

Universidad de Cuenca

Facultad de Ingeniería



Informe

Trabajo 1: Configuración de servidores

Nombre

María Belén Sarmiento Quezada

Asignatura

Programación web

Docente

Ing. Priscila Cedillo PhD

Carrera

Computación

Fecha

Lunes, 01 de abril de 2024

1. Objetivos

- Comprender la configuración de servidores web locales y de servidores web de aplicaciones.
- Entender la diferencia entre un servidor web local y un servidor que permita despliegue en la nube.

2. Marco teórico

2.1. Servidor web

Un servidor web es un ordenador que se encarga de almacenar, procesar y transmitir archivos de sitios web a los usuarios a través de un navegador. Está formado por hardware y software, utiliza el Protocolo de Transferencia de Hipertexto (HTTP) para responder a las peticiones de los usuarios de la web realizadas a través de la World Wide Web [1]. Un servidor web está permanentemente en espera de una solicitud de información [2].

Ejemplos:

- **Apache:** Es un servidor web de código abierto, multiplataforma y gratuito que lleva en funcionamiento desde 1995. Por sus múltiples beneficios, es uno de los más utilizados en el mundo. Apache consigue que la comunicación entre cliente y servidor sea fluida y constante [3]. Entre sus ventajas están: la carga dinámica de módulos, su amplia integración con otras herramientas de software y su sólida compatibilidad con medios. [15]
- **Nginx:** Servidor web caracterizado por su optimización del rendimiento en el manejo de la alta concurrencia. En lugar de crear nuevos procesos para cada solicitud web, usa un enfoque asíncronico basado en eventos donde las solicitudes se manejan en un solo hilo. [4]
- **Microsoft IIS:** Caracterizado por su arquitectura modular que ofrece extensibilidad y administración de componentes independientes que se pueden agregar y quitar fácilmente [5].

2.2. Servidor de aplicaciones

Es un tipo de servidor diseñado para instalar, operar y hospedar aplicaciones y servicios web. Su función principal es proporcionar un entorno para el desarrollo, ejecución y gestión de aplicaciones, permitiendo conectar aplicaciones con bases de datos, software y hardware relacionado [6]. Situado entre el servidor web y el nivel de backend del servidor de bases de datos (Ver figura 1).

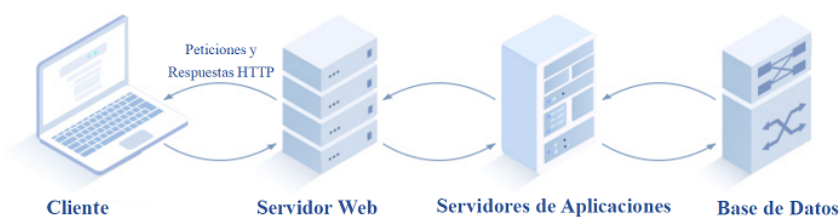


Figura 1: Arquitectura del sistema web

Ejemplos:

- **Apache Geronimo:** Proporciona bibliotecas para implementar especificaciones para Java Enterprise Edition y Jakarta Enterprise Edition. Además proporciona metadatos para usar con el framework de Java OSGi [7].
- **Apache Tomcat:** Es una implementación de código abierto de las especificaciones de la plataforma Jakarta EE y Java EE. Se desarrolla en un entorno participativo con colaboración de los mejores desarrolladores del mundo. Se publica bajo la licencia de Apache versión 2 [8].
- **Zope:** Es un servidor de aplicaciones gratuito y de código abierto escrito en Python para crear aplicaciones altamente escalables. Tiene un conjunto de bibliotecas destinadas a ser reutilizadas para desarrollar aplicaciones o marcos web [9].

2.3. Hosting

Es un servicio online que permite publicar nuestro sitio web para que sea accesible en todo momento desde cualquier dispositivo conectado a Internet. Al contratar este servicio, básicamente se alquila un espacio en un servidor físico donde se pueden almacenar todos los archivos y datos necesarios para el correcto funcionamiento del sitio web. El proveedor de hosting es el responsable de garantizar la seguridad y funcionamiento del sitio web [10].

2.4. Cloud Computing

Es una tecnología que permite acceso remoto a software, almacenamiento de archivos y procesamiento de datos a través de la conectividad y gran escala de Internet. Es una alternativa a la ejecución en una computadora personal o servidor local. No hay la necesidad de instalar aplicaciones localmente [11]. Los modelos de servicio de Cloud Computing se basan en el concepto de compartir recursos informáticos y software, incluidos los servicios de computación, almacenamiento y redes [12].

Por definición, la nube es un entorno donde se ejecutan las aplicaciones. Según la compañía Red Hat [13] existen varios tipos de nubes:

- **Públicas:** Entornos creados a partir de recursos ajenos al usuario final que pueden redistribuirse a otros inquilinos.
- **Privadas:** Entorno diseñado solo para el usuario final, generalmente dentro del firewall del usuario.
- **Híbridas:** Varios entornos de nube con cierto nivel de portabilidad, organización y gestión de las cargas de trabajo entre ellos.
- **Multiclouds:** Sistemas de TI que incluyen más de una nube, pública o privada y que pueden conectarse en red (o no).

2.5. Despliegue continuo

Es una estrategia de desarrollo de software en la que los cambios aplicados al código de una aplicación se publican automáticamente en el entorno de producción. La automatización se basa en una serie de pruebas predefinidas. Una vez que las actualizaciones pasan esas pruebas, el sistema envía la aplicación con los cambios directamente a los usuarios finales [14].

3. Práctica

Se configuran tres servidores en una máquina virtual usando la distribución Ubuntu de Linux.

3.1. Configurar APACHE

Se usa XAMPP porque es multiplataforma, mientras que WAMP solo está disponible para Windows. El proceso para configurar Apache es el siguiente:

1. Instalar Apache

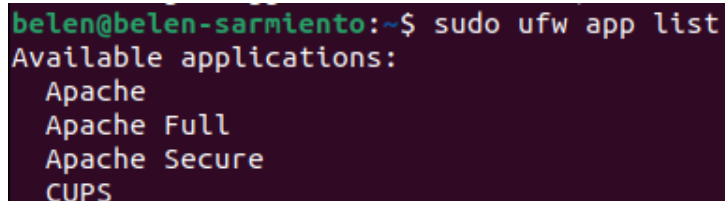
Ejecutar los comandos:

```
sudo apt update  
sudo apt install apache2
```

2. Ajustar el firewall

Se debe modificar los ajustes del firewall para permitir el acceso externo a los puertos web predeterminados. Durante la instalación, Apache se registra con UFW (Uncomplicated Firewall) para proporcionar algunos perfiles de aplicación que pueden habilitar o deshabilitar el acceso a Apache. Para ver los perfiles de aplicación, ejecutar el comando:

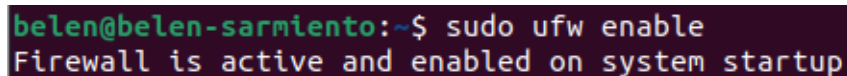
```
sudo ufw app list
```



```
belen@belen-sarmiento:~$ sudo ufw app list  
Available applications:  
  Apache  
  Apache Full  
  Apache Secure  
  CUPS
```

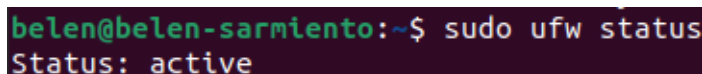
Figura 2: Perfiles de aplicación

Apache abre el puerto 80, Apache Full abre los puertos 80 y 443, mientras que Apache Secure abre el puerto 443. El puerto 80 es de tráfico web normal no cifrado y el 443 es de tráfico TLS/SSL cifrado. Se habilita el cortafuegos y se verifica el cambio (Ver figuras 3 y 4).



```
belen@belen-sarmiento:~$ sudo ufw enable  
Firewall is active and enabled on system startup
```

Figura 3: Habilitar cortafuegos



```
belen@belen-sarmiento:~$ sudo ufw status  
Status: active
```

Figura 4: Comprobar estado activo

3. Comprobar servidor web

Para verificar si el servicio está en ejecución, usar:

```
sudo systemctl status apache2
```

```
belen@belen-sarmiento:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Wed 2024-03-27 07:53:32 -05; 4min 38s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 5876 (apache2)
      Tasks: 55 (limit: 4573)
   Memory: 4.7M
      CPU: 137ms
    CGroup: /system.slice/apache2.service
            └─5876 /usr/sbin/apache2 -k start
              └─5877 /usr/sbin/apache2 -k start
                └─5879 /usr/sbin/apache2 -k start
```

Figura 5: Lista de tráfico HTTP permitido

Ver la dirección IP de la máquina y probar en el navegador con `http://ip`

```
belen@belen-sarmiento:~$ hostname -I
192.168.204.128
```

Figura 6: IP de la máquina

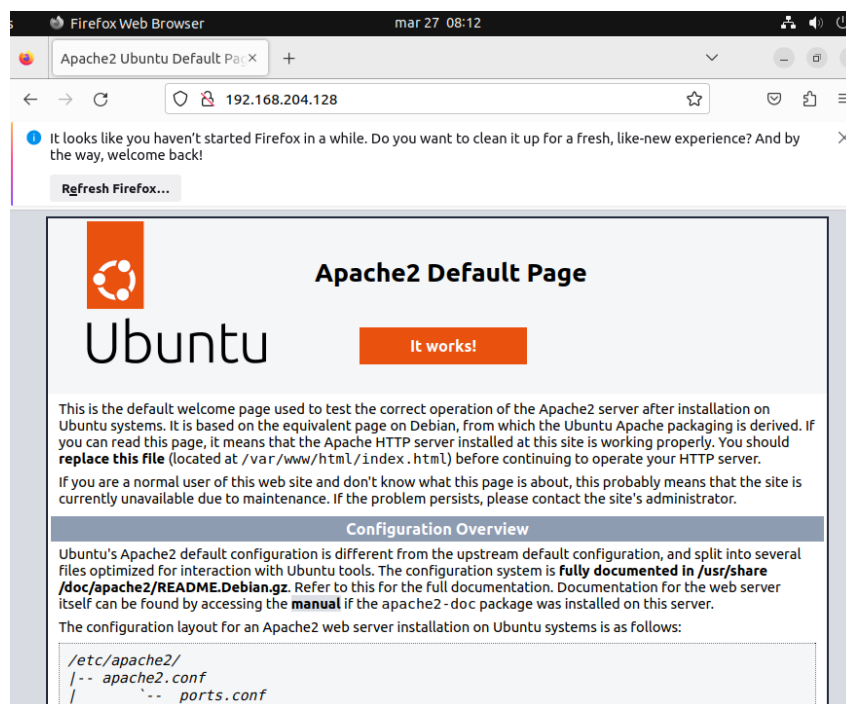


Figura 7: Comprobar funcionamiento

Algunos comandos para administrar el proceso Apache son:

```
sudo systemctl stop apache2
sudo systemctl start apache2
sudo systemctl restart apache2
sudo systemctl reload apache2
sudo systemctl disable apache2
sudo systemctl enable apache2
```

3.2. Configurar APACHE TOMCAT

1. Instalar Java

Es necesario instalar Java porque Tomcat desde su creación tenía la función de aceptar y responder servidores de aplicaciones Java. OpenJDK es el desarrollo predeterminado de Java en Ubuntu. Para instalarlo se usa el comando:

```
sudo apt install default-jdk
```

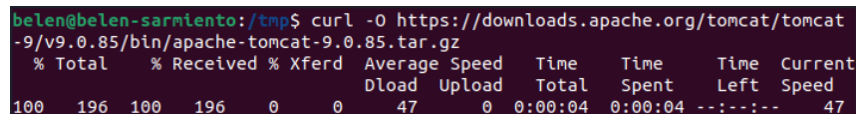
2. Crear un usuario de Tomcat

Por seguridad es necesario crear un usuario único dentro de un grupo. Luego se crea el usuario tomcat estableciendo un shell que no le permite iniciar sesión porque no es necesario. El directorio de inicio del usuario es /opt/tomcat. Para esto se usan los comandos:

```
sudo groupadd tomcat
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

3. Instalar Tomcat en Ubuntu

Se debe copiar el enlace que está en el sitio web después del comando curl para verificar la conectividad de la URL y descargar desde el enlace copiado.



```
belen@belen-sarmiento:~/tmp$ curl -O https://downloads.apache.org/tomcat/tomcat-9/v9.0.85/bin/apache-tomcat-9.0.85.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left     Speed
100  196  100  196    0    0   47      0  0:00:04  0:00:04 --:--:--   47
```

Figura 8: Descargar Apache Tomcat

4. Permisos de actualización

Se otorgan los permisos respectivos para que el usuario tomcat tenga acceso completo a la instalación y el directorio /opt/tomcat donde se descomprimirá la descarga.

```
sudo mkdir /opt/tomcat
cd /opt/tomcat
sudo tar xzvf apache-tomcat-9.0.87.tar.gz -C /opt/tomcat
--strip-components=1
sudo chgrp -R tomcat /opt/tomcat
```

Se otorga el acceso al directorio conf para ver su contenido y ejecutar el acceso al directorio en sí. Además se establece que el usuario de Tomcat sea el propietario de los directorios de apps, work, temp y logs con los comandos:

```
sudo chmod -R g+r conf
sudo chmod g+x conf
sudo chown -R tomcat webapps/work/temp/logs
```

5. Crear un archivo de unidad systemd

Crear un archivo tomcat.service en /etc/systemd/system/ con el contenido que se observa a continuación.

```

1      [Unit]
2      Description=Apache Tomcat Web Application Container
3      After=network.target
4      [Service]
5      Type=forking
6
7      Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre
8      Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
9      Environment=CATALINA_HOME=/opt/tomcat
10     Environment=CATALINA_BASE=/opt/tomcat
11     Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+
        UseParallelGC '
12     Environment='JAVA_OPTS.awt.headless=true -Djava.security.egd=file:/
        dev/v/urandom '
13
14     ExecStart=/opt/tomcat/bin/startup.sh
15     ExecStop=/opt/tomcat/bin/shutdown.sh
16
17     User=tomcat
18     Group=tomcat
19     UMask=0007
20     RestartSec=10
21     Restart=always
22
23     [Install]
24
25     WantedBy=multi-user.target

```

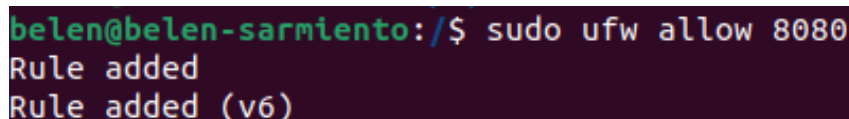
Listing 1: Archivo de configuración tomcat.service

Notificar al sistema que se ha creado un nuevo archivo emitiendo el siguiente comando:

```
sudo systemctl daemon-reload
```

6. Ajustar el Firewall

Para que las solicitudes lleguen al servicio se ajusta el firewall. Por defecto se usa el puerto 8080 utilizando UFW.



```

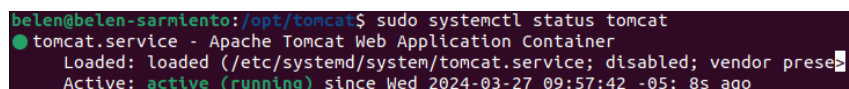
belen@belen-sarmiento:/$ sudo ufw allow 8080
Rule added
Rule added (v6)

```

Figura 9: archivo tomcat.service

7. Comprobar funcionamiento

Iniciar el servidor y comprobar el estado de Apache Tomcat.



```

belen@belen-sarmiento:/opt/tomcat$ sudo systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor prese
   Active: active (running) since Wed 2024-03-27 09:57:42 -05; 8s ago

```

Figura 10: Comprobar el estado del servidor

Acceder desde el navegador con `http://ip`

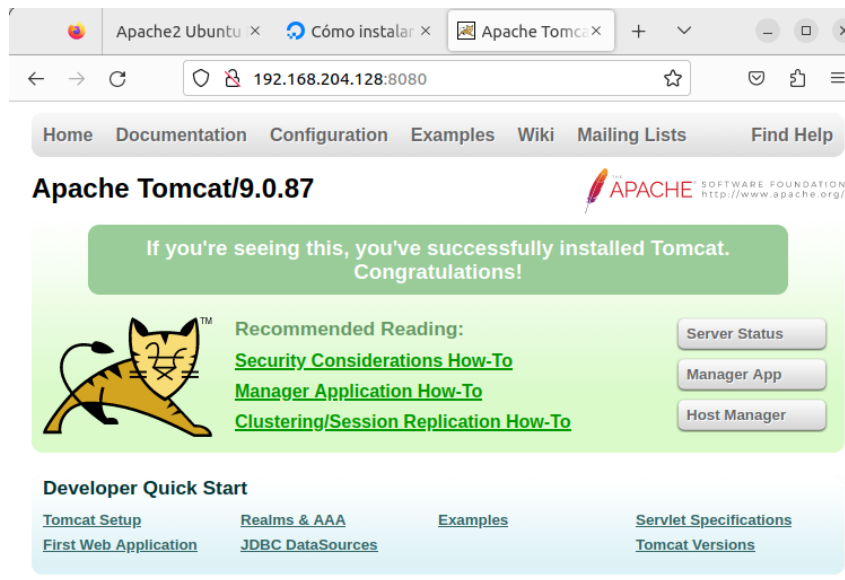


Figura 11: Prueba en el navegador

3.3. Configurar un servidor en Cloud Computing

El servidor en la nube escogido es Render. Se ha desarrollado una aplicación web simple usando Node.js. El proyecto se encuentra en un repositorio de GitHub. El proceso a seguir para configurar el servidor en la nube es:

1. Crear una cuenta en Render vinculada a GitHub.
2. Subir el nuevo proyecto a un repositorio de GitHub.
3. Crear un nuevo servicio web en Render

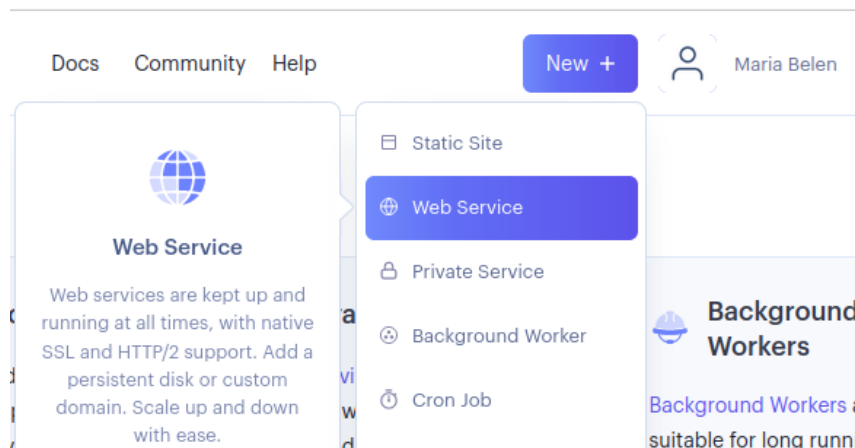


Figura 12: Creación de un nuevo Web Service

4. Seleccionar el repositorio de GitHub donde se encuentra el proyecto para desplegar la aplicación.



for these repositories:

☐ **All repositories**

This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

☒ **Only select repositories**

Select at least one repository.
Also includes public repositories (read-only).

Select repositories ▾

Selected 1 repository.

msbq14/desplegar_render



Figura 13: Seleccionar repositorio

5. Realizar las configuraciones correspondientes:

- El nombre del proyecto
- Región: Escoger la más cercana al país actual.
- Rama: La rama de git a utilizar.
- Tiempo de ejecución: Plataforma sobre la cual se ejecutará la aplicación.
- Comando de construcción: El comando que instala las librerías, migraciones o recursos de compilación.
- Comando de inicio: Se ha especificado el comando necesario para iniciar la aplicación, en este caso, npm start que ejecuta el script start definido en el package.json.

Name A unique name for your web service.	hello_world_nodejs
Region The region where your web service runs.	Oregon (US West) ▾
Branch The repository branch used for your web service.	main ▾
Root Directory <small>Optional</small> Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.	e.g., src
Runtime The runtime for your web service.	Node ▾
Build Command This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.	\$ npm install
Start Command This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.	\$ npm start

Figura 14: Configuraciones del proyecto

6. Configurar las variables de entorno necesarias para la aplicación. En este caso el puerto.

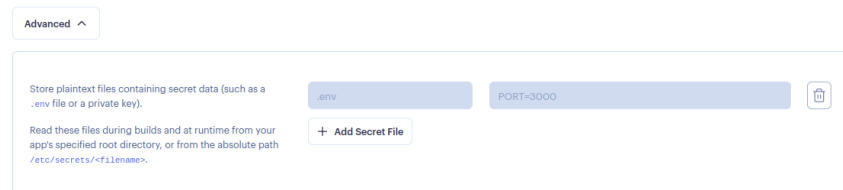


Figura 15: Configurar variables de entorno

7. Una vez configurado, Render ha construido y desplegado la aplicación correctamente.

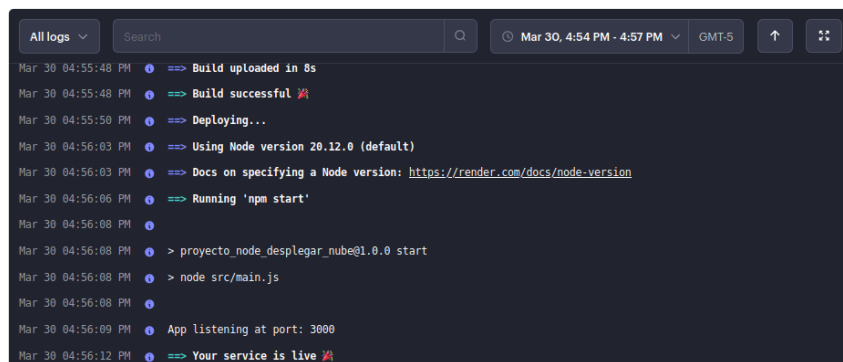


Figura 16: Construcción exitosa

8. Comprobar funcionamiento: Al estar en la nube, no es necesario que la aplicación esté corriendo localmente para su uso. Se accede a la aplicación con un enlace generado por el servidor de la nube.

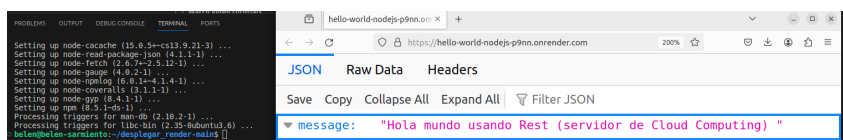


Figura 17: Funcionamiento en el navegador

4. Ejercicios a realizar

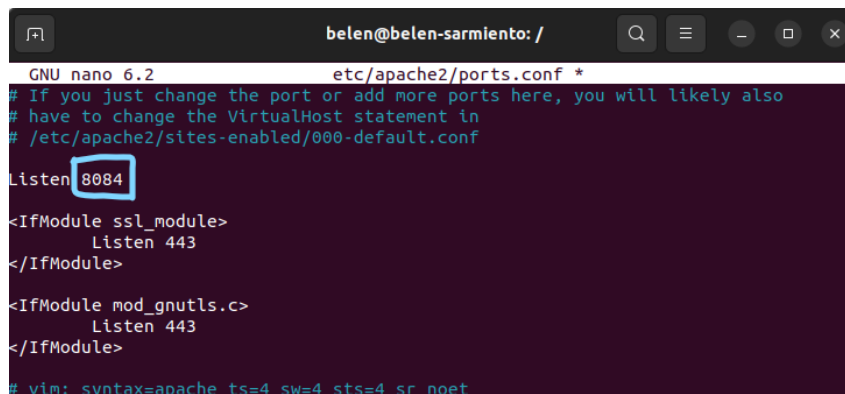
4.1. Servidores locales

Ejercicio 1.

Cambiar los puertos para que funcionen en un puerto desde el 8080 hasta el 8089 (elegir).

- APACHE

Para Apache el puerto escogido es el **8084**. Para cambiarlo, se edita la parte del puerto del archivo `/etc/apache2/ports.conf`.



```
GNU nano 6.2      etc/apache2/ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8084

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 18: Cambio del puerto 8080 al 8084

Para permitir el tráfico de la red a través del firewall, se ejecuta:

```
sudo ufw allow 8084
```

Se comprueba el funcionamiento cambiando en la ruta `http://ip:8084` como se observa en la Figura 19:

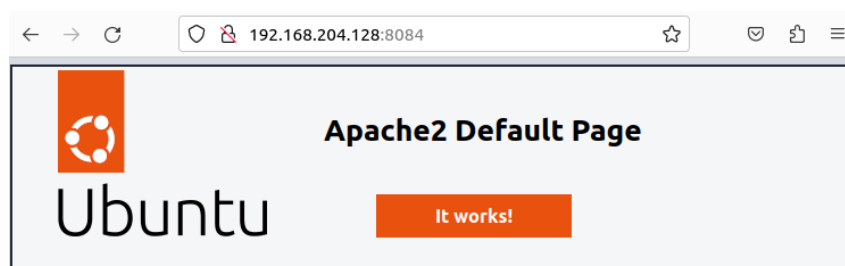


Figura 19: Comprobar funcionamiento

■ APACHE TOMCAT

Para Apache Tomcat el puerto escogido es el **8087**. Se debe cambiar el archivo `server.xml`.

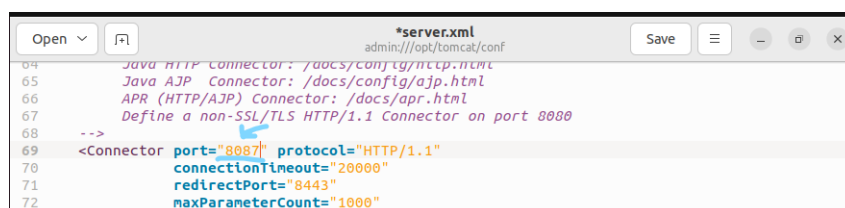


Figura 20: Cambiar al puerto 8087

Para reiniciar la configuración, usar el comando:

```
sudo systemctl restart tomcat
```

Se comprueba el funcionamiento cambiando en la ruta `http://ip:8087` como se observa en la Figura 21:

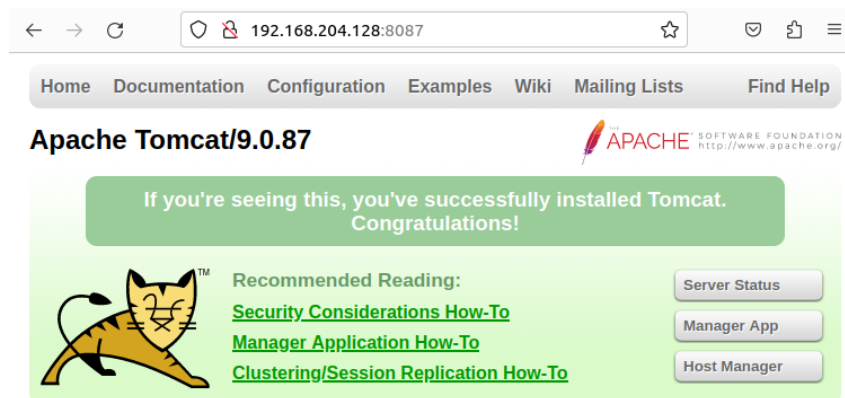


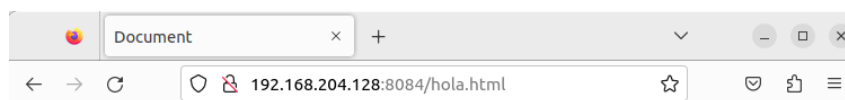
Figura 21: Comprobar funcionamiento

Ejercicio 2

Desplegar una página Web sencilla de hola mundo en cada uno.

■ APACHE

El archivo index.html que se ejecuta por defecto, se encuentra en `/var/www/html/`. Se debe crear un archivo con html en esta carpeta. Para observarlo en el navegador se usa `http://192.168.204.128:8084/archivo.html`. En este caso el archivo se llama `hola.html`.

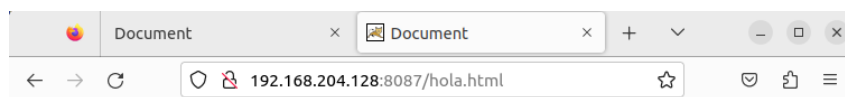


Hola mundo

Figura 22: Página web en Apache

■ APACHE TOMCAT

El archivo index.jsp es el que se ejecuta por defecto y se encuentra en el directorio `/opt/tomcat/` creado anteriormente para descomprimir Apache Tomcat. De igual manera se crea el archivo con extensión `.html` en ese directorio y se llama con la ruta `http://192.168.204.128:8087/hola.html`



Hola mundo

Figura 23: Página web en Apache Tomcat

Ejercicio 3

Crear una carpeta que se llame `carpeta_publica` y hacer que sea la carpeta de publicación

Paso 1: Crear una carpeta llamada `carpeta_publica` en `/var/` y dentro un archivo llamado `index.html`.

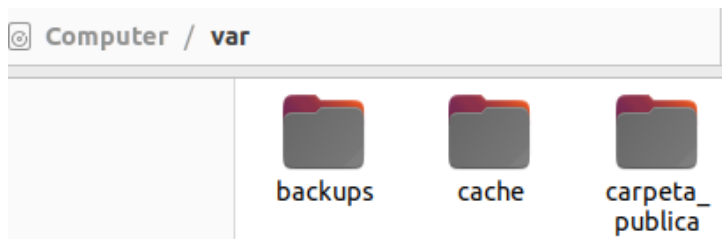


Figura 24: Carpeta pública

Paso 2: Modificar el archivo 000-default.conf, ubicado en el directorio etc/apache2/sites-available/

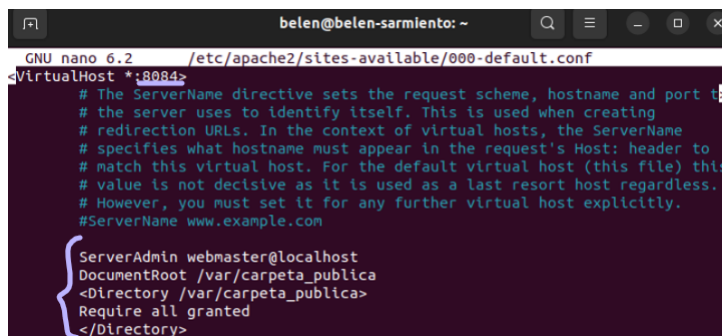


Figura 25: Cambios aplicados

Paso 3: Verificar funcionamiento



Figura 26: Cambios aplicados

Ejercicio 4

Buscar la estructura recomendada de un árbol de archivos para publicación. Crear la estructura de carpetas y probar que hacer para publicar cada una de ellas.

En el desarrollo web no existe una estructura de carpetas estándar. Sin embargo, hay algunas convenciones comunes que se utilizan para organizar los archivos de una aplicación web. Es recomendable que el front-end y el back-end estén en diferentes servidores por seguridad y para que sea más fácil la detección y corrección de errores. La estructura puede estar basada en el tipo de archivo, funcionalidad o módulos.

- src
 - components/ (Componentes reutilizables)
 - styles/ (Archivos CSS o preprocesados)
 - scripts/ (Archivos JavaScript)
 - images/ (Imágenes)
 - fonts/ (Fuentes)
 - utils/ (Utilidades o funciones auxiliares)
 - pages/ (Páginas principales)
 - ...

Listing 2: Estructura basada en el tipo de archivo

```

- src
  auth/    (Autenticación y gestión de usuarios)
  home/     (Página de inicio)
  clients/  (Clientes)
  orders/   (Órdenes de compra)
  ...

```

Listing 3: Estructura basada en la funcionalidad

```

- src
  modules/
    clients/
      components/
      services/
      utils/
    orders/
      components/
      services/
      utils/
  ...

```

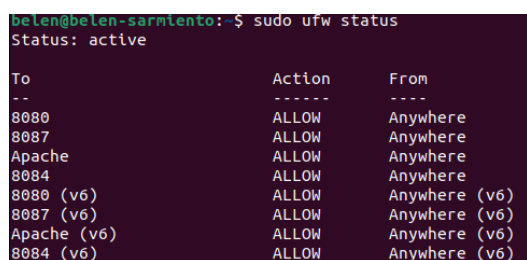
Listing 4: Estructura basada en los módulos

Ejercicio 5

Configurar el firewall para que sirva exclusivamente en el puerto que se quiera indicar.

En este caso están habilitados los puertos 8080, 8084 y 8087. Se pretende que el único puerto que sirva sea el 8084. Para desactivar los demás puertos se usa el comando:

```
sudo ufw deny puerto
```



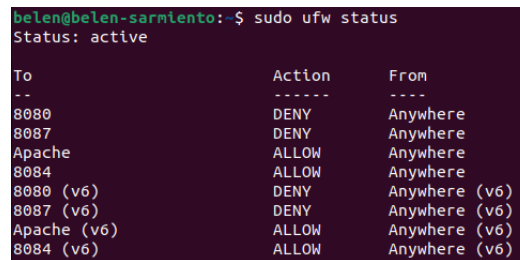
```

belen@belen-sarmiento:~$ sudo ufw status
Status: active

To      Action    From
--      -
8080    ALLOW     Anywhere
8087    ALLOW     Anywhere
Apache  ALLOW     Anywhere
8084    ALLOW     Anywhere
8080 (v6) ALLOW     Anywhere (v6)
8087 (v6) ALLOW     Anywhere (v6)
Apache (v6) ALLOW     Anywhere (v6)
8084 (v6) ALLOW     Anywhere (v6)

```

Figura 27: Estado antes de denegar



```

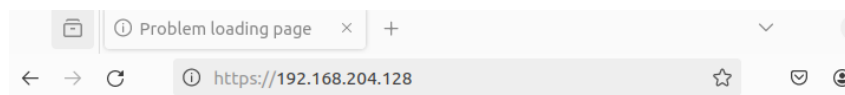
belen@belen-sarmiento:~$ sudo ufw status
Status: active

To      Action    From
--      -
8080    DENY      Anywhere
8087    DENY      Anywhere
Apache  ALLOW     Anywhere
8084    ALLOW     Anywhere
8080 (v6) DENY      Anywhere (v6)
8087 (v6) DENY      Anywhere (v6)
Apache (v6) ALLOW     Anywhere (v6)
8084 (v6) ALLOW     Anywhere (v6)

```

Figura 28: Estado con un solo puerto

El puerto por defecto es el 8080, es decir no se lo debe especificar en la URL, pero como se puede evidenciar ya no sirve porque se ha deshabilitado el firewall para el mismo.



Unable to connect

An error occurred during a connection to 192.168.204.128.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

Figura 29: Comprobar puerto único

4.2. Servidor Cloud

Buscar un servidor con trial o transacciones limitadas, revisar los costos, restricciones de tecnología. Ver que ofrecen y hacer una reflexión sobre dicho contenido

Render es un espacio en la nube que permite alojar de todo, desde prototipos hasta aplicaciones con muchos servicios. Cuenta con cuatro planes mensuales dependiendo de las necesidades. Todos los planes ofrecen sitios estáticos, PostgreSQL administrado, copias de seguridad automatizadas, discos persistentes, implementaciones automáticas desde Git, vistas previas del servicio, construcciones de Docker, comando previo al despliegue, dominios personalizados, HTTP/2 y HTTP/3, redes privadas, etc. A continuación, se presentan los precios y restricciones de cada plan [18].

Característica por versión	Individual	Equipo	Organización	Empresa
Descripción	Para aficionados y estudiantes	Para equipos pequeños	Para equipos más grandes	Para máxima potencia y personalización.
Precio por usuario al mes	\$0	\$19	\$29	Depende
Ancho de banda libre	100 GB	500 GB	1 TB	Depende
Minutos de canalización gratuitos	500 al mes	500 por usuario al mes	500 por usuario al mes	Depende
Reversiones instantáneas	5 compilaciones retenidas	15 compilaciones retenidas	30 compilaciones retenidas	30 compilaciones retenidas
Retención de registros	7 días	14 días	30 días	30 días
Miembros del equipo	1	Hasta 10	Ilimitado	Depende

Cuadro 1: Comparación de planes en Render

En el caso del plan empresarial, el precio va variando en función de las necesidades del cliente. Fácilmente se alcanza el límite gratuito, sin embargo, usar un servidor en la nube elimina la necesidad de infraestructura física, propensa a daños, y es accesible desde cualquier lugar, una ventaja destacable. Es importante seleccionar un proveedor adecuado, considerando que los precios varían según la necesidad del cliente.

5. Recomendaciones

- Realizar la práctica usando una distribución del sistema operativo Linux permitió comprender de mejor manera el proceso para configurar los servidores.
- Usar una máquina virtual para configurar los servidores ayuda a proteger al sistema operativo host de posibles daños.

Referencias

- [1] Betania, V. (2022, febrero 9). ¿Qué es un servidor web y cómo funciona? Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-servidor-web>
- [2] de Souza, I. (2019, junio 14). ¿Qué es un servidor web y cuáles son sus características? Rock Content - ES; Rock Content. <https://rockcontent.com/es/blog/que-es-un-servidor/>
- [3] Martínez, G. (2020, agosto 19). Qué es Apache y cómo funciona. Webempresa. <https://www.webempresa.com/hosting/que-es-servidor-apache.html>
- [4] ¿Qué Es Nginx y Cómo Funciona? (2018, marzo 25). Kinsta®; Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
- [5] Introducción al servidor web IIS. (s/f). Microsoft.com. Recuperado el 26 de marzo de 2024, de <https://learn.microsoft.com/es-es/iis/get-started/introduction-to-iis/iis-web-server-overview>
- [6] Jose. (2024, enero 12). ¿Qué son los servidores de aplicaciones y por qué los necesitas? Blog de tecnología; Sistemas Ibertrónica. <https://ibertronica.es/blog/actualidad/que-son-los-servidores-de-aplicaciones-y-por-que-los-necesitas/>
- [7] The Apache Software Foundation. (s/f). Apache Geronimo. Apache.org. Recuperado el 26 de marzo de 2024, de <https://geronimo.apache.org/>
- [8] Apache Tomcat Project. (s/f). Apache tomcat®. Apache.org. Recuperado el 27 de marzo de 2024, de <https://tomcat.apache.org/>
- [9] The world of zope — zope project and community documentation. (s/f). Zope.dev. Recuperado el 27 de marzo de 2024, de <https://www.zope.dev/world.html>
- [10] Gustavo, B. (2018, junio 8). ¿Qué es un hosting y cómo funciona? Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-hosting>
- [11] ¿Qué es Cloud Computing? (s/f). Salesforce. Recuperado el 27 de marzo de 2024, de <https://www.salesforce.com/mx/cloud-computing/>

- [12] ¿Qué es cloud computing? (s/f). Google Cloud. Recuperado el 27 de marzo de 2024, de <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>
- [13] ¿Qué es cloud computing o computación en la nube? (s/f). Redhat.com. Recuperado el 27 de marzo de 2024, de <https://www.redhat.com/es/topics/cloud>
- [14] ¿Qué es el despliegue continuo? (s/f). Ibm.com. Recuperado el 27 de marzo de 2024, de <https://www.ibm.com/es-es/topics/continuous-deployment>
- [15] Glass, E. (2020, mayo 21). Cómo instalar el servidor web Apache en Ubuntu 20.04. Digitalocean.com; DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>
- [16] Deyimar, A. (2020, enero 13). Cómo instalar Tomcat en Ubuntu 18.04. Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/como-instalar-tomcat-en-ubuntu>
- [17] Precios. (s/f). Google Cloud. Recuperado el 28 de marzo de 2024, de https://cloud.google.com/functions/pricing?hl=es&_gl=1*1j0c7w7*_up*MQ..&gclid=CjwKCAjwh4-wBhB3EiwAeJsppI7gE5t0rCwqoDeB9kx1mO-Lrvl0Zb0zL0lw26axw12bHe1Zb3DeZBoCetoQAvD_BwE&gclsrc=aw.ds
- [18] Cloud application hosting for developers. (s/f). Render.com. Recuperado el 1 de abril de 2024, de <https://render.com/>