

6.L Servicios Web en Linux (Apache)

- No estudiaremos cómo crear contenido, sino aspectos relativos a:
 - Instalación
 - Configuración
 - Seguridad
 - Administración

6.L Instalación

- `yum install httpd`
- `firewall-cmd --permanent --add-port=80/tcp`
- `firewall-cmd --add-port=443/tcp`
- `firewall-cmd --reload`

6.L Arranque

- Iniciar manualmente

```
systemctl start httpd.service
```

- Para que arranque al iniciar el servidor

```
systemctl enable httpd.service
```

- Para comprobar el status del servicio

```
systemctl status httpd.service
```

- Para detener el servicio

```
systemctl stop httpd.service
```

6.L Comentarios

- El servicio se llama “httpd”
- No es un único proceso; se lanza una batería de procesos (comprobar con `ps | grep httpd`)
- El paquete instala un fichero `httpd.conf` de ejemplo con una configuración mínima. Una vez instalado, al acceder a <http://localhost> se muestra una página de prueba
- No es necesario que el servidor esté en modo gráfico para servir páginas web

APACHE HTTP SERVER

Test Page

This page is used to test the proper operation of the [Apache HTTP server](#) after it has been installed. If you can read this page it means that this site is working properly. This server is powered by [CentOS](#).

Just visiting?

The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name “webmaster” and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to webmaster@example.com.

Important note:

The CentOS Project has nothing to do with this website or its content, it just provides the software that makes the website run.

6.L httpd.conf

- El fichero httpd.conf contiene un gran número de directivas que indican dónde se alojan en el filesystem los contenidos, los scripts, los tipos de contenido (estático, activo) que se sirven, las opciones de seguridad, los hosts virtuales (un mismo servidor puede alojar páginas con diferentes direcciones IP), los alias (diferentes directorios en la dirección de la página pueden referirse a localizaciones no relacionadas con esos nombres), el tipo de encriptación, etc.
- También se puede especificar que el servidor sea proxy (redirija las consultas a otras páginas), que haga caché de los contenidos, etc.
- Gran parte de las posibilidades del servidor están disponibles a través de módulos que se cargan dinámicamente

6.L Directivas Básicas de Configuración

- Las directivas básicas son
 - ServerAdmin (email del responsable)
 - ServerName (p.e. www.midominio.com)
 - ServerRoot (directorio donde se almacenan las páginas)
 - ServerType (standalone, en general)
 - Port (generalmente el 80)

6.L Manejo de procesos

- Apache no crea un proceso por cada consulta, para no sobrecargar el sistema. Se lanzan varios procesos independientes y se balancea la carga entre ellos. Hay un límite de transacciones simultáneas.
 - *MinSpareServers, MaxSpareServers*: mínimo/máximo número de procesos inactivos
 - *StartServers*: Lanzados inicialmente
 - *MaxRequestWorkers*: Máximo número de conexiones que se procesarán de forma simultánea
 - *MaxConnectionsPerChild*: máximo número de conexiones que un proceso hijo manejará durante su vida

6.L Configuración inicial

1. Backup del archivo de configuración por defecto:

```
cp /etc/httpd/conf/httpd.conf ~/httpd.conf.backup
```

2. Modificar httpd.conf para que apunte al directorio que contiene los datos del servidor web y para que tenga los parámetros deseados de número de procesos

```
DocumentRoot "/var/www/html/example.com/public_html"
```

```
...
```

```
<IfModule prefork.c>  
    StartServers 5  
    MinSpareServers 20  
    MaxSpareServers 40  
    MaxRequestWorkers 256  
    MaxConnectionsPerChild 5500  
</IfModule>
```


6.L Directivas frecuentes

- La directiva *ServerAdmin* indica la dirección de contacto del administrador
- La directiva *ServerName* indica el nombre del servidor y el puerto en que se establece la comunicación
- La directiva *ServerAlias* indica un nombre alternativo para el servidor (se usa para definir hosts virtuales)
- La directiva *DocumentRoot* define el directorio en que se almacena el árbol de documentos: p.e. *DocumentRoot* “/var/apache/htdocs”
- La directiva *ErrorLog* define la localización donde se almacenan los mensajes de error (se puede completar con la directiva *ErrorLogFormat*)
- La directiva *CustomLog* define la localización de almacenamiento de los mensajes de traza (en combinación con la directiva *LogFormat*)

6.L Directivas de traza (logging)

- Las directivas de trazado determinan que se almacena ante cada consulta y cada error.
- Se puede hacer que ante ciertos eventos se almacenen mensajes en varios archivos diferentes (p.e. %404{User-agent}i)

```
ErrorLog /var/apache/logs/error_log
```

```
LogLevel warn
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

```
CustomLog /var/apache/logs/access_log common
```

6.L Ejemplo de uso de directivas: Hosts virtuales

1. En el subdirectorio conf.d se crea un archivo /etc/httpd/conf.d/vhost.conf para almacenar las configuraciones de los hosts virtuales:

```
NameVirtualHost *:80
```

```
<VirtualHost *:80>  
    ServerAdmin webmaster@example.com  
    ServerName example.com  
    ServerAlias www.example.com  
    DocumentRoot /var/www/html/example.com/public_html/  
    ErrorLog /var/www/html/example.com/logs/error.log  
    CustomLog /var/www/html/example.com/logs/access.log combined  
</VirtualHost>
```

6.L Alias

- Un alias asocia un directorio en el nombre de la página con un directorio en el sistema de archivos del servidor

```
Alias /icons/ "/var/apache/icons/"
```

```
Alias /manuals/ "/usr/apache/htdocs/manual/"
```

```
ScriptAlias /cgi-bin/ "/var/apache/cgi-bin/"
```

6.L Contenedores

- Para aplicar diferentes directivas solamente a un directorio, se crea un contenedor para esas directivas, mediante las órdenes que siguen:
 - `<Directory pathname>` crea un contenedor con instrucciones hasta el próximo `</Directory>` (directorios del filesystem)
 - `<Location document>` crea un contenedor con directivas aplicables a un documento (una página, que puede contener varios archivos)
 - `<Files filename>` un fichero determinado

6.L Contenedores (II)

- *Order* define cómo se evalúan las reglas de control de acceso
- *Deny/Allow*: hosts prohibidos (por IP). Deny all indica que todos los hosts están prohibidos salvo por las excepciones indicadas después en allow

```
<Directory "/var/apache/htdocs/internal">  
    Order deny,allow  
    Deny from all  
    Allow from hostpermitido.com  
</Directory>
```

6.L Opciones de Seguridad

- Las opciones de configuración de seguridad de cada directorio se definen mediante la orden *AccessFileName* *.htaccess*. Si se encuentra el fichero *.htaccess* en un directorio, se puede indicar en ese fichero quién puede acceder a esa página y opcionalmente incluir usuario y contraseña.
- La directiva *AllowOverride* permite que un fichero *.htaccess* altere la configuración de seguridad por defecto de ese directorio

6.L Ejemplo de directiva <Directory>

1. En el ejemplo siguiente, se habilitan los índices en los los archivos del directorio indicado y en sus subdirectorios:

```
<Directory "/usr/local/httpd/htdocs">  
    Options Indexes FollowSymLinks  
</Directory>
```


6.L Ejemplo de control de acceso de usuarios

```
<Directory "/var/apache/htdocs/internal/accounting">  
  AuthName "Accounting"  
  AuthType Basic  
  AuthUserFile /etc/apache/http.passwords  
  AuthGroupFile /etc/apache/http.groups  
  Require hdqtrs rec bill pay  
  Order deny,allow  
  Deny from all  
  Allow from Limit>  
</Directory>
```

- Contraseñas para acceder a recursos. Con diferentes módulos, se puede combinar la autenticación con la del sistema.

6.L Proxy servers y caching

- Los proxies son servidores intermedios entre los clientes y los servidores web
- Las opciones que controlan el caching son:
 - CacheNegotiatedDocs (permite a un proxy que almacene nuestras páginas)
 - ProxyRequests (poner a on para convertir el servidor en un proxy)
 - CacheRoot, CacheSize: directorio de almacenamiento de cache
 - etc.

6.L squid

- **squid** es un proxy+cache que ofrece control de acceso, autorización y logging. También puede configurarse como un proxy inverso, para redireccionar las conexiones al puerto 80. Cuando actúa como un proxy inverso, el contenido cacheado se sirve desde el servidor proxy al cliente sin exponer el origen de los datos en la red interna.
- Instalación: **sudo dnf install squid**
- Ejecución: **sudo systemctl enable --now squid**

6.L squid

```
#
# Recommended minimum configuration:
#
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
acl localnet src 10.0.0.0/8           # RFC 1918 local private network (LAN)
acl localnet src 100.64.0.0/10        # RFC 6598 shared address space (CGN)
acl localnet src 169.254.0.0/16       # RFC 3927 link-local (directly plugged) machines
acl localnet src 172.16.0.0/12        # RFC 1918 local private network (LAN)
acl localnet src 192.168.0.0/16       # RFC 1918 local private network (LAN)
acl localnet src fc00::/7            # RFC 4193 local private network range
acl localnet src fe80::/10           # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80               # http
acl Safe_ports port 21               # ftp
acl Safe_ports port 443              # https
acl Safe_ports port 70               # gopher
acl Safe_ports port 210              # wais
acl Safe_ports port 1025-65535       # unregistered ports
acl Safe_ports port 280              # http-mgmt
acl Safe_ports port 488              # gss-http
acl Safe_ports port 591              # filemaker
acl Safe_ports port 777              # multiling http
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

"/etc/squid/squid.conf" 76L, 2553C
```

El archivo de configuración es
/etc/squid/squid.conf

Cada lista de control de acceso (ACL) consiste en un nombre, un tipo y un valor. Por ejemplo, para configurar hosts en el segmento 192.168.10.0/24 se haría:

**acl miredlocal src
192.168.10.0/24**

Esto crea una ACL llamada miredlocal que especifica los hosts de dicha red

6.L squid

- Una vez se ha definido una ACL, se puede referenciar para permitir o denegar el acceso a una función del cache. Por ejemplo:

http_access allow miredlocal

- El fichero de configuración se lee de arriba a abajo y el orden de las directivas es relevante
- Para bloquear el acceso a sitios específicos puede crearse un archivo que define los dominios bloqueados, y añadir una ACL indicándolo:

acl sitiosbloqueados dstdomain "/etc/squid/sitios-bloqueados.squid"

http_access deny sitiosbloqueados

http_access allow miredlocal

6.L squid

- También puede denegarse el acceso mediante keywords. Se crea un fichero con palabras prohibidas:

```
vi /etc/squid/palabras-prohibidas.squid
```

```
juego
```

```
poker
```

```
apuestas
```

- y se añade al archivo de configuración:

```
acl sitiosbloqueados dstdomain "/etc/squid/sitios-bloqueados.squid"
```

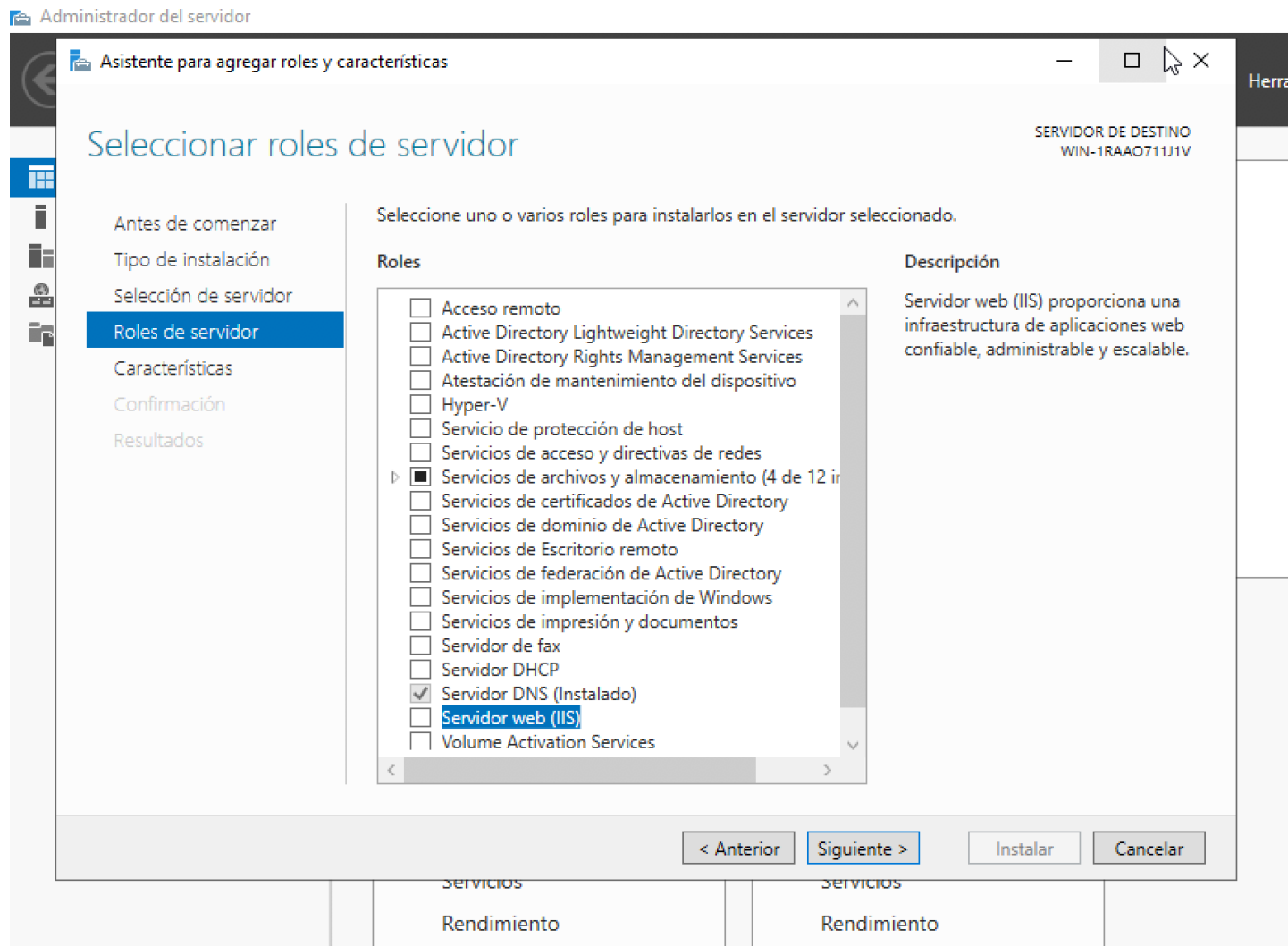
```
http_access deny sitiosbloqueados
```

```
http_access deny palabras-prohibidas
```

```
http_access allow miredlocal
```

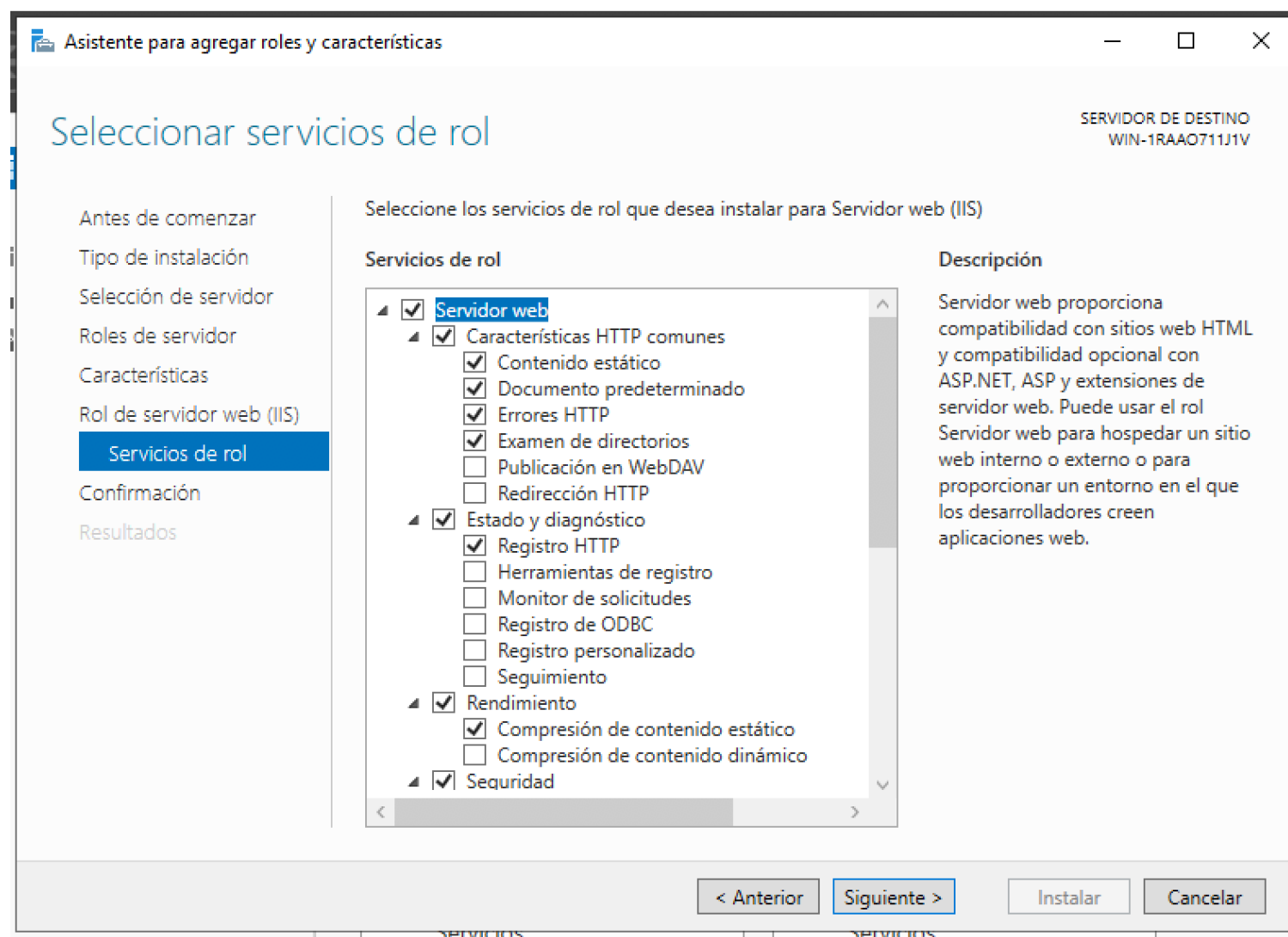
6.W: Internet Information Server

- Se selecciona el rol "Servidor web"



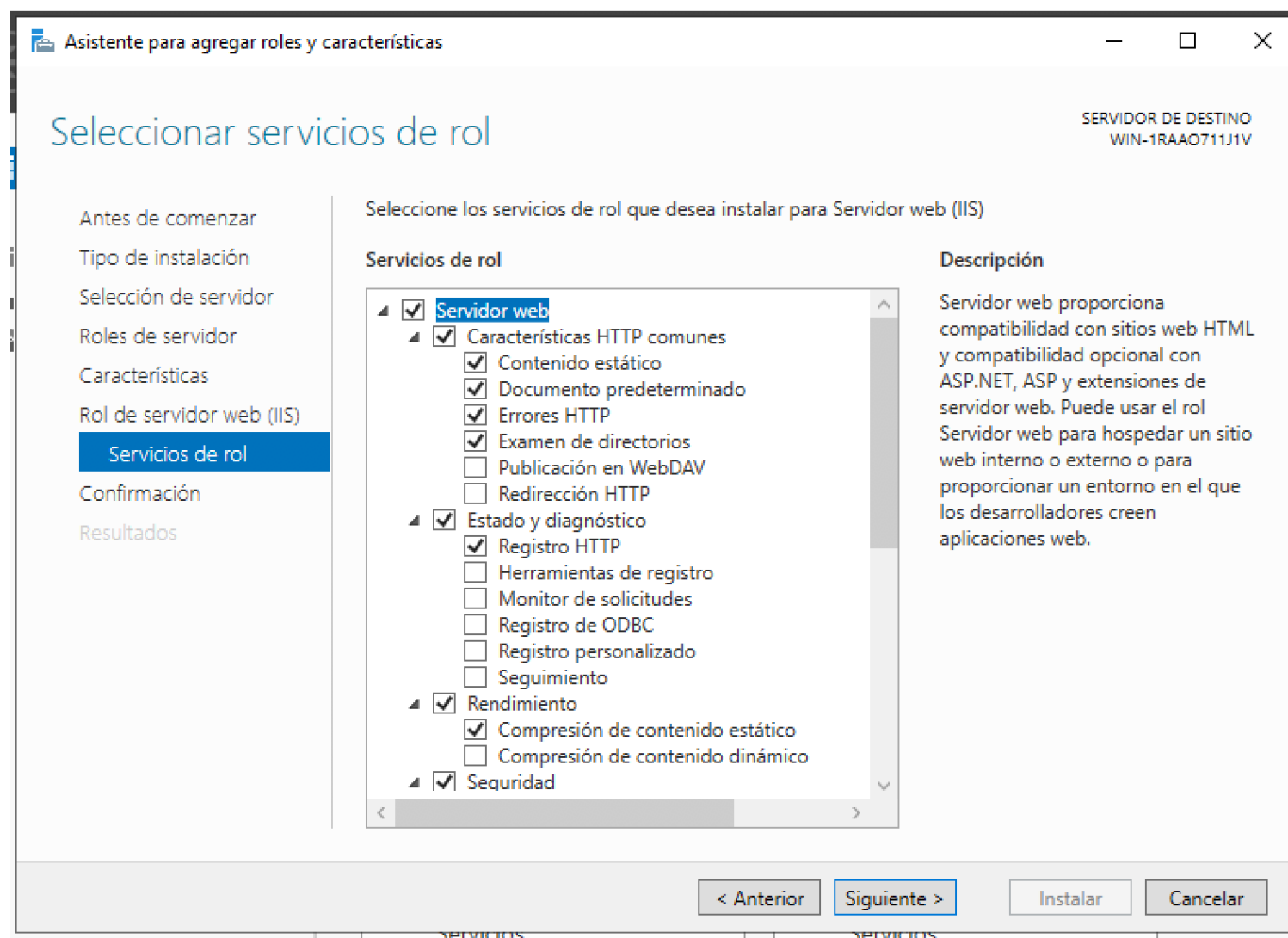
6.W: Internet Information Server

- Se seleccionan los servicios de rol deseados



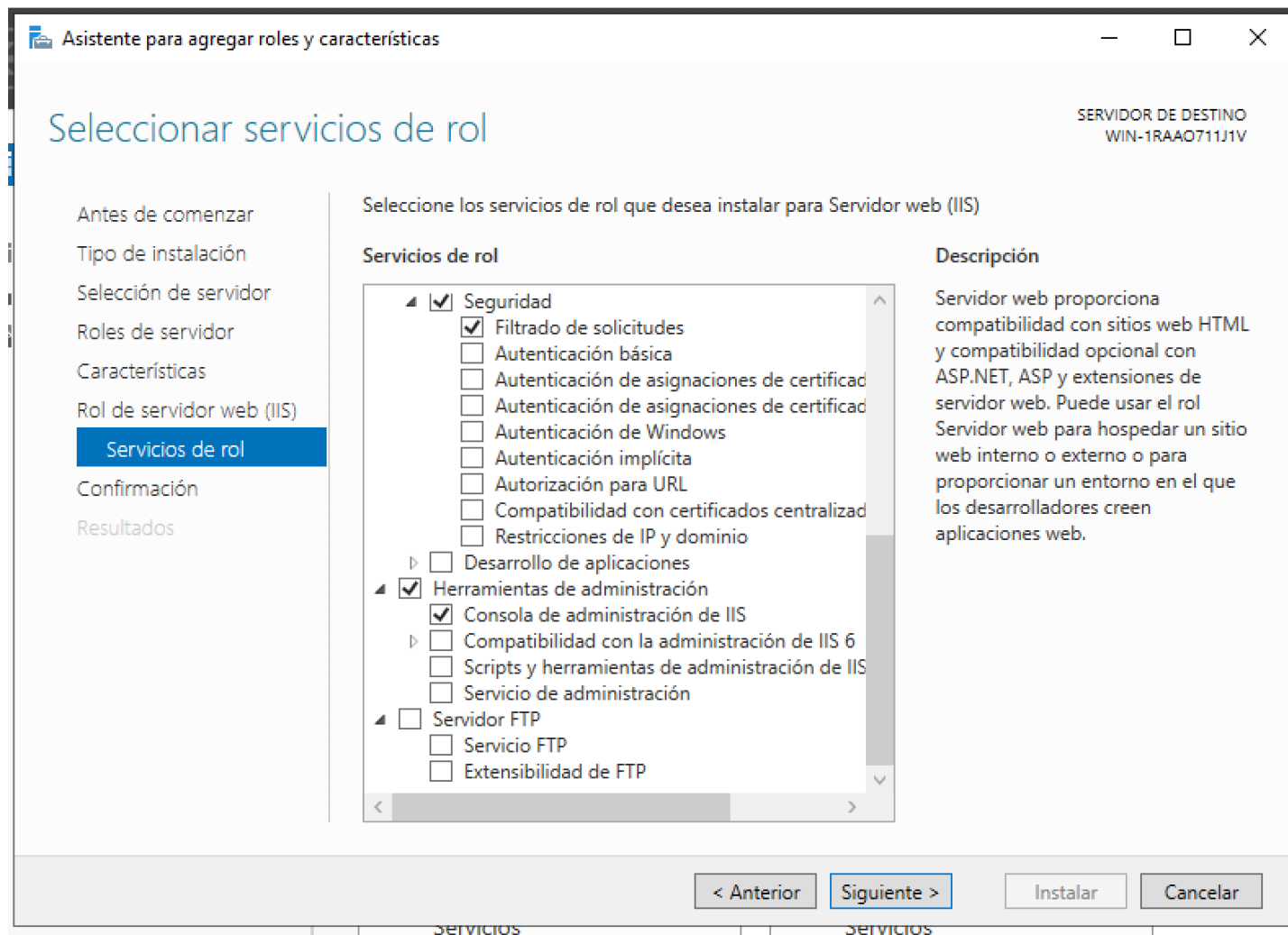
6.W: Internet Information Server

- Se seleccionan los servicios de rol deseados



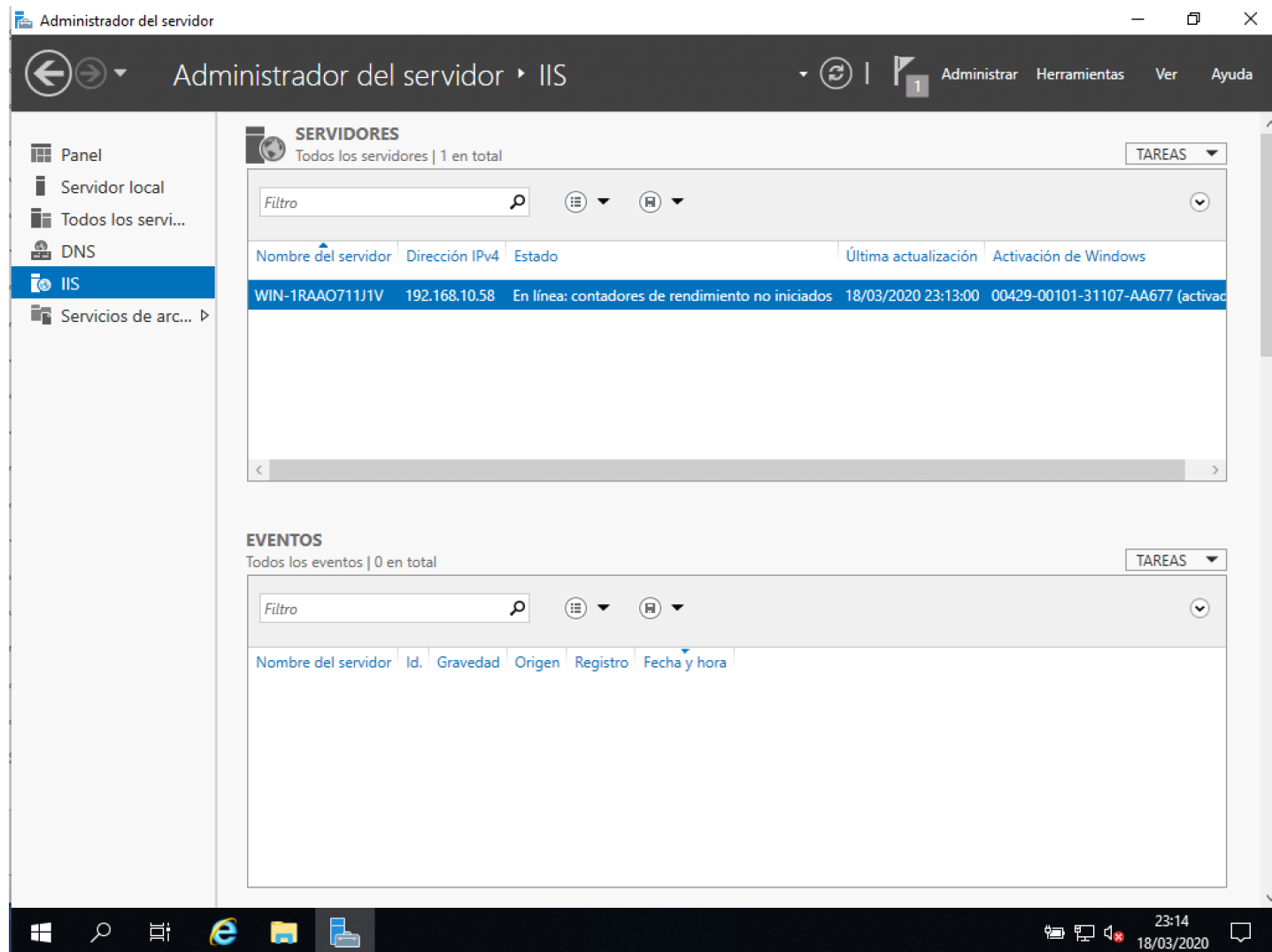
6.W: Internet Information Server

- Se incluye la consola de administración y opcionalmente el servidor FTP

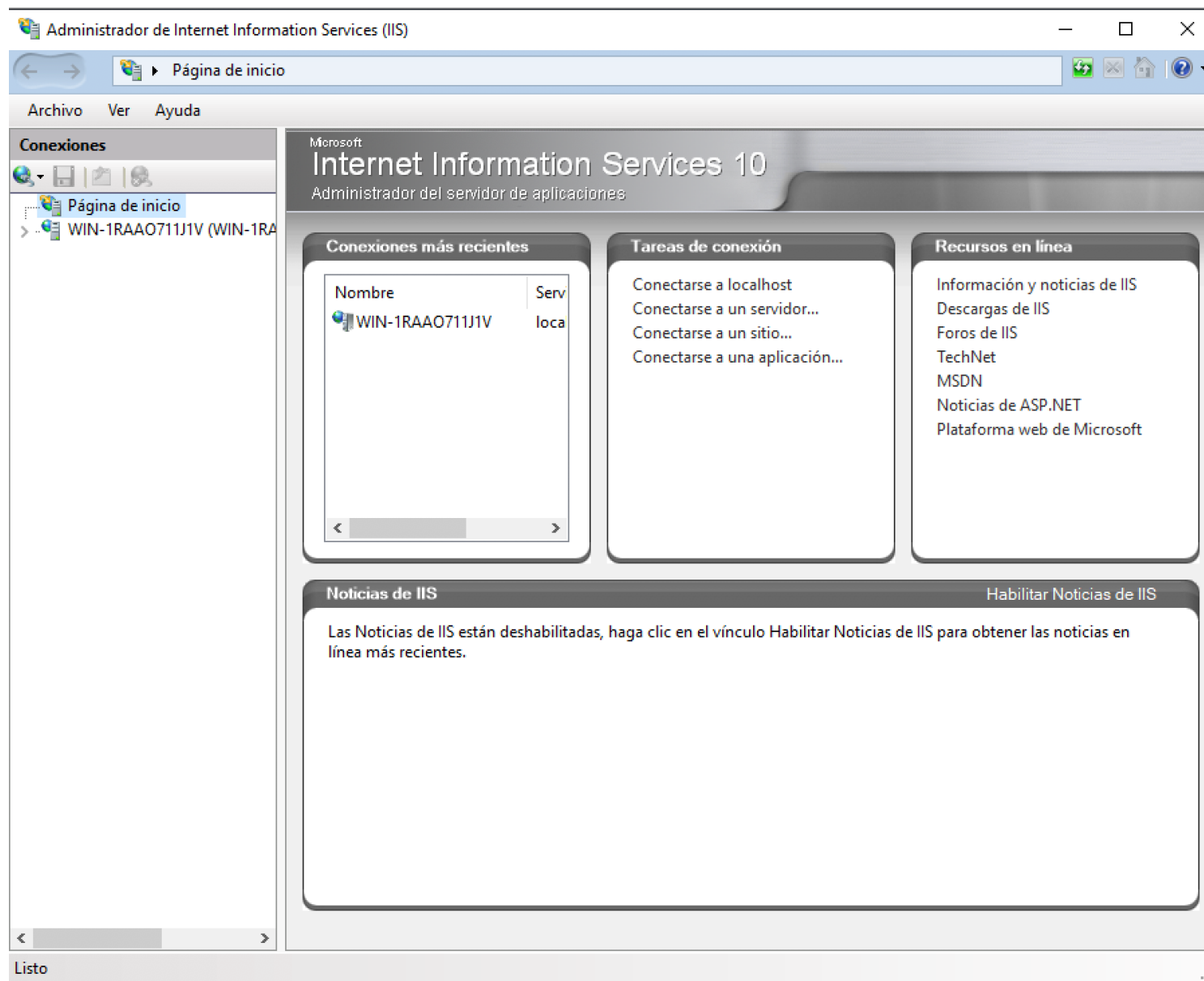


6.W: Internet Information Server

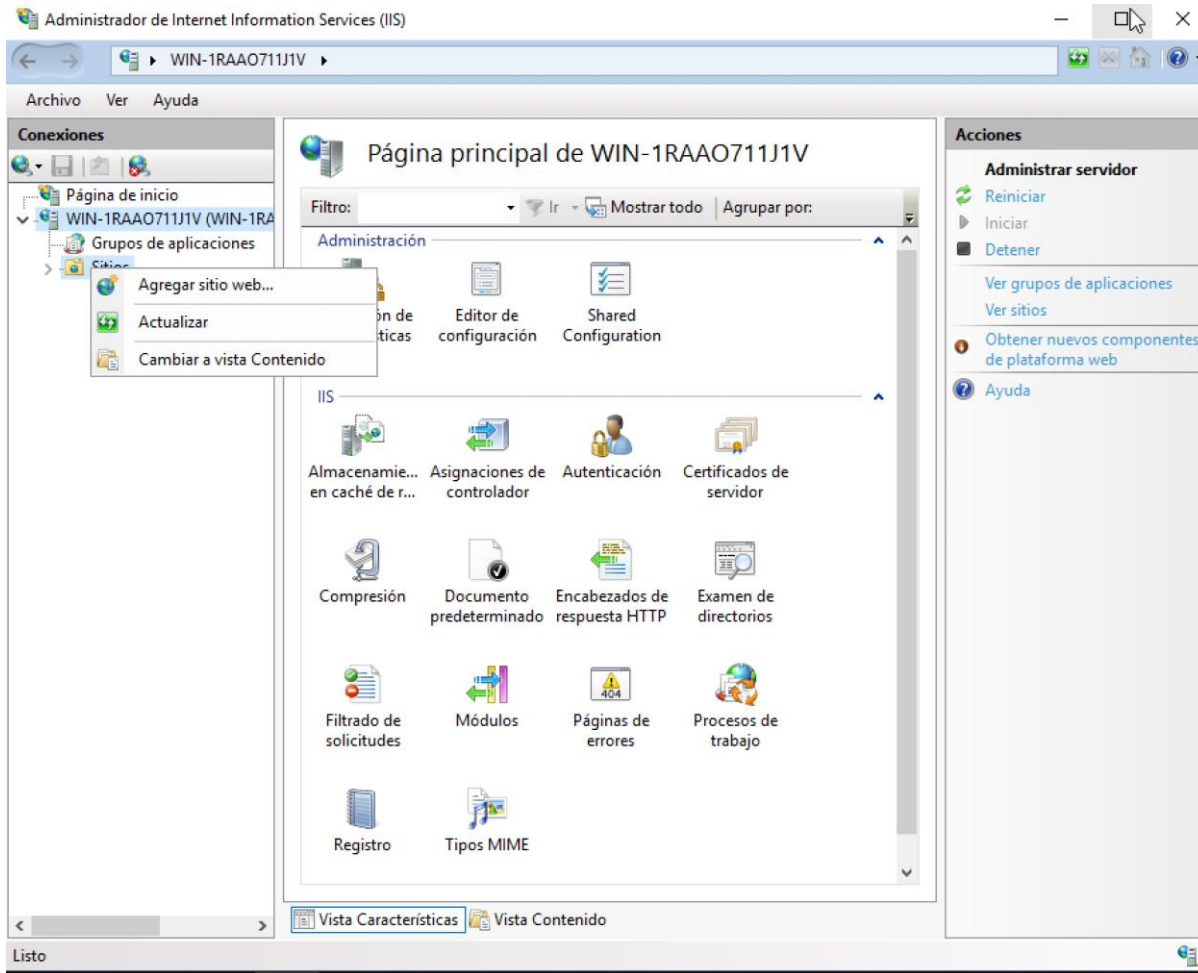
- IIS desde administración del servidor



6.W. Administrador de IIS



6.W Agregar nuevo sitio



- Botón secundario del ratón en el nodo "Sitios"
- Se indica el directorio donde se almacenan los archivos
- Se indica la dirección IP y el nombre del host

6.W Agregar nuevo sitio

Agregar sitio web

Nombre del sitio: MiNuevoSitio

Grupo de aplicaciones: MiNuevoSitio Seleccionar...

Directorio de contenido

Ruta de acceso física: C:\MiNuevoSitio ...

Autenticación de paso a través

Conectar como... Probar configuración...

Enlace

Tipo: http Dirección IP: 192.168.10.58 Puerto: 80

Nombre de host: www.minuevositio.com

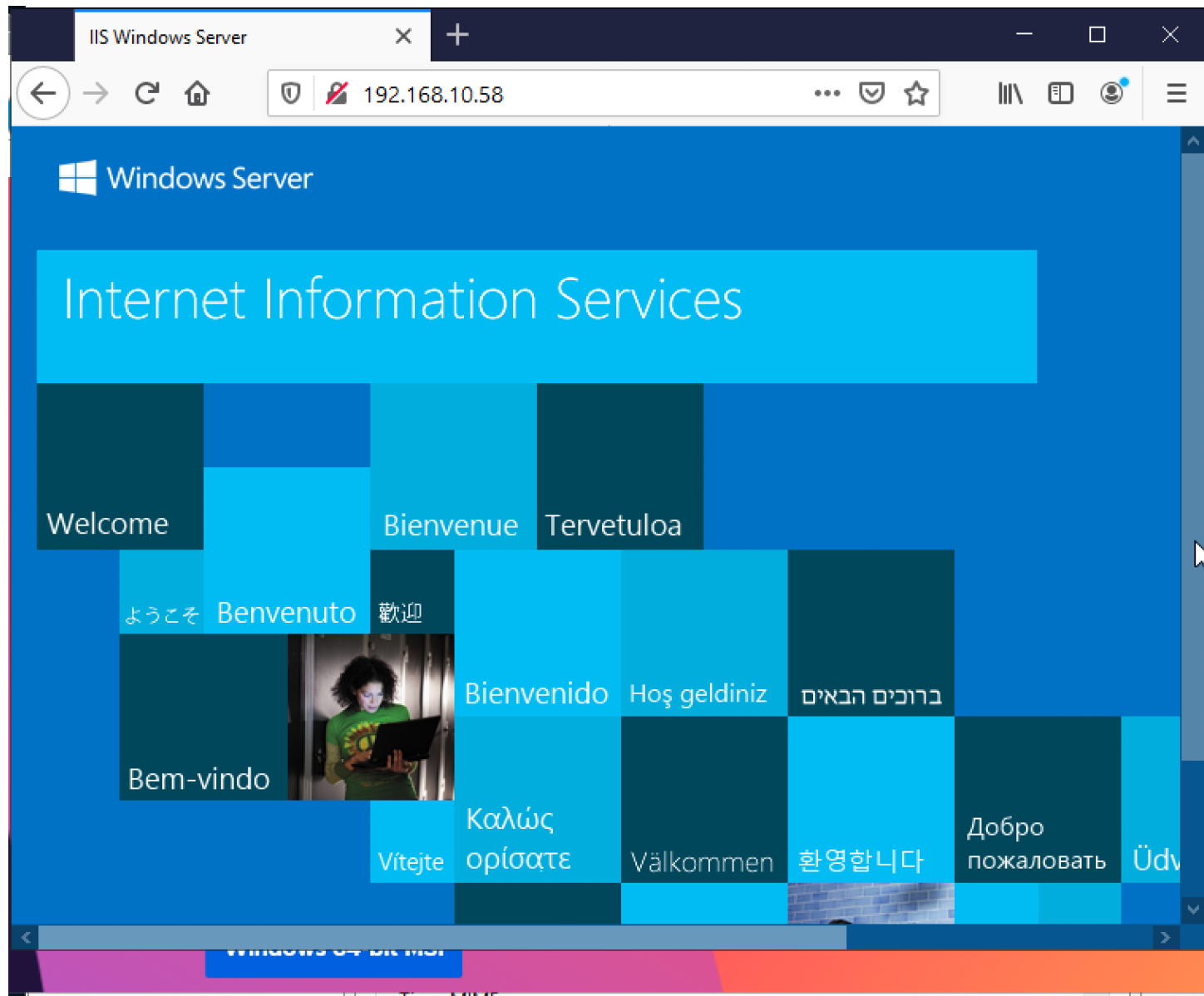
Ejemplo: www.contoso.com o marketing.contoso.com

☒ Iniciar sitio web inmediatamente

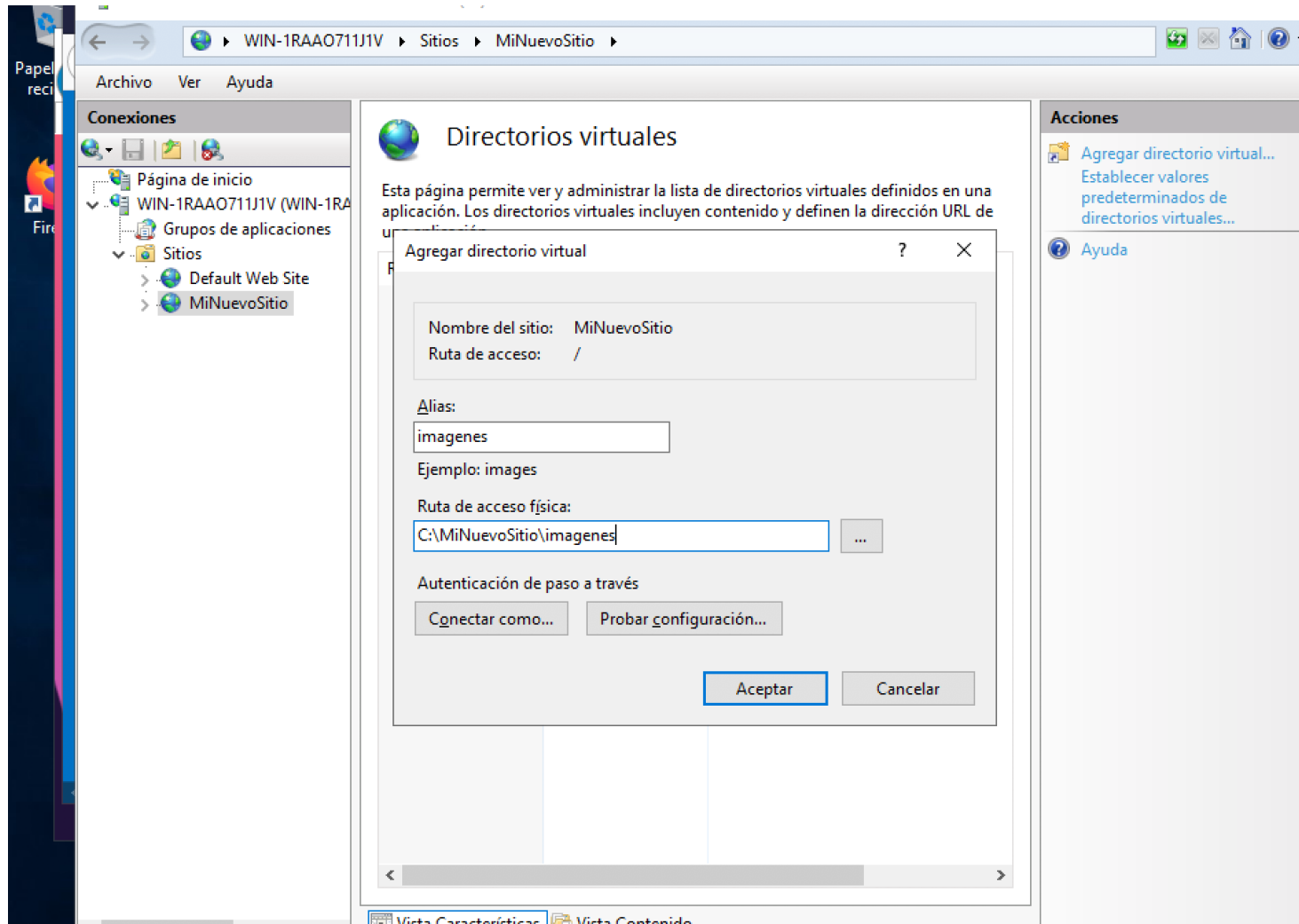
Aceptar Cancelar

- Botón secundario del ratón en el nodo "Sitios"
- Se indica el directorio donde se almacenan los archivos
- Se indica la dirección IP y el nombre del host

6.W Agregar nuevo sitio



Directorio Virtual

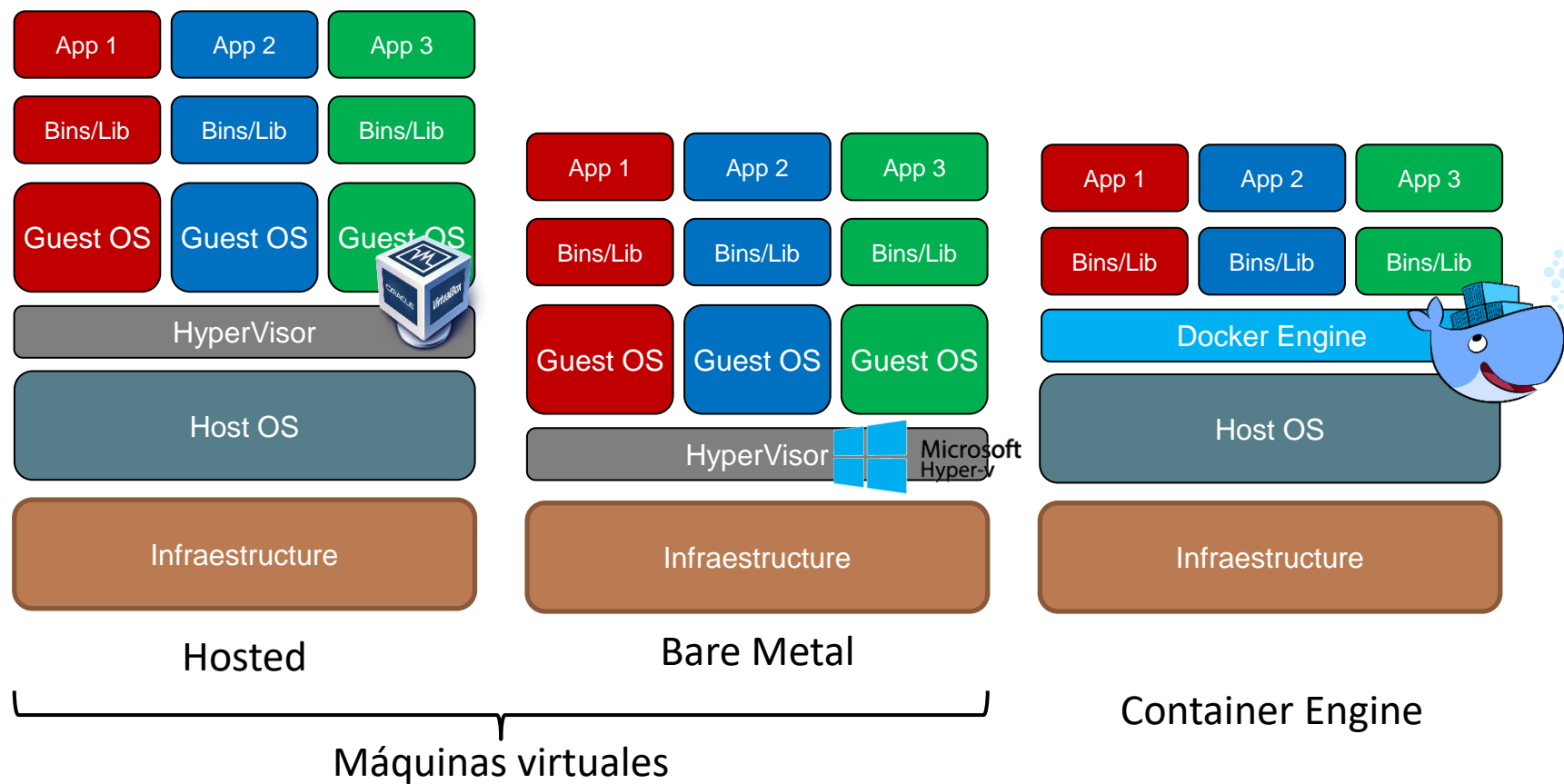


- Los alias (un nombre alternativo para un directorio no contenido en el directorio principal) se crean también desde el nodo sitios (menú contextual, añadir directorio virtual)

6.C Problemas del entorno de operaciones

- Administrar un sistema puede causar problemas
 - Actualizar una librería puede hacer que los servicios que estén en producción dejen de funcionar
 - Olvidarse de abrir puertos puede causar problemas
 - ...
- Replicar un entorno exactamente igual es complicado: máquinas librerías, red, aplicaciones, ...
 - En el servidor de pruebas funcionaba, no entiendo por qué no funciona en producción
 - Me está fallando, voy a probar a instalar la nueva versión de java
 - Quizás haya que instalarle una librería
 - En Reddit dicen que hay que instalar Tomcat para que funcione
 - Igual se necesita abrir el puerto 8080
 - ...
- Virtualizar entornos y servicios puede simplificar varios problemas

6.C Virtualización



6.C Contenedores

- Facilita la administración de (micro)servicios:
 - Arranque, Actualización, Orquestación,...
- Aplicaciones/servicios autocontenidos
 - Librerías, puertos, software,...
- Fáciles de desplegar y portar
 - Se pueden almacenar en repositorios
- Fáciles de gestionar y actualizar
 - Son reemplazables
 - Se les puede cambiar la memoria, cpu, etc
- Seguros y aislados
- “Estándar”



* “[docker](#)” by [Bo-Yi Wu](#) is licensed under [CC BY 2.0](#)

6.C Terminología Docker

- Imagen: “An executable package that includes everything needed to run an application -the code, a runtime, libraries, environment variables, and configuration files.” Docker.com
- Contenedor: “A runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process” Docker.com
- Otra terminología:
 - Volumen:
 - Archivos y directorios que contienen los datos del contenedor
 - Si eliminamos un contenedor, el volumen sigue existiendo
 - Permiten separar procesamiento (contenedor) de datos (volumen)

6.C Instalación Docker

- En RHEL-based:

```
sudo yum install -y yum-utils device-mapper-persistent-  
data lvm2  
sudo yum-config-manager --add-repo  
https://download.docker.com/linux/centos/docker-ce.repo  
sudo yum install docker-ce docker-ce-cli containerd.io
```
- En algún RHEL-based hay conflictos: `sudo yum remove podman buildah`
- Inicializar Docker: `sudo systemctl start docker`

6.C Instalación Docker

- Instalación snap (ej. en Ubuntu): Resuelve automáticamente las dependencias y queda todo instalado

```
Featured Server Snaps [ Help ]

These are popular snaps in server environments. Select or deselect with SPACE,
press ENTER to see more details of the package, publisher and versions
available.

[ ] microk8s           Lightweight Kubernetes for workstations and appliance ▶
[ ] nextcloud          Nextcloud Server - A safe home for all your data ▶
[ ] wekan              Open-Source kanban ▶
[ ] kata-containers    Lightweight virtual machines that seamlessly plug int ▶
[*] docker             Docker container runtime ▶
[ ] canonical-livepatch Canonical Livepatch Client ▶
[ ] rocketchat-server  Group chat server for 100s, installed in seconds. ▶
[ ] mosquitto          Eclipse Mosquitto MQTT broker ▶
[ ] etcd              Resilient key-value store by CoreOS ▶
[ ] powershell        PowerShell for every system! ▶
[ ] stress-ng          A tool to load, stress test and benchmark a computer ▶
[ ] sabnzbd            SABnzbd ▶
[ ] wormhole           get things from one computer to another, safely ▶
[ ] aws-cli            Universal Command Line Interface for Amazon Web Servi ▶
[ ] google-cloud-sdk   Command-line interface for Google Cloud Platform prod ▶
[ ] slcli              Python based SoftLayer API Tool. ▶
[ ] doctl              The official DigitalOcean command line interface ▶
[ ] conjure-up          Package runtime for conjure-up spells ▶
[ ] minidlna-escoand    server software with the aim of being fully compliant ▶
[ ] postgresql10        PostgreSQL is a powerful, open source object-relation ▶
[ ] heroku             CLI client for Heroku ▶
[ ] keepalived          High availability VRRP/BFD and load-balancing for Lin ▶
[ ] prometheus          The Prometheus monitoring system and time series data ▶
[ ] juju               A model-driven operator lifecycle manager ▶

[ Done      ]
[ Back      ]
```

6.C Ejemplo despliegue de contenedor

- Aplicación Hello world

- Descargamos la imagen de hello-world

```
docker pull hello-world
```

Es una imagen que sólo hace hello-world y está en el repositorio de Docker

- Creamos un contenedor

```
docker create --name miHelloWorld hello-world
```

Nombre del contenedor

- Arrancamos el contenedor

```
docker start --attach --interactive miHelloWorld
```

STDIN, STDOUT y STDERR de la terminal del anfitrión pasa a ser la del contenedor

Permite tener la entrada estándar abierta esté o no atacheado

6.C Ejemplo despliegue de contenedor

- Aplicación Hello world
 - Descargamos la imagen de hello-world
 - Creamos un contenedor
 - Arrancamos el contenedor
 - Salida:

```
[moranjesus@localhost ~]$ sudo docker start --attach --interactive miHelloWorld
```

```
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.
(amd64)
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.

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://cloud.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/engine/userguide/
```

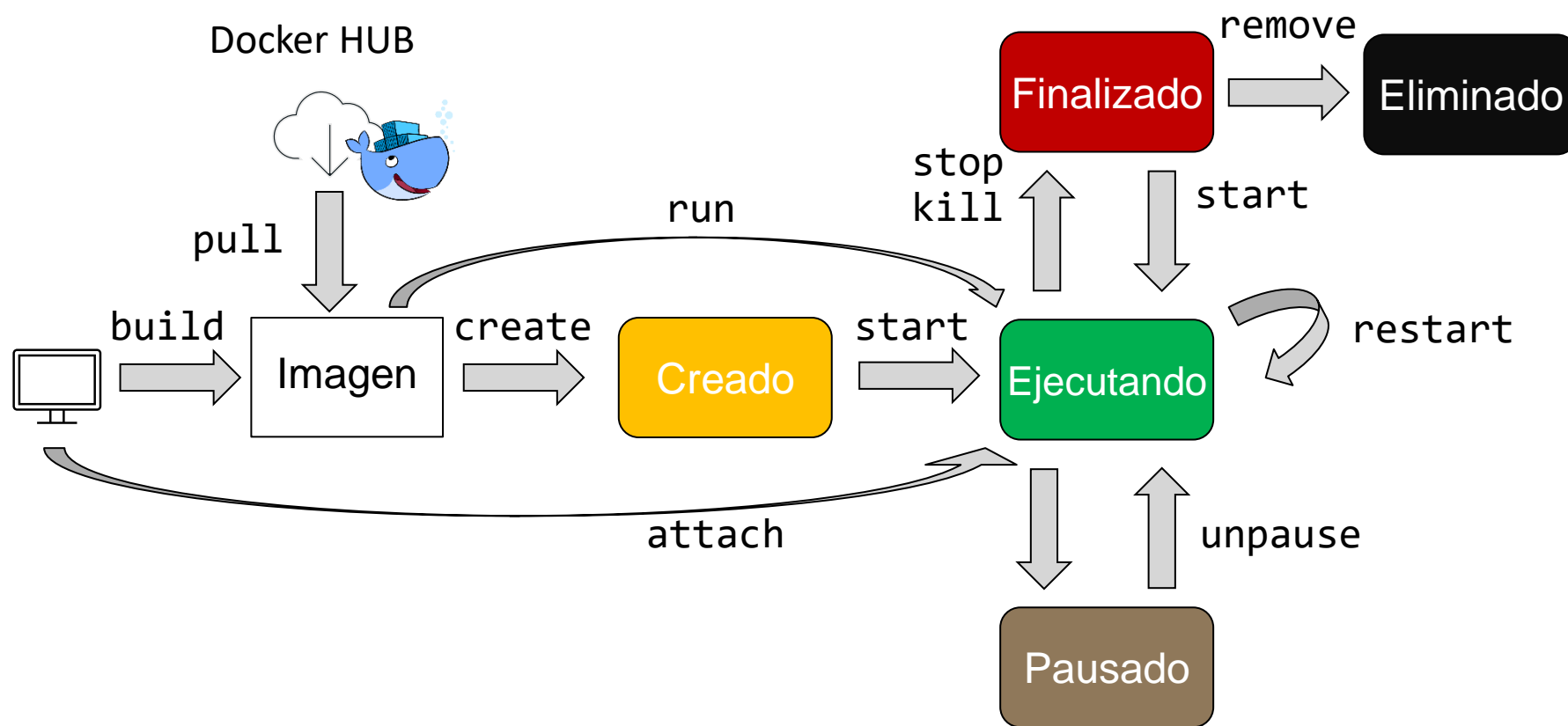
```
[moranjesus@localhost ~]$ █
```

→ **Volvemos al anfitrión**

**Comando ejecutado
en el anfitrión**

**Lo que printeo el
contenedor**

6.C Ciclo de vida de un contenedor



6.C Ejemplo contenedor Ubuntu

- Descargar la imagen de ubuntu

```
docker pull ubuntu
```

- Crear el contenedor

Se le asigna una pseudoterminal
para que nos podamos conectar

```
docker create --tty --name miUbuntu ubuntu
```

- Arrancar el contenedor

Conecta a la tty del contenedor

```
[moranjesus@localhost ~]$ sudo docker start --attach --interactive miUbuntu  
root@acfc24995cd2:/#
```

Prompt del contenedor

6.C Comandos del ciclo de vida (I)

- Arrancar

```
docker start --attach --interactive MiUbuntu
```

- Reiniciar

```
docker restart miUbuntu
```

- Pausar

```
docker pause miUbuntu
```

- Des-pausar

```
docker unpause miUbuntu
```

6.C Comandos del ciclo de vida (II)

- Apagar

```
docker stop MiUbuntu
```

- Matar

```
docker kill miUbuntu
```

- Conectarse

```
[moranjesus@localhost ~]$ sudo docker attach miUbuntu  
root@acfc24995cd2:/#
```

- Ejecutar un comando

```
[moranjesus@localhost ~]$ sudo docker exec miUbuntu cat /etc/passwd | grep mor*  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
[moranjesus@localhost ~]$ █
```

6.C Compartir datos

- Opción mount

```
docker create --name miUbuntu --tty --interactive  
--mount type=bind,  
        source=/home/moranjesus/misDatosEnHost,  
        target=/misDatosEnContenedor,  
        bind-propagation=shared ubuntu
```

Host

```
[moranjesus@localhost ~]$ ls -l /home/moranjesus/misDatosEnHost/  
total 8  
-rw-rw-r--. 1 moranjesus moranjesus 13 Apr 12 10:40 archiv01  
-rw-r--r--. 1 root      root      13 Apr 12 10:51 archiv02  
[moranjesus@localhost ~]$ █
```

Contenedor miUbuntu

```
root@4c3238aa97bb:/# ls -l /misDatosEnContenedor/  
total 8  
-rw-rw-r--. 1 1000 1000 13 Apr 12 08:40 archiv01  
-rw-r--r--. 1 root root 13 Apr 12 08:51 archiv02  
root@4c3238aa97bb:/#
```

- Opción volumen: menos verbosa

6.C Contenedor de servidor Apache

- Descargar la imagen del servidor Apache

```
docker pull httpd
```

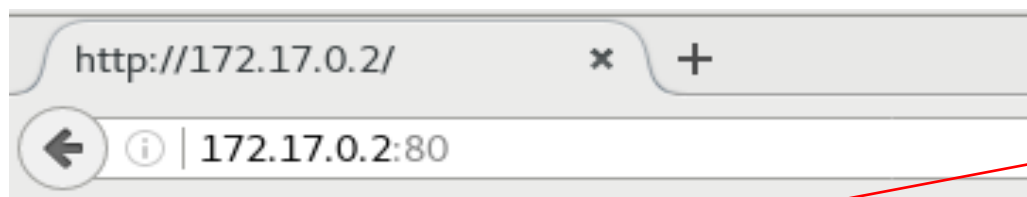
- Crear el contenedor exponiendo puerto

```
docker create --name miServidorWeb --expose 80 httpd
```

Se expone el puerto 80: se abre el puerto en la IP del contenedor

- Arrancar el contenedor

```
docker start miServidorWeb
```



Sólo se puede acceder a través de la IP del contenedor

It works!

6.C Contenedor de servidor Apache

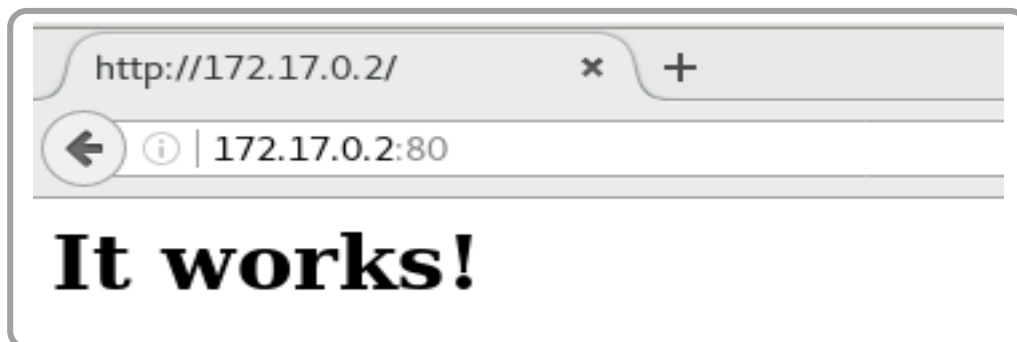
- Crear el contenedor publicando el puerto

```
docker create --name miServidorWeb --publish 1111:80 httpd
```

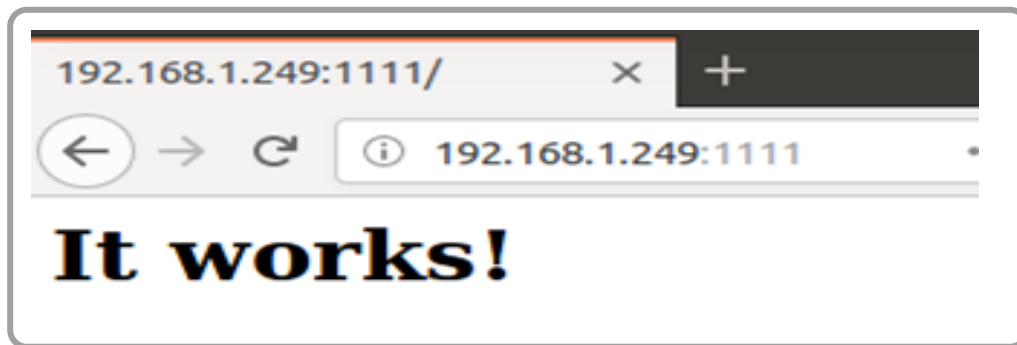
Mapea el puerto 80 del contenedor con el puerto 1111 del anfitrión

- Arrancar el contenedor

```
docker start miServidorWeb
```



Se puede acceder a través del puerto 80 del contenedor [está expuesto]



Se puede acceder a través del puerto 1111 del anfitrión [está publicado]

6.C Crear imagen desde contenedor

- Ejemplo de un contenedor que tenga un Ubuntu con Maven instalado

- Arrancar un contenedor

```
docker start --attach --interactive miUbuntu
```

- Modificar el contenedor hasta que tenga lo que queramos

```
root@50ab61106e5a:/# mvn -version  
Warning: JAVA_HOME environment variable is not set.  
Apache Maven 3.3.9  
Maven home: /usr/share/maven
```

Se han ejecutado todos los comandos para instalar maven en el contenedor

- Crear una imagen a partir del contenedor

La imagen se guardará en nuestro repositorio local

```
docker commit miUbuntu ubuntuconmaven
```

Ahora ya se pueden crear contenedores que tengan Ubuntu con maven

6.C Crear imagen con Docker File

- Crear un Docker File llamado milimagen:

```
FROM ubuntu  
MAINTAINER Jesus Moran  
RUN apt-get update  
RUN apt-get install -y maven
```

Tiene que tener todas las librerías, códigos, etc que se necesiten

En este ejemplo tendremos un Ubuntu con maven instalado

- Crear la imagen a partir del Docker File

```
docker build --file milimagen -t ubuntu-maven-desde-dockerfile  
/home/moranjesus
```

- Otras opciones del Docker File: FROM, RUN, ENTRYPOINT, CMD, LABEL, EXPOSE, COPY, ...

6.C Crear imagen con Docker File

- Crear un Docker File llamado `miImagen`:

```
FROM ubuntu
MAINTAINER Jesus Moran
RUN apt-get u
RUN apt-get i
```

- Primera instrucción que se ejecutará al arrancar el contenedor
 - El proceso tendrá el PID1
- En Docker el PID 1 no suele ser el proceso `init`
 - Docker está pensado para ejecutar un único servicio
- Si queremos instalar `systemd` en un contenedor:
 - Ejecutar contenedor con las capabilities de administración: `--cap-add SYS_ADMIN`
 - (Alternativa) Ejecutar docker en modo privilegiado: `--privileged=true`
 - Crear un volumen con el anfitrión: `-v /sys/fs/cgroup:/sys/fs/cgroup:ro`
 - (Opcional) Evitar servicios innecesarios: eliminar archivos `/etc/systemd/system/*wants/` y `/lib/systemd/system/*wants/`
 - Instalar `systemd` y ejecutar como `entrypoint /sbin/init`

- Crear la imagen

```
docker build --
/home/moranj
```

- Otras opciones del Docker File: `FROM`, `RUN`, `ENTRYPOINT`, `CMD`, `LABEL`, `EXPOSE`, `COPY`, ...

6.C Crear imagen de un programa

- Crear un programa holaMundo y lo compilamos
- Crear un Docker File llamado milimagenJava

```
FROM alpine  
ADD holaMundo.class holaMundo.class  
RUN apk --update add openjdk8-jre  
ENTRYPOINT ["java", "holaMundo"]
```

Es un Linux ligero

Añadimos el programa que hemos creado

Se necesita java para poder ejecutarlo

Indicamos que se ejecute el programa java

- Crear la imagen a partir del Docker File

```
docker build --file milimagenJava -t holamundojava /home/moranjesus
```

Se pueden crear contenedores y tendrán todo lo necesario: java y el programa holaMundo

6.C Gestión de contenedores e imágenes

- Buscar imágenes en repositorio remoto

```
C:\Users\crist>docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL
ubuntu	Ubuntu is a Debian-based Linux operating sys...	10416	[OK]
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	385	
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offi...	240	

- Ver imágenes en repositorio local

```
C:\Users\crist>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
selenium/vnc	chrome_78.0	67312aa40b2f	2 months ago	897MB
eexit/mirror-http-server	latest	9290012d0cef	4 months ago	285MB
aerokube/selenium	1.8.4	ec8a96615a09	13 months ago	13.1MB
mysql	5.7.22	6bb891430fb6	18 months ago	372MB

- Ver contenedores en ejecución

```
C:\Users\crist>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d229c3effb1b	codeurjc/full-teaching:2.3.0	"/bin/sh -c '/wait.s..."	2 months ago	Up 39 seconds	5000/tcp

6.C Gestión de contenedores e imágenes

- Buscar imágenes en repositorio remoto

```
C:\Users\crist>docker ps --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
d229c3effb1b	codeurjc/full-teaching:2.3.0	"/bin/sh -c '/wait.s..."	2 months ago	Exited (1) 4 minutes ago
197c9ff3a74b	openvidu/openvidu-server-kms:2.3.0	"/usr/bin/supervisord"	2 months ago	Exited (0) 2 months ago
5c339a9520aa	mysql:5.7.22	"docker-entrypoint.s..."	2 months ago	Exited (0) 2 months ago
ceb57b9083f4	codeurjc/full-teaching:2.3.0	"/bin/sh -c '/wait.s..."	3 months ago	Exited (137) 3 months ago

- Eliminar contenedor

```
docker rm miUbuntu
```

- Ver información estática de contenedores

```
docker inspect miUbuntu
```

- Ver información dinámica de contenedores

```
docker stats miUbuntu
```

6.C Limitar recursos

- Activar cgroups v1:
- Activarlo como opción de arranque:

Archivo `/etc/default/grub`

```
GRUB_CMDLINE_LINUX_DEFAULT="cgroup_enable=memory swapaccount=1"
```

- Actualizar el grub:

```
jesus@injtest:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-91-generic
Found initrd image: /boot/initrd.img-5.4.0-91-generic
Found linux image: /boot/vmlinuz-5.4.0-90-generic
Found initrd image: /boot/initrd.img-5.4.0-90-generic
Found linux image: /boot/vmlinuz-5.4.0-86-generic
Found initrd image: /boot/initrd.img-5.4.0-86-generic
Adding boot menu entry for UEFI Firmware Settings
done
jesus@injtest:~$
```

- Reiniciar la máquina

6.C Limitar recursos

- Activar cgroups v2:
 - Sólo para SO con kernel ≥ 4.15
 - Sólo para Docker ≥ 20.10
 - Activar cgroups v2 como opción de arranque:

Archivo /etc/default/grub

```
GRUB_CMDLINE_LINUX="systemd.unified_cgroup_hierarchy=1"
```

- Actualizar el grub
- Reiniciar la máquina
- Nota: si queremos utilizar cgroups v1:

```
GRUB_CMDLINE_LINUX="systemd.unified_cgroup_hierarchy=0"
```

6.C Limitar recursos

- Tras activar cgroup:

```
jesus@injtest:~$ sudo docker run -it --rm --memory 512m --memory-swap 512m --name  
"alpine_limitado" alpine:latest /bin/sh  
/ #
```

docker stats

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
d1ad67693f04	8_65536-131072_0_9	100.60%	3.764GiB / 11GiB	34.22%	1.01kB / 0B	18.4MB / 0B	34
9396294b26d9	7_65536-131072_3_10	100.75%	4.204GiB / 11GiB	38.22%	1.15kB / 0B	0B / 0B	34
924a4266b58a	7_65536-131072_3_9	101.38%	4.068GiB / 11GiB	36.98%	1.15kB / 0B	11.5MB / 0B	34
3a78d55a8951	6_65536-131072_3_11	100.03%	8.983GiB / 11GiB	81.67%	1.57kB / 0B	0B / 0B	34
a93e5955f517	prometheus	0.00%	241.9MiB / 47.41GiB	0.50%	4.94GB / 86.5MB	164MB / 2.7GB	23
5804c90ed26f	caddy	0.00%	33.06MiB / 47.41GiB	0.07%	481MB / 470MB	34.8MB / 0B	22
35757cb3d043	cadvisor	7.98%	59.8MiB / 47.41GiB	0.12%	87.9MB / 5.4GB	95.2MB / 0B	42
82c67db4f94e	alpine_limitado	0.00%	1.684MiB / 512MiB	0.33%	726B / 0B	0B / 0B	1

Vemos que está limitado a 512MB