctl enable rad
```

AÑADIR AL DNS

Dentro de la MV ns1 añadimos en /var/nsd/zones/4.ff.es.eu.org.directo

> router2 IN AAAA 2001:470:736b:410::2

Dentro de /var/nsd/zones/4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.

> 2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1 IN PTR router2.4.ff.es.eu.org.

Para comprobar y aplicar los cambios ejecutamos los comandos:

```
nsd-control reload 4.ff.es.eu.org
nsd-control reload 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.
nsd-control zonestatus 4.ff.es.eu.org
nsd-control zonestatus 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.
```

Para comprobar que funciona correctamente:

```
nslookup -type=AAAA router2.4.ff.es.eu.org 2001:470:736b:420::4
```

```
host -t PTR 2001:470:736b:410::2 2001:470:736b:420::4
```

CREACIÓN IMAGEN BASE CENTOS

En este apartado se crea la imagen base de CentOS siendo esta la empleada para crear posteriormente las MVs:ipa41,ipa42,nfs41,cliente41,cliente42. Para generar posteriormente dichas MVs se emplearan imagenes diferenciale spor lo que en esta solo debemos realizar unas pequeñas configuraciones generales, siendo estas la creación de un usuario, su añadido a sudo y la copia de la sshkey empleada para el acceso de las MVs.

Copiamos la imagen y configuración base, modificamos en la configuración el nombre, la dirección MAC y el file source.

Definimos la máquina en qemu:

```
virsh -c qemu+ssh://a821259@155.210.154.207/system define c74.xml
```

Añadimos permisos de escritura de grupo a c74.qcow2

Lanzamos la MV

Creamos nuestro usuario ejecutando los siguientes comandos:

```
useradd a821259 -g wheel
passwd a821259
```

Editamos el fichero /etc/sudoers.d comentando en el la linea que convierte a wheel en sudoer y descomentando la que lo convierte en sudoer sin necesidad de introducir contraseñas.

Ejecutamos los siguientes comandos para verificar que todo ha funcionado:

```
su a821259
sudo whoami
mkdir .ssh
touch .ssh/authorized_keys
```

Añadimos en el fichero /etc/sysctl.conf lo siguiente para deshabilitar ip del interfaz ethernet:

> net.ipv6.conf.eth0.use_tempaddr = 0
> net.ipv6.conf.eth0.autoconf = 0
> net.ipv6.conf.eth0.accept_ra = 0

Eliminamos la definición de la máquina base

```
virsh -c qemu+ssh://a821259@155.210.154.207/system undefine centOsBase
```

Le quitamos los permisos de escritura de grupo a c74.qcow2

CREACIÓN CLIENTE1

En este apartado realizamos la creación báswica del cliente1, siendo esta muy importante dado que en ella vamos a definir la VLAN y a configurarla como cliente de nuestro DNS, la importancia reside en que para agilizar el desarrolo todas las demás MVs de las VLAN 411 y 412 van a ser copias de esta.

Ejecutamos los siguientes comandos:

```
qemu-img create -f qcow2 -o backing_file=c74.qcow2 cliente41.qcow2
chmod g+w cliente41.qcow2
cp c74.xml cliente41.xml
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos las MV

```
virsh -c qemu+ssh://a821259@155.210.154.207/system define cliente41.xml
```

Ejecutamos

```
sysctl -p
modprobe 8021q
```

En el fichero /etc/hostname escribimos lo siguiente:

> cliente41.4.es.eu.ff.org

En el fichero /etc/sysconfig/network-script/ifcfg-eth0 cambiamos las siguientes lineas:

> IP6INIT="no"
> IP6_AUTOCONF="no"
> IP6_DEFROUTE="no"

Copiamos /etc/sysconfig/network-script/ifcfg-eth0 a /etc/sysconfig/network-script/ifcfg-eth0.412

> IPV6INIT="yes"
> IPV6_AUTOCONF="yes"
> NAME="VLAN 412"
> DEVICE="eth0.412"
> ONBOOT="yes"
> IPV6_DEFAULTGW="2001:470:736b:412::1"
> VLAN="yes"

Añadimos en /etc/mygate

> 2001:470:736b:411::1

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
/etc/init.d/networking restart
```

Realizamos las pruebas de ping para verificar que todo funciona que debería:

- [x] ~~ping6 cliente41 -> orouter41(411)~~
- [x] ~~ping6 cliente41 -> orouter4(410)~~

- [x] ~~ping6 cliente41 -> central~~
- [x] ~~ping6 cliente41 -> ntp~~

Para establecer que los clientes empleen el servidor NTP ya existente comentamos los servidores del fichero /etc/chrony.conf y escribimos lo siguiente:

> server 2001:470:736b:4ff::2

Ejecutamos los comandos:

```
systemctl restart chronyd.service
chronyc sources
```

CREACIÓN Y CONFIGURACIÓN DE MVS DE VLAN 411

En este apartado se realiza la creación de las MVs de la VLAN 411 así como su configuración básica de red

Ejecutamos los siguientes comandos:

```
cp cliente41.xml ipa41.xml
cp cliente41.qcow2 ipa41.qcow2

cp cliente41.xml ipa42.xml
cp cliente41.qcow2 ipa42.qcow2

cp cliente41.xml nfs41.xml
cp cliente41.qcow2 nfs41.qcow2
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos las máquinas:

```
virsh -c qemu+ssh://a821259@155.210.154.208/system define {ipa4{1,2},nfs41}.xml
```

Encendemos las máquinas y en cada una de ellas hacemos la siguiente modificación cambiamos el nombre de /etc/sysconfig/network-script/ifcfg-eth0.412 a /etc/sysconfig/network-script/ifcfg-eth0.411 y hacemos que este sea su contenido
sustituir Z por el valor correspondiente para cada MV

> IPV6INIT="yes"
> IPV6_AUTOCONF="no"
> IPV6ADDR="2001:470:736b:411::Z"
> NAME="VLAN 411"
> DEVICE="eth0.411"
> ONBOOT="yes"
> IPV6_DEFAULTGW="2001:470:736B:411::1"
> VLAN="yes"

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
/etc/init.d/networking restart
```

Pruebas realizadas(las demás funcionan por transitividad):

- [x] ~~ping6 ipa41 -> orouter4(410)~~
- [x] ~~ping6 orouter4 -> ipa41~~
- [x] ~~ping6 ipa42 -> orouter4(410)~~
- [x] ~~ping6 orouter4 -> ipa42~~
- [x] ~~ping6 nfs41 -> orouter4(410)~~
- [x] ~~ping6 orouter4 -> nfs41~~

Modificamos el fichero /etc/hostname escribimos lo siguiente:
siendo S el nombre de la MV

> S

Añadimos las nuevas máquinas en:

- Variable de entorno de central

- .u/hosts
- Scripts de arranque y apagado

CONFIGURACIÓN ZONA IPA DNS

DNS UNBOUND

Añadimos al fichero /var/unbound/etc/unbound.conf lo siguiente:

> forward-zone:
> name: "1.4.ff.es.eu.org."
> forward-addr: 2001:470:736b:411::2
> forward-first: yes

Ejecutamos el comando `unbound-control reload`

DNS MASTER

Añadimos al fichero /var/nsd/zones/4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa. lo siguiente:

> 2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa IN PTR ipa.1.4.ff.es.eu.org.
> 3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa IN PTR ipa1.1.4.ff.es.eu.org.
> 4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa IN PTR nfs.1.4.ff.es.eu.org.

Ejecutamos el comando `nsd-control reload 4.0.b.6.3.7.0.7.4.0.1.0.0.2.ip6.arpa.`

CONFIGURACIÓN IPA MASTER

En este apartado se realizan las modificaciones necesarias y se pone en funcionamiento el servicio freeIPA en la máquina ipa1.

Actualizamos el contenido del fichero /etc/hosts añadiendo

> 2001:470:736b:411::2 ipa.1.4.ff.es.eu.org ipa

Actualizamos el contenido del fichero /etc/hostname escribiendo

> ipa.1.4.ff.es.eu.org

Añadimos al fichero /etc/sysconfig/network

> HOSTNAME=ipa.1.4.ff.es.eu.org

Instalamos IPA con los siguientes comandos:

```
yum install freeipa-server
yum install freeipa-server-dns
ipa-server-install -p relativamentelargo -a relativamentelargo --domain-level 0
```

Durante la instalación elegir las siguientes opciones:

- DNS integrado:SI
- SERVERHostname: ipa.1.4.ff.es.eu.org
- Las contraseñas de directory manager y IPA admin son las mismas que las de root
- No dns forwarder
- No reverse zones

Comprobamos que esta correctamente instalado ejecutando los siguientes comandos:

```
kinit admin
ipa domainlevel-get
```

La salida debe ser la siguiente:

```
----------------------
Current domain level: 0
----------------------
```

Comprobamos que la salida del comando `ipactl status` es la siguiente:

```
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
ntpd Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

Comprobamos que la salida del comando `ipa user-find admin` es la siguiente:

```
--------------
1 user matched
--------------
  User login: admin
```

```
  Last name: Administrator
  Home directory: /home/admin
  Login shell: /bin/bash
  Principal alias: admin@1.4.FF.ES.EU.ORG
  UID: 86600000
  GID: 86600000
  Account disabled: False
----------------------------
Number of entries returned 1
----------------------------
```

CONFIGURACIÓN IPA EXCLAVO

Actualizamos el contenido del fichero /etc/hosts añadiendo

> 2001:470:736b:411::3 ipa1.1.4.ff.es.eu.org ipa1

Actualizamos el contenido del fichero /etc/hostname escribiendo

> ipa1.1.4.ff.es.eu.org

Añadimos al fichero /etc/sysconfig/network

> HOSTNAME=ipa1.1.4.ff.es.eu.org

Cambiamos el dns por defecto en /etc/resolv.conv por:

> 2001:470:736b:420::4

Ejecutamos el comando:

```
/ect/init.d/network restart
```

Ejecutamos los siguientes comandos en freeIPA master:

```
sudo ipa-replica-prepare ipa1.1.4.ff.es.eu.org
scp /var/lib/ipa/replica-info-ipa1.1.4.ff.es.eu.org.gpg a821259@ipa1.1.4.ff.es.eu.org:~/
```

Instalamos IPA con los siguientes comandos:

```
yum install freeipa-server
ipa-replica-install replica-info-ipa1.1.4.ff.es.eu.org.gpg
```

Para comprobar que efectivamente ha funcionado ejecutamos el comando `ipa dnsrecord-find 1.4.ff.es.eu.org.` en master y en replica y comprobamos que sus salidas son idénticas.

Para comprobar que todo se actualiza como es debido ejecutamos el comando `ipa dnsrecord-add 1.4.ff.es.eu.org. nfs.1.4.ff.es.eu.org. --aaaa-rec 2001:470:736b:411::4` en IPA MASTER y comprobamos con `ipa dnsrecord-find 1.4.ff.es.eu.org.` en la replica que se ha actualizado.

Como la replica no tiene por defecto el DNS ejecutamos:

```
yum install freeipa-server-dns
ipa-dns-install
```

Comprobamos que funciona mediante `dig -6 @2001:470:736b:411::3 nfs.1.4.ff.es.eu.org`.

CONFIGURACIÓN NFS

Actualizamos el contenido del fichero /etc/hosts añadiendo

> 2001:470:736b:411::4 nfs.1.4.ff.es.eu.org nfs

Actualizamos el contenido del fichero /etc/hostname escribiendo

> nfs.1.4.ff.es.eu.org

Añadimos al fichero /etc/sysconfig/network

> HOSTNAME=nfs.1.4.ff.es.eu.org

Cambiamos el dns por defecto en /etc/resolv.conv por:

> 2001:470:736b:420::4

Instalamos IPA con los siguientes comandos:

```
yum install ipa-client
yum install nfs-utils
ipa-client-install
kinit admin
ipa service-add nfs/nfs.1.4.ff.es.eu.org
ipa-getkeytab -s ipa.1.4.ff.es.eu.org -p nfs/nfs.1.4.ff.es.eu.org -k /etc/krb5.keytab
```

Comprobamos que todo ha funcionado de momento con `ipa service-show nfs/nfs.1.4.ff.es.eu.org`, la salida esperada es la siguiente:

```
Nombre principal: nfs/nfs.1.4.ff.es.eu.org@1.4.FF.ES.EU.ORG
Principal alias: nfs/nfs.1.4.ff.es.eu.org@1.4.FF.ES.EU.ORG
Keytab: True
Managed by: nfs.1.4.ff.es.eu.org
```

Ejecutamos `ipa-client-automount`.

En /etc/exports escribimos:

```
/home  *(rw,sec=krb5:krb5i:krb5p)
```

Ejecutamos:

```
systemctl enable nfs-server.service
systemctl start nfs-server.service
exportfs -rv
showmount -e

ipa automountmap-add-indirect default auto.home --mount=/home
ipa automountkey-add default auto.home --key "*" --info "nfs.1.4.ff.es.eu.org:/home/&"

ipa user-add --first a --last a --password --shell=/bin/bash
mkhomedir_helper a 077
touch /home/a/test.txt
chown a: /home/a/test.txt
```

Comprobamos que todo esta bien con los siguiientes comandos:

```
ipa automountlocation-find
ipa automountmap-find
ipa automountkey-find
```

CONFIGURACIÓN DEL CLIENTE 1

En este apartado se realiza la configuración del cliente1 como cliente de freeIPA

Cambiamos el dns por defecto en /etc/hostname por:

> cliente.4.ff.es.eu.org

Cambiamos el dns por defecto en /etc/resolv.conv por:

> 2001:470:736b:420::4

Ejecutamos el comando:

```
/etc/init.d/network restart
```

Ejecutamos el siguiente comando para poder instalar el cliente IPA:

```
yum install ipa-client
ipa-client-install --force-ntpd
ipa-client-automount
```

Comprobamos que la instalación es correcta:

```
ssh a@localhost
ls
```

CREACIÓN MVS GLUSTERFS

En este apartado realizamos la creación básica del glusternfs, se va a emplear para todas las demás MVs de GlusterFS.

Ejecutamos los siguientes comandos:

```
qemu-img create -f qcow2 -o backing_file=c74.qcow2 glusterfs.qcow2
chmod g+w glusterfs.qcow2
cp c74.xml glusterfs.xml
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos las MV

```
virsh -c qemu+ssh://a821259@155.210.154.205/system define glusterfs.xml
```

Ejecutamos

```
sysctl -p
```

En el fichero /etc/hostname escribimos lo siguiente:

> glusterfs.4.es.eu.ff.org

En el fichero /etc/sysconfig/network-script/ifcfg-eth0 cambiamos las siguientes lineas:

> IP6INIT="no"
> IP6_AUTOCONF="no"
> IP6_DEFROUTE="no"

Copiamos /etc/sysconfig/network-script/ifcfg-eth0 a /etc/sysconfig/network-script/ifcfg-eth0.411

> IPV6INIT="yes"
> IPV6_AUTOCONF="no"
> IPV6ADDR="2001:470:736b:411:5"
> NAME="VLAN 411"
> DEVICE="eth0.411"
> ONBOOT="yes"
> IPV6_DEFAULTGW="2001:470:736b:411::1"
> VLAN="yes"

Añadimos en /etc/mygate

> 2001:470:736b:411::1

En el archivo /etc/resolv.conf fijamos como servidor DNS la ip `2001:470:736b:420::4`

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
/etc/init.d/networking restart
```

Realizamos las pruebas de ping para verificar que todo funciona que debería:

- [x] ~~ping6 glusterfs -> orouter41(411)~~
- [x] ~~ping6 glusterfs -> orouter4(410)~~

- [x] ~~ping6 glusterfs -> central~~
- [x] ~~ping6 glusterfs -> ntp~~

Para establecer que los clientes empleen el servidor NTP ya existente comentamos los servidores del fichero /etc/chrony.conf y escribimos lo siguiente:

> server 2001:470:736b:4ff::2

Ejecutamos los comandos:

```
systemctl restart chronyd.service
chronyc sources
```

Apagamos la MV y ejecutamos los siguientes comandos desde central para crear sus copias:

```
cp glusterfs.qcow2 glusterfs1.qcow2
cp glusterfs.qcow2 glusterfs2.qcow2
cp glusterfs.xml glusterfs1.xml
cp glusterfs.xml glusterfs2.xml
chmod g+w glusterfs1.qcow2
chmod g+w glusterfs2.qcow2
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos las MV

```
virsh -c qemu+ssh://a821259@155.210.154.205/system define glusterfs1.xml
virsh -c qemu+ssh://a821259@155.210.154.205/system define glusterfs2.xml
```

Modificamos las ips internas de cada uno en el fichero `/etc/sysconfig/network-scripts/ifcfg-eth0.411` para que no se solapen entre si, cambiamos también el hostname.

Copiamos una de las imagenes para usarla para el 2º cliente con `cp glusterfs.qcow2 cliente42.qcow2`

Definimos las nuevas ips en los scripts de arranque y parada

Se ha tratado de configurar el sistema glusterfs mediante distintas guías obteniendo siempre como resultado un error en el proceso de instalación del servidor gluster.

CONFIGURACIÓN CLIENTE2

En este apartado realizamos la creación báswica del cliente2.

Ejecutamos los siguientes comandos:

```
qemu-img create -f qcow2 -o backing_file=c74.qcow2 cliente42.qcow2
chmod g+w cliente42.qcow2
cp c74.xml cliente42.xml
```

Editamos los xml modificando los siguientes flag:

- uuid
- nombre
- source
- MAC

Definimos la MV

```
virsh -c qemu+ssh://a821259@155.210.154.205/system define cliente42.xml
```

Ejecutamos

```
sysctl -p
modprobe 8021q
```

En el fichero /etc/hostname escribimos lo siguiente:

> cliente42.4.ff.es.eu.org

En el fichero /etc/sysconfig/network-script/ifcfg-eth0 cambiamos las siguientes lineas:

> IP6INIT="no"
> IP6_AUTOCONF="no"
> IP6_DEFROUTE="no"

Copiamos /etc/sysconfig/network-script/ifcfg-eth0 a /etc/sysconfig/network-script/ifcfg-eth0.412

> IPV6INIT="yes"
> IPV6_AUTOCONF="yes"
> NAME="VLAN 412"
> DEVICE="eth0.412"
> ONBOOT="yes"
> IPV6_DEFAULTGW="2001:470:736b:412::1"
> VLAN="yes"

Añadimos en /etc/mygate

> 2001:470:736b:411::1

Ejecutamos el siguiente comando para reiniciar y aplicar todos los cambios:

```
/etc/init.d/networking restart
```

Realizamos las pruebas de ping para verificar que todo funciona que debería:

- [x] ~~ping6 cliente42 -> orouter41(411)~~
- [x] ~~ping6 cliente42 -> orouter4(410)~~
- [x] ~~ping6 cliente42 -> central~~
- [x] ~~ping6 cliente42 -> ntp~~

En este apartado se realiza la configuración del cliente2 como cliente de freeIPA

Cambiamos el /etc/hostname por:

> cliente2.4.ff.es.eu.org

Cambiamos el dns por defecto en /etc/resolv.conv por:

> 2001:470:736b:420::4

Ejecutamos el comando:

```
/etc/init.d/network restart
```

Ejecutamos el siguiente comando para poder instalar el cliente IPA:

```
yum install ipa-client
ipa-client-install --force-ntpd
ipa-client-automount
```

Comprobamos que la instalación es correcta:

```
ssh a@localhost
ls
```

DESPLIEGUE REAL

Para simular un despliegue real se ha seguido la siguiente distribución de MVs para máquinas físicas

MVs de las VLANs 410 y 411 y máquinas NFS/GLUSTERFS en la 155.210.154.205
MVs de clientes en 155.210.154.204
IPA Master 155.210.154.203
IPA Replica 155.210.154.202

Pruebas realizadas:

- Acceso a las MVs relevantes
- Creación de usuario en NFS, acceso desde cliente1 creación de fichero y acceso al fichero desde cliente2 posteriormente

```
#en nfs
ipa user-add --first b --last b --password --shell=/bin/bash
mkhomedir_helper b 077
#en cliente1
echo 'a' > a.txt
#en cliente2
cat a.txt
```

- Comprobar acceso y funcionamiento en un cliente con IPA master caido.
- Comprobar que pasa al tratar de crear usuarios con IPA master caido

```
#en nfs
ipa user-add --first c --last c --password --shell=/bin/bash
# como cabía esperar lanza error de KDC no encontrado
```

ANEXO I

Nueva versión del script de ruby:

```
#!/usr/local/bin/ruby -w
require 'net/ping/tcp'
#require "net/ssh"
require 'optparse'

OptionParser.new do |opts|
  opts.banner = "Uso: u.rb [p | s 'comando en shell']"

  opts.on("-h", "--h", "-help", "--help") do |h|
    puts "Configurar previamente en ~/.u/hosts un fichero con una IP por linea
Uso: u.rb [p | s 'comando en shell'
Este programa con la opción p realiza un ping a dichas IPs
Con la opción s ejecuta via ssh el comando introducido en dichas IPs"
  end
end.parse!

f = File.open(ENV["HOME"] + "/.u/hosts")
ips = f.readlines.map(&:chomp)
f.close

case ARGV[0]
when "p"
    puts "Ejecutando ping sobre las máquinas definidas:"
    def accion(ip)
        t = Net::Ping::TCP.new("#{ip}",22,0.1) # puerto 22, timeout 0.02 s
        if t.ping?
            puts "#{t.host}:Funcionando con respuesta en #{t.duration} segundos"
        else
            puts "#{t.host}:Falla"
        end
    end

when "s"
    puts "Ejecutando el comando #{ARGV[1]} sobre las máquinas definidas via ssh"
    def accion(ip)
        t = Net::Ping::TCP.new("#{ip}",22,0.1) # puerto 22, timeout 0.02 s
        if t.ping?
            puts "Ejecutando en: #{t.host}"
            system "ssh -n a821259@#{t.host} #{ARGV[1]}"
        else
            puts "#{t.host}:Falla"
        end
    end

else
    puts "Opción #{ARGV[0]} es incorrecta emplea 'p' o 's'\n"
        system exit 1
end

for ip in ips do
    accion(ip)
end
```

La diferencia con la versión anterior es que esta ya no depende de la biblioteca ssh sino que emplea el ssh nativo del sistema, adicionalmente se ha creado una sshkey de tipo rsa en el router1 y se ha distribuido a todas las máquinas

ANEXO II

Scripts bash para automatizaciones:

- Para apagado remoto de las MVs:

```
#!/usr/bin/bash
for mv in 411::{4..2}
do
        ssh -6 -n a821259@2001:470:736b:${mv} "sudo shutdown -h now"
    echo "Apagando internamente la máquina $mv"
done
ssh -6 -n a821259@2001:470:736b:410::2 "doas shutdown -h now"
echo "Apagando internamente la máquina router41"
for mv in 420::{4..1}
do
        ssh -6 -n a821259@2001:470:736b:${mv} "doas shutdown -h now"
    echo "Apagando internamente la máquina $mv"
done
sleep 10
for mv in {o4ff{2..4},orouter4,orouter41}
do
    virsh -c qemu+ssh://a821259@155.210.154.${1}/system destroy $mv --graceful
    echo "Apagando forzadamente la máquina $mv"
done
```

- Para desdefinir las MVs de una máquina física:

```
#!/usr/bin/bash
for mv in {orouter4,o4ff{2..4},orouter41,ipa4{1..2},nfs41}
do
    echo "Eliminado la máquina $mv"
    virsh -c qemu+ssh://a821259@155.210.154.${1}/system undefine $mv
done
```

- Para definir las MVs en una máquina física:

```
#!/usr/bin/bash
for mv in /misc/alumnos/as2/as22022/a821259/{orouter4,o4ff{2..4},orouter41,ipa4{1..2},nfs41}.xml
do
    echo "Definiendo la máquina $mv"
    virsh -c qemu+ssh://a821259@155.210.154.${1}/system define $mv
done
```

- Para encender todas las MVs:

```
#!/usr/bin/bash
for mv in {orouter4,o4ff{2..4},orouter41,ipa4{1..2},nfs41}
do
    echo "Encendiendo la máquina $mv"
    virsh -c qemu+ssh://a821259@155.210.154.${1}/system start $mv
done
```

- Para corregir los permisos de las MVs si se corrompen:

```
#!/usr/bin/bash
for mv in /misc/alumnos/as2/as22022/a821259/{orouter4,o4ff{2..4},orouter41,ipa4{1..2},nfs41}.qcow2
```

```
do
    echo "Fixeando la máquina $mv"
        cp ${mv} ${mv}copia
        rm ${mv}
        mv ${mv}copia ${mv}
        chmod g+w ${mv}
done
```

ANEXO III

Metodología de resolución de problemas:

1. Buscar en las guías de autodocumentación los cambios desde la última práctica.
2. Revisar que los ficheros en la MV en cuestión tienen el contenido adecuado.
3. Emplear los comandos pertinentes para obtener el estado actual configurado.
4. Una vez localizado el problema buscar la posible causa.
5. Ejecutamos los comandos necesarios para solucionar el problema.
6. Comprobamos mediante los comandos necesarios que efectivamente el problema se ha resulto.

Solución alternativa: En caso de no encontrar el problema restaurar una versión anterior del sistema, la cual funcione, para comparar los ficheros y configuraciones.

Problemas encontrados a lo largo de la práctica:

- El dns que redirige las peticiones a nuestra zona desde fuera(google) sigue redirigiendo las peticiones a la IP6 anterior del dns es decir 2001:$70:736b:4ff::4 en cuenta de a la nueva 2001:470:736b:420::4, este problema escapa de nuestro control ya que lo deber arreglar el administrador del DNS que realiza los glue-records
- Tras cambias todas las MVs de máquina física a otra maquina, fallaban todas las comunicaciones con el exterior a pesar de que la red seguía siendo la misma, después de depurar con ayuda del comando netstat -r, el fallo resulto ser que en dicha máquina seguía presente el fallo de la primera práctica en la cual los paquetes enviados a traves de bridge-utils no llegaban a central dado que estos desapareción. La solución en este caso fue volver a cambiar de máquina física a una que si tenía el parche necesario para que esto no sucediese.
- Tras mover las MVs de maquina fisica con los comandos `define` o `undefine` se corrompen los permisos de escritura de la imagen qcow2, este problema se localiza al tratar de encender dichas MVs, la solución es simplemente corregir los permisos mediante `chmod`
- En la instalación de freeIPA en la máquina ipa1 ocurria el siguiente fallo:`Allocated memory error`, este fallo causo tantos problemas que esta en el ANEXO V

ANEXO IV

Variables de entorno definidas para agilizar el workflow desde central.cps.unizar.es en el fichero .bashrc

```
export as2Fol="/misc/alumnos/as2/as22022/a821259"
export subRedAS2="2001:470:736b:420::"
export router="${subRedAS2}1"
export ntp="${subRedAS2}2"
export ns1="${subRedAS2}3"
export ns2="${subRedAS2}4"
export router1="2001:470:736b:410::2"
export ipa="2001:470:736b:411::2"
export ipa1="2001:470:736b:411::3"
export nfs="2001:470:736b:411::4"
```

ANEXO V

Durante el proceso de instalación de freeIPA en la máquina ipa1 ocurría el siguiente fallo:`Allocated memory error`, tras buscar si era un problema de los parametros introducidos durante la configuración o un error de la MV sin mucho éxito se opto por comprobar en tiempo real que estaba ocurriendo tanto en el almacenamiento de la MV como en su memoria volátil.

Para ello se emplearon 2 terminales en dicha MV de forma concurrente, una para monitorizar los valores y otra para ejecutar el proceso de instalación, los comandos ejecutados son los siguientes:

```
watch -n 0.1 "free -h;df -h"

sudo ipa-server-install -p=relativamentelargo -a=relativamentelargo
```

Tras localizar el fallo, en este caso que se agota la memoria RAM disponible en cierto punto de la instalación simplemente para arreglarlo apagamos la MV, borramos su definición desde QEMU, duplicamos la memoria RAM disponible de la MV editando para ello el archivo ipa41.xml y redefinimos la máquina mediante QEMU.

Tras este fix el proceso de instalación se ejecuta de principio a fin sin ningún problema.

Dado que en ipa42 iba a suceder el mismo error en un futuro, para prevenirlo también realizamos el fix sobre ipa42.xml

```
watch --interval '0,1' 'free -h;df -h'
```

ANEXO VI

El script empleado para monitorizar mediante el uso del motd y de erb es el siguiente:

```
<% puts "ESTADO ACTUAL DEL SISTEMA:\n";File.foreach("/home/a821259/.u/hosts") { |ip| ip.chomp!;next if ip.empty?;
puts (system("ping6 -c 1 #{ip} >/dev/null 2>&1") ? "#{ip} - ALIVE" : "#{ip} - DEAD") ;}; %>

Ultima ejecucion <% puts Time.now %>
Se ha conectado a Router, el cual sirve como punto de monitorizacion y de ejecucion de comandos remotos.
```