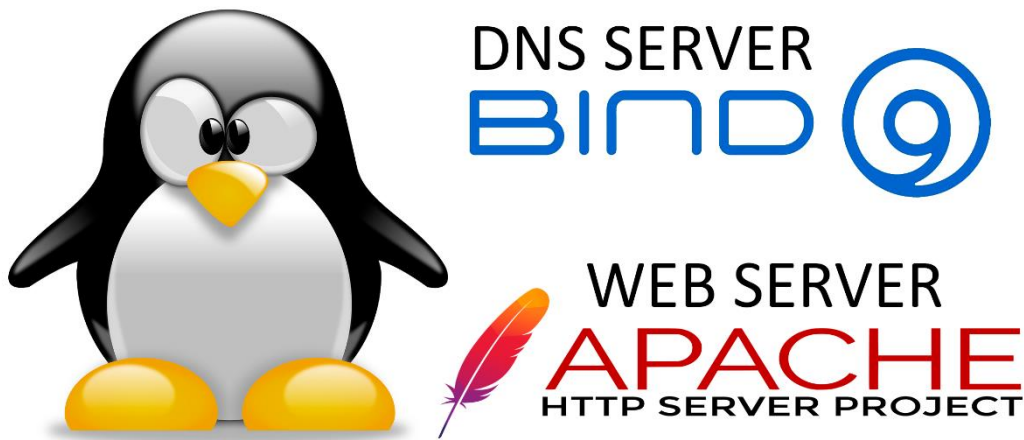


Despliegue de aplicaciones Web



Tema 2: Sitios **Web Virtuales**

Índice

- 1.- Introducción
- 2.- Montar la máquina virtual
- 3.- Instalar y configurar como servidor DNS el Bind9
 - 3.1.- Zona Directa
 - 3.2.- Zona Inversa
- 4.- Instalar y configurar Apache
 - 4.1.- Comprobar el servidor Web
 - 4.2.- Comandos básicos Apache
 - 4.3- Configurar hosts virtuales

1.- Introducción

En este ejercicio vamos a montar un servidor web Apache con dos sitios virtuales:


www.jasejitu.com

www.jagarcia.com

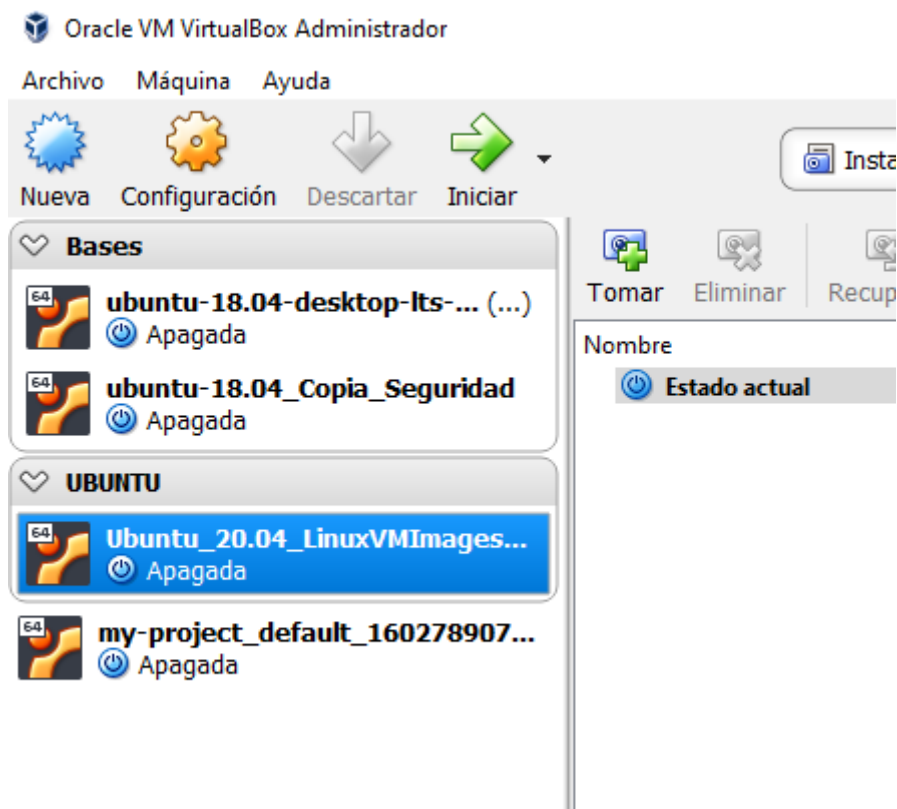
Además de instalar y configurar un servidor DNS para ambos dominios.

2.- Montar la máquina virtual

Para este ejercicio vamos a descargarnos una imagen de Ubuntu 20.04 para el virtual box.

 Ubuntu_20.04_LinuxVMImages.com

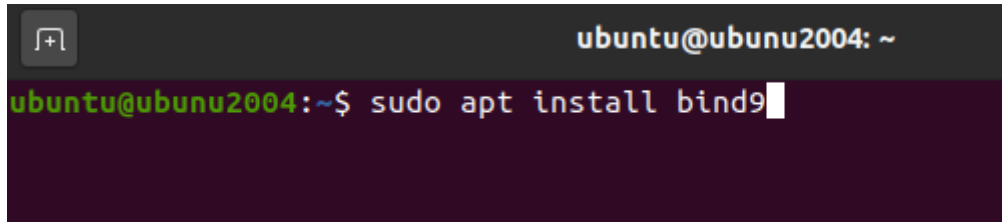
Abrimos VirtualBox e importamos la imagen.



Levantamos la máquina de Ubuntu 20.04 (Iniciar).

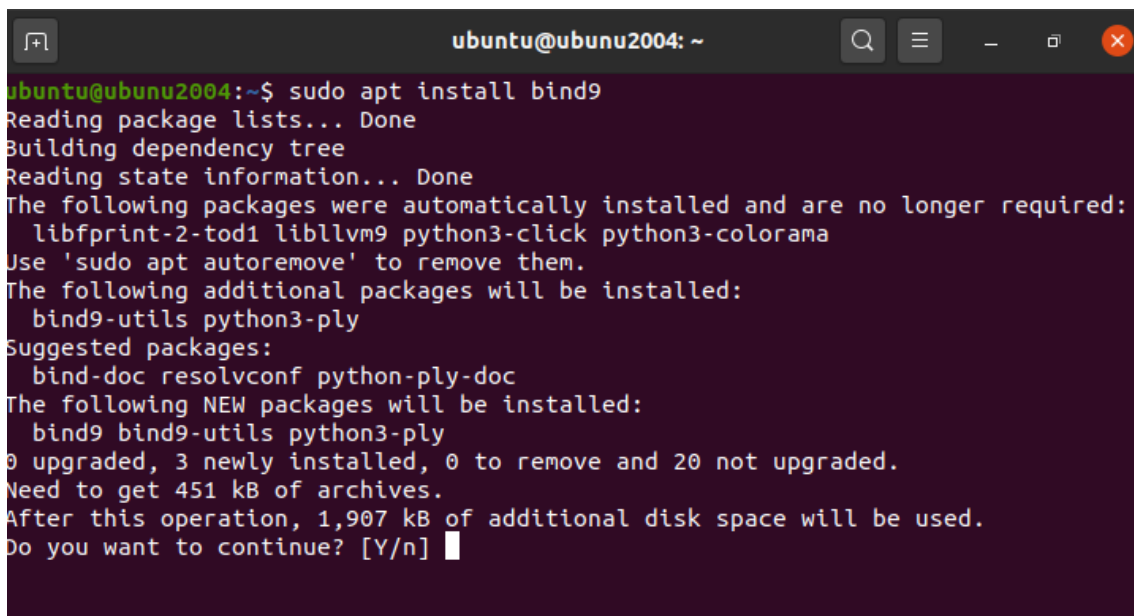
3.- Instalar y configurar Bind9

Abrimos el terminal CTRL+ALT+T en Linux, y escribimos “**sudo apt install bind9**” para instalar el servidor DNS.



```
ubuntu@ubunu2004: ~  
ubuntu@ubunu2004:~$ sudo apt install bind9
```

Nos pedirá una verificación y si estamos de acuerdo escribimos “**Y**”, pulsamos a “**Enter**”.

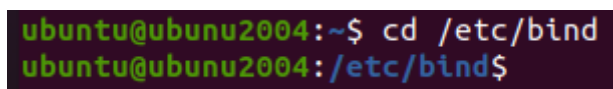


```
ubuntu@ubunu2004: ~  
ubuntu@ubunu2004:~$ sudo apt install bind9  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libfprint-2-tod1 libllvm9 python3-click python3-colorama  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  bind9-utils python3-ply  
Suggested packages:  
  bind-doc resolvconf python-ply-doc  
The following NEW packages will be installed:  
  bind9 bind9-utils python3-ply  
0 upgraded, 3 newly installed, 0 to remove and 20 not upgraded.  
Need to get 451 kB of archives.  
After this operation, 1,907 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

3.1.- Zona Directa

Una vez instalado nos vamos al fichero de configuración en la siguiente ruta

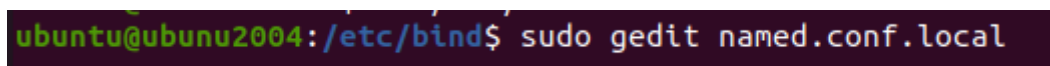
“**cd /etc/bind**”



```
ubuntu@ubunu2004:~$ cd /etc/bind  
ubuntu@ubunu2004:/etc/bind$
```

Editamos el fichero de configuración **named.conf.local**

“**sudo gedit named.conf.local**”



```
ubuntu@ubunu2004:/etc/bind$ sudo gedit named.conf.local
```

En este fichero vamos a indicar las zonas donde están configurados los ficheros para el **DNS**

```
Open ▾ /etc/bind *named.conf.local
1 //
2 // Do any local configuration here
3 //
4
5 // Consider adding the 1918 zones here, if they are not used in your
6 // organization
7 //include "/etc/bind/zones.rfc1918";
8 zone "jasejitu.com" {
9     type master;
10    file "/etc/bind/db.jasejitu";
11 };
12
```

En el “**file**” le vamos a indicar donde va a estar nuestro fichero de configuración que importamos.

Salvamos el fichero “**Save**”.

Vamos a crear el fichero de configuración a partir de uno ya creado (“**db.local**”), indicando el nombre que va a tener dicha copia.

“**sudo cp db.local db.jasejitu**”

Editamos el fichero “**sudo gedit db.jasejitu**” para modificar el **localhost** por el nombre del dominio que hemos elegido **jasejitu.com**

```
Open ▾ /etc/bind db.jasejitu
1 ;
2 ; BIND data file for local loopback interface
3 ;
4 $TTL      604800
5 @         IN      SOA      jasejitu.com. root.jasejitu.com. (
6             2          ; Serial
7             604800     ; Refresh
8             86400      ; Retry
9             2419200    ; Expire
10            604800 )     ; Negative Cache TTL
11 ;
12 @         IN      NS       jasejitu.com.
13 @         IN      A        192.168.1.38
14 www      IN      CNAME     jasejitu.com.
15
```

Salvamos el fichero y ahora reiniciamos el proceso **bind** con el comando “**sudo service bind9 restart**”

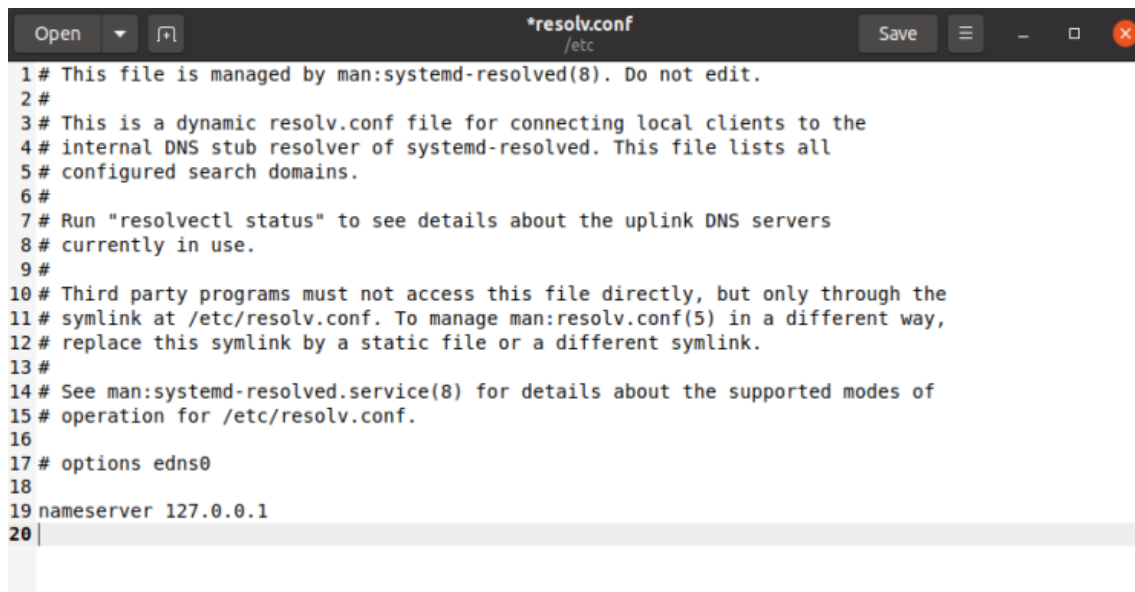
```
ubuntu@ubunu2004:/etc/bind$ sudo service bind9 restart
ubuntu@ubunu2004:/etc/bind$
```

Una vez reiniciado el servicio editamos el fichero de configuración **resolv.conf** para que se utilice este servidor DNS que acabamos de configurar.

“**sudo gedit /etc/resolv.conf**”

```
ubuntu@ubunu2004:/etc/bind$ sudo gedit /etc/resolv.conf
```

Cambiamos la línea **nameserver** 127.0.0.53 por **nameserver** 127.0.0.1 en el fichero



```
*resolv.conf
/etc
Open Save
1 # This file is managed by man:systemd-resolved(8). Do not edit.
2 #
3 # This is a dynamic resolv.conf file for connecting local clients to the
4 # internal DNS stub resolver of systemd-resolved. This file lists all
5 # configured search domains.
6 #
7 # Run "resolvectl status" to see details about the uplink DNS servers
8 # currently in use.
9 #
10 # Third party programs must not access this file directly, but only through the
11 # symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
12 # replace this symlink by a static file or a different symlink.
13 #
14 # See man:systemd-resolved.service(8) for details about the supported modes of
15 # operation for /etc/resolv.conf.
16
17 # options edns0
18
19 nameserver 127.0.0.1
20 |
```

Salvamos.

Probamos que todo esté correcto:

“host jasejitu.com”



```
ubuntu@ubuntu2004: /etc/bind
ubuntu@ubuntu2004:/etc/bind$ sudo gedit /etc/resolv.conf
(gedit:5119): Tepl-WARNING **: 17:51:32.805: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
ubuntu@ubuntu2004:/etc/bind$ host jasejitu.com
jasejitu.com has address 192.168.1.38
ubuntu@ubuntu2004:/etc/bind$
```

Repetimos el mismo proceso para el **host jagarciamateos.com**

Editamos el fichero **named.conf.local**



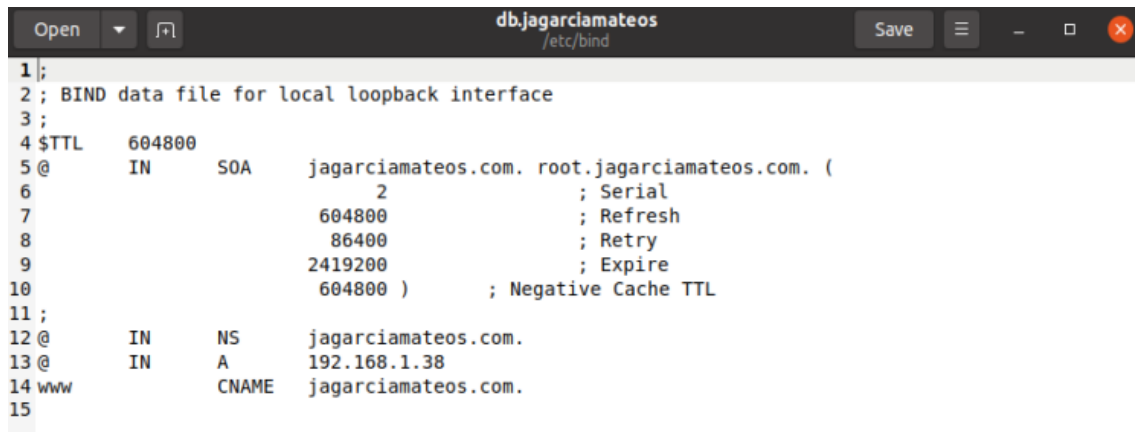
```
*named.conf.local
/etc/bind
Open Save
1 //
2 // Do any local configuration here
3 //
4
5 // Consider adding the 1918 zones here, if they are not used in your
6 // organization
7 //include "/etc/bind/zones.rfc1918";
8 zone "jasejitu.com" {
9     type master;
10    file "/etc/bind/db.jasejitu";
11 };
12
13 zone "jagarciamateos.com" {
14     type master;
15     file "/etc/bind/db.jagarciamateos";
16 };
17
```

Vamos a crear el fichero de configuración a partir de uno ya creado (“db.jasejitu”).

“sudo cp db.jasejitu db.jagarciamateos”

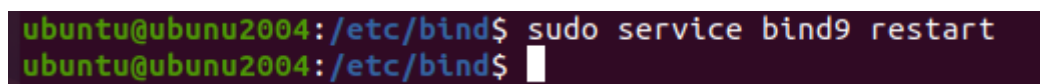
Editamos el fichero **db.jagarciamateos** para modificar el **jasejitu.com** por el nombre del dominio que hemos elegido **jagarciamateos.com**

“**sudo gedit db.jagarciamateos**”



```
1;
2; BIND data file for local loopback interface
3;
4 $TTL      604800
5 @         IN      SOA      jagarciamateos.com. root.jagarciamateos.com. (
6                               2             ; Serial
7                               604800        ; Refresh
8                               86400         ; Retry
9                               2419200       ; Expire
10                              604800 )      ; Negative Cache TTL
11;
12 @         IN      NS       jagarciamateos.com.
13 @         IN      A        192.168.1.38
14 www       IN      CNAME    jagarciamateos.com.
15
```

Salvamos el fichero y ahora reiniciamos el proceso **bind** con el comando “**sudo service bind9 restart**”



```
ubuntu@ubunu2004:/etc/bind$ sudo service bind9 restart
ubuntu@ubunu2004:/etc/bind$
```

Probamos que todo esté correcto:

“**host jasejitu.com**”

“**host jagarciamateos.com**”



```
ubuntu@ubunu2004:/etc/bind$ host www.jatuapellido.com
www.jatuapellido.com is an alias for jatuapellido.com.
jatuapellido.com has address 192.168.1.38
ubuntu@ubunu2004:/etc/bind$
```

3.2.- Zona Inversa

Vamos a configurar el servidor DNS para que resuelva los dominios a la inversa, poniendo la dirección IP y que nos devuelva el dominio que pertenece a dicha dirección IP. Para ello vamos a volver a modificar el fichero **named.conf.local**

“**sudo gedit named.conf.local**”

Y añadimos la siguiente zone

zone “1.168.192.in-addr.arpa” IN{

type master;

file “/etc/bind/db.192”;

allow-update {none; };

```
allow-transfer {192.168.1.38 };  
  
};
```

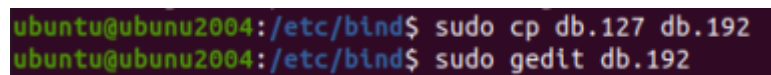


```
1 //  
2 // Do any local configuration here  
3 //  
4  
5 // Consider adding the 1918 zones here, if they are not used in your  
6 // organization  
7 //include "/etc/bind/zones.rfc1918";  
8 zone "jasejitu.com" {  
9     type master;  
10    file "/etc/bind/db.jasejitu";  
11 };  
12  
13 zone "jagarciamateos.com" {  
14     type master;  
15     file "/etc/bind/db.jagarciamateos";  
16 };  
17  
18 zone "1.168.192.in-addr.arpa" IN{  
19     type master;  
20     file "/etc/bind/db.192";  
21     allow-update {none;};  
22     allow-transfer {192.168.1.38; };  
23 };  
24
```

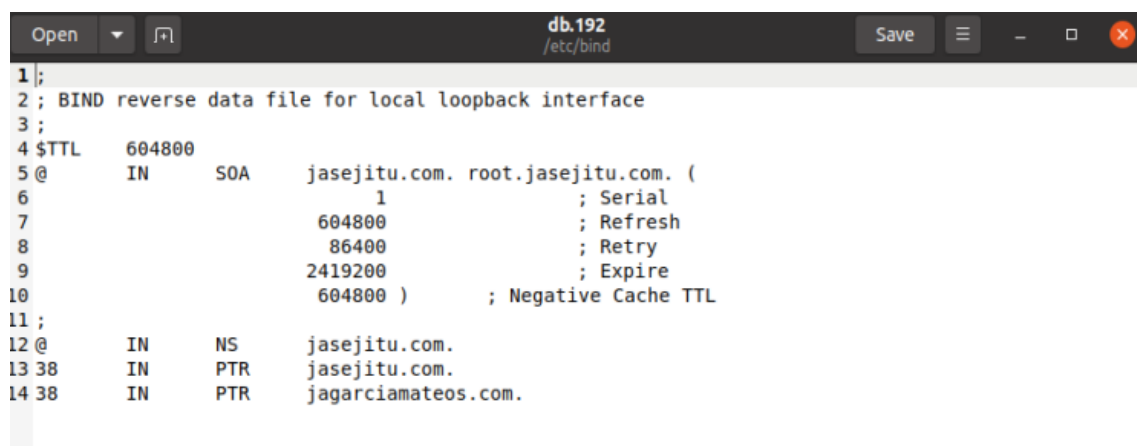
Vamos a crear el fichero de configuración a partir de uno ya creado (“**db.127**”), indicando el nombre que va a tener dicha copia (“**db.192**”) y editarlo.

“**sudo cp db.127 db.192**”

“**sudo gedit db.192**”



```
ubuntu@ubunu2004:/etc/bind$ sudo cp db.127 db.192  
ubuntu@ubunu2004:/etc/bind$ sudo gedit db.192
```



```
1 ;  
2 ; BIND reverse data file for local loopback interface  
3 ;  
4 $TTL      604800  
5 @        IN      SOA      jasejitu.com. root.jasejitu.com. (  
6          1          ; Serial  
7          604800     ; Refresh  
8          86400      ; Retry  
9          2419200    ; Expire  
10         604800 )      ; Negative Cache TTL  
11 ;  
12 @        IN      NS       jasejitu.com.  
13 38       IN      PTR      jasejitu.com.  
14 38       IN      PTR      jagarciamateos.com.
```

Cambiamos el localhost por el host de **jasejitu** y añadimos el de **jagarciamateos**

Ponemos el último octeto de la **ip** y la **PTR** donde apunte.

PTR -> PoinTeR de correspondencia IP – nombre del host.

Comprobamos con el siguiente comando:

“host 192.168.1.38”

```
ubuntu@ubuntu2004:/etc/bind$ host 192.168.1.38
38.1.168.192.in-addr.arpa domain name pointer jasejitu.com.
38.1.168.192.in-addr.arpa domain name pointer jagarciamateos.com.
```

Con el comando: **“dig @192.168.1.38 -x 192.168.1.38”** que nos devolverá algo parecido a esto

```
ubuntu@ubuntu2004:/etc/bind$ dig @192.168.1.38 -x 192.168.1.38

; <<>> DiG 9.16.1-Ubuntu <<>> @192.168.1.38 -x 192.168.1.38
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 21031
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 66a183578e672f2e010000005f9f1b361b9bbcb039e8c382 (good)
;; QUESTION SECTION:
;38.1.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
38.1.168.192.in-addr.arpa. 604800 IN      PTR      jasejitu.com.
38.1.168.192.in-addr.arpa. 604800 IN      PTR      jagarciamateos.com.

;; Query time: 0 msec
;; SERVER: 192.168.1.38#53(192.168.1.38)
;; WHEN: Sun Nov 01 21:31:50 CET 2020
;; MSG SIZE rcvd: 137

ubuntu@ubuntu2004:/etc/bind$
```

4.- Instalar y configurar Apache

Vamos a necesitar un servidor Web como apache para Linux para poder crear nuestros hosts virtuales.

Comenzamos actualizando el índice de paquetes locales para que reflejen los últimos cambios anteriores con el comando **“sudo apt update”** y luego **“sudo apt upgrade”**

Instalaremos apache desde un terminal con el siguiente comando **“sudo apt install apache2”**

```
ubuntu@ubuntu2004:/etc/bind$ sudo apt install apache2
```

Enumeramos los perfiles de la aplicación **ufw** con el comando **“sudo ufw app list”**

```
ubuntu@ubuntu2004:/etc/bind$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  Bind9
  CUPS
  OpenSSH
ubuntu@ubuntu2004:/etc/bind$
```

Tenemos tres perfiles disponibles para Apache:

- **Apache:** este perfil abre solo el puerto 80 (tráfico web normal no cifrado)
- **Apache Full:** este perfil abre el puerto 80 (tráfico web normal no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)
- **Apache Secure:** este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

Se recomienda habilitar el perfil más restrictivo, pero en este caso, solo deberemos permitir el tráfico en el puerto 80:

“sudo ufw allow ‘Apache’”

Para ver el cambio escribimos el siguiente comando:

“sudo ufw status”

4.1.- Comprobar el servidor Web

Comprobamos el servidor web con el comando **“sudo systemctl status apache2”**

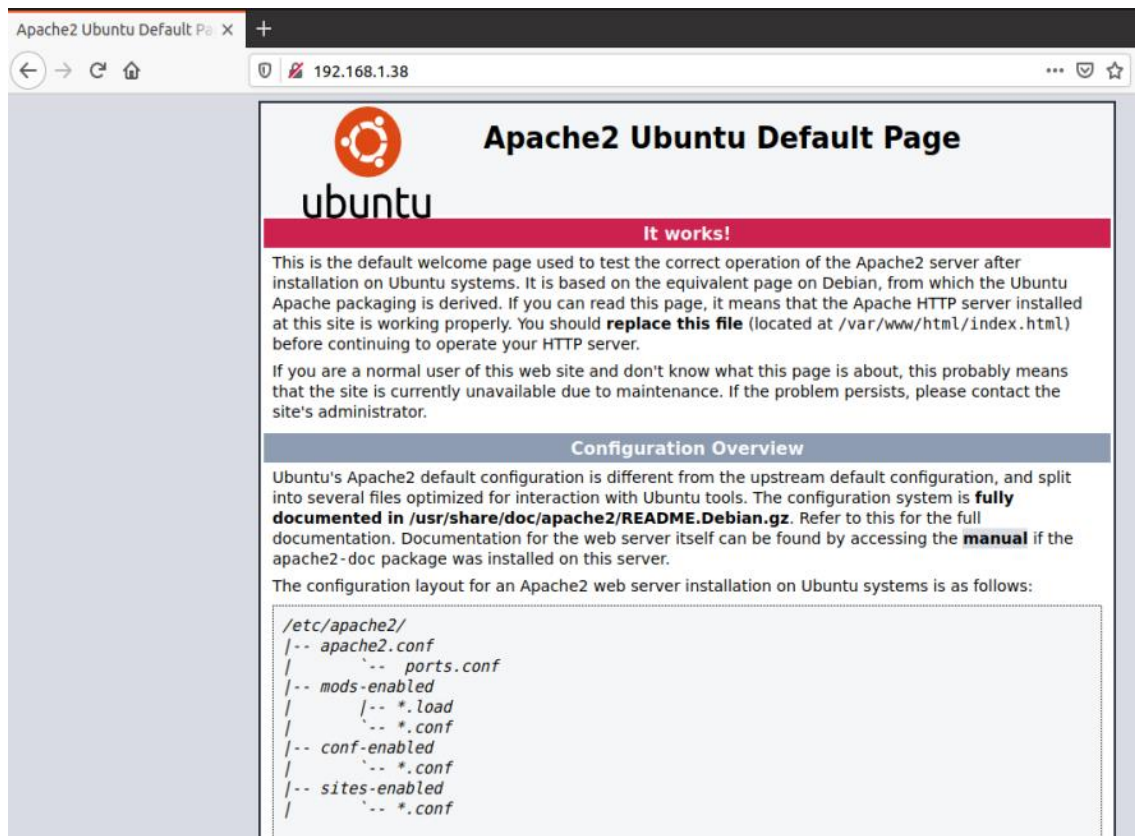
```
ubuntu@ubuntu2004:/etc/bind$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Sun 2020-11-01 19:32:55 CET; 4min 22s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3449 (apache2)
      Tasks: 55 (limit: 7069)
     Memory: 5.0M
    CGroup: /system.slice/apache2.service
            └─3449 /usr/sbin/apache2 -k start
              └─3451 /usr/sbin/apache2 -k start
                └─3452 /usr/sbin/apache2 -k start

Nov 01 19:32:54 ubuntu2004 systemd[1]: Starting The Apache HTTP Server...
Nov 01 19:32:55 ubuntu2004 systemd[1]: Started The Apache HTTP Server.
lines 1-14/14 (END)
```

Escribimos el siguiente comando **“sudo hostname -I”** para obtener algunas direcciones separadas por espacios para poder probarlas en el navegador web y ver que funcionan.

```
ubuntu@ubuntu2004:/etc/bind$ sudo hostname -I
192.168.1.38
```

Introducimos la **ip** en la **url** del navegador:



Obtenemos la web predeterminada de Apache en Ubuntu.

4.2.- Comandos básicos Apache

- Para detener el servidor web: **"sudo systemctl stop apache2"**
- Para iniciar el servidor web cuando no esté activo: **"sudo systemctl start apache2"**
- Para detener y luego iniciar el servicio de nuevo: **"sudo systemctl restart apache2"**
- Para recargar sin cerrar conexiones: **"sudo systemctl reload apache2"**
- Para deshabilitar el inicio automático: **"sudo systemctl disable apache2"**
- Para volver a habilitar el servicio de modo que se cargue en el inicio: **"sudo systemctl enable apache2"**

4.3- Configurar hosts virtuales

Creamos primero la estructura de directorios para cada dominio

"sudo mkdir /var/www/jasejitu"

"sudo mkdir /var/www/jagarciamateos"

```
ubuntu@ubunu2004:/etc/bind$ sudo mkdir /var/www/jasejitu  
ubuntu@ubunu2004:/etc/bind$ sudo mkdir /var/www/jagarciamateos
```

Ahora asignamos la propiedad del directorio con la variable entorno \$USER

"sudo chown -R \$USER:\$USER /var/www/jasejitu"

"sudo chown -R \$USER:\$USER /var/www/jagarciamateos"

```
ubuntu@ubuntu2004:/etc/bind$ sudo chown -R $USER:$USER /var/www/jasejitu
ubuntu@ubuntu2004:/etc/bind$ sudo chown -R $USER:$USER /var/www/jagarciamateos
```

Nos aseguramos que los permisos sean los correctos y se pueda permitir al propietario leer, escribir y ejecutar los archivos, y a la vez conceder solo permisos de lectura y ejecución a los grupos y terceros

"sudo chmod -R 755 /var/www/jasejitu"

"sudo chmod -R 755 /var/www/jagarciamateos"

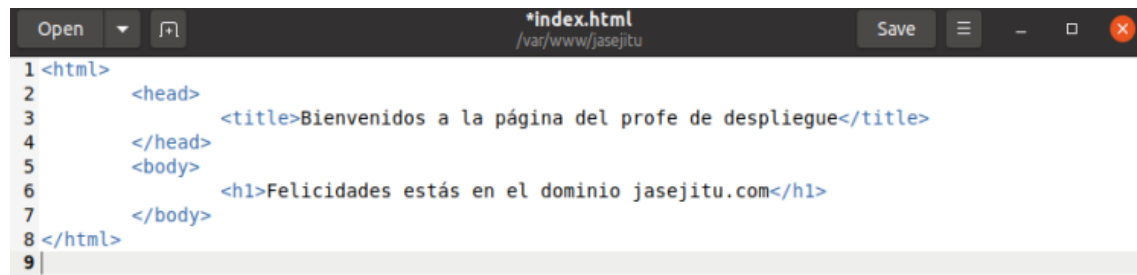
```
ubuntu@ubuntu2004:/etc/bind$ sudo chmod -R 755 /var/www/jasejitu
ubuntu@ubuntu2004:/etc/bind$ sudo chmod -R 755 /var/www/jagarciamateos
```

A continuación crearemos una página de ejemplo index.html para cada hosts virtual utilizando un editor.

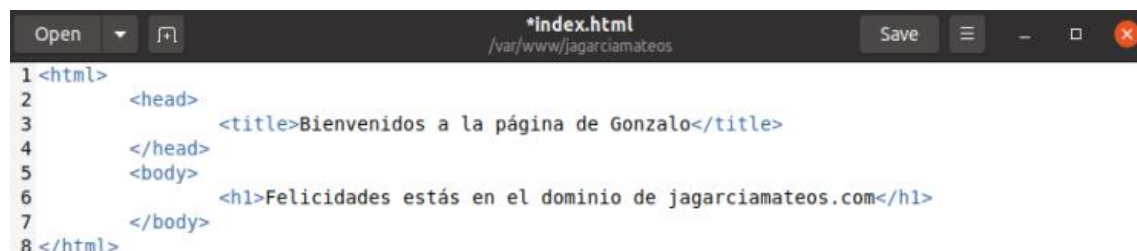
"sudo gedit /var/www/jasejitu/index.html"

"sudo gedit /var/www/jagarciamateos/index.html"

```
ubuntu@ubuntu2004:/etc/bind$ sudo gedit /var/www/jasejitu/index.html
```



```
ubuntu@ubuntu2004:/etc/bind$ sudo gedit /var/www/jagarciamateos/index.html
```



Para que Apache proporcione este contenido, es necesario crear un archivo de configuración del host virtual con las directivas correctas.

"sudo gedit /etc/apache2/sites-available/ jasejitu.conf" y añadimos lo siguiente:



```
*jasejitu.conf
/etc/apache2/sites-available

1 <VirtualHost *:80>
2     ServerAdmin jasejitu@jasejitu.com
3     ServerName jasejitu.com
4     ServerAlias www.jasejitu.com
5     DocumentRoot /var/www/jasejitu
6     ErrorLog ${APACHE_LOG_DIR}/error.log
7     CustomLog ${APACHE_LOG_DIR}/access.log combined
8 </VirtualHost>
9
```

Salvamos y salimos.

“**sudo gedit /etc/apache2/sites-available/ jagarciamateos.conf**” y añadimos lo siguiente:



```
*jagarciamateos.conf
/etc/apache2/sites-available

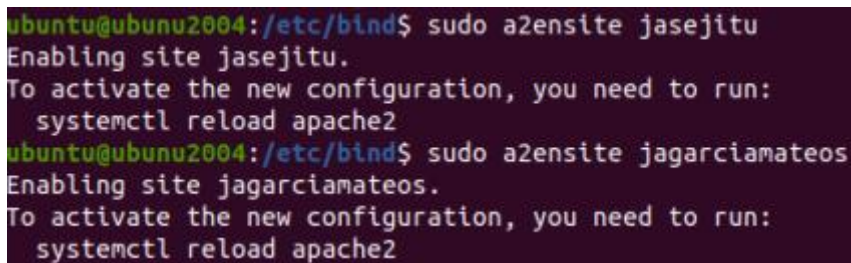
1 <VirtualHost *:80>
2     ServerAdmin jagarciamateos@jagarciamateos.com
3     ServerName jagarciamateos.com
4     ServerAlias www.jagarciamateos.com
5     DocumentRoot /var/www/jagarciamateos
6     ErrorLog ${APACHE_LOG_DIR}/error.log
7     CustomLog ${APACHE_LOG_DIR}/access.log combined
8 </VirtualHost>
9
```

Salvamos y salimos.

Habilitaremos el archivo con la herramienta a2ensite:

“**sudo a2ensite jasejitu**”

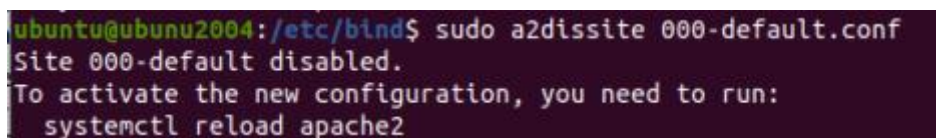
“**sudo a2ensite jagarciamateos**”



```
ubuntu@ubuntu2004:/etc/bind$ sudo a2ensite jasejitu
Enabling site jasejitu.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ubuntu2004:/etc/bind$ sudo a2ensite jagarciamateos
Enabling site jagarciamateos.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Deshabilitamos el sitio predeterminado definido en 000-default.conf con el siguiente comando:

“**sudo a2dissite 000-default.conf**”



```
ubuntu@ubuntu2004:/etc/bind$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Hacemos una prueba para ver que no existan errores de configuración con el siguiente comando:

“**sudo apache2ctl configtest**”

```
ubuntu@ubuntu2004:/etc/bind$ sudo apache2ctl configtest  
Syntax OK
```

Reiniciamos Apache para implementar los cambios:

“sudo systemctl restart apache2”

```
ubuntu@ubuntu2004:/etc/bind$ sudo systemctl restart apache2
```

Probamos introduciendo el dominio en la url del navegador: **jasejitu.com**



Probamos introduciendo el dominio en la url del navegador: **www.jasejitu.com**



Probamos introduciendo el dominio en la url del navegador: **jagarciamateos.com**



Probamos introduciendo el dominio en la url del navegador: **www.jagarciamateos.com**



Archivos subidos a mi repositorio de github:

https://github.com/Kurtgon/2DAW_Despliegue_Aplicaciones_Web/tree/main/Tema2