



PROGRAMA MODULAR EN ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

PRÁCTICO

Proyecto

Título: PERRAURI EN PI

Autor/a: Rafael Jiménez Camacho

Tutor/a: Sergio Martín Gutierrez

Curso académico: 2020 - 21



Agradecimientos

Quiero dedicar este proyecto a mi abuela, gracias a ella descubrí el maravilloso mundo de la informática con ocho años, dejándome enfrente de un salvapantallas para Mac, para mí fue como descubrir un universo.

A mis padres por regalarme un Spectrum y dejarlo en mi cuarto, todo lo que pude trastear y asentar mi pasión por la informática.

A mi pareja, Laura, porque sin su incondicional apoyo, amor y su energía yo nunca habría llegado hasta aquí.

Por último, a mis hijos, por dadme las fuerzas que necesito cuando me faltan con una simple sonrisa.

Sin todo esto no existiría tal como soy y como entiendo este maravilloso mundo de la informática.

Índice de contenido

1.	Resumen/Summary	VI
1.1.	Resumen	VI
1.2.	Summary.....	VI
2.	Antecedentes.....	VII
3.	Objetivos generales y específicos y alcance del proyecto	VIII
3.1.	Objetivos generales	VIII
3.2.	Objetivos específicos	VIII
3.3.	Alcance del proyecto	VIII
4.	Definiciones	IX
5.	Desarrollo del trabajo final	1
5.1.	Adquisición y montaje de Raspberry Pi 4B	3
5.1.1.	Estudio comparación y adquisición Raspberry Pi 4B	4
5.1.2.	Montaje de componentes	5
5.2.	Puesta en marcha con Ubuntu.....	6
5.2.1.	Instalación del sistema operativo Ubuntu.....	6
5.2.2.	Actualización del sistema	7
5.2.4.	Instalación de paquetes adicionales	11
5.2.5.	Dynamic DNS - DDNS.....	12
5.3.	Creación contenedores virtualización ligera con Docker	14
5.3.1.	Introducción a Docker	14
5.3.2.	Creación contenedores VM con Docker	15
5.4.	Gestión contenedores con Portainer	19
5.4.1.	Introducción a Portainer	19
5.4.2.	Gestión contenedores con Portainer	20
5.5.	Monitorización de estadísticas con NetData	22
5.5.1.	Introducción a NetData	22
5.5.2.	Configuración NetData	23
5.6.	Creación contenedores página web y configuración	26
5.6.1.	Base de Datos MariaDB.....	30

5.6.2.	Gestor PhpMyadmin	31
5.6.3.	Wordpress	33
5.7.	Diseño página Web	34
5.8.	Creación y configuración VPN con OpenVPN	36
5.8.1.	Servidor VPN OpenVPN.....	36
5.8.2.	Clientes VPN	38
5.9.	Creación y configuración almacenamiento en nube Nextcloud.....	42
5.9.1.	Configuración clientes Nextcloud.....	45
5.9.1.1.	Configuración cliente Windows.....	45
5.9.1.2.	Configuración clientes Android	46
5.10.	Creación copia de seguridad a Pendrive	47
5.10.1.	Configuración dispositivo externo USB pendrive	47
5.10.2.	Configuración contenedor software backup Duplicati	49
5.10.3.	Configuración copia de seguridad con Duplicati	51
5.11.	Seguridad.....	55
5.11.1.	Introducción a la seguridad en el servidor	55
5.11.2.	Seguridad en los backups	57
5.11.3.	Seguridad en el router cliente	57
6.	Presupuesto.....	61
7.	Conclusiones y recomendaciones	62
8.	Bibliografía.....	64
9.	Anexos.....	66
9.1.	Anexo 1. Texto codificación Open VPN.....	66
9.2.	Anexo 2. Curriculum Vitae	69

Índice de figuras

Imagen 1. Empresa Perrauri proyecto	1
Imagen 2. Raspberry Pi 4B.....	2
Imagen 3. Kit Montaje Raspberry Pi 4B.....	3
Imagen 4. Raspberry Pi 4B montada	5
Imagen 5. Tarjeta Modulo GPIO Raspberry Pi 4B	6
Imagen 6. Software Raspberry Pi Imager 1.4	7
Imagen 7. Log consola tareas creación usuario perrauri.....	8
Imagen 8. Eliminación usuario Ubuntu	8
Imagen 9. Cambio hostname y comprobación.....	9
Imagen 10. Interfaces de red Raspberry	9
Imagen 11. Configuración estática netplan	10
Imagen 12. Respuesta sensores temperatura Raspberry.....	11
Imagen 13. Parámetros DDNS perrauri.ddns.net	12
Imagen 14. Configuración DDNS en el router Livebox +.....	13
Imagen 15. Logo Docker	14
Imagen 16. Pila virtualización clásica (izquierda) vs pila Docker (derecha).....	15
Imagen 17. Instalación Docker.io	16
Imagen 18. Información Docker	16
Imagen 19. Contenedor Hello World.....	17
Imagen 20. Instalación de Docker-compose.....	18
Imagen 21. Instalación de Docker-compose.....	18
Imagen 22. Logo Portainer.....	19
Imagen 23. Instalación Portainer.....	20
Imagen 24. Inicio Portainer Web	21
Imagen 25. Interfaz Portainer	21
Imagen 26. Logo Netdata.....	22
Imagen 27. Infografía funciones e integración Netdata	23
Imagen 28. Ejecución contenedor Net-data	25
Imagen 29. Página web Netdata.....	26
Imagen 30. Creación pila Perrauri y su configuración	29
Imagen 31. Administración phpMyAdmin.....	32
Imagen 32. Visualización del contenedor Wordpress	34
Imagen 33. Imagen de Easy Cookies Policy	35
Imagen 34. Página Web “Perrauri”	35
Imagen 35. Salida comando contenedor VPN	37
Imagen 36. Configuración perfil OpenVPN Android.....	38
Imagen 37. Salida IP a través de la VPN Imagen 38. Datos conexión VPN	39
Imagen 39. Instalación cliente Windows OpenVPN	40

Imagen 40. Importación .ovpn cliente OpenVPN	41
Imagen 41. Cliente OpenVPN conectado correctamente.....	41
Imagen 42. Nextcloud.....	42
Imagen 43. Aplicaciones instaladas con Nextcloud.....	44
Imagen 44. Aplicaciones Nextcloud en Windows configurada	45
Imagen 45. Aplicaciones Nextcloud en Android configurada.....	46
Imagen 46. Listado de discos conectados Ubuntu.	47
Imagen 47. Información pendrive USB.	48
Imagen 48. Montaje de unidades en /etc/fstab.	48
Imagen 49. Logo software Duplicati	49
Imagen 50. Menú Duplicati.....	51
Imagen 51. Primer paso creación backup Duplicati	52
Imagen 52. Segundo paso, elección destino Duplicati	53
Imagen 53. Detalles copia de seguridad programada	54
Imagen 54. Activación cortafuegos UFW.	56
Imagen 55. Estado cortafuegos y puertos.	56
Imagen 56. Página principal acceso al router.	57
Imagen 57. Activación notificaciones acceso al router.....	58
Imagen 58. Cambio DNS servidor Raspberry.....	59

Índice de tablas

Tabla 1. Tabla comparativa mini servidores.....	4
Tabla 2. Puertos abiertos en el router cliente Perrauri.	60
Tabla 3. Costes hardware del proyecto.....	61
Tabla 4. Costes laborales del proyecto.	61

1. Resumen/Summary

1.1. Resumen

Este proyecto trata de la digitalización y proyección online de un negocio unipersonal. Para ello se abordará desde el punto de vista de un empresario sin apenas recursos económicos para poder invertir en tecnología, enfoque considero muy habitual hoy en día. Se parte de la compra de un servidor económico, e implantación para gestionar máquinas virtuales. Sobre esta base se confecciona el tejido necesario para sostener los servicios que se consideran imprescindibles hoy en día, página web para su visibilidad, gestión de base de datos para los clientes, capacidad de alojamiento de archivos en nube y offline sin cuotas y sus respectivos backups.

1.2. Summary

This project deals with the digitalization and online projection of a sole proprietorship business. For this, it will be approached from the point of view of an entrepreneur with hardly economic resources to be able to invest in technology, an approach I consider habitually today. It starts with the purchase of an economic server, and implementation to manage virtual machines. On this basis, the necessary fabric is made to support the services that are considered essential today, website for visibility, database management for customers, capacity to host files in cloud and offline without fees and their respective backups.

2. Antecedentes

Estamos en un mundo donde toda empresa pequeña se debe plantear si debe progresar en su digitalización más allá de un móvil una Tablet o un Ordenador de escritorio. Además, actualmente existen en nuestro país muchas pequeñas empresas con uno o pocos trabajadores que no pueden invertir grandes cantidades de dinero en servidores o cuotas grandes en servicios en nubes.

Entre los servicios necesarios, una página web que publicite online los servicios de estas empresas ya que con visibilidad en Internet hoy en día podemos ganar en clientes al tener mayor presencia; cubriendo posibles clientes que desconozcan nuestros servicios, fidelizando los mismos al poder tener una base de datos con los datos de estos clientes y pudiendo realizar acciones comerciales con ellos además de posicionar nuestra marca, entre otras ventajas.

Otros servicios imprescindibles hoy en día es poder disponer de nuestros archivos, de manera segura en cualquier sitio y dispositivo aprovechando la nube, pero a bajo costo y sin cuotas que estrangulen la economía.

Con el peligro que supone poder perder nuestros datos y ficheros, se aprovecha la implantación de un servidor para realizar copias de seguridad en este caso offline en un simple pendrive de bajo coste. De esta manera podremos recuperar antes desastres, virus y/o ataques informáticos nuestros datos sin problemas.

Gracias a la tecnología actual, es posible con una inversión mínima poder abastecer de esas necesidades a este tipo de empresas. En este caso se desarrollará sobre una plataforma consolidada como es la virtualización, aprovechándonos de ella. Con un simple miniordenador podremos construir una serie de servidores que nos provean de los servicios que requieren.

3. Objetivos generales y específicos y alcance del proyecto

Los objetivos generales del proyecto y específicos se detallan a continuación, basándose en la idea de una puesta en marcha de una empresa unipersonal:

3.1. Objetivos generales

Montaje de un servidor de bajo coste con página web y su base de datos. Además, se creará almacenamiento accesible en nube y backups offline.

3.2. Objetivos específicos

Los objetivos específicos son:

- Adquisición de mini servidor Raspberry pi 4
- Puesta en marcha con Ubuntu
- Configuración servicios DDNS
- Creación de contenedores VM con Docker necesarios
- Creación de página web WordPress y configuración
- Creación contenedores BB.DD. y PhpMyadmin
- Creación Nextcloud y configuración.
- Creación VPN y configuración.
- Creación copia de seguridad a pendrive.

3.3. Alcance del proyecto

Estaría fuera del alcance de este proyecto la idoneidad de realizarlo en algún servidor más potente o bien en nube ya que lo que se pretende es demostrar que con una mínima inversión se puede digitalizar y promover vía web una pequeña empresa.

El proyecto trata exclusivamente el trabajo de arrancar los servicios básicos digitales de una empresa unipersonal, que es el que suele contar con menos presupuesto, y más cuando no se trata de una empresa tecnológica sino de servicios como en este supuesto caso una peluquería canina.

4. Definiciones

Docker – Es un programa de código abierto que sirve para el despliegue de contenedores y sus aplicaciones dentro de ellas aislados, pero con posibilidad de conexión entre ellos.

Raspberry Pi – Es un ordenador de bajo coste y dimensiones reducidas utilizado tanto para electrónica como para informática. Actualmente en su modelo básico están por la cuarta versión.

LTS – Abreviatura de Long Term Support, soporte a largo plazo en castellano. Es una versión más estable y con mayor duración que la normal.

URL- Abreviatura de Uniform Resource Locator, dirección específica de recursos de red.

DNS – Abreviatura de Domain Name System, sistema de nombres de dominio; es un servicio que habilita enlaces entre nombres y sus direcciones de red asociadas.

Pip- Abreviatura de Python Package Index, es un sistema de gestión de paquetes escritos en Python.

PGID- Abreviatura de Process Group Id. - Identidad de proceso de grupo.

BackEnd- En diseño Web se refiere a la visualización del sitio desde el lado del administrador/desarrollador del mismo.

Cookies- Archivos con información de navegación acerca de los sitios web visitados.

5. Desarrollo del trabajo final

Perrauri surge como empresa de peluquería canina gestionada por una única persona la cual se tiene que centrar en su negocio y ha dejado de lado todos los temas relacionados con el mismo a nivel digital. En la Imagen 1 se muestra el logo de la empresa “Perrauri” del proyecto.



Imagen 1. Empresa Perrauri proyecto

Esta empresa contacta con un informático para extender su negocio en Internet, y para ello solicita tener los servicios básicos explicados, a saber, una página web básica que pueda escalar en un futuro e incorporar más adelante una base de datos de clientes, algún tipo de servidor para sus ficheros y que pueda acceder desde el móvil y el portátil, copias de seguridad y poder conectarse de manera segura si hiciese falta a su local bien para en un futuro integrar cámaras o bien para acceder a algún tipo de recurso. Para ello y debido a la pandemia, no cuenta apenas con recursos y además de esto, no quiere que esté del todo en internet, ya que no le interesa agregar suscripciones mensuales de servicios sino hacer un esfuerzo de implementación y quizás en un futuro poder replantearse la situación si mejora el negocio o si tiene que contratar algún trabajador.

Con esto se procede a implementar el proyecto con un servidor dedicado en Raspberry Pi 4B ya que nos dará la flexibilidad suficiente para instalar todo lo necesario y ejecutarlo con niveles aceptables para las necesidades de la empresa por un coste muy bajo para lo que ofrece. Para tener una gran versatilidad, se procederá a instalar contenedores basados en virtualización ligera Docker en este caso. A partir de aquí se procederá a la instalación de todo lo necesario para el montaje de los servicios necesarios para la empresa. En la Imagen 2 se muestra una Raspberry Pi modelo 4B-.



Imagen 2. Raspberry Pi 4B.

5.1. Adquisición y montaje de Raspberry Pi 4B

Adquisición de Raspberry Pi 4 con su comparación. El coste total de inversión es de 122,49 € que se divide en:

- Ordenador Raspberry Pi 4B 8GB RAM 82.51€
- Kit LABISTS Caja 19,99€
- Tarjeta SanDisk Extreme 128GB 19,99€

En el kit viene con carcasa, adaptador de corriente, cable Micro HDMI, 3 disipadores de calor de metal y ventilador. En la Imagen 3 se muestra el kit adquirido de Raspberry Pi 4B y sus complementos.



Imagen 3. Kit Montaje Raspberry Pi 4B

5.1.1. Estudio comparación y adquisición Raspberry Pi 4B

La decisión de adquirir y montar Raspberry Pi 4B para Perrauri ha venido motivada por la siguiente comparativa de modelos Synology, Servidor HP y Raspberry que era lo mínimo en cuanto a precio y recursos se puede exigir en el montaje de nuestro servidor. En la Tabla 1 se muestra la comparación entre mini servidores.

	SYNOLOGY NAS DS118	RASPBERRY PI 4B 8GB	HP PROLIANT GEN10 5420
CPU	REALTEK 4 NÚCLEOS 1.4 GHz	BROADCOM 4 NÚCLEOS 1,5 GHz	Intel G5420 2 NÚCLEOS 3.8 GHz
RAM	1GB DDR4	8GB DDR4	8gb DDR4
ARQUITECTURA	ARM 64 Bits	ARM 64 Bits	X86 64 Bits
ALMACENAMIENTO	1 HDD 2,5 SATA SSD 120 GB	1 SD XDHG 128GB	1 HDD 2,5 SATA SSD 120 GB
CONECTIVIDAD LAN	1x 1000 Ethernet	1x 1000 Ethernet 1x Wifi AC	4x 1000 Ethernet
CONSUMO	9.4W Acceso	15.3 W máximo 3.4W Acceso	74.1W 70% CPU 15.2W acceso
PRECIO	~200€ con SSD	~130€ con SD	~425€ con SSD

Tabla 1. Tabla comparativa mini servidores.

Como se observa, por apenas un poco más de cien euros la Raspberry nos da recursos suficientes para todo el montaje de servicios que se va a realizar en comparación con las otras dos opciones, siendo la más económica de todas ellas.

5.1.2. Montaje de componentes

Se procede al montaje de los componentes adquiridos, empezando por la carcasa de la Raspberry,

Montaje de componentes: Montaje carcasa. Montaje ventilador en modo silencioso a 3.3V + ground (tierra) en modo extracción. Montajes disipadores 1 CPU Socket, 1 para RAM y 1 para ethernet. En la Imagen 4 se muestra el kit adquirido de Raspberry Pi 4B una vez montado listo para conectar.



Imagen 4. Raspberry Pi 4B montada

Procedemos a conectar el ventilador que trae el kit a los pines 1, correspondiente a 3,3 Voltios al cable rojo del ventilador y 6 correspondiente a tierra o ground con el cable negro del ventilador para que funcione en modo “silencioso”, que será suficiente para nuestro cometido.

En la Imagen 5 se muestra la tarjeta de conexionado de modulo GPIO Raspberry Pi 4B, en rojo los pines conectados.

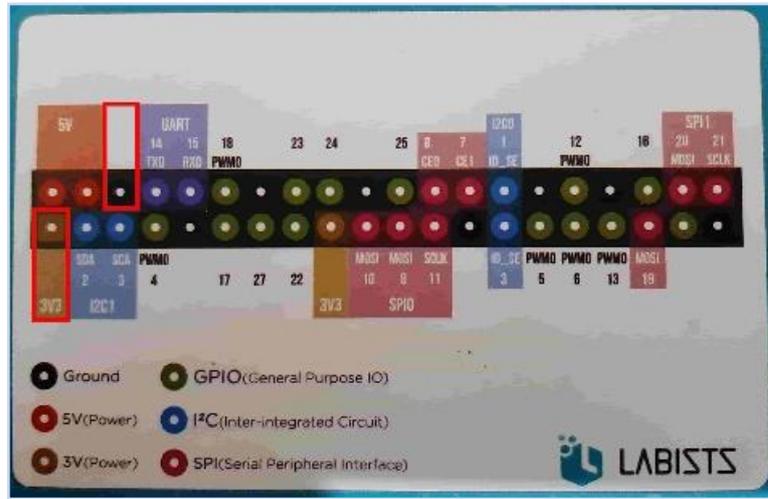


Imagen 5. Tarjeta Modulo GPIO Raspberry Pi 4B

5.2. Puesta en marcha con Ubuntu

Vamos a proceder con la puesta en marcha del sistema operativo elegido, Ubuntu 20.04 LTS versión 64 bits.

5.2.1. Instalación del sistema operativo Ubuntu

Se decide la instalación en Ubuntu ya que a pesar de haber otros sistemas operativos preparados para Raspberry como Raspbian encontraremos más apoyo por la comunidad con Ubuntu al estar más extendido para la implantación de servidor y de virtualización ligera.

Desde la versión 20.04 LTS Canonical ha certificado la versión para la Raspberry, además, al ser una versión LTS suele ser más estable y comprobada, con soporte de cinco años, lo que nos llevaría hasta el 2025 así que se procede a su instalación versión 20.04 LTS 64 bits.

Pudiéndose hacer de diferentes formas, instalaremos el sistema operativo en la tarjeta SD ayudándose de la herramienta Raspberry Pi Imager 1.4, desde la URL: <https://www.raspberrypi.org/software/> . En la Imagen 6 se muestra una captura de pantalla del software Raspberry Pi Imager utilizado para crear una imagen de sistema Ubuntu.



Imagen 6. Software Raspberry Pi Imager 1.4

Eligiendo Ubuntu Server 20.04.1 LTS para la SD. Una vez finalizado el proceso, ya tenemos el sistema operativo en la tarjeta. La introducimos en la Raspberry e iniciamos la misma.

5.2.2. Actualización del sistema

Se realiza una actualización del sistema para que busque e instale los últimos parches:

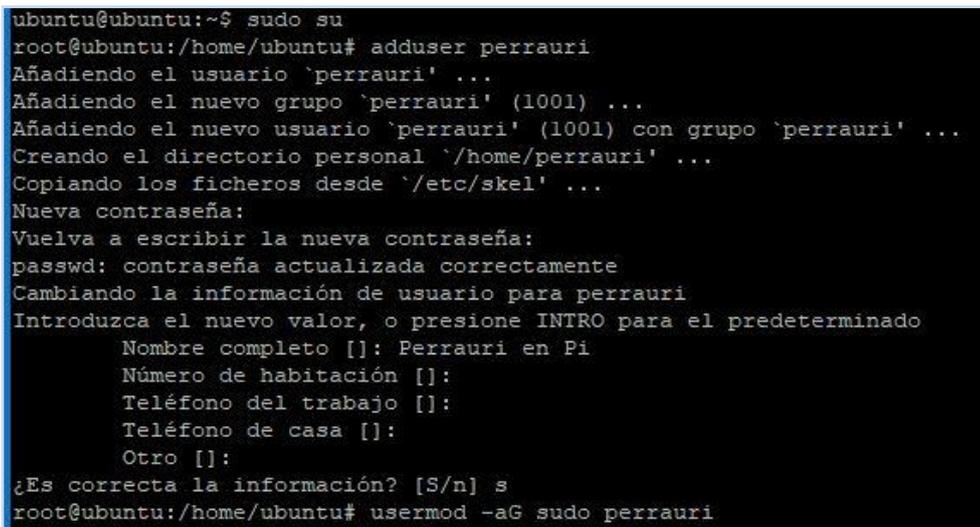
```
ubuntu@ubuntu:~$ sudo apt update && upgrade
```

5.2.3. Configuración básica del sistema

Una vez instalada la distribución lo primero que deberemos hacer es cambiar el usuario y contraseña ya que suponen una vulnerabilidad.

Como root, se crea un usuario nuevo y se conceden privilegios administrativos agregándolo al grupo sudo. En la Imagen 7 se muestra una captura de pantalla con la creación del usuario perrauri.

```
ubuntu@ubuntu:~$ sudo su
ubuntu@ubuntu:~$ adduser perrauri
ubuntu@ubuntu:~$ usermod -aG sudo perrauri
```

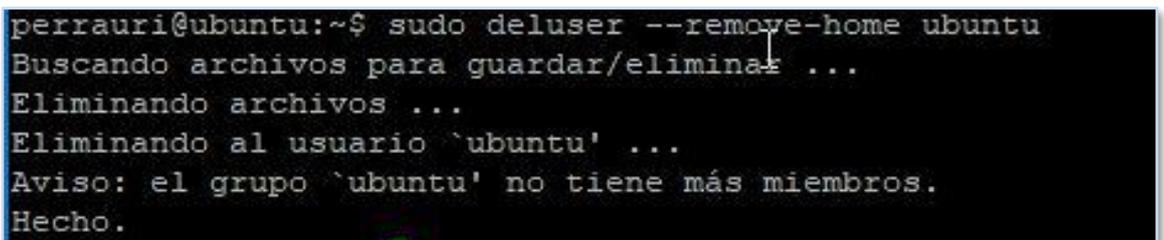


```
ubuntu@ubuntu:~$ sudo su
root@ubuntu:/home/ubuntu# adduser perrauri
Añadiendo el usuario `perrauri' ...
Añadiendo el nuevo grupo `perrauri' (1001) ...
Añadiendo el nuevo usuario `perrauri' (1001) con grupo `perrauri' ...
Creando el directorio personal `/home/perrauri' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para perrauri
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Perrauri en Pi
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
root@ubuntu:/home/ubuntu# usermod -aG sudo perrauri
```

Imagen 7. Log consola tareas creación usuario perrauri

Eliminamos el usuario que trae por defecto el sistema, de nombre “ubuntu” junto con su directorio de usuario para dejar el sistema limpio. En la Imagen 8 se muestra una captura de pantalla con la eliminación del usuario Ubuntu.

```
ubuntu@ubuntu:~$ sudo deluser --remove-home ubuntu
```



```
perrauri@ubuntu:~$ sudo deluser --remove-home ubuntu
Buscando archivos para guardar/eliminar ...
Eliminando archivos ...
Eliminando al usuario `ubuntu' ...
Aviso: el grupo `ubuntu' no tiene más miembros.
Hecho.
```

Imagen 8. Eliminación usuario Ubuntu

A partir de este momento se trabajará con el usuario “perrauri”. Se va a cambiar también el nombre del equipo por “pservervm” para más tarde reiniciar el equipo. En la Imagen 9 se muestra una captura de pantalla con el cambio de equipo y el estado del sistema

```
perrauri@ubuntu:~$ sudo hostnamectl set-hostname pservervm
```

```
perrauri@ubuntu:~$ sudo hostnamectl set-hostname pservervm
perrauri@ubuntu:~$ hostname
pservervm
perrauri@ubuntu:~$ hostnamectl status
  Static hostname: pservervm
            Icon name: computer
            Machine ID: 263b35cd509f40afa20e48a136a75c1d
            Boot ID: 9d715bd2c78f42879859fc5c4079b209
  Operating System: Ubuntu 20.04.1 LTS
            Kernel: Linux 5.4.0-1023-raspi
            Architecture: arm64
perrauri@ubuntu:~$
```

Imagen 9. Cambio hostname y comprobación

El siguiente paso correspondería a cambiar el direccionamiento de red y sus DNS de dinámicos a estáticos, ya que, al tratarse de un servidor, requiere por funcionalidad una dirección IP estática.

Primero de todo revisamos con el comando `ls /sys/class/net` la interfaz a la que vamos a cambiar la configuración. En la Imagen 10 se muestra una captura de pantalla el listado del directorio `/sys/class/net` para ver los interfaces de red.

```
perrauri@pservervm:~$ ls /sys/class/net
br-a577c76c673d  eth0          veth3c78fdf   veth58595fb   wlan0
br-fa9357a19f98  lo           veth4b89d1a   veth6434f75
docker0         veth2bdd394  veth4e4d736   vethd27660d
perrauri@pservervm:~$
```

Imagen 10. Interfaces de red Raspberry

El interfaz que interesa en esta ocasión es eth0 ya que es la conexión cableada, mucho más estable que la inalámbrica (wlan0), como sólo tiene un conector ethernet, se configura.

Se edita el archivo yaml situado en /etc/netplan poniendo en false el valor dhcp4, eliminando la fila optional ya que no haría falta al tener dirección IP estática y se añaden los valores de la red, en este caso:

- IP 192.168.1.1
- Máscara de subred 255.255.255.0 (/24)
- Pasarela o Gateway 192.168.1.1
- DNS's en este caso se añaden públicos 8.8.8.8 que corresponden a Google y los del propio router 192.168.1.1 para que asigne los de la compañía de comunicaciones, en este caso Jazztel.

En la Imagen 11 se muestra una captura de pantalla con el fichero netplan modificado

```
network:
  ethernets:
    eth0:
      dhcp4: false
      addresses: [192.168.1.11/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8,192.168.1.1]
  version: 2
```

Imagen 11. Configuración estática netplan

Una vez configurado y guardado, aplicamos y ya tendríamos estático el direccionamiento.

```
perrauri@pservervm:~$ sudo netplan apply
```

5.2.4. Instalación de paquetes adicionales

Se instalan paquetes adicionales que se consideran útiles para varios cometidos, empezando, “lm-sensors” para poder monitorizar la temperatura del sistema.

```
perraui@pservervm:~$ sudo apt-get install lm-sensors
```

Comprobamos que las temperaturas del sistema están en valores normales, en caso de la medición 41,4 grados centígrados sin carga. En la Imagen 12 se muestra una captura de pantalla con los sensores térmicos funcionando.

```
perraui@pservervm:~$ watch -n 5 sensors
```



```
Cada 5,0s: sensors

cpu_thermal-virtual-0
Adapter: Virtual device
temp1:          +41.4°C

rpi_volt-isa-0000
Adapter: ISA adapter
in0:            N/A
```

Imagen 12. Respuesta sensores temperatura Raspberry

5.2.5. Dynamic DNS - DDNS

En la empresa Perrauri se ha instalado una conexión a internet de fibra óptica con dirección IP externa dinámica, de tal forma que irá cambiando de dirección con el paso de los días lo que supondría un problema para el direccionamiento de la página web. Para poder asignar un DNS a la empresa con IP dinámica, utilizaremos un servicio de direccionamiento dinámico o DDNS sin coste. Entre los posibles candidatos, se elige No-IP. En la Imagen 13 se muestra una captura de pantalla con el registro de dominio en la página web no-ip de perrauri.ddns.net

Se registra el dominio “perrauri.ddns.net”

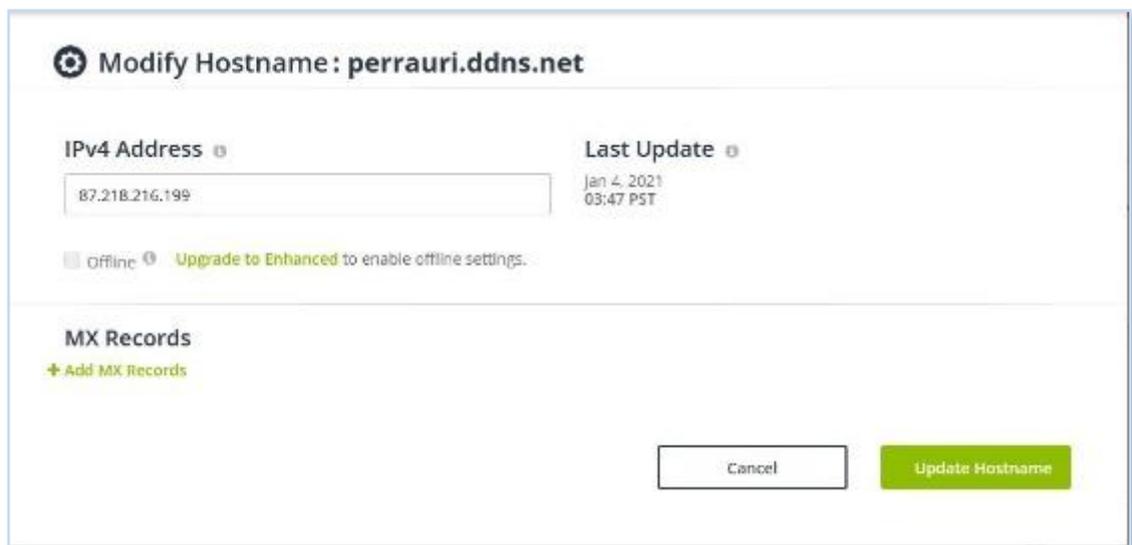


Imagen 13. Parámetros DDNS perrauri.ddns.net

Una vez registrado y comprobado que funciona correctamente, se debe configurar un renovador de IP para el servicio DDNS. Se plantean tres posibilidades, introducirlo en un contenedor Docker, instalarlo como servicio directamente en la Raspberry o en este caso por una solución más sencilla.

Existen actualmente multitud de routers con el servicio de actualización DDNS integrado así que, para minimizar recursos, se procede sobre a configurarlo sobre el mismo. En la sección configuración avanzada de este modelo (Livebox +) se introducen los datos en el apartado “DynDNS”. En la Imagen 14 se muestra una captura de pantalla con la configuración del router del ddns perrauri.ddns.net.

Esta página te permite configurar el servicio DynDNS. Usa un DynDNS para asociar a la dirección IP pública de una URL larga y un subdominio fácil de recordar como mydomain.dyndns.org. De esta manera, podrás acceder Livebox o servidor en tu LAN fácilmente.

Utilizar DynDNS puede ser útil si alojas un sitio web, servidor ftp o cualquier tipo de servidor en la LAN. Puedes fácilmente un nombre como thisiscool.dyndns.org

configuración del Servidor de Nombres Dinámico (DDNS)

servicio	nombre de host completo	nombre de usuario email	contraseña	última actualización	
dyndns	<input type="text"/>	<input type="text"/>	<input type="password"/>		guardar
NoIP	perrauri.ddns.net	<input type="text"/>	*****	05/01/21 07:25:25	borrar

Imagen 14. Configuración DDNS en el router Livebox +

Se comprueba que corresponde la dirección IP externa a la DDNS perrauri.ddns.net con el comando curl y ping.

```
perrauri@pservervm:~$ curl ifconfig.me
87.218.217.232perrauri@pservervm:~$ ping perrauri.ddns.net
```

```
PING perrauri.ddns.net (87.218.217.232) 56(84) bytes of data.
64 bytes from 232.217.218.87.dynamic.jazztel.es (87.218.217.232): icmp_seq=1
ttl=64 time=0.693 ms
```

5.3. Creación contenedores virtualización ligera con Docker

Para la creación de los contenedores que se utilizarán más adelante, se va a instalar la base de las máquinas virtuales ligeras, Docker.

5.3.1. Introducción a Docker

Se elige Docker ya que es un sistema de virtualización ligera que trabaja con contenedores, lo que nos permite a bajo coste de inversión poder tener varios servicios en nuestra empresa. En la Imagen 15 se muestra una imagen de Docker con su logo.



Imagen 15. Logo Docker

Es una plataforma de código abierto que nos va a permitir crear contenedores con aplicaciones o servicios que se pueden ejecutar de manera autosuficiente al margen de su sistema anfitrión, lo que se denomina virtualización ligera. Utiliza para ello el kernel de Linux del propio sistema anfitrión por lo que no requiere de un sistema operativo propio dentro de su contenedor, liberando por tanto de recursos.

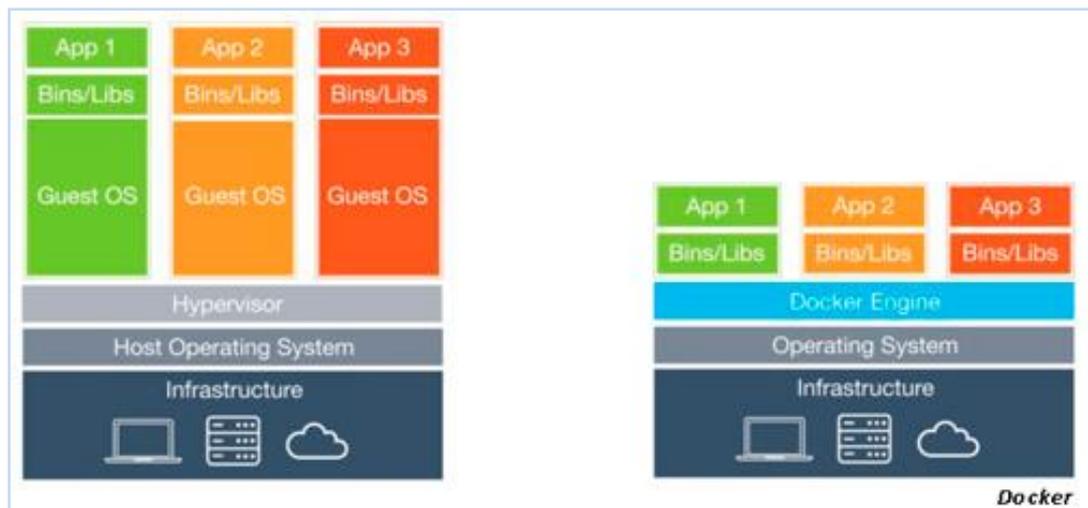


Imagen 16. Pila virtualización clásica (izquierda) vs pila Docker (derecha)

En la imagen 16 se observa la diferencia entre la virtualización clásica que requiere para cada aplicación o servicio de un sistema operativo cliente y otro para el anfitrión y la virtualización ligera con Docker que únicamente requiere el sistema operativo de la máquina anfitrión. De esta forma es como se ahorran recursos.

Dentro de estos contenedores es donde están los servicios o aplicaciones que se irán desplegando para los requisitos de esta empresa. Éstos pueden estar aislados totalmente o conectados entre sí para compartir recursos, lo cual nos brinda las oportunidades de montar servicios que requieren de varios recursos a la vez, cómo una página web con base de datos, por ejemplo.

5.3.2. Creación contenedores VM con Docker

Se procede a la instalación de Docker, para ello primero actualizamos el sistema:

```
perrauri@pservervm:~$ sudo apt-get update && sudo apt-get upgrade
```

Para a continuación instalar Docker.

```
perrauri@pservervm:~$ sudo apt-get install docker.io
```

En la Imagen 17 se muestra una captura de pantalla con la instalación del paquete docker.io a través de apt-get.

```
perraauri@pservervm:~$ sudo apt-get install docker.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
Paquetes sugeridos:
  ifupdown aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
Se instalarán los siguientes paquetes NUEVOS:
  bridge-utils cgroupfs-mount containerd docker.io pigz runc ubuntu-fan
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 48,0 MB de archivos.
Se utilizarán 270 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Imagen 17. Instalación Docker.io

Comprobamos el estado e información de Docker:

```
perraauri@pservervm:~$ sudo docker info
```

```
perraauri@pservervm:~$ sudo docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.8
```

Imagen 18. Información Docker

En la Imagen 18 se muestra una captura de pantalla con la información de estado de Docker. Opcionalmente, también se podría revisar con el comando `systemctl status docker` para saber si está ejecutándose.

Se procede también a activar el servicio “docker” en el arranque del equipo para no tener que hacerlo manualmente en un reinicio o apagado:

```
perrauri@pservervm:~$ sudo systemctl enable docker
```

Si necesitamos parar/iniciar el servicio de docker manualmente ejecutaríamos:

```
perrauri@pservervm:~$ sudo systemctl stop docker
```

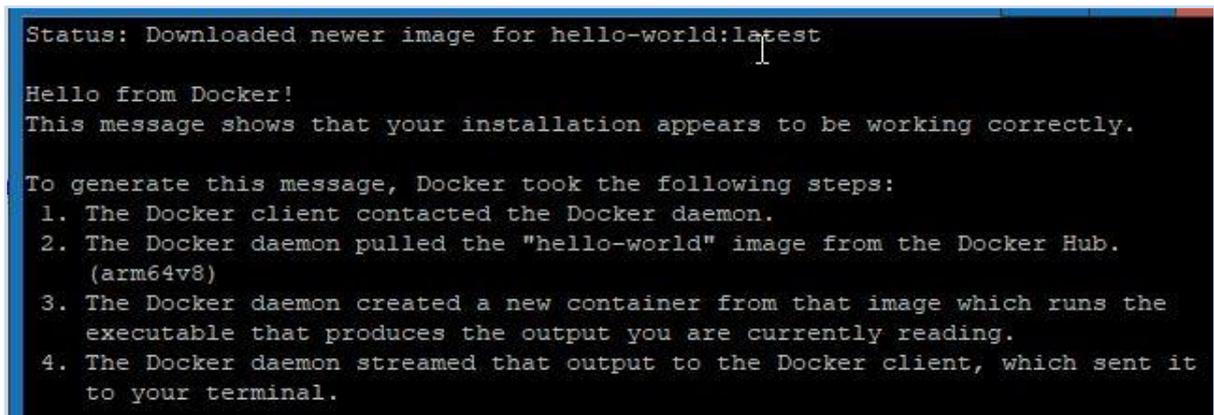
 para parar docker

```
perrauri@pservervm:~$ sudo systemctl start docker
```

 para iniciar docker

Para la comprobación del sistema docker se hace una prueba con el contenedor “Hello World” que nos dirá si está funcionando correctamente. En la Imagen 19 se muestra una captura de pantalla el estado del contenedor de prueba “Hello World” descargada

```
perrauri@pservervm:~$ sudo docker run hello-world
```



```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

Imagen 19. Contenedor Hello World

Para proceder a continuación con la instalación de docker-compose el cual va a servir para poder trabajar con múltiples contenedores pudiendo unirlos entre sí. En la Imagen 20 se muestra una captura de pantalla con la instalación de docker-compose vía apt-get

```
perrauri@pservvm:~$ sudo apt-get install docker-compose
```

```
perrauri@pservvm:~$ sudo apt-get install docker-compose
[sudo] contraseña para perrauri:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
python3-cached-property python3-docker python3-dockerpty python3-docopt python3-texttable python3-websocket
Se instalarán los siguientes paquetes NUEVOS:
docker-compose python3-cached-property python3-docker python3-dockerpty python3-docopt python3-texttable
python3-websocket
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 262 kB de archivos.
Se utilizarán 1.616 kB de espacio de disco adicional después de esta operación.
```

Imagen 20. Instalación de Docker-compose

Para no tener que estar continuamente introduciendo el comando “sudo” se va a incluir el usuario “perrauri” al grupo “docker”

```
perrauri@pservvm:~$ sudo usermod -aG docker perrauri
```

Se comprueba la versión instalada, en este caso la 1.25.0. En la Imagen 21 se muestra una captura de pantalla la comprobación de la versión de docker-compose.

```
perrauri@pservvm:~$ docker-compose --version
```

```
perrauri@pservvm:~/usr/local/bin$ docker-compose version
docker-compose version 1.25.0, build unknown
docker-py version: 4.1.0
CPython version: 3.8.5
OpenSSL version: OpenSSL 1.1.1f  31 Mar 2020
```

Imagen 21. Instalación de Docker-compose

Desde aquí ya se tiene la base para poder instalar un gestor de contenedores por página web para poder administrar de una manera más “amigable” los contenedores.

5.4.Gestión contenedores con Portainer

Se procede a la instalación y configuración del gestor de contenedores Portainer bajo Docker.

5.4.1. Introducción a Portainer

Se elige este gestor de contenedores ya que se puede trabajar con él fácil y gráficamente a través de página web. Es una herramienta de código abierto desde donde vamos a desplegar, gestionar los contenedores y sus características en este caso con Docker, aunque es compatible con otro tipo de contenedores.

Además de esto, permite opciones como la parada, ejecución o revisión de logs de los contenedores, por lo que también para el mantenimiento nos brinda herramientas suficientes para el control de los contenedores liberándonos de la consola del sistema lo cual redundará en mayor agilidad y rapidez a la hora de trabajar con contenedores. En la Imagen 22 se muestra el logo del gestor de contenedores “portainer.io”



Imagen 22. Logo Portainer.

5.4.2. Gestión contenedores con Portainer

Se procede a la configuración y creación del contenedor Portainer. Para ello, se comienza creando un volumen con el contenedor:

```
perrauri@pservvm:~$ sudo docker volume create portainer data
```

Se configuraría con las siguientes opciones:

```
perrauri@pservvm:~$ sudo Docker run -d -p 8000:8000 -p 9000:9000 --  
name=portainer --restart=always -v /var/run/docker.sock:/var/run/Docker.sock -v  
portainer_data:/data portainer/portainer
```

El detallado del código sería el siguiente:

- `docker run -d` → Ejecuta Docker y arranca en segundo plano
- `-p 8000:8000 -p 9000:9000` → Utilizará los puertos 8000 y 9000 para conectarse
- `-name=portainer` → Ejecutará o si no la tiene en repositorio descargará la imagen de nombre “portainer”
- `restart: always` → Se inicia siempre automáticamente.
- `-v /var/run/docker.sock:/var/run/Docker.sock -v portainer_data:/data portainer/portainer` → las variables de las rutas donde alojará y conectará portainer.

En la Imagen 23 se muestra una captura de pantalla con la creación del volumen y contenedor de Docker.io

```
perrauri@pservvm:~$ sudo docker volume create portainer_data  
portainer_data  
perrauri@pservvm:~$ sudo docker run -d -p 8000:8000 -p 9000:9000 --name=portai  
ner --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_d  
ata:/data portainer/portainer  
Unable to find image 'portainer/portainer:latest' locally  
latest: Pulling from portainer/portainer  
d1e017099d17: Pull complete  
4e1a79253c37: Pull complete  
Digest: sha256:f8c2b0a9ca640edf508a8a0830cf1963a1e0d2fd9936a64104b3f658e120b868  
Status: Downloaded newer image for portainer/portainer:latest  
bd3236abe36fb6e7c8c535734250fdal3cb9aa3dbd65e04d3d37283d41bb7715  
perrauri@pservvm:~$
```

Imagen 23. Instalación Portainer

Ya estaría creado y configurado container y podemos acceder desde un navegador web a la URL de la Raspberry por el puerto :9000. En la Imagen 24 se muestra una captura de pantalla con pantalla de inicio y petición de credenciales de Portainer.

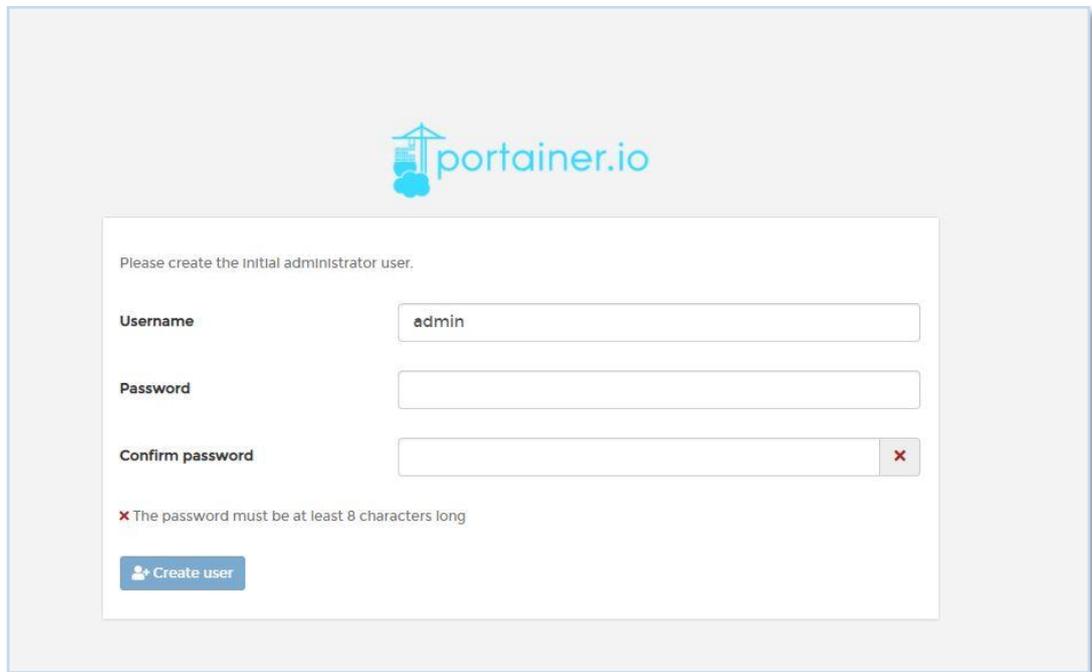


Imagen 24. Inicio Portainer Web

Conectamos a nivel Local el entorno docker con Portainer y ya tendríamos finalizada la instalación básica de Portainer. En la Imagen 25 se muestra una captura de pantalla con la pantalla de cuadro de mando de Portainer conectado con docker.

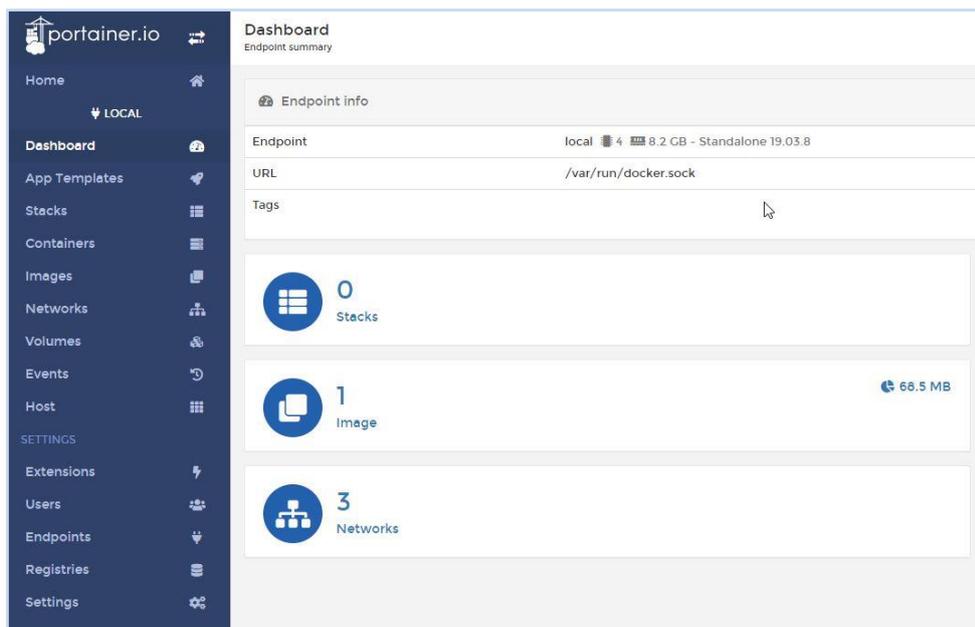


Imagen 25. Interfaz Portainer

Si queremos parar el contenedor, podemos desde el propio portainer por web o bien desde la Shell:

```
perrauri@pservervm:~$ sudo docker stop portainer
```

5.5. Monitorización de estadísticas con NetData

A la hora de poder administrar este mini servidor se necesita poder medir su rendimiento en tiempo real en una aplicación que podamos ejecutar según necesidad y parar para dejar de consumir recursos. Se opta para Perrauri por Netdata.

5.5.1. Introducción a NetData

Netdata es una aplicación de código abierto que permite entre otras funciones, la monitorización de los recursos de todo tipo del sistema, desde el consumo de CPU, de RAM, etc, pasando por consumo de recursos de los contenedores de manera totalmente visual y gráfica. En la Imagen 26 se muestra el logo de Netdata



Imagen 26. Logo Netdata

La fortaleza de esta herramienta reside en que se integra perfectamente con el sistema, así como con todo tipo de aplicaciones y es capaz de medir miles de variables.

También su consumo propio de recursos es muy bajo, por lo que no perjudica apenas el sistema donde se integra.

Además, tiene un tiempo de respuesta de las métricas para monitorizar muy elevado, por lo que se puede visualizar en tiempo casi real las métricas, permitiéndonos identificar problemas sin demora o bien observando los cambios que apliquemos en el sistema o contenedores de forma inmediata.

En la Imagen 27 se muestra la infografía de compatibilidad con Netdata.

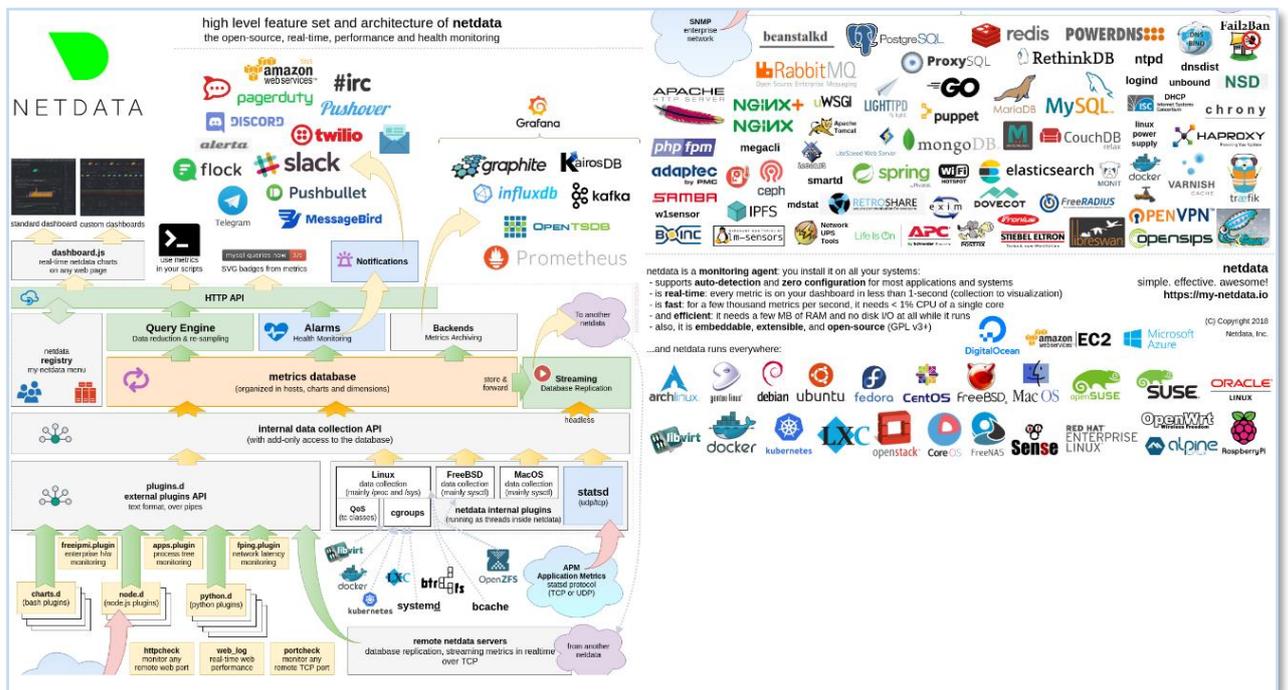


Imagen 27. Infografía funciones e integración Netdata

5.5.2. Configuración NetData

El primer paso para su instalación es averiguar el PGID del servicio de Docker para poder otorgárselo a las variables de Netdata. Para ello se ejecuta el siguiente comando.

```
perraui@pservervm:~$ grep docker /etc/group | cut -d ':' -f 3
```

La salida nos devuelve el ID de proceso Docker PGID 134, que se insertará en el código para despliegue. Se procede al despliegue de Netdata, en esta ocasión por Docker, aunque se podría hacer opcionalmente por docker-expose o desde Portainer.

El código para el contenedor es el siguiente:

```
docker run -d --name=netdata \  
-p 19999:19999 \  
-e PGID=134 \  
-v netdataconfig:/etc/netdata \  
-v netdatalib:/var/lib/netdata \  
-v netdatacache:/var/cache/netdata \  
-v /etc/passwd:/host/etc/passwd:ro \  
-v /etc/group:/host/etc/group:ro \  
-v /proc:/host/proc:ro \  
-v /sys:/host/sys:ro \  
-v /etc/os-release:/host/etc/os-release:ro \  
-v /var/run/docker.sock:/var/run/docker.sock:ro \  
--restart unless-stopped \  
--cap-add SYS_PTRACE \  
--security-opt apparmor=unconfined \  
netdata/netdata
```

El detallado del sería el siguiente:

- docker run -d → Ejecuta Docker y arranca en segundo plano
- -name=netdata -> Ejecutará o si no la tiene en repositorio descargará la imagen de nombre “netdata”
- -p 19999:19999 -> Utilizará el puerto 19999 y lo enviará al mismo.
- -e PGID=134 utiliza la variable PGID vista anteriormente del servicio Docker.
- Variables netdataconfig, netdatalib netdata caché:
 - v netdataconfig:/etc/netdata \
 - v netdatalib:/var/lib/netdata \
 - v netdatacache:/var/cache/netdata \Son las rutas donde carga la configuración librerías y caché de netdata.
- Resto de variables:
 - v /etc/passwd:/host/etc/passwd:ro \
 - v /etc/group:/host/etc/group:ro \
 - v /proc:/host/proc:ro \
 - v /sys:/host/sys:ro \
 - v /etc/os-release:/host/etc/os-release:ro \
 - v /var/run/docker.sock:/var/run/docker.sock:ro \Son las rutas desde donde va a leer netdata las estadísticas como usuarios grupos, procesos, sistema, etcétera. restart: always -> Se inicia siempre automáticamente.

- --cap-add SYS_PTRACE -> activa la capacidad de trazar y seguir procesos.
- --security-opt apparmor=unconfined -> Inicia el contenedor sin perfil de seguridad del módulo del kernel, que no lo confine AppArmor.
- netdata/netdata -> descarga la imagen netdata/netdata para el contenedor.

En la Imagen 28 se muestra una captura de pantalla con el código de creación del contenedor netdata y su configuración.

```
perraauri@pservervm:~$ docker run -d --name=netdata \
> -p 19999:19999 \
> -e PGID=134 \
> -v netdataconfig:/etc/netdata \
> -v netdatalib:/var/lib/netdata \
> -v netdatacache:/var/cache/netdata \
> -v /etc/passwd:/host/etc/passwd:ro \
> -v /etc/group:/host/etc/group:ro \
> -v /proc:/host/proc:ro \
> -v /sys:/host/sys:ro \
> -v /etc/os-release:/host/etc/os-release:ro \
> -v /var/run/docker.sock:/var/run/docker.sock:ro \
> --restart unless-stopped \
> --cap-add SYS_PTRACE \
> --security-opt apparmor=unconfined \
> netdata/netdata
26e2d6939d3d76ab06e28de3f3bff30f6459b4235c7a3261d36bb124e66d3c60
```

Imagen 28. Ejecución contenedor Net-data

Con esto estaría creado el contenedor, que habilita por página web al puerto mapeado 19999, por lo que abrimos una página web desde LAN; bien porque estamos en ella directamente bien porque nos conectamos por OpenVPN que se verá más adelante. No se le dará salida WAN por carecer de autenticación el contenedor. Se observa que ya tenemos los valores de netdata, pudiendo tener las gráficas de CPU, carga, disco, RAM, contenedores, etc.

Se observa a través de página web que ya tenemos los valores de netdata, pudiendo visualizar las gráficas de CPU, carga, disco, RAM, contenedores, etc. En la Imagen 29 se muestra una captura de pantalla con la página web de Netdata mostrando las estadísticas del sistema.

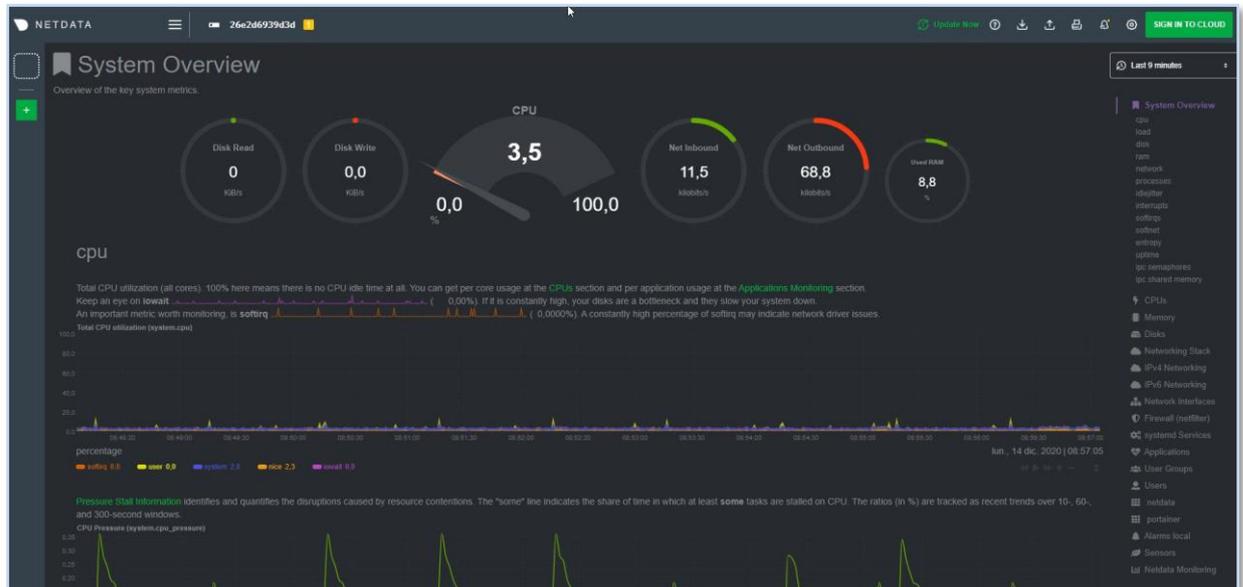


Imagen 29. Página web Netdata

Este contenedor podemos levantarlo desde portainer y desactivarlo cuando no nos haga falta bien por seguridad, bien para liberar recursos. Para ello, desde el apartado Containers seleccionamos contenedor y pulsamos Stop. También podemos realizarlo desde SSH con el comando:

```
docker stop netdata
```

5.6. Creación contenedores página web y configuración

Una vez creado estos contenedores básicos, se empieza la configuración de los contenedores necesarios para la creación de la página web.

En este caso ya que se trata de una página sencilla promocional, se va a montar con Wordpress con una base de datos MariaDB y un administrador de bases de datos phpMyAdmin se toma esta decisión al tratarse de software libre.

Se procede para ello a crear una pila en portainer de nombre "Perrauri" y se utiliza el editor Web, aunque también se puede crear un archivo de tipo .yaml y subirlo bien con portainer bien con docker compose herramienta también instalada para su gestión.

El código para crear los tres contenedores en la pila es el siguiente:

```
#Rafa 2021 Contenedores MariaDB, Wordpress y phpMyAdmin
version: '2'
```

```
services:
  # Base de datos MariaDB
  db:
    image: mariadb:latest
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    ports:
      - '3306:3306'
    environment:
      MYSQL_ROOT_PASSWORD: En2021-RootPi
      MYSQL_DATABASE: wordpress
      MYSQL_USER: perrauri
      MYSQL_PASSWORD: En2021-WordPi
    networks:
      - wpsite
```

```
  # Gestor phpmyadmin
  phpmyadmin:
    depends_on:
      - db
    image: phpmyadmin:latest
    restart: always
    ports:
      - '8085:80'
    environment:
      PMA_HOST: db
      PMA_ARBITRARY: 1
      MYSQL_ROOT_PASSWORD: En2021-RootPi
    networks:
      - wpsite
```

```
# Wordpress
wordpress:
  depends_on:
    - db
  image: wordpress
  restart: always
  ports:
    - '8008:80'
  volumes: ['./:/var/www/html']
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: perrauri
    WORDPRESS_DB_PASSWORD: En2021-WordPi
  networks:
    - wpsite
networks:
  wpsite:
  volumes:
    db_data:
```

Una vez desplegado, nos aparecerá una pila (stack), tres contenedores, tres imágenes (images) y una red de nombre “perrauri_wpsite” que gestiona las uniones entre los tres contenedores. En la Imagen 30 se muestra una captura de pantalla con la creación de los tres contenedores bajo el editor web de portainer.

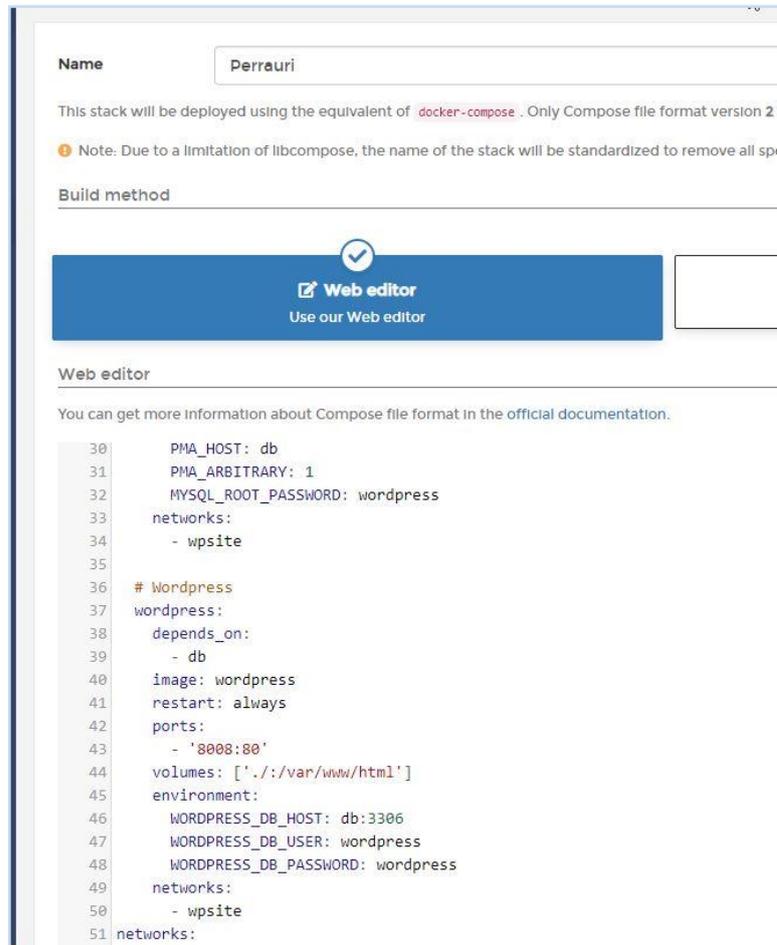


Imagen 30. Creación pila Perrauri y su configuración

5.6.1. Base de Datos MariaDB

En el conjunto de montaje de esta pila, se integra por su versatilidad MariaDB como punto de partida a la página web que se desarrolla inicialmente y que, en un futuro se puede utilizar como base de datos en este u otros desarrollos de Perrauri. De esta forma, la base ya estaría disponible.

Se explica a continuación la parte de código de la base de datos.

```
# Base de datos MariaDB
db:
  image: mariadb:latest
  volumes:
    - db_data:/var/lib/mysql
  restart: always
  ports:
    - '3306:3306'
  environment:
    MYSQL_ROOT_PASSWORD: En2021-RootPi
    MYSQL_DATABASE: wordpress
    MYSQL_USER: perrauri
    MYSQL_PASSWORD: En2021-WordPi
  networks:
    - wpsite
```

Con respecto a las opciones de este contenedor serían las siguientes:

- image: mariadb:latest → Utiliza la imagen última de Mariadb.
- Volumes: -db_data:/var/lib/mysql -> Utiliza esa ruta para el montaje
- Restart: always -> Se inicia siempre automáticamente.
- Ports: -'3306:3306' -> utilizará el puerto 3306 como conexión
- Environment: MYSQL_ROOT_PASSWORD: -> La contraseña que utilizará como raíz
- Environment: MYSQL_DATABASE: -> La base de datos que va a crear inicialmente
- Environment: MYSQL_PASSWORD: -> La contraseña de esa bb.dd. (wordpress)
- Networks: -wpsite -> la red que crea y utilizará en conjunto con el resto de la pila.

5.6.2. Gestor PhpMyadmin

Como añadido a esta pila y de cara a poder mantener las bases de datos de manera fácil, se utilizará phpMyAdmin que es una herramienta vía web de mantenimiento de BB.DD. De esta forma cualquier tarea en la BB.DD. actual o futura se gestionará desde aquí.

Para ello, se utilizará el siguiente código:

```
# Gestor phpmyadmin
phpmyadmin:
  depends_on:
    - db
  image: phpmyadmin:latest
  restart: always
  ports:
    - '8085:80'
  environment:
    PMA_HOST: db
    PMA_ARBITRARY: 1
    MYSQL_ROOT_PASSWORD: En2021-RootPi
  networks:
    - wpsite
```

Con respecto a las opciones de este contenedor serían las siguientes:

- Depends_on: -db → Utilizará como base la base de datos MariaDB creada anteriormente
- Image:phpmyadmin:latest -> Utilizará la última imagen de phpmyadmin dockerizada
- Restart: always -> Se inicia siempre automáticamente
- Ports: -'8085:80' -> Utilizará el puerto internamente 80 y lo reenviará al 8085 externo
- Environment: PMA_HOST: db -> el host de la base de datos MariaDB
- Environment: PMA_ARBITRARY:1 -> phpMyAdmin nos solicitará el contenedor de conexión.
- Environment: MYSQL_ROOT_PASSWORD: -> contraseña de conexión a BB.DD. MariaDB
- Networks: -wpsite -> la red que crea y utilizará en conjunto con el resto de la pila.

Una vez desplegado, al gestor PhpMyAdmin podemos entrar introduciendo desde el navegador la dirección IP 192.168.1.11 o bien la dirección DNS perrauri.dyndns.net a través del puerto que hemos especificado al crear el contenedor :8085, logándonos con el usuario root y la contraseña especificada también en la creación. En la Imagen 31 se muestra una captura de pantalla con el gestor vía web phpMyAdmin.

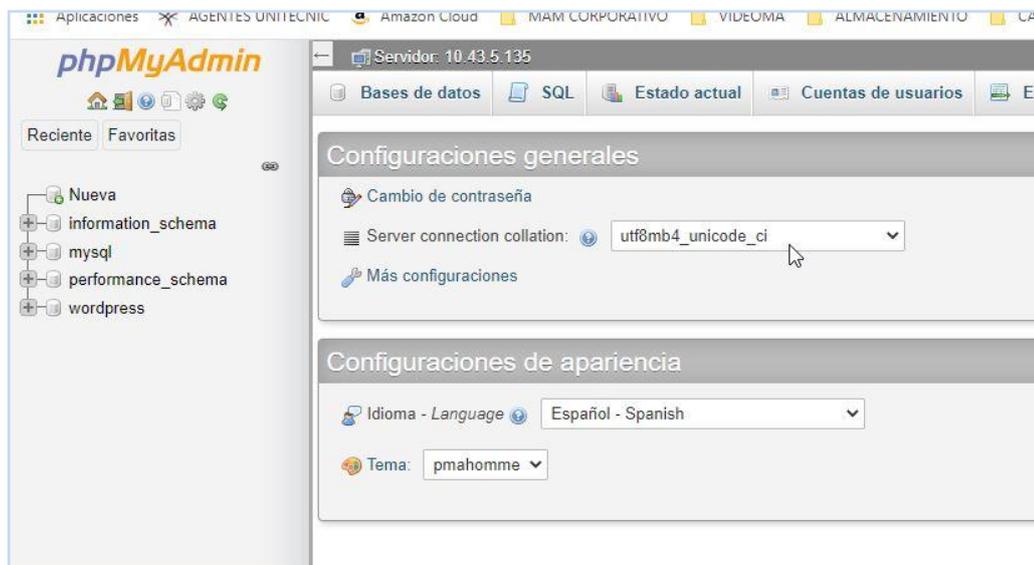


Imagen 31. Administración phpMyAdmin

Una vez dentro del sistema, se puede crear y mantener BB.DD., gestionar código SQL, usuarios y demás.

5.6.3. Wordpress

Se va a utilizar un contenedor Wordpress, que es referencia en el mundo de las páginas web y que, por su manejo, facilidad y versatilidad, encaja perfectamente en la creación de la página web de peluquería canina para la empresa Perrauri.

Con respecto a la parte del código del contenedor sería el siguiente detallado:

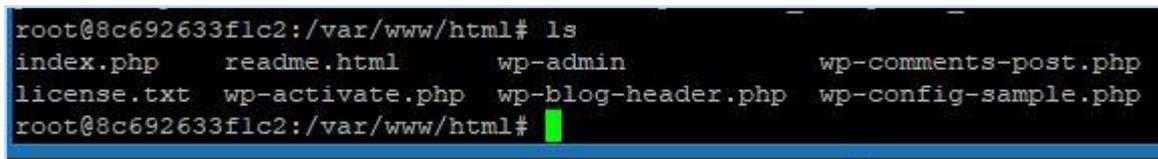
```
# Wordpress
wordpress:
  depends_on:
    - db
  image: wordpress
  restart: always
  ports:
    - '8008:80'
  volumes: ['./var/www/html']
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: perrauri
    WORDPRESS_DB_PASSWORD: En2021-WordPi
  networks:
    - wpsite
  networks:
    wpsite:
  volumes:
    db_data:
```

- Depends_on: -db → Utilizará la base de datos MariaDB creada anteriormente.
- Image:wordpress → Utilizará la imagen disponible de nombre WordPress.
- Restart: always → Se inicia siempre automáticamente.
- volumes: ['./var/www/html'] → Utilizará el volumen definido /var/www/html
- Environment: WORDPRESS_DB_HOST: db:3306 → Utilizará el servidor de bb.dd. de nombre db a través de una conexión por el puerto 3306.
- Environment: WORDPRESS_DB_USER: → La conexión con la base de datos será con el usuario indicado.
- Environment: WORDPRESS_DB_PASSWORD: → La conexión con la base de datos será con la contraseña usuario indicada.
- Networks: -wpsite y volumes: db_data: → la conexión con los volúmenes y la red

A pesar de que el desarrollo de la página web se va a realizar desde el propio BackEnd de Wordpress, si queremos entrar por consola para modificar algún archivo de WordPress o cargar algún complemento manualmente se procedería de la siguiente manera:

```
perrauri@pservervm:~$ docker exec -t -i perrauri_wordpress_1 bash
```

Esto le indica el nombre del contenedor para entrar con la consola bash y ya se podría operar desde dentro del contenedor. En la Imagen 32 se muestra una captura de pantalla desde dentro del contenedor Wordpress listando los archivos con el comando “ls”.



```
root@8c692633flc2:/var/www/html# ls
index.php      readme.html    wp-admin      wp-comments-post.php
license.txt    wp-activate.php wp-blog-header.php wp-config-sample.php
root@8c692633flc2:/var/www/html#
```

Imagen 32. Visualización del contenedor Wordpress

5.7. Diseño página Web

Para el proyecto de Perrauri, se nos solicita una página sencilla ya que únicamente quiere dar a conocer su labor.

Para ello, entramos a la página web por el backend: <http://perrauri.ddns.net/wp-admin/index.php>

Comprobamos que tiene actualizaciones pendientes de Temas, así como de plugins, por lo que se actualizan desde la sección “Actualizaciones”.

Una vez resuelto las actualizaciones pendientes, vamos a utilizar el tema de Wordpress “Roseta” <https://wordpress.org/support/theme/roseta/reviews/> para aplicarle forma a la página web.

Una vez subidas al servidor, así como definidas las imágenes, logotipos y contenido, se aplica y publica el contenido.

Para cumplir con la política de privacidad de datos, se va a instalar el complemento “Easy Cookies” el cual nos permite instalar una barra de advertencia acerca de la política de cookies. En la Imagen 33 se muestra una captura de pantalla con el plugin para Wordpress “Easy Cookies Policy” usado para advertir a los usuarios acerca de las cookies.



Imagen 33. Imagen de Easy Cookies Policy

Por otro lado, de cara al mantenimiento de las noticias y novedades de la página web, se crea para el usuario de la empresa Perrauri un perfil de editor, de esta forma, puede subir contenido sin tener que acudir al administrador del sitio.

A partir de aquí, ya se entregaría a la empresa la página web para que continúe añadiendo contenido. La base ya estaría. En la Imagen 34 se muestra una captura de pantalla de navegador Web con la página web cargada y funcional “http://perrauri.ddns.net”

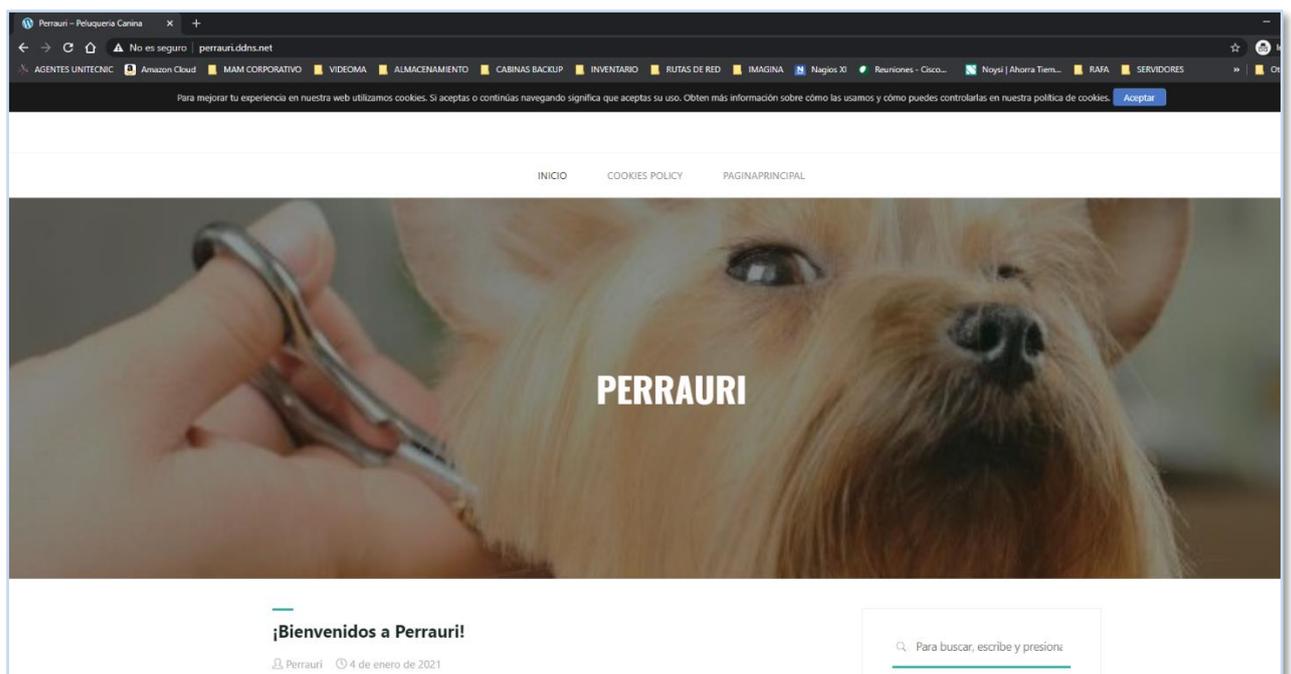


Imagen 34. Página Web “Perrauri”

5.8. Creación y configuración VPN con OpenVPN

Para la empresa va a ser necesario dada la privacidad de alguna información y en caso de tener que trabajar desde algún sitio externo que tenga poca fiabilidad o seguridad como algún punto wifi o demás. Con este fin se va a configurar un contenedor con un servidor VPN en OpenVPN. Se elige OpenVPN al tratarse de Open source y utilizar seguridad SSL/TLS, además de otras características.

5.8.1. Servidor VPN OpenVPN

Lo primero de todo es crear un contenedor, para ello utilizaremos la imagen docker `evolvedm/openvpn-rpi` que es compatible con nuestra Raspberry pi.

```
perrauri@pservervm:~$ docker run -v ovpn_data:/etc/openvpn --rm evolvedm/openvpn-rpi ovpn_genconfig -u udp://perrauri.ddns.net
```

En las opciones de creación, hemos utilizado la dirección WAN de nuestra red: `perrauri.ddns.net`. El siguiente paso es configurar la entidad que nos va a emitir el certificado:

```
perrauri@pservervm:~$ docker run -v ovpn_data:/etc/openvpn --rm -it evolvedm/openvpn-rpi ovpn_initpki nopass
```

En la Imagen 35 se muestra una captura de pantalla con la creación del contenedor Openvpn vía docker y su configuración básica.

```
00529eabes01 portainer/portainer /portainer 4 weeks ago up 19 hours 0.0.0.0:8000->8000/tcp, 0.0.0.0:8000->8000/tcp
perrauri@pservervm:~/contenedores$ docker run -v ovpn_data:/etc/openvpn --rm evolvedm/openvpn-rpi ovpn_genconfig -u udp://perrauri.ddns.net
Unable to find image 'evolvedm/openvpn-rpi:latest' locally
latest: Pulling from evolvedm/openvpn-rpi
Image docker.io/evolvedm/openvpn-rpi:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility
ee4626e9f304: Pull complete
a3ed95caeb02: Pull complete
49388e30a4d6: Pull complete
6f3afab76d27: Pull complete
Digest: sha256:02d8d6c2bceadf58939c182d1ce61555e65ce7837e2b90cffc7dd07be83d303d
Status: Downloaded newer image for evolvedm/openvpn-rpi:latest
Successfully generated config
perrauri@pservervm:~/contenedores$ docker run -v ovpn_data:/etc/openvpn --rm -it evolvedm/openvpn-rpi ovpn_initpki nopass

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/pki

Generating a 2048 bit RSA private key
.+++
.....+++
writing new private key to '/etc/openvpn/pki/private/ca.key.XXXXXHfDd'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/pki/ca.crt

Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....+.....
```

Imagen 35. Salida comando contenedor VPN

A continuación, inicio del servicio con las opciones del puerto 1194 en UDP, añadiendo la red NET_ADMIN, que reinicie siempre y poniéndole nombre al volumen:

```
perrauri@pservervm:~$ docker run -v ovpn_data:/etc/openvpn --rm -it evolvedm/openvpn-rpi ovpn_initpki nopass
```

Ahora se utiliza una variable de nombre perrauri y después se genera el perfil para openvpn en el contenedor donde se asigna la contraseña "P3rrauri2021@" para el perfil.

```
perrauri@pservervm:~$ CLIENTNAME="perrauri"
```

```
perrauri@pservervm:~$ docker run -v ovpn_data:/etc/openvpn --rm -it evolvedm/openvpn-rpi easyrsa build-client-full
```

Una vez termina se exporta el certificado a un fichero de nombre perraui.ovpn.

```
perraui@pservervm:~$ docker run -v ovpn_data:/etc/openvpn --rm  
evolvedm/openvpn-rpi openvpn_getclient $CLIENTNAME > $CLIENTNAME.ovpn
```

Este archivo se deja en el sistema operativo de la raspberry en la ruta “/home/perraui/contenedores/vpn” y además de ello lo copiamos a Nextcloud y demás equipos.

5.8.2. Clientes VPN

Openvpn necesita aplicación propia para gestionar la VPN. Existen varios clientes y se va a utilizar en un móvil Android y en un dispositivo con Windows.

En Android, en la tienda directamente se puede localizar:

https://play.google.com/store/apps/details?id=net.openvpn.openvpn&hl=es_VE&gl=US

Se procede a su configuración con los parámetros asignados, se selecciona el archivo .ovpn que hemos guardado en Nextcloud o local, usuario perraui, etc y ya estaría listo para conexión. En la Imagen 36 se muestra una captura de pantalla de móvil Android con la configuración inicial de OpenVPN y la importación del perfil.

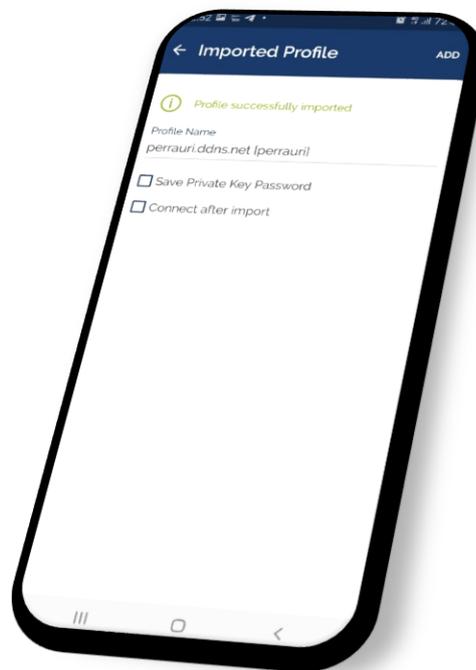


Imagen 36. Configuración perfil OpenVPN Android

Una vez se conecta ya se puede comprobar que estamos en la misma red, con la misma dirección IP Pública y responden los dispositivos en el entorno de la VPN. Además, con la aplicación para Android nos otorga datos estadísticos como velocidad, duración, dirección IP local, etc. En la Imagen 37 se muestra una captura de pantalla de móvil Android con la IP Pública que coincide con la del sistema VPN mientras que en la Imagen 38 se muestra una captura de pantalla de móvil Android con las estadísticas de conexión de OpenVPN.



Imagen 37. Salida IP a través de la VPN

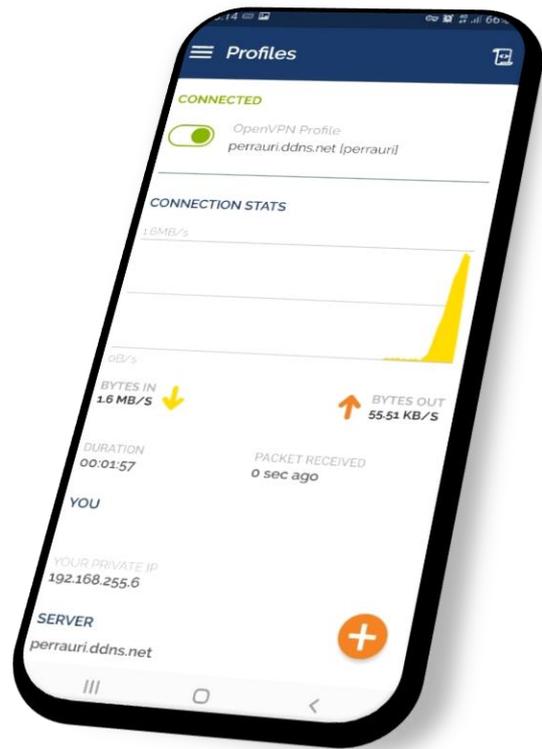


Imagen 38. Datos conexión VPN

En Windows el procedimiento sería similar. Se procede a descargar el cliente desde: <https://openvpn.net/client-connect-vpn-for-windows/> que sirve para Windows 7 o superior. En el asistente de instalación se elige el driver “Tap Driver” al estar recomendado y se terminaría la instalación pulsando en “install”. En la Imagen 39 se muestra una captura de pantalla la instalación del cliente OpenVPN bajo Windows.

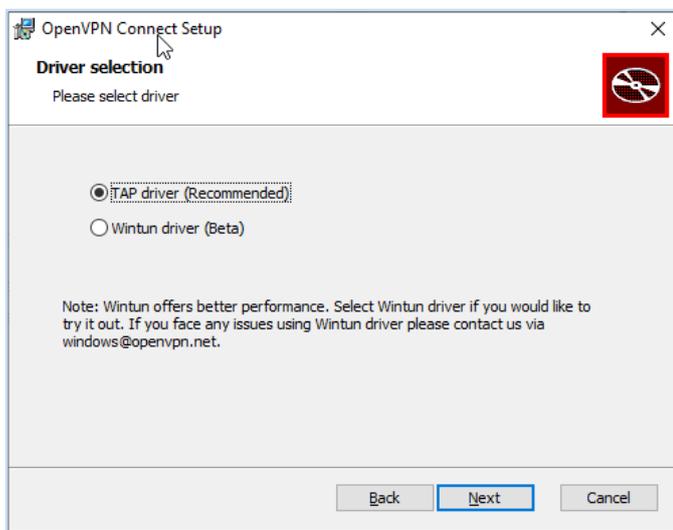


Imagen 39. Instalación cliente Windows OpenVPN

Una vez terminada, ejecutamos el programa “OpenVpn connect” y procedemos a importar el archivo .ovpn que hemos creado previamente y a “connect”. En la Imagen 40 se muestra una captura de pantalla de Windows con la configuración inicial de OpenVPN y la importación del perfil, mientras que en la Imagen 41 se muestra una captura de pantalla de Windows con las estadísticas de conexión de OpenVPN.

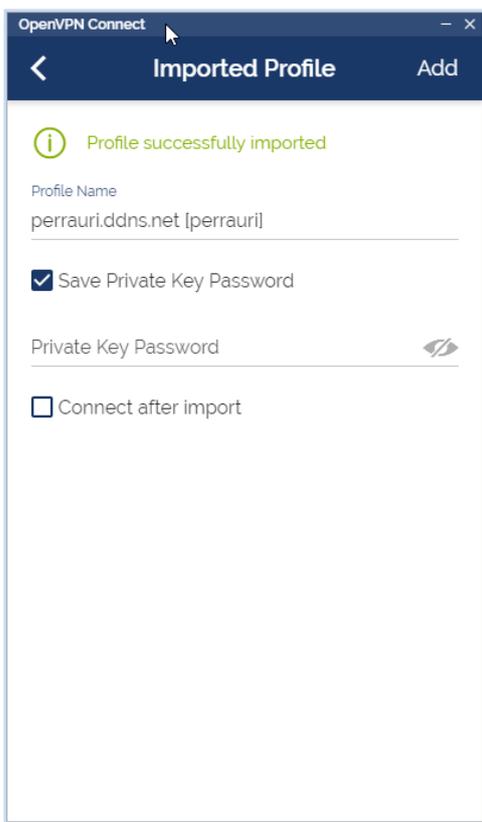


Imagen 40. Importación .ovpn cliente OpenVPN

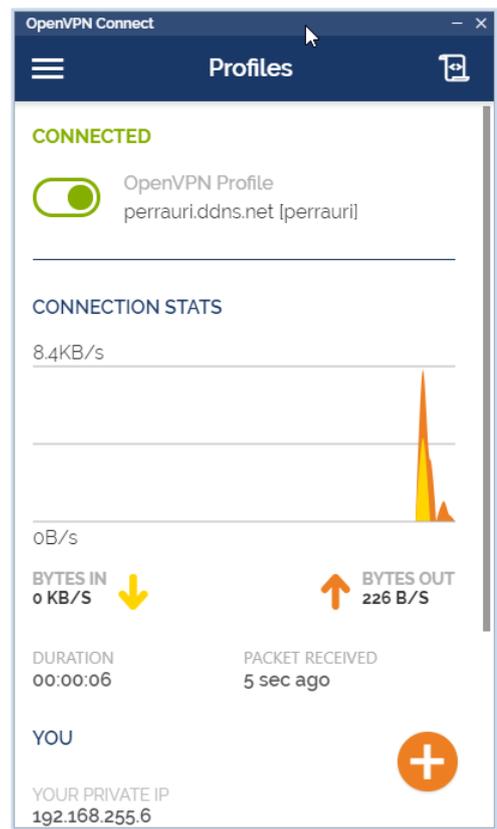


Imagen 41. Cliente OpenVPN conectado correctamente

Con esto ya estaríamos conectados desde el ordenador a través de OpenVPN a nuestra red interna.

5.9. Creación y configuración almacenamiento en nube Nextcloud

Para poder gestionar los archivos de la empresa, así como imágenes y demás, se opta por la instalación de software que nos permita tener almacenamiento en nube. En esta ocasión, se elige Nextcloud, al tratarse de una aplicación de código abierto.

Ésta nos va a permitir trabajar con nuestros archivos en nube, pero desde nuestro servidor. Siendo además el punto de partida para gestionar aplicaciones como calendario, tareas, email, y otros. En la Imagen 42 se muestra una imagen de Nextcloud con su logo e iconos



Imagen 42. Nextcloud

Para ello se crea una pila desde Portainer (opcionalmente un fichero de tipo “.yaml” y se gestiona con docker-compose).

```
#Rafa 2021 Contenedor Nextcloud
version: "2"
services:
  nextcloud:
    image: nextcloud
    container_name: nextcloud
    environment:
      - PUID=1001
      - PGID=1001
      - TZ=Europe/Madrid
      - NEXTCLOUD_TRUSTED_DOMAINS=perrauri.ddns.net
    volumes:
      - ~/nextcloud/config:/config
      - ~/nextcloud/data:/data
    ports:
      - 8081:80
      - 8082:443
    restart: always
```

El detallado del código sería el siguiente:

- Versión: “2” -> Utiliza la versión “2” de Docker compose.
- Services: nextcloud -> Carga el servicio de nombre nextcloud.
- Image: nextcloud -> Utilizará la imagen de nombre “Nextcloud” la descargará si no la tiene de internet.
- Container_name: nextcloud -> Al contenedor le otorga el nombre “nextcloud”.
- Environment: PUID=1001: utilizará el id del usuario 1001 que corresponde al usuario Linux perrauri del sistema.
- Environment: PGID=1001: utilizará el id del grupo del usuario 1001 que corresponde al usuario Linux perrauri del sistema.
- TZ=Europe/Madrid -> Asigna a la zona horaria de Madrid.
- NEXTCLOUD_TRUSTED_DOMAINS=perrauri.ddns.net -> asigna el dominio de confianza a perrauri.ddns.net que es el de la empresa creado, de esta forma cargará cuando sea llamado externamente.
- volumes - ~/nextcloud/config:/config y volumes - ~/nextcloud/data:/data -> Utiliza sendas rutas para la configuración y volúmenes de Netdata

- ports: 8081:80 – 8082:443 -> Utilizará los puertos internos 80 y 443 y lo reenviará al 8081 y 8082.
- Restart: always -> La política de reinicio se establece en que se inicie siempre automáticamente.

En la Imagen 43 se muestra una captura de pantalla de Nextcloud funcionando con las aplicaciones adicionales instalándose.



Imagen 43. Aplicaciones instaladas con Nextcloud

5.9.1. Configuración clientes Nextcloud

Con la aplicación ya instalada y configurada, descargamos para el usuario la aplicación para escritorio y para móvil desde las direcciones web: <https://nextcloud.com/install/#install-clients>. En este caso es para un móvil Android así que sería desde la tienda <https://play.google.com/store/apps/details?id=com.nextcloud.client>

5.9.1.1. Configuración cliente Windows

Ejecutamos el asistente para la instalación en Windows descargado de la página descrita anteriormente, en este caso el archivo Nextcloud-3.1.1x64.msi versión de 64 Bits para Windows 10. Con la dirección web y puerto de nuestro servidor. Una vez concedido permiso de acceso, Configuramos sincronización según cliente y se asigna la carpeta local para el replicado. Una vez hecho ya dispone del cliente configurado y ejecutándose en el sistema, podrá subir y descargar archivos desde el equipo. En la Imagen 44 se muestra una captura del cliente Nextcloud para windows configurado con el usuario “perraui”



Imagen 44. Aplicaciones Nextcloud en Windows configurada

5.9.1.2. Configuración clientes Android

Con la aplicación instalada desde Android PlayStore, se introducen los credenciales y dirección web de nuestro servidor para facilitar el acceso. En la Imagen 45 se muestra una captura de pantalla de móvil Android con el cliente Nextcloud configurado con el usuario “perrauri” ya funcionando y sincronizado.

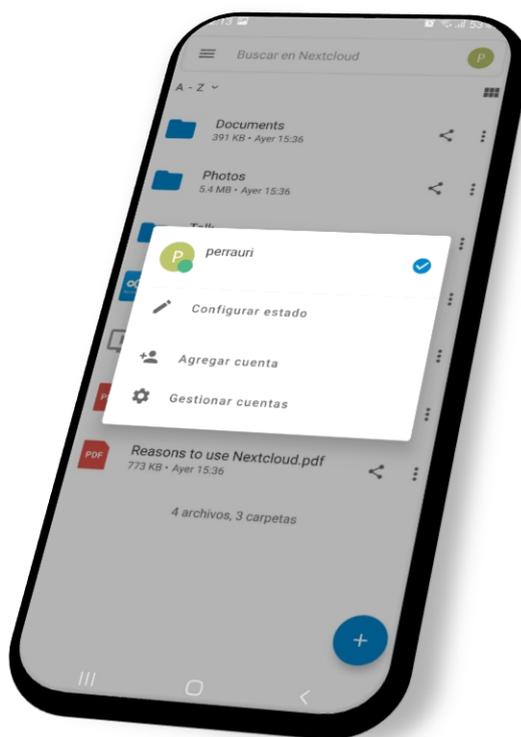


Imagen 45. Aplicaciones Nextcloud en Android configurada

Una vez logado en el servidor, el usuario ya podrá disponer de su almacenamiento en nube sincronizando los archivos desde/hacia Android.

5.10. Creación copia de seguridad a Pendrive

Para poder realizar copias de seguridad y que además sean programadas, se va a utilizar el software “Duplicati” en un contenedor junto con un pendrive USB de 32 GB recién adquirido.

5.10.1. Configuración dispositivo externo USB pendrive

Lo primero de todo es proceder al montaje del pendrive. Se introduce en una ranura de la Raspberry Pi y creamos un directorio para el montaje de la unidad en /media/pendrivebackups.

```
perrauri@pservvm:~$ sudo cd /media
perrauri@pservvm:~$ sudo mkdir pendrivebackups
```

Ahora, se listan los dispositivos para comprobar cual corresponde con el pendrive . En la Imagen 46 se muestra una captura de pantalla de Ubuntu con el comando para listar dispositivos en busca del id del pendrive USB.

```
perrauri@pservvm: /media/pendrivebackups$ ls -l /dev/disk/by-uuid/*
```



```
perrauri@pservvm:/media/pendrivebackups$ ls -l /dev/disk/by-uuid/*
lrwxrwxrwx 1 root root 10 abr  1  2020 /dev/disk/by-uuid/3D78-4A1B -> ../../sda1
lrwxrwxrwx 1 root root 15 feb  3 14:45 /dev/disk/by-uuid/483efb12-d682-4daf-9b34-6e2f774b56f7 -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 15 abr  1  2020 /dev/disk/by-uuid/B726-57E2 -> ../../mmcblk0p1
perrauri@pservvm:/media/pendrivebackups$
```

Imagen 46. Listado de discos conectados Ubuntu.

Sabiendo que el dispositivo es 3D78-4A1B y corresponde a /sda1, se procede a visualizar la información del dispositivo para prepararlo. En la Imagen 47 se muestra una captura de pantalla de Ubuntu con la información del pendrive USB y su formato.

```
perraUri@pservervm: sudo fdisk -l /dev/sda1
```

```
Disco /dev/sda: 28,67 GiB, 30765219840 bytes, 60088320 sectores
Disk model: SanDisk 3.2Gen1
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x00000000

Dispositivo Inicio Comienzo      Final Sectores Tamaño Id Tipo
/dev/sda1                32 60088319 60088288  28,7G  c W95 FAT32 (LBA)
perraUri@pservervm:~$
```

Imagen 47. Información pendrive USB.

Nos da la información que el pendrive ya está preparado con formato FAT32, por tanto, al no tener que formatearlo, se procede directamente al montaje en el fichero del sistema FSTAB (/etc/fstab) de la unidad con formato VFAT. En la Imagen 48 se muestra una captura de pantalla de Ubuntu con el contenido del fichero fstab ya configurado para el Pendrive.

```
perraUri@pservervm:~$ sudo nano /etc/fstab
```

```
GNU nano 4.8
LABEL=writable /          ext4 defaults        0 0
LABEL=system-boot /boot/firmware vfat defaults          0 1
/dev/sda1      /media/pendrivebackups vfat defaults          0 0
```

Imagen 48. Montaje de unidades en /etc/fstab.

La explicación del montaje y sus opciones serían:

- /dev/sda1: dirección del USB
- /media/pendrivebackups: montaje del pendrive
- vfat: Formato FAT32 del pendrive
- defaults: Opciones por defecto del montaje.
- primer 0: No hace backup del sistema de ficheros
- segundo 0: No ejecuta check del sistema

Una vez guardado el archivo montamos:

```
perrauri@pservervm:~$ sudo mount -a
```

Y ya estaría listo el pendrive para su uso.

5.10.2. Configuración contenedor software backup Duplicati

Una vez procedido al montaje del pendrive, se va a crear una pila o stack dentro de Portainer con el contenedor Duplicati. En la Imagen 49 se muestra el logo de Duplicati



Imagen 49. Logo software Duplicati

Dicha pila o stack tendría la siguiente configuración:

```
version: "2.1"  
services:  
  duplicati:  
    image: ghcr.io/linuxserver/duplicati  
    container_name: duplicati
```

```

environment:
  - PUID=0
  - PGID=0
  - TZ=Europe/Madrid
  - CLI_ARGS= #optional
volumes:
  - /home/perrauri/contenedores/duplicati/config:/config
  - /home/perrauri/contenedores/duplicatibackups:/backups
  - /home/perrauri:/source
  - /media/pendrivebackups:/pendrive
  - /var/lib/docker:/dockers
ports:
  - 8200:8200
restart: unless-stopped

```

El detallado del código sería el siguiente:

- Versión: “2.1” -> Utiliza la versión “2.1” de Docker compose.
- Services: duplicati -> Carga el servicio de nombre nextcloud.
- Image: ghcr.io/linuxserver/duplicati -> Utilizará la imagen de nombre “duplicati” la descargará si no la tiene de internet en ghcr.io/linuxserver/duplicati.
- Container_name: duplicati -> Al contenedor le otorga el nombre “duplicati”.
- Environment: PUID=0: utilizará el id del usuario 0 que corresponde al usuario Linux root del sistema.
- Environment: PGID=0: utilizará el id del grupo del usuario 0 que corresponde al usuario Linux root del sistema.
- TZ=Europe/Madrid -> Asigna a la zona horaria de Madrid.
- CLI_ARGS=#optional -> cargará las opciones adicionales en este caso los volúmenes.
- volumes:
 - /home/perrauri/contenedores/duplicati/config:/config
 - /home/perrauri/contenedores/duplicatibackups:/backups
 - /home/perrauri:/source
 - /media/pendrivebackups:/pendrive
 - /var/lib/docker:/dockers

Cargaría dichas rutas para ver/montar, importante entre ellas /var/lib/Docker como ruta de los contenedores desde donde luego haremos copia de seguridad, /media/pendrivebackups:/pendrive que sería la ruta del pendrive USB para el destino de los backups.
- Ports: 8200:8200 -> Utilizará el puerto interno 8200 y lo reenviará al 8200.
- Restart: unless-stopped -> La política de reinicio se establece en que se inicie siempre automáticamente excepto si se detiene manualmente.

5.10.3. Configuración copia de seguridad con Duplicati

Una vez desplegado, podemos acceder por página web a: <http://perrauri.ddns.net:8200/> procediendo al login con el usuario especificado. En la Imagen 50 se muestra una captura de pantalla del panel web con las opciones de Duplicati.



Imagen 50. Menú Duplicati

Una vez logados, procederíamos al primer paso, “Añadir copia de seguridad”. En este paso, se configura una nueva copia de seguridad, en la cual, tenemos que asignarle un nombre, una descripción, así como elegir si queremos cifrar la copia de seguridad.

En este caso, se procede a elegir el cifrado de tipo “AES-256”, el cual hemos de completar con una frase de seguridad para el cifrado. En la Imagen 51 se muestra una captura de pantalla de la configuración de la copia de seguridad de Duplicati vía web.

1 General 2 Destino 3 Datos de Origen 4 Horario 5 Opciones

Configuración general de la copia de seguridad

Nombre:

Descripción (opcional):

Cifrado:

Frase de seguridad:

Repita la frase de seguridad:

Ocultar | Generar | Strength: Muy fuerte

Imagen 51. Primer paso creación backup Duplicati

El siguiente paso, 2, sería el destino (normalmente es el origen, pero Duplicati lo gestiona de esta forma), en tipo de almacenamiento elegiríamos “Unidad o carpeta local”, y la ruta del pendrive, cual es pendrive\backups. En un futuro se podría desplegar almacenamiento cloud con algunos de los servicios aceptados, como Onedrive, Dropbox o similar. En la Imagen 52 se muestra una captura de pantalla con la elección del destino de la copia de seguridad.



Imagen 52. Segundo paso, elección destino Duplicati

En el paso 3, seleccionaríamos los datos de origen, en este caso haremos copia de los contenedores y sus volúmenes, seleccionando dentro de la carpeta dockers (que en la configuración le hemos asignado el nombre “dockers” a la ruta /var/lib/docker) al menos los directorios “containers” y “volumes”

A partir de aquí, en el paso 4 se procede a la programación si se desea automática con el horario elegible en días, horas, etc. En este caso una vez cada 3 días y a las 3 de la mañana ya que será cuando menos carga probablemente tenga el sistema.

En el paso 5 se procede a la política de conservación de datos en “Retención de copias inteligente” para que Duplicati vaya eliminando automáticamente, pero conservando una Copia de seguridad cada siete días por uno cada cuatro semanas por uno cada doce meses. De esta forma siempre se tienen suficientes copias de seguridad como para protegerse ante daños.

El tamaño empleado en las copias de seguridad será de aproximadamente 800 MB con el material del usuario, por lo que tendríamos un total de al menos 29 imágenes ocupando un espacio en el pendrive de ~23 GB; restando 6 GB (espacio una vez formateado) para futuros aumentos de tamaño en la copia de seguridad.

Si aumentase drásticamente el uso habría que ampliar recomendablemente a 128 GB, dejando el de 32 para por ejemplo una copia anual fría.

Una vez finalizado y aceptado, nos saldrán los datos de los backups programados en el inicio donde se puede observar los datos y estadísticas de este. En la Imagen 53 se muestra una captura de pantalla con el resumen de la tarea de copia de seguridad, detalles y programación en Duplicati via web.



Imagen 53. Detalles copia de seguridad programada

5.11. Seguridad

Se crea este apartado del proyecto para definir independientemente la seguridad dado que muchos de los servicios requieren conexión a internet.

5.11.1. Introducción a la seguridad en el servidor

Se va a configurar el firewall que trae integrado el propio Ubuntu, “UFW” o Uncomplicated Firewall. Siendo este el cortafuegos por defecto del sistema operativo. Para ello lo primero de todo es conocer el estado de este, con el comando:

```
perrauri@pservvm:~$ sudo ufw status
```

```
Estado: inactivo
```

Como podemos observar, el sistema nos devuelve el estado, en este caso inactivo. A partir de aquí procedemos con los valores por defecto:

```
perrauri@pservvm:~$ sudo ufw default deny incoming
```

```
perrauri@pservvm:~$ sudo ufw default allow outgoing
```

Con estos comandos, lo que hacemos es por un lado denegar por defecto las conexiones entrantes y por otro lado permitir el tráfico saliente, de tal forma que, sobre esta base, iremos abriendo únicamente los puertos de entrada para los servicios y el resto estará cerrado de tal forma que se aumente la seguridad.

El siguiente paso es permitir las conexiones remotas al resto de puertos y servicios:

```
perrauri@pservvm:~$ sudo ufw allow 22
```

```
perrauri@pservvm:~$ sudo ufw allow 80
```

```
perrauri@pservvm:~$ sudo ufw allow 443
```

```
perrauri@pservvm:~$ sudo ufw allow 1194
```

```
perrauri@pservvm:~$ sudo ufw allow 3306
```

```
perrauri@pservvm:~$ sudo ufw allow 8000
```

```
perrauri@pservvm:~$ sudo ufw allow 8008
```

```
perrauri@pservvm:~$ sudo ufw allow 8081
```

```
perrauri@pservvm:~$ sudo ufw allow 8085
```

```
perrauri@pservvm:~$ sudo ufw allow 8200
```

```
perrauri@pservvm:~$ sudo ufw allow 9000
```

```
perrauri@pservvm:~$ sudo ufw allow 19999
```

Los puertos especificados son los necesarios para los contenedores, así como conexiones ssh (puerto 22) para entrar desde otra máquina a configurar la raspberry ya que esta se quedará cuando se termine la configuración guardada junto al router de la empresa Perrauri sin teclado ni monitor, ocupando el mínimo espacio.

Una vez configurada, se procede a activar el firewall:

```
perrauri@pservvm:~$ sudo ufw enable
```

```
perrauri@pservvm:~$ sudo ufw enable
El comando puede interrumpir las conexiones ssh existentes. ¿Continuar con la operación (s|n)? s
El cortafuegos está activo y habilitado en el arranque del sistema
perrauri@pservvm:~$
```

Imagen 54. Activación cortafuegos UFW.

En la Imagen 54 se muestra una captura de pantalla con la habilitación del firewall UFW en Ubuntu.

El sistema devuelve correctamente la habilitación del firewall, así como el inicio cada vez que arranque el sistema.

Se procede a reiniciar el sistema para verificar que reanuncia correctamente y, una vez reiniciado, se comprueba con:

```
perrauri@pservvm:~$ sudo ufw status verbose
```

El sistema nos devuelve los puertos permitidos. En la Imagen 55 se muestra una captura de pantalla en Ubuntu con el estado de los puertos configurado bajo UFW

```
perrauri@pservvm:~$ sudo ufw status
Estado: activo

Hasta      Acción     Desde
-----      -
22         ALLOW     Anywhere
80         ALLOW     Anywhere
443        ALLOW     Anywhere
1194       ALLOW     Anywhere
3306       ALLOW     Anywhere
8000       ALLOW     Anywhere
8008       ALLOW     Anywhere
8081       ALLOW     Anywhere
8085       ALLOW     Anywhere
8200       ALLOW     Anywhere
9000       ALLOW     Anywhere
19999     ALLOW     Anywhere
```

Imagen 55. Estado cortafuegos y puertos.

5.11.2. Seguridad en los backups

Para evitar el tan temido ransomware, que recordemos es un programa maligno que bloquea los archivos codificándolos y haciéndolos inservibles a menos que se obtenga vía pago el desbloqueo, se ha procedido en este proyecto a la creación de copias de seguridad en un dispositivo USB extraíble, de tal forma que una vez copiado, se puede extraer y guardar lo que se denomina copia en frío. De esta forma, los contenedores y su contenido estarán protegidos eventualmente y, en caso de infección, se podrá montar ágilmente con estas copias en este u otro servidor de contenedores por lo que el tiempo de recuperación de este desastre será bastante rápido.

Teniendo en cuenta que un pendrive es fácilmente hurtable, estas copias de seguridad a su vez se cifran con el protocolo AES de 256 bits. Este es uno de los cifrados más robustos actualmente ya que su clave al ser de 256 bits es muy larga en comparación con otros cifrados. Dicho de otra manera, será difícil que puedan romper y acceder a estos archivos.

5.11.3. Seguridad en el router cliente

El router siempre es una parte fundamental de la seguridad ya que es el punto final de nuestra red y la puerta a Internet. En la empresa Perrauri se contrató una línea de fibra con el distribuidor Jazztel, por lo que se hizo entrega de un router de nombre “Livebox+”. No habiendo presupuesto en este proyecto para sustituir por un router neutro, se va a configurar con las opciones disponibles para aumentar su seguridad.

Desde un navegador web, se accede por IP a la 192.168.1.1 siendo esta la puerta de acceso del router. En la Imagen 56 se muestra una captura de pantalla con la página principal de acceso al router.

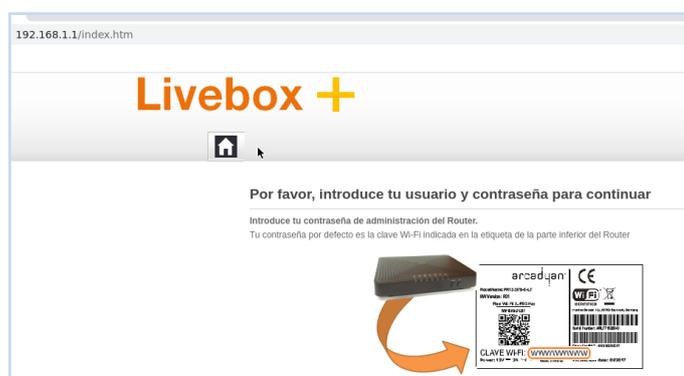


Imagen 56. Página principal acceso al router.

Lo primero de todo sería cambiar la contraseña por defecto ya que es la primera barrera de seguridad en la configuración del router.

Se cambia por tanto en configuración avanzada -> administración. Además, se activa la opción “Usar exclusivamente HTTPs en el acceso a la interfaz web” para de este modo vaya encriptada la comunicación en la comunicación con la página web de configuración.

Además, se van a activar las notificaciones por email en caso de que alguien acceda a la configuración, activándola en notificaciones por e-mail y luego en nuevo inicio de sesión a la GUI. Para ello, se introduce el email al que se enviará las notificaciones. En la Imagen 57 se muestra una captura de pantalla de la página web con las notificaciones activadas en el router.

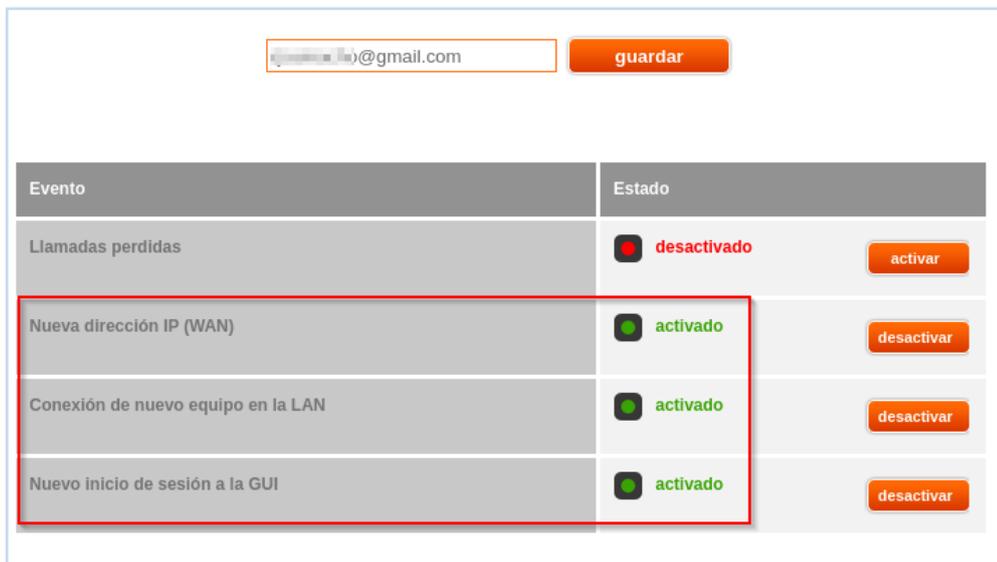
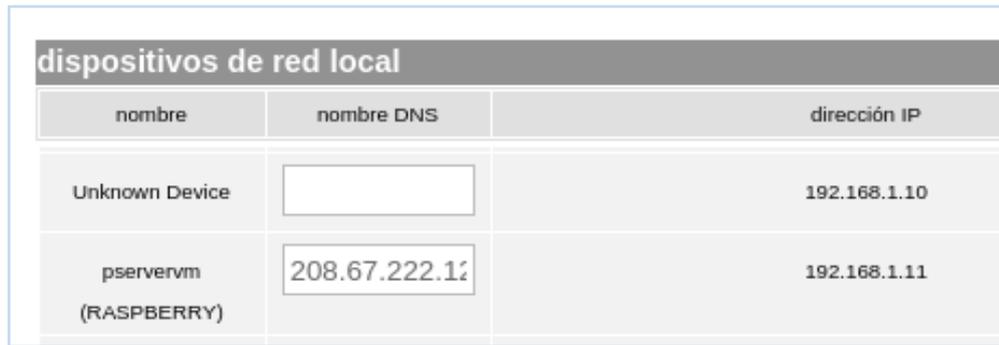


Imagen 57. Activación notificaciones acceso al router.

Se aprovecha además para que notifique de equipos nuevos en LAN, así como nos notifique de cambios en la dirección de salida WAN para que, en caso de fallar el DDNS, podamos acceder por la dirección IP.

Para protegerse de phishing y páginas web maliciosas, se va a cambiar las DNS's con las que se entrega el router (87.216.1.65 y 87.216.1.66) por las de OpenDNS, que tiene filtrado de sitios maliciosos y/o phishing gratuitamente y que se puede comprobar en la dirección web www.opendns.com. Se ponen por tanto las DNS's 208.67.222.123 y 208.67.220.123.



dispositivos de red local		
nombre	nombre DNS	dirección IP
Unknown Device	<input type="text"/>	192.168.1.10
pservervm (RASPBERRY)	208.67.222.123	192.168.1.11

Imagen 58. Cambio DNS servidor Raspberry.

En la Imagen 58 se muestra una captura de pantalla de la página web del router con la configuración DNS para la Raspberry Pi 4B

Como configuración extra para aumentar la seguridad o rendimiento del router.

- Se deshabilita el acceso remoto al router para que únicamente se pueda acceder desde la LAN, aprovechando OpenVPN en caso de que se esté fuera de la red.
- Se deshabilita el servidor de impresora y el servidor de archivos, ya que son servicios innecesarios en nuestra empresa tal y como se ha configurado.
- Aunque no es parte de este proyecto ya que los servidores y Raspberry están sobre cable ethernet, se configura la Wifi:
 - Se cambia el SSID y se oculta
 - Se cambia clave principal
 - El modo de seguridad pasa a ser WPA2 AES, debido a que AES es la opción más segura, debido a que AES es un protocolo más moderno y seguro que la otra opción que nos permite (TKIP)
 - Una vez se tengan los dispositivos con los que se va a trabajar por wifi (móviles y demás) se activa el filtrado MAC y se definen las MACs de los dispositivos que se van a permitir.
 - Se define un horario para la red wifi coincidiendo con el horario de apertura y cierre de Perrauri, dejando cierto margen de entrada y salida (1 hora). De esta forma, se apaga automáticamente la señal inalámbrica por lo que se minimizan ataques en este rango que no hay personal.

Con respecto a la apertura de puertos para poder acceder desde internet a los diferentes servicios de Perrauri vía traducción NAT de los mismos, se realiza apertura de puertos y su configuración mostrada en la siguiente Tabla 2, donde se muestran los puertos abiertos en el router cliente Perrauri su descripción y destino

Tabla 2. Puertos abiertos en el router cliente Perrauri.

PUERTO	DESCRIPCIÓN	PUERTO DESTINO
9000	ACCESO A PORTAINER	9000
1194	ACCESO VPN OPENVPN	1194
80	PAGINA WEB PERRAURI WORDPRESS	8008
8081	ACCESO NEXTCLOUD	8081
8085	ACCESO A PHPMYADMIN	8085
8200	ACCESO A DUPLICATI	8200

6. Presupuesto

En este apartado se desglosa el presupuesto de hardware, software, así como el laboral.

Con respecto al coste del Hardware se muestran en la Tabla 3, donde se muestran los costes de adquisición del hardware necesario para el proyecto:

Tabla 3. Costes hardware del proyecto.

UNIDADES	DESCRIPCIÓN	COSTE
1	RASPBERRY PI 4B 8GB RAM	82.51€
1	KIT LABIST PARA RASPBERRY (CAJA, VENTILADOR, CARGADOR, CABLE HDMI, DISIPADORES, ETC)	19.99€
1	PENDRIVE SANDISK 32 GB	4,99 €
1	TARJETA MICROSD SANDISK EXTREME 128 GB	19,99

Por tanto, el precio a nivel de hardware sería tan sólo de **127,48€**

Con respecto al Software:

Se ha utilizado en todo momento software libre para abaratar costes, por lo que el gasto de esta partida sería **0€**

Con respecto al coste laboral:

Se establece el precio por hora de servicios de un informático en 30€, la siguiente tabla 4 refleja el coste por partidas a nivel laboral del proyecto empleando un único trabajador.

Tabla 4. Costes laborales del proyecto.

HORAS	DESCRIPCIÓN	COSTE
5	BÚSQUEDA, ADQUISICIÓN HARDWARE Y MONTAJE	150€
50	CREACION CONTENEDORES Y DESPLIEGUE	1500€
25	REVISIÓN SISTEMAS, DISEÑO PÁGINA WEB Y BACKUPS	750€

Por tanto, el precio a nivel de costes de personal sería de **2400€**

El coste total del proyecto ascendería por tanto a un total de **2527,48€**

7. Conclusiones y recomendaciones

Se ha de reconocer que hoy en día con hardware poco exigente gracias a la virtualización ligera se puede hacer un despliegue bastante coherente de servicios. Además, teniendo en cuenta las posibilidades que nos permiten los contenedores de una migración a hardware más potente de manera fácil.

Se encuentran los siguientes beneficios a la hora de trabajar con docker en lugar de la instalación tradicional bajo servicios en un servidor:

- Se pueden desplegar contenedores con menos recursos, lo que facilita ahorro de hardware.
- Se pueden gestionar fácilmente servicios del mismo tipo independientes entre sí simplemente duplicando contenedores, lo que facilita tanto el testeado como configuraciones múltiples.
- El despliegue es robusto y rápido una vez se domina la técnica de imágenes y su código de implementación.
- La gestión con gestores de contenedores como “portainer” facilita mucho el arranque y parada de servicios, más que con un servidor clásico.
- Permite la migración a nube y a otros servidores ya de forma empaquetada sin apenas problemas.
- El aislamiento de las aplicaciones permite la identificación y reparación de problemas de manera más precisa.

A pesar de esto, hay que tener en cuenta la arquitectura para la que se despliegan algunos contenedores ARM en este caso.

Se han encontrado diversas dificultades a la hora de la implementación del proyecto. De un lado, la parte hardware se ha desarrollado sin complicaciones; no así como su plataforma, de tipo ARM, que hace que, a la hora de componer y diseñar los contenedores, haya que localizar imágenes adecuadas para una Raspberry de tipo ARM habiendo para diversidad de sistemas y no siempre teniendo disponible las deseadas. Con esto se tiene una visión fragmentada de Docker, ya que, al tratarse de virtualización ligera, no llega a ser de tipo universal como virtualización clásica, sino que hay que adecuarla a la plataforma desde la que se van a montar los contenedores.

De otro lado, se encuentra también dificultad en la localización de contenido y contenedores totalmente actualizados en cuanto a parches de seguridad. Es cierto que para muchos se aplican mejoras y actualizaciones, pero se observa que no al mismo nivel que si se tratase de la misma aplicación para el sistema operativo fuera de un contenedor. Esto lleva a estar más tiempo expuesto a vulnerabilidades y, al tratarse de una empresa digitalizada con presencia en internet y servicios WAN, conlleva un mayor riesgo.

Mi opinión es que, para empezar, recomendaría a la hora de instalar y desplegar servicios en una empresa pequeña por este tipo de tecnologías para abaratar costes y luego ir escalando en hardware más potente o bien migrar a la nube, pero siempre teniendo en cuenta el coste que supone mensualmente para una empresa pequeña.

Acerca de futuros trabajos, creo que se debería centrar en mejorar la seguridad de todos los contenedores, poniendo un proxy inverso entre medias y aplicando medidas de contención como un firewall, etc.

Además, para mejorar la estabilidad de la informática de la empresa, recomendaría tener las imágenes creadas subidas a algún repositorio de imágenes tipo GitHub o similar, de tal forma que, si hay problemas, se pueda recomponer rápidamente, además de facilitar la migración a plataformas totalmente Cloud, AWS, etc.

8. Bibliografía

- [1] 6 RAZONES DEL PORQUÉ CREAR UNA PÁGINA WEB <https://www.nextu.com/blog/6-razones-del-por-que-crear-una-pagina-web/> visitada 20/11/2020
- [2] Container Station: Docker y virtualización ligera <https://naseros.com/2015/11/23/container-station-docker-y-virtualizacion-ligera/> visitada 02/12/2020
- [3] Manejar GPIO Raspberry Pi | Aprendiendo Arduino <https://aprendiendoarduino.wordpress.com/2020/03/04/manejar-gpio-raspberry-pi/> visitada el 02/12/2020
- [4] Linux Monitoring CPU and temperatura – Unix & Linux Stack Exchange <https://unix.stackexchange.com/questions/419028/monitoring-cpu-and-temperature> visitada el 03/12/2020
- [5] FAQ: ¿Qué significa LTS? | Ubuntu Fácil <http://www.ubuntufacil.com/2014/03/faq-que-significa-lts/> visitada el 03/12/2020
- [6] Cómo cambiar el hostname en Ubuntu 18.04 <https://www.hostinger.es/tutoriales/como-cambiar-hostname-ubuntu> visitada el 07/12/2020
- [7] Portainer.io – Easily manage Docker containers in Linux <http://www.linuxandubuntu.com/home/portainer-io-manage-docker-containers-easily> visitada el 07/12/2020
- [8] Docker daemon doesn't start on boot on CoreOS <https://serverfault.com/questions/743087/docker-daemon-doesnt-start-on-boot-on-coreos/743097> visitada el 07/12/2020
- [9] Installing Docker and Docker Compose on the Raspberry Pi in 5 Simple Steps <https://dev.to/rohansawant/installing-docker-and-docker-compose-on-the-raspberry-pi-in-5-simple-steps-3mgj> visitada el 10/12/2020
- [10] Front end y back end – Wikipedia, la enciclopedia libre https://es.wikipedia.org/wiki/Front_end_y_back_end visitada el 30/12/2020
- [11] Configurar OpenVPN en docker sobre nuestra Raspberry Pi(Raspbian) -itcloud- @itcloud _es <https://itcloud.es/configurar-openvpn-en-docker-sobre-nuestra-raspberry-pi-raspbian/> visitada el 05/01/2021
- [12] linuxserver/duplicati - Docker Hub <https://hub.docker.com/r/linuxserver/duplicati> visitada el 14/01/2021

[13] How To Set Up a Firewall with UFW on Ubuntu 14.04

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-14-04> visitada el 09/03/2021

9. Anexos

Integro aquí los anexos correspondientes al proyecto que por motivos de opcionalidad no se han incluido en el texto principal.

9.1. Anexo 1. Texto codificación Open VPN

Se adjunta como texto plano el archivo generado para Openvpn con su encriptación para “perrauri”:

```
client
nobind
dev tun
remote-cert-tls server
remote perrauri.ddns.net 1194 udp

<key>
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIoF939SDB7KYCAggA
MBQGCCqGSIB3DQMHBAijrAMvz/IU0wSCBMhf6dbDeEFfm4wLzR24feO7YShKgDvK
KAzdoYKA6M/23iEM21EJOWOCWa5830iLC9Wrrz4Ae+duDwB/LOG9KjQR3v9ne9PS
EYP4zSvo/d4qG+WBJKp1/0wKfotMV2IMIEa0liFDRJihf0dc6lzuapbJda/NtOL1
fsISyBKCnWISpuzXUoLyQB8LM/1VaELFz4uqcx/Bqdlw4HkJmOEHISWel9OMjti
9flq8VQjGGP5jh0/WaAXB63c0mQWVbtIgpbh2tXoFLO/Z7qwqnnnciBNB2+vbjft
1VvlpBoxuJN08UIBELiD5s/6c3Aawl+yT6EejIAEM8aWybicEh4c/OooFANag0jb
dCn42Mo3lckGAn2JJZIV2VVSsn1U4EY0Qa2WVhiM+xRqGAYzkxB/Uc6POROiDWOtc
HkoUjrcAlOphePOSik2FyAZ2tr67CBEWLRf8Z4cRKwpBC2A8A5ZJ4wncRJst6zx
OFppom53rI3Wz3mTqjQCijwsF1ze2A9Z5sC0D9E2mqTvrQPSzqkE3wg/VuTxYRzx
hgUtRd0ge+hFzq5R74jADU6qr4vB5uJAEaeEz6hSd2zg+ASdVXXVvJj1rGT/LOjN
RPXjU9N5e+mKjtf5fKbp3vOBPCIfclJl8Z+twy8wfl3PioUtdPJ+UIxWcsZWUU38
BwWo6cJ9atnuamJXgffloRPSXOuGPbb6nKxSWxgTC4WQYVs3us7jyLaAQgj3yZaq
lZYjqM0Q4lQf42T6G5/P1LjX5Y20a9JXil11hgptla4b4tQz6zcSRf8Nr3AxvoYA
UvSTnJve0MY0FX1CQte2Gfp5jJZ/0bETxlqUhr1hb6bxxQoBWOWyeYfzZY5McSW
VQTVe/YO/dC4HwrB1jlhJA7cUrWfM+SWNtbwFn5eRkLB9y8oBs7FuDD+TMo34AjM
wW15eUFh36e3/3JUhcXNZtgT8jo7HV0NH0QP87kvih3csQuid30n4LFA6Y4cEbHk
J364op6nuR7JVvqGWq34TcN/XUnjLi9aqC9JfyWwOkEVGIC9uwmY8xKDugsnL0L4
tUaol+lmKXHKlBzP0w77M2TaiQZrAWBmOslo486zLJKOWam6aMcntSSIkKfM5YnC
1jAaDh/bwgzJni1rZoYF+yChs/autplQAQRByK07Nk6EbQ3HJeQDguFzYAYNiUWC
```

K4ud7lmCG1537UTCf5fDuNslIAV4vIRXQaMPVu3jtu/2mOXpSTWncsUWxmhsKBWU
+y7bAxc9Aa53A9mbvZf1H3yguvOqrKiLhoyuYhrdKxnXNvfvzswtyo0PvUO39xYp
u+kw62pZKFADSTZhSDMYQfWotgQwtEKzqRdmm9dKjn0Y6+UDvx+B4hwgSvQeG6zz
mCIT+vkNgPVDskNrltrZm00Jy7C+vC+A6qhh18Sv9BfcNylHPHaAi0jwvlnhzev
j0IA2gvXRzC10nM8aW3slnq0TzSXNrkYfgvB4RGzuPdHiYFLamCgFcn3JKLnuPp
5NVPEu5xtSHTFk4J2ISVgVk47Marx4MdSWhua1RldEqPYgMsRrhHqTx59NnYRDuW
2+fue46kGTtpoV9keDkrwgxVx6KeKOLnuzrDiaqtA6j9trfyBzcX4BI1W2Rxnnyl
UY4=

-----END ENCRYPTED PRIVATE KEY-----

</key>

<cert>

-----BEGIN CERTIFICATE-----

MIIDPDCCAiSgAwIBAgIBAJANBgkqhkiG9w0BAQsFADAWMRQwEgYDVQQDDAtFYXN5
LVJTQSBDQTAeFw0yMTAxMDUxMTM4NDFaFw0zMTAxMDMxMTM4NDFaMBMxETAPBgN
V

BAMMCHBlcnJhdXJpMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAv4y9
TwZfWgeSsN+J1KaX2+FZYqMkA1vVyXF6lGTiaaX8Xs6zbzlBxZbwSIHdOzYsZs5f
/TfrhlirAx4gzBHPTeiAKRPz+4LYlwLcrKorO7btYRaHWTdVTFyk1RSD6gyxMZy
JtU/CH74XpG8gCRbHahf2cknD5+HsEFK5yEiowZfwZgOSPts7YZ3gZggDmzimxds
qe7ISfRmlnHZ58tDCeZ5bd+0hfQ14rXzqyBqWPBAu6QL7jcEUhT+4yGX0EoLK1C
z1AUidn/2cdr1DkXn+caunem508FLlru8uusTM7dn/qdryQpBgtClr0ZRhMcmxsj
OeOIJu8nDjywbjhbOQIDAQABo4GXMIGUMAKGA1UdEwQCMAAwHQYDVR0OBBYEFESm
oLKMLCHQQ99ZNh2A8K0xDaQDMEYGA1UdIwQ/MD2AFKeMkY0oHe5oWFgBh4E8gJYC
K4+FoRqkGDAWMRQwEgYDVQQDDAtFYXN5LVJTQSBDQYIJAIBnTnVyLOQxMBMGA1Ud
JQQMMAoGCCsGAQUFBwMCMAAsGA1UdDwQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQE

A

b/I83v2XUDhgVjY1PczJlAl0kwz41tqd/FQI7ZvruEphxdLlWzXm883z4RvA56R8
VbHWHB7CcSk+dUQrIdwdKdjfVlbqPTH1/ulvGoJnTCMnKENfjfO4pqeqyxj4tylK
kSblMz/YC5JdBb9fZzsrMDMKdjwYtEqOMetymg2WrjTlKxLKdD4fw0XXXrLo3bum
AMenq9INs5QyRo8WtdPEUtRc0+DkelSWwHEneA7CHtCDfixXkWJGcKksDBjQQ3Ti
JP/7BINrp4oNRyhlPjsvr2IJ3nL+dKLNf7qm0DkEsogHoekIIH7548YtEEueatN
zDEuVBGebOpdMa9JmjKnMA==

-----END CERTIFICATE-----

</cert>

<ca>

-----BEGIN CERTIFICATE-----

MIIDNTCCAh2gAwIBAgIJAIBnTnVyLOQxMA0GCSqGSIb3DQEBCwUAMBYxFDASBgNV
BAMMCOVhc3ktUINBIENBMB4XDTIxMDEwNTEwMjQxOVoXDTMxMDEwMzExMjQxOVoVow
FjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQDQHCEmks+VqP8DSCD6pYE3dVuw8GdWd1vHFhgbT1XnpKIKCejWoCu1
nXT00lqdxm+MDuc5u7+tWvbOqMUQn9F37pdIuAw7unVRT+ge7QseUVUQAy23/qR2

75blgcbJPiKLwpkOvGI90cdovZ4ZzfmzlgvOYBY1I3WtG/4PI2qIWirpXX6K0EwD
9XU95ea/Gkr6n0033xdTqk/njGLJNyZEhd8caKZswU3KsfM8jIWTvt19A0ONFV9I
wNu3ZYj4cbcTVzrjH4fiEmYdrcFehPOFss70GToU4u+BcCvQExC20uwspl+t943g
KCm1Y4e5Nr0ue/5Dv7oDen442nCmD+P5AgMBAAGjYUwgYlWlHQYDVR0OBBYEFKeM
kY0oHe5oWFgBh4E8gJYCK4+FMEYGA1UdlwQ/MD2AFKeMkY0oHe5oWFgBh4E8gJYC
K4+FoRqkGDAWMRQwEgYDVQQDDAtFYXN5LVJlTQSBQYIJAIBnTnVyLOQxMAwGA1Ud
EwQFMAMBAf8wCwYDVR0PBAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQC79NjM845f
p0szldYjmFGtABkV80k3w5ULThQ37gM8x1pzLolor+hmcl48ld2KTGgJD6I4x6AM
Lj2Z28hy1k8aDRUSCP0DXx6suSgt8JruVpVPp0UjXP3Q1Zyzny3qYJN0IX7Dh3X3
OivvJy+gZ94p6FqpwVPHMa6RUIqJVK0hIwHuNBEWvYDgOPzka6VljKDoNG8v+N3e
fTF9/Zt/a/VWRUPBiP5DEtUmll3B/U4/9WjUwAoSKsBb0ukzq4y+/PVaS0PFbsvZ
1KPtDzBJPBJ32sQvO6nFTDi/rY3Lpf5rDmo7lI15b7teefAS7XclfbFzRc1a4zUB
ObOF0HW8BuPB

-----END CERTIFICATE-----

</ca>

<tls-auth>

#

2048 bit OpenVPN static key

#

-----BEGIN OpenVPN Static key V1-----

a8378b3fd7d2c280f280bf404eb5148d
d3607393a4486c44b5432bbf05bd0069
7e825e8e05028236ecf06cbf42339dfb
7dec396cd80b0ae605d95661d8567f03
7659e9268d5a4f5f3519588f782c74aa
4773d9f23c0f8c4d57791cff3c31f00a
9bb35c0adc4e1f44cc47f64d661a607a
08b80aab61607b3f6687ad9ddfb4c468
31b872ae119516bba324e30d0ce831c4
41b8f1cbfd2c812cba0a7c48984e6935
95970c4c5fad3f484f126dae4d99562d
0c68b128b1c947c2e5d869a9d350b04f
796a56b537fcf16e1338047b027838d0
77a001d155ab216d109c016aaff0c33a
e0dce3e4e24d6b5b0e50e0ac86e76cbd
bdc201245d50488d3d465242eb6d5afa

-----END OpenVPN Static key V1-----

</tls-auth>

key-direction 1

redirect-gateway def1

9.2. Anexo 2. Curriculum Vitae



Rafael Jimenez Camacho

(+34) 628702671 | rjcamacho@gmail.com | Skype: rjcamacho80 |

28814, Daganzo de Arriba, España

EXPERIENCIA LABORAL

IT – DEPARTAMENTO DE SOPORTE – SERVICIOS INTEGRALES UNITECNIC SL

1 ENE 18 – ACTUAL – Madrid, España

Tareas de soporte dpto. de informática en cliente Globomedia Imagina Media.

Soporte en:

Video: AVID, Nexis, MAM.

Redes: Electrónica de red en capas 1-3 OSI en Cisco y Dell Powerconnect principalmente.

Sistemas: Servidores Windows - Linux. AD, DHCP, DNS, SMB, etc. NAS

Helpdesk: Incidencias nivel dos en entornos Windows - Linux - Mac OS

IT - DPTO. DE INFORMÁTICA – GLOBOMEDIA - IMAGINA MEDIA

1 ENE 09 – 31 DIC 17 – Madrid, España

Funciones de sistemas: Windows Server / Linux + AD + Backup Exec + Amazon S3 / Glacier + pfSense + Nagios, VMware ESX, Hyper-V, etc.

Funciones de redes: Redes de infraestructura: capas 1, 2 y 3 de OSI en hardware CISCO, HP, Dell, Dlink, etc. cobre y panel de parcheo de fibra.

Gestión de QoS, VLAN's, entorno corporativo WiFi, etc.

Funciones de incidencias, de forma remota y física. CAU Nivel 1: Incidencias bajo el sistema de tickets OTRS a nivel de usuario, Windows, VPN, RDP, etc. remoto. Gestión de telefonía PBX Samsung, VoIP, Trixbox

Trabajando eventualmente en oficinas ubicadas en Zaragoza, Barcelona, Sevilla, Valencia, Bruselas.

IT - NETWORKING – MEDIAPRO

27 FEB 06 – DIC 08 – Pozuelo de Alarcón, España

Departamento de redes - networking.

Mantenimiento y despliegue de infraestructura de red CISCO en entorno OSPF, telefonía IP, etc. Firewall: Nokia Firewall, SonicWall

Servidores: RADIUS, SNMP, LOG's

ENCARGADO DEPARTAMENTO DE VENTAS INFORMÁTICA – CARREFOUR

NOV 02 – FEB 06 – Majadahonda, España

Encargado departamento de ventas sección informática y videojuegos.

EDUCACIÓN Y FORMACIÓN

MÓDULO MEDIO DE SMR - SISTEMAS MICROINFORMÁTICOS Y REDES – Tres Olivos

1 SEP 13 – 30 JUN 15 – Madrid, España

CISCO CNAP – Cas-Training

07 – 08 – Madrid

ITIL FOUNDATION V3 REV.2011 – PEOPLECERT

OCT 17 ID: GR750341077RJ

ISO 20000 – EXIN

ID: EXN4749222

COMPETENCIAS DE IDIOMAS

Inglés nivel: B1

COMPETENCIAS DIGITALES

Servidores

Windows Servers, ACTIVE DIRECTORY, Linux (Debian, Fedora, Ubuntu)

Conocimientos de entornos Virtualizados (VirtualBox, Vmware, Hyper-V).

Instalación y configuración de servicios de red (DHCP, DNS, FTP, correo)

Escritorio

Programa de gestión de clientes (OTRS)
Ticketing Systems Jira
Soporte IT Flexibilidad con diversos O.S. (Windows, MacOS, Linux)
Ofimática entorno Office

Redes

CISCO - CCNA Routing and Switching
Mantenimiento, soporte, diseño e implementación de redes

PUBLICACIONES

[LIBRO ANÁLISIS DEL MERCADO DE PRODUCTOS DE COMUNICACIONES UF1869](#)

ISBN- 978-84-16173-58-7 Año: 2014

Publicación Informática:
Libro formativo UF1869 ajustado a certificado de profesionalidad
"Análisis del mercado de productos de comunicaciones"
Autor(es): Rafael Jiménez Camacho
Páginas: 318
ISBN: 978-84-16173-58-7
Editorial publicación: IC Editorial

[CURSO TALLER PERSONALIZACIÓN Y RECUPERACIÓN DE MÓVILES BAJO ENTORNO ANDROID.](#)

Semana de la ciencia 2014 Tres Olivos – Ponente

HABILIDADES LABORALES

Trabajo en equipo y adaptación, proactivo.

He trabajado en multitud de equipos, a veces colaborando físicamente con diferentes empresas en Europa incluyendo diferentes ubicaciones en España, Francia, Bélgica y en ocasiones de forma telemática en Argentina, Portugal, USA, EAU, China, etc.