

**Gestió i administració de xarxes:**  
**Pràctica 3**

**Carles Costas Mateu - 1491578**  
**Maksym Lakhmanets - 1495282**  
**18/11/2021**

## 1. Instal·lació d'Apache

Instal·larem en la màquina master el paquet de apache2 con la comanda apt install apache2:

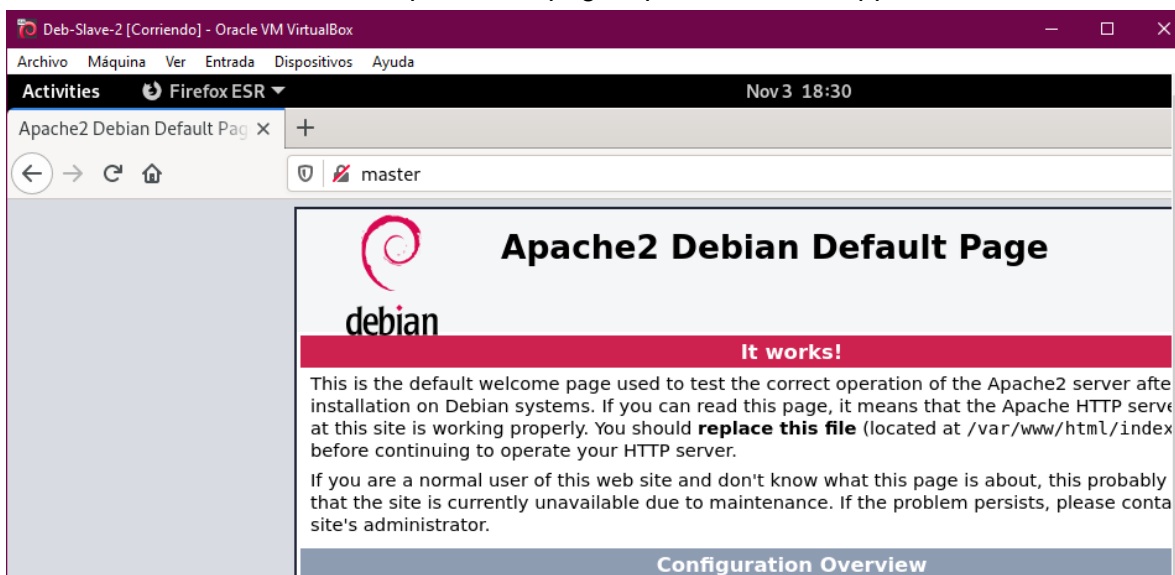
```
root@master:~# apt install apache2
```

Una vez descargado, iniciamos el servicio de apache2 y comprobamos que el estado es running.

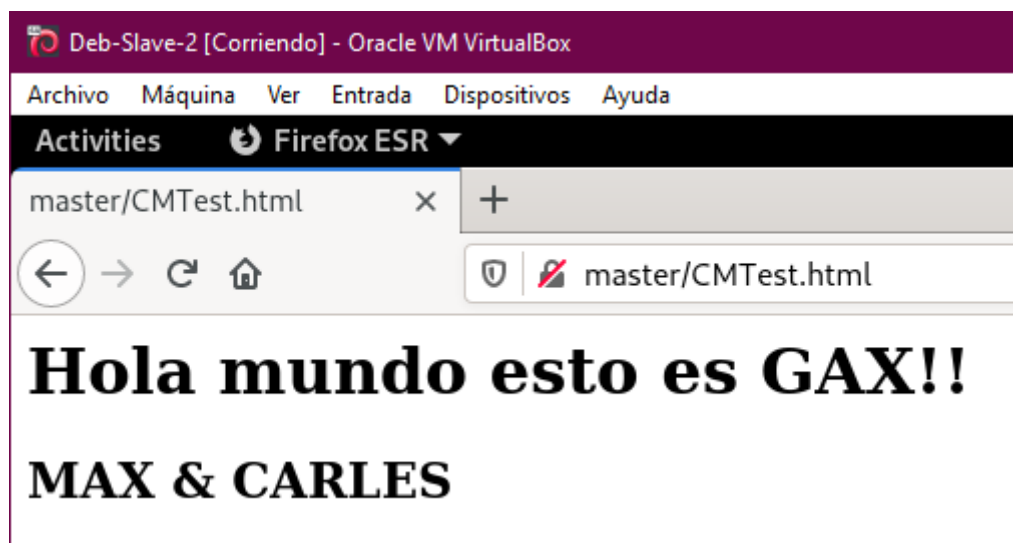
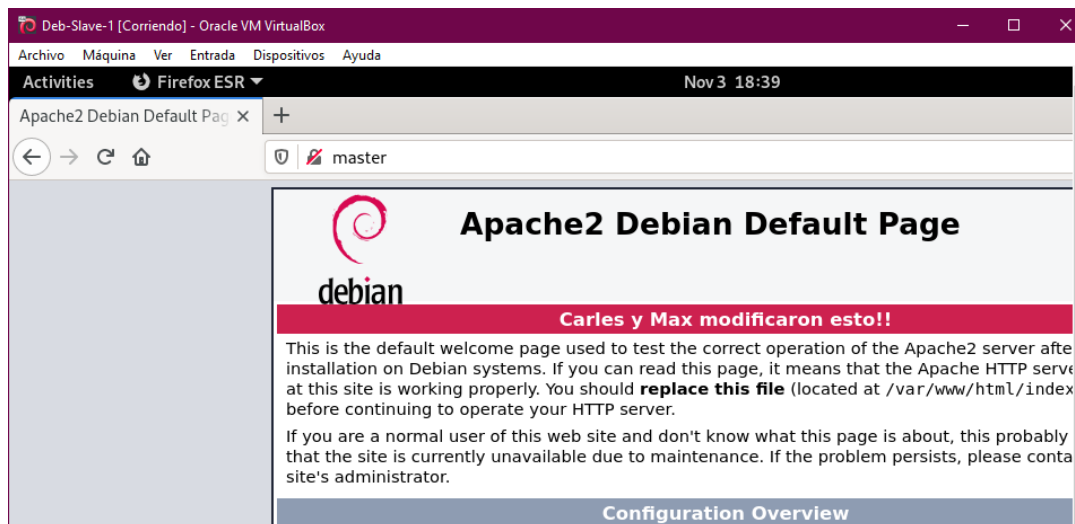
```
root@master:~# systemctl start apache2
root@master:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-11-03 18:23:13 CET; 2min 3s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 2525 (apache2)
      Tasks: 55 (limit: 1117)
     Memory: 11.1M
        CPU: 193ms
    CGroup: /system.slice/apache2.service
            └─2525 /usr/sbin/apache2 -k start
              └─2527 /usr/sbin/apache2 -k start
                └─2528 /usr/sbin/apache2 -k start

Nov 03 18:23:12 master systemd[1]: Starting The Apache HTTP Server...
Nov 03 18:23:13 master systemd[1]: Started The Apache HTTP Server.
root@master:~#
```

Accedemos a las maquinas slave1 y 2 y desde el buscador web accedemos a la url: master:80 donde nos debería aparecer la página por defecto de Appache2.



Accedemos al directorio /var/www/html/ donde encontraremos el fichero index.html el cual contiene el código html de la página de inicio de apache. Aquí podemos crear un nuevo html modificar el original.



Dado que tenemos el servicio en una máquina virtual, para poder acceder a la página por defecto de apache2 deberemos realizar un port forwarding de la interfaz nat de nuestra máquina virtual de la siguiente manera:

master - Configuración

**Red**

Adaptador 1   Adaptador 2   Adaptador 3   Adaptador 4

☒ Habilitar adaptador de red

Conectado a: NAT

Nombre:

Avanzadas

Tipo de adaptador: Intel PRO/1000 MT Desktop (82540EM)

Modo promiscuo: Denegar

Dirección MAC: 080027E48458

☒ Cable conectado

Reenvío de puertos

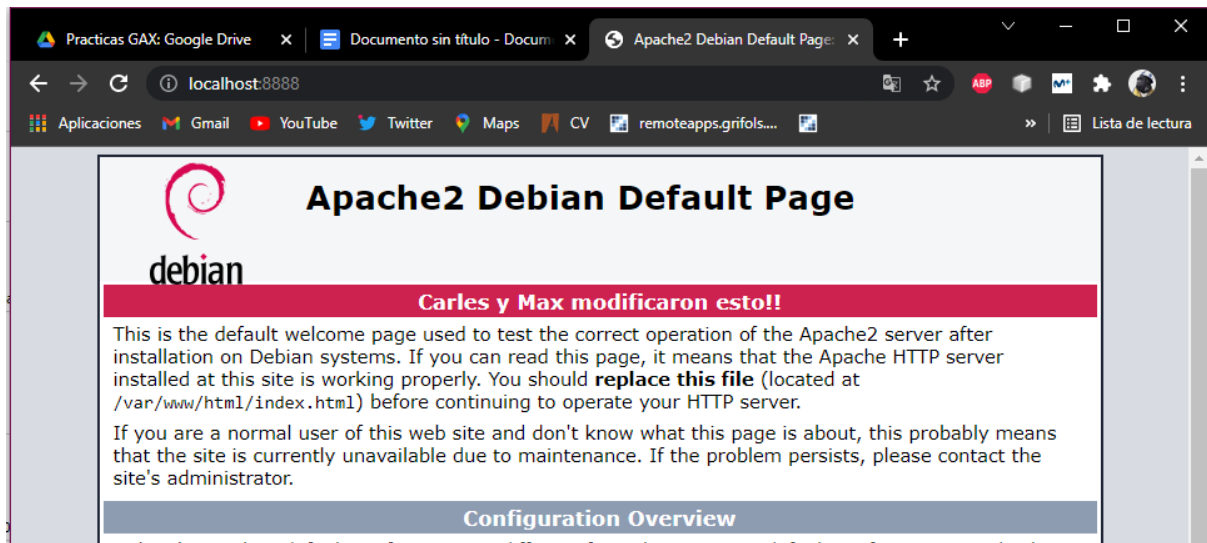
Aceptar Cancelar

Reglas de reenvío de puertos

Nombre	Protocolo	IP anfitrión	Puerto anfitrión	IP invitado	Puerto invitado
Rule 1	TCP		8888		80

Aceptar Cancelar

Una vez realizado el port forwarding, podemos comprobar que desde la máquina host, podemos acceder a la página web, desde el puerto 8888.



## 2. Mòduls d'Apache

Ahora procederemos a instalar los módulos de php y python para apache.

```
root@master:/var/log/apache2# apt install libapache2-mod-php7.4
root@master:/var/log/apache2# apt install libapache2-mod-python
```

Ejecutamos la siguiente comanda para que apache pueda ejecutar python y python un script dentro de una página web.

```
root@master:/var/log/apache2# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@master:/var/log/apache2# systemctl restart apache2
root@master:/var/log/apache2#
```

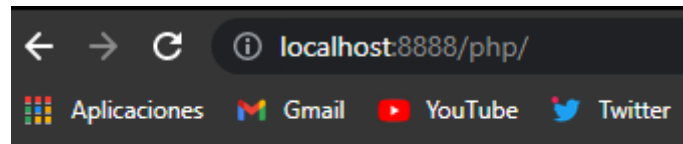
Verificamos que los modulos estan correctamente instalados con apachectl -M

```
root@master:/var/log/apache2# apachectl -M
Loaded Modules:
  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
  unixd_module (static)
  access_compat_module (shared)
  alias_module (shared)
  auth_basic_module (shared)
  authn_core_module (shared)
  authn_file_module (shared)
  authz_core_module (shared)
  authz_host_module (shared)
  authz_user_module (shared)
  autoindex_module (shared)
  cgi_module (shared)
  deflate_module (shared)
  dir_module (shared)
  env_module (shared)
  filter_module (shared)
  mime_module (shared)
  mpm_prefork_module (shared)
  negotiation_module (shared)
  php7_module (shared)
  python_module (shared)
  reqtimeout_module (shared)
  setenvif_module (shared)
  status_module (shared)
root@master:/var/log/apache2#
```

Creamos la carpeta php y un fichero index.php que ejecute un código php y muestre el texto de la imagen.

```
root@master:/var/www/html# mkdir php
root@master:/var/www/html# cd php/
root@master:/var/www/html/php# vi index.php
root@master:/var/www/html/php# cat index.php
<?php
print "Hola GAX des de PHP!";
?>
```

Ahora si cargamos la página con /php/ veremos que ejecuta el php correctamente y se visualiza el contenido.



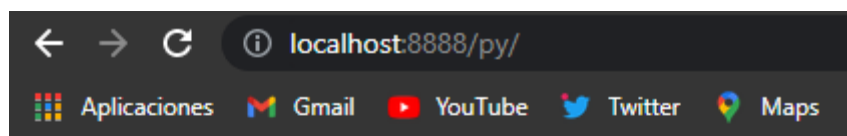
Hola GAX des de PHP!

Realizaremos los mismos pasos para python, esta vez tendremos que dar permisos de ejecución al script, pero como no tenemos interprete el .py no se ejecutara y nos mostrara el directorio.



```
root@master:/var/www/html# mkdir py
root@master:/var/www/html# cd py/
root@master:/var/www/html/py# vi
```

```
root@master:/var/www/html/py# cat index.py
#!/usr/bin/python3
print("Content-Type: test/html\n\n")
print('Hola GAX des de Python!')
root@master:/var/www/html/py#
```

```
root@master:/var/www/html/py# rm index.html
root@master:/var/www/html/py# chmod +x index.py
root@master:/var/www/html/py#
```



## Index of /py

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">index.py</a>	2021-11-03 19:10	89	

*Apache/2.4.51 (Debian) Server at localhost Port 8888*

### 3. Hosts virtuales configurats per IP

Dado que Apache no es capaz de interpretar python, vamos a modificar la configuración para que ejecute los archivos .py con un intérprete de python. Para ello desactivaremos el site por defecto y crearemos dos sites nuevos uno para php y otro para python. Los cuales llamaremos fuera.conf y dentro.conf

```

<VirtualHost 10.0.2.15:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName mv-a-gax.org

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/py

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/html/py>
        DirectoryIndex index.py
        Options Indexes FollowSymLinks Multiviews ExecCGI
        AllowOverride None
        Order allow,deny
        allow from all
        AddHandler cgi-script .py
    </Directory>

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

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

```

Configuramos el archivo fuera.conf para que vaya por la ip 10.0.2.15 y index.py. Añadimos AddHandler cgi-script .py para permitir la interpretación de python por parte del servidor apache.



```

VirtualHost 172.16.1.1:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName mv-a-gax.org

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/php

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/html/php>
        DirectoryIndex index.php
        Options Indexes FollowSymLinks Multiviews ExecCGI
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

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

```

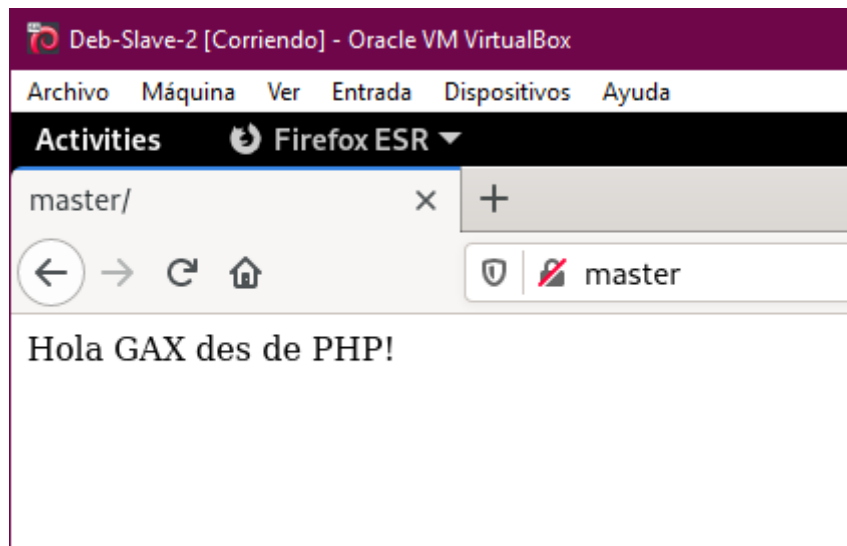
Indicamos la ip y el puerto en el que se tendra el site 172.16.1.1:80 dentro.conf

```

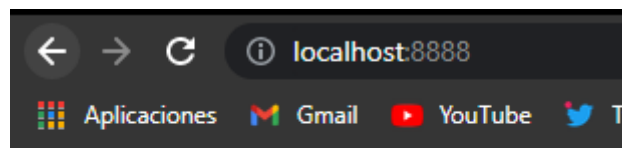
root@master:/etc/apache2/sites-available# a2ensite fuera.conf
Enabling site fuera.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@master:/etc/apache2/sites-available# a2ensite dentro.conf
Enabling site dentro.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@master:/etc/apache2/sites-available# █

```

Ejecutamos un a2ensite para levantar las páginas python (fuera) y php (dentro). Con systemctl reload apache2 reiniciamos apache para aplicar los cambios.



Accedemos a la master desde la red interna



Hola GAX des de Python!

Accedemos al localhost desde fuera del virtualbox.

#### 4. Certificats i HTTPS

Para crear nuestro certificado usaremos make-ssl-cert, el cual usará una plantilla para generar nuestro certificado autoasignado.

```
root@master:/var/www/html# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/private/master.crt
root@master:/var/www/html#
```

Una vez generado el certificado, deberemos añadir la ruta donde se encuentra, en el fichero de configuración de un nuevo site que estará disponible por SSL. Tal como se muestra a continuación se modifica la ruta del certificado en la variable SSLCertificateFile, se habilita el site con la comanda a2ensite y se hace un reload del servicio para aplicar los cambios.

Finalmente podemos comprobar que desde la máquina slave2 si accedemos por https, nos advierte que es una página web con certificado auto-signado.

```

<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile      /etc/ssl/private/master.crt
    #SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
    # certificate chain for the server certificate. Alternatively
    # the referenced file can be the same as SSLCertificateFile
    # when the CA certificates are directly appended to the server
    # certificate for convinience.
    #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

    # Certificate Authority (CA):
    # Set the CA certificate verification path where to find CA
    # certificates for client authentication or alternatively one
    # huge file containing all of them (file must be PEM encoded)
    # Note: Inside SSLCACertificatePath you need hash symlinks
    #       to point to the certificate files. Use the provided
    #       Makefile to update the hash symlinks after changes.
    #SSLCACertificatePath /etc/ssl/certs/
    #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

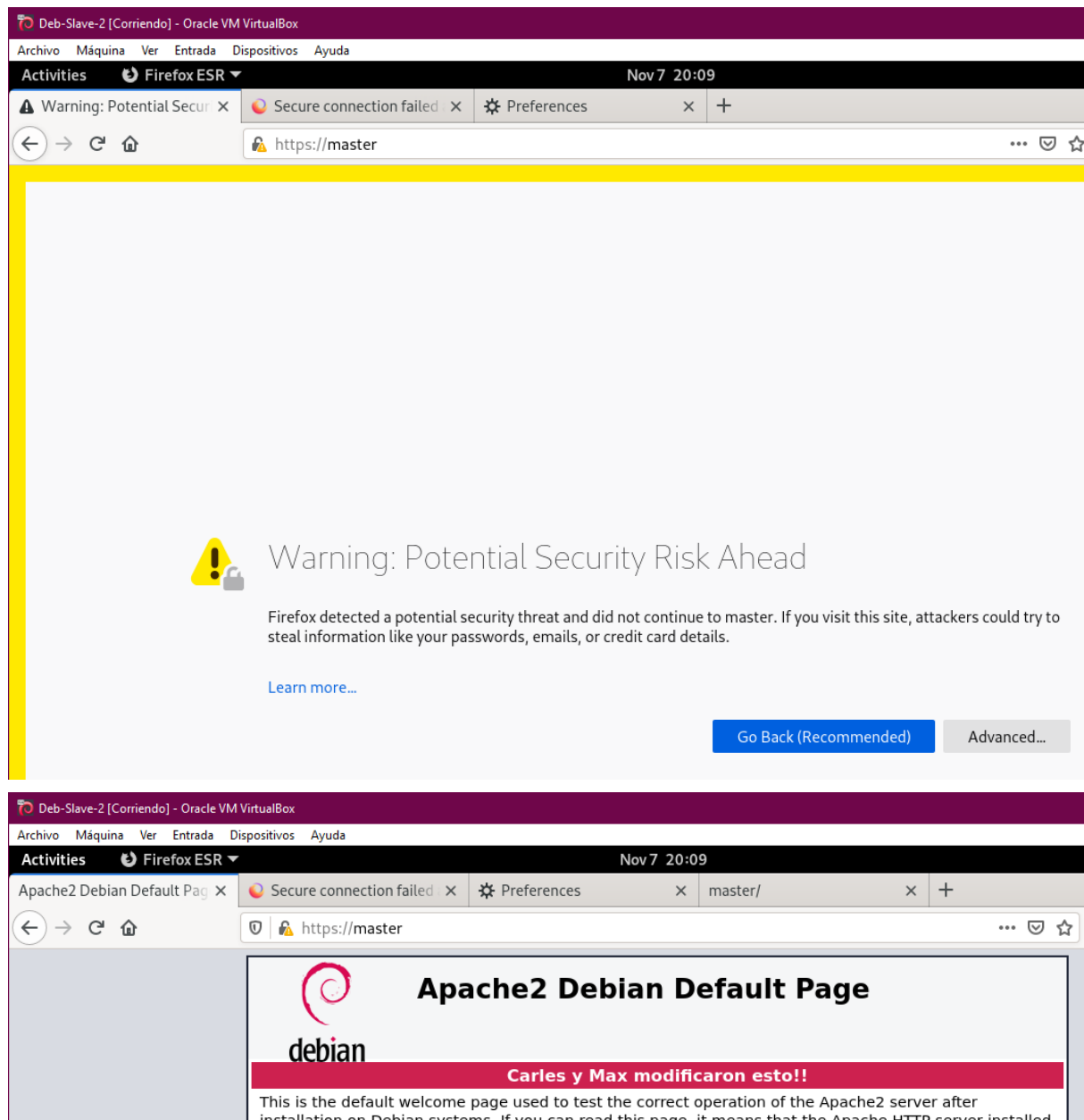
```

"gax-ssl.conf" 134L, 6330B

```

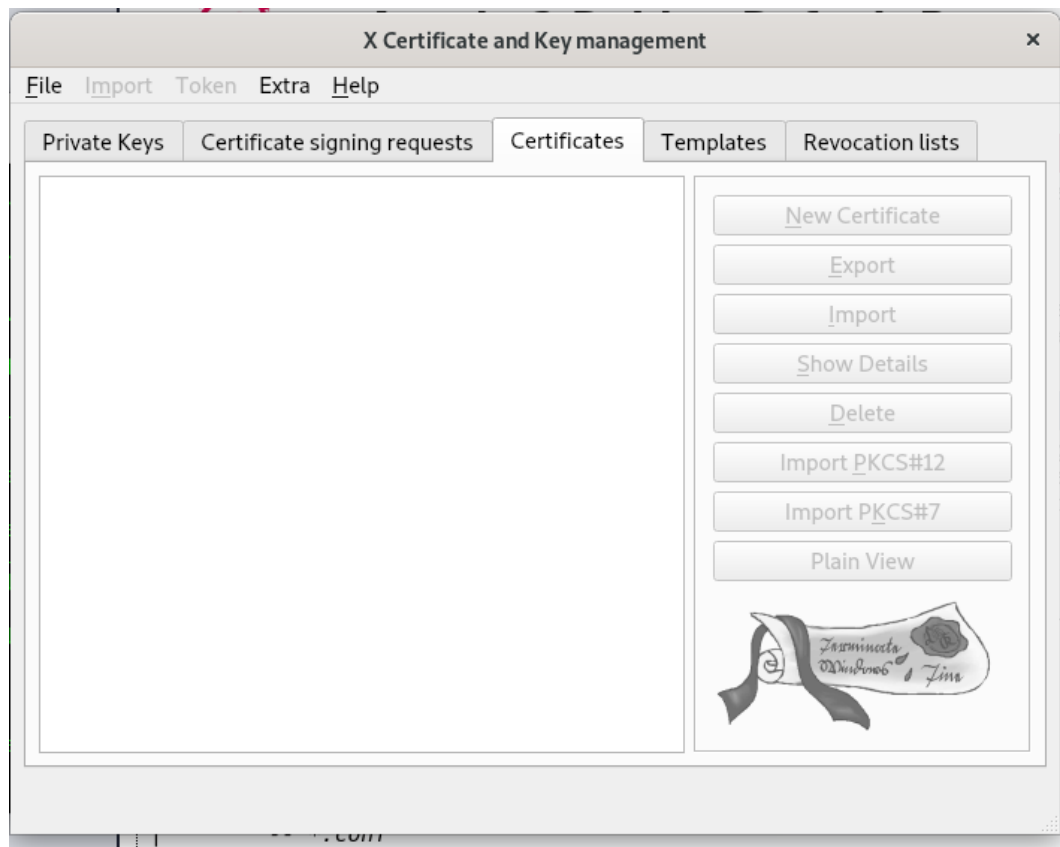
root@master:/etc/apache2/sites-available# a2ensite gax-ssl.conf
Enabling site gax-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@master:/etc/apache2/sites-available# systemctl reload apache2.service

```

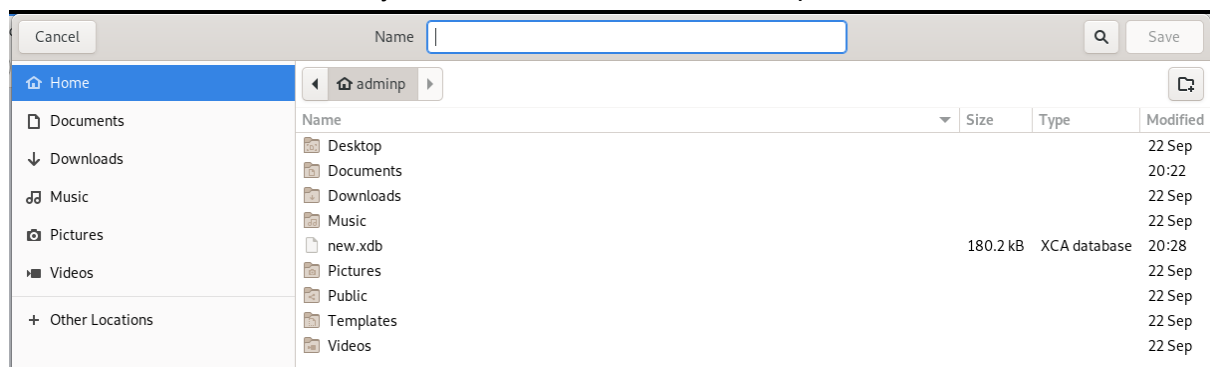


Dado que el certificado es auto-signado, nos advierte que no es segura la página web, por lo que deberíamos instalar una entidad certificadora.

Instalamos xca y abrimos la aplicación:



Creamos la nueva database y introducimos una contraseña “pnimda”



Ahora vamos a crear el certificado RootCA, este certificado es el que instalaremos en el navegador para que valide la cadena de confianza. Primero de todo deberemos crear un New Certificate, se nos abrirá una pantalla Create X509 Certificate donde podremos configurar la información de identificación. También generamos la Key con el tipo RSA.

X Certificate and Key management

### Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

**Signing request**

☐ Sign this Certificate signing request

☒ Copy extensions from the request

☐ Modify subject of the request

**Signing**

☒ Create a self signed certificate

☐ Use this Certificate for signing

**Signature algorithm** SHA 256

**Template for the new certificate**

[default] Empty template

Apply extensions Apply subject Apply all

Cancel OK

X Certificate and Key management

### Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

Distinguished name

countryName

stateOrProvinceName

localityName

organizationName

Type

Private key

Used keys too Generate a new key

Cancel OK

X Certificate and Key management

#### New Key

Please give a name to the new key and select the desired keysize

Key properties

Name UAB

Keytype RSA

Keysize 2048 bit

☐ Remember as default

Cancel Create

X Certificate and Key management

### Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

Distinguished name

countryName	<input type="text" value="UAB"/>	organizationalUnitName	<input type="text" value="UAB"/>
stateOrProvinceName	<input type="text" value="UAB"/>	commonName	<input type="text" value="UAB"/>
localityName	<input type="text" value="UAB"/>	emailAddress	<input type="text" value="UAB@uab.cat"/>
organizationName	<input type="text" value="UAB"/>		

Type

--	--

Private key

☐ Used keys too

X Certificate and Key management

Successfully created the RSA private key 'UAB'

X Certificate and Key management

### Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

X509v3 Basic Constraints

Type  Path length

☐ Critical

Key identifier

☐ X509v3 Subject Key Identifier

☐ X509v3 Authority Key Identifier

Validity

Not before  Not after

Time range

☐ Midnight ☐ Local time ☐ No well-defined expiration

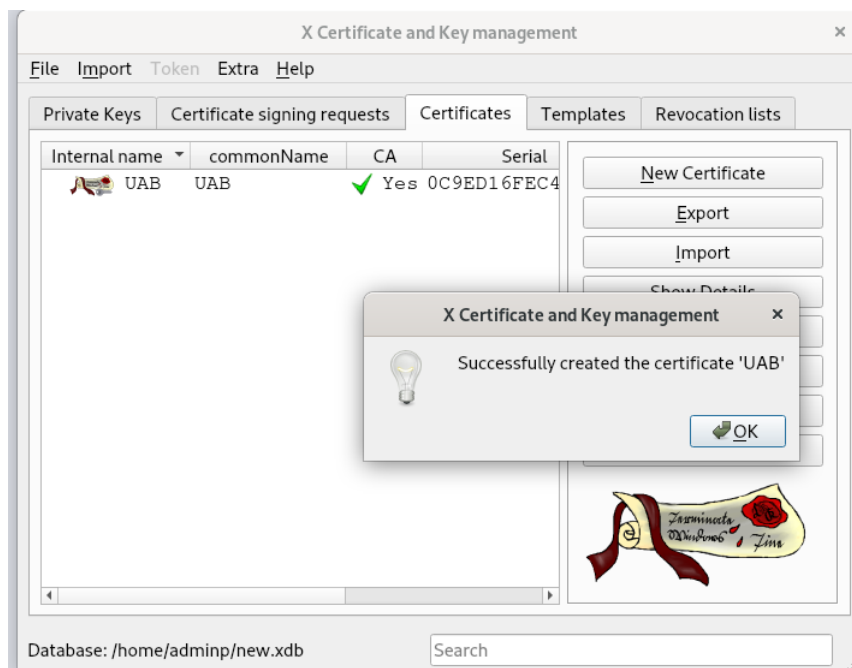
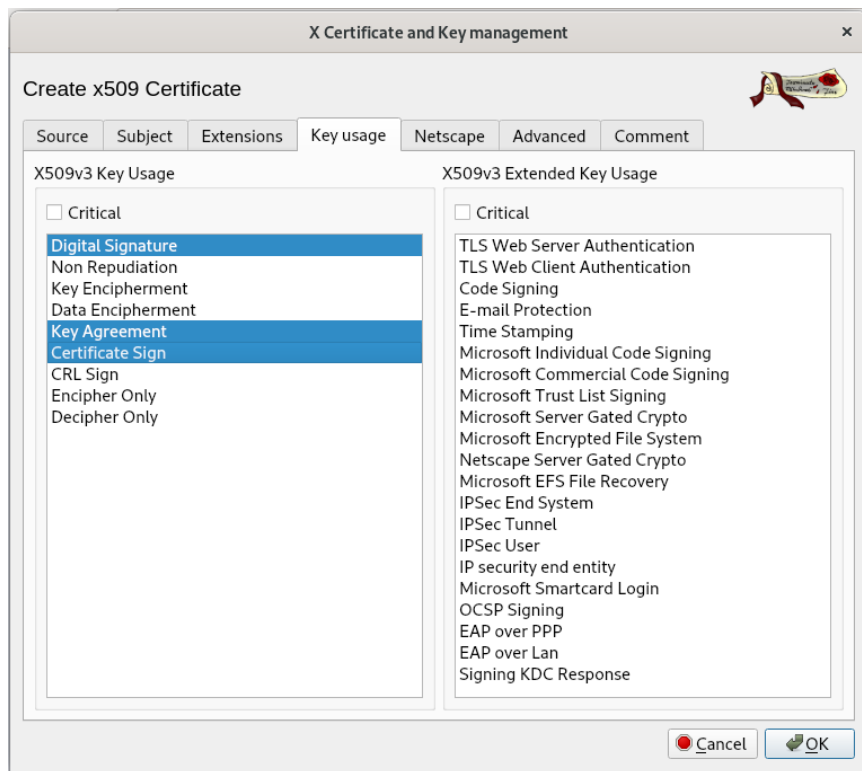
X509v3 Subject Alternative Name

X509v3 Issuer Alternative Name

X509v3 CRL Distribution Points

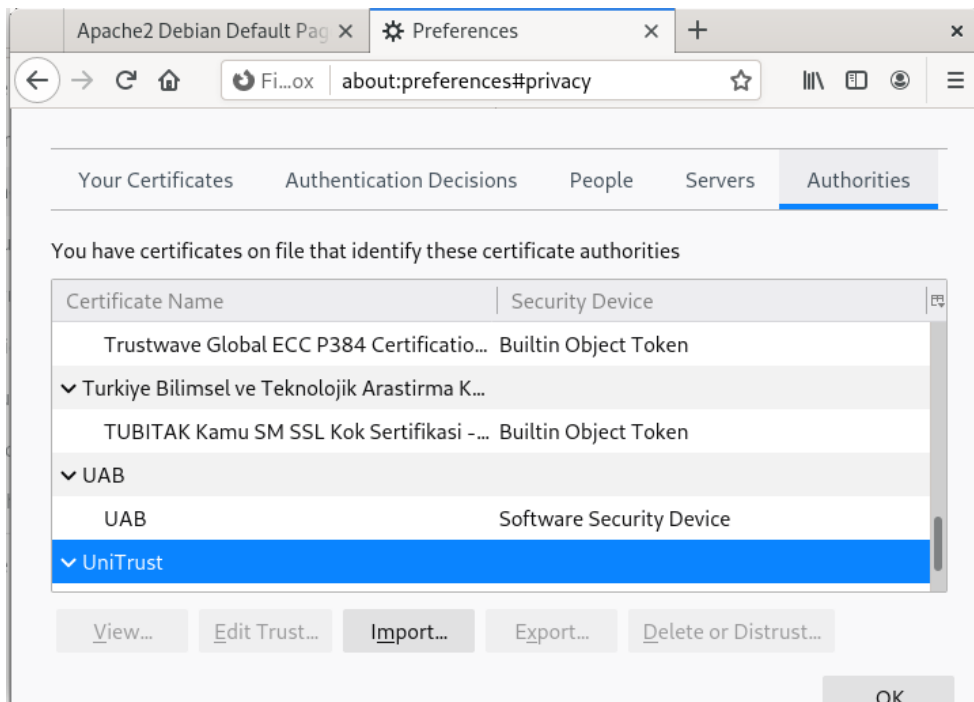
Authority Information Access

☐ OCSP Must Staple

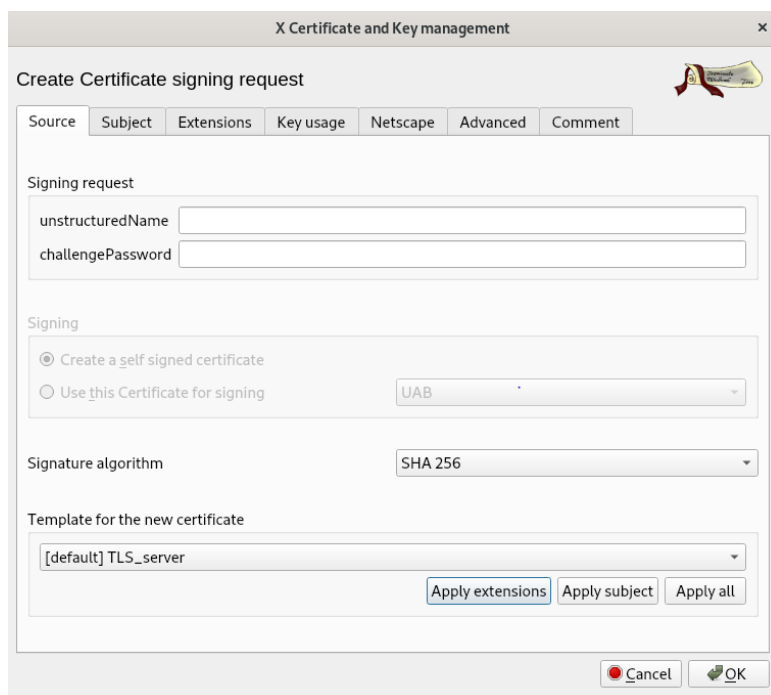


Una vez tengamos el certificado creado, si hemos creado correctamente el certificado, podremos exportar el certificado e importar el certificado en el explorador Web. En este caso lo importamos a Firefox. Tal como vemos en la siguiente captura el certificado se ha importado correctamente.





Ahora procederemos a la creación del certificado SSL y la clave privada para el servidor. Para realizar esto primero de todo deberemos ir a la pestaña Certificate Signing Request y vamos a crear una nueva request, se nos abra una nueva ventana donde configuraremos el certificado para TLS o SSL. Acabamos de configurar la pestaña subject con cuidado de poner el FDQN en el name.



X Certificate and Key management

### Create Certificate signing request

Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

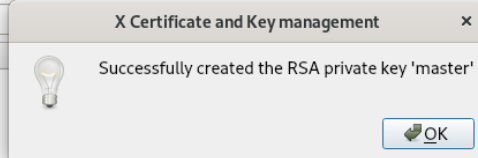
Distinguished name

countryName	ES	organizationalUnitName	UAB
stateOrProvinceName	UAB	commonName	master
localityName	UAB	emailAddress	UAB@uab.cat
organizationName	UAB		

Type

Private key

master (RSA:2048 bit) ☐ Used keys too [Generate a new key](#)



X Certificate and Key management

### Create x509 Certificate

Source Extensions Key usage Netscape Advanced Comment

Signing request

☒ Sign this Certificate signing request master

☒ Copy extensions from the request [Show request](#)

☐ Modify subject of the request

Signing

☐ Create a self signed certificate

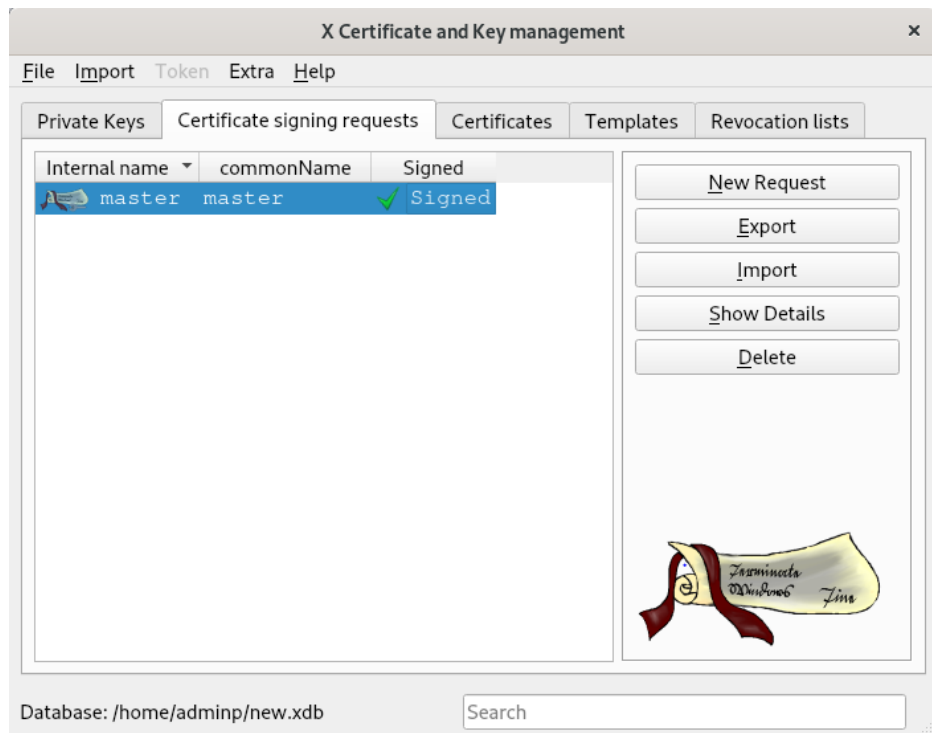
☒ Use this Certificate for signing UAB

Signature algorithm SHA 256

Template for the new certificate

[default] Empty template

[Apply extensions](#) [Apply subject](#) [Apply all](#)



Finalmente podremos exportar el certificado SSL y la clave privada que deberemos indicar la ruta en la configuración del apache. Crearemos un nuevo site copia de 00.ssldefault.conf y modificaremos el SSLCertificateFile donde indicaremos el certificado SSL y SSLCertificateKeyFile donde indicaremos la key privada del servidor. Tendremos que desactivar el site anterior que funcionaba con el certificado autofirmado y activar el site nuevo con el certificado y la clave privada.

```

<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    #
    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    #
    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile /etc/ssl/private/UABSigned.crt
    SSLCertificateKeyFile /etc/ssl/private/master.pem


    #
    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
    # certificate chain for the server certificate. Alternatively
    # the referenced file can be the same as SSLCertificateFile
    # when the CA certificates are directly appended to the server
    # certificate for convinience.
    #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt


    #
    # Certificate Authority (CA):
    # Set the CA certificate verification path where to find CA
    # certificates for client authentication or alternatively one
    # huge file containing all of them (file must be PEM encoded)
    # Note: Inside SSLCACertificatePath you need hash symlinks
    # to point to the certificate files. Use the provided
    # Makefile to update the hash symlinks after changes.
    #SSLCACertificatePath /etc/ssl/private/UABSigned.crt
    #SSLCACertificateFile /etc/ssl/private/master.pem


```


Llegado a este punto no conseguimos que nos funcionara, se nos levantaba el site y nos dejaba acceder por SSL, pero no nos marcaba con un tic verde la página web. Creemos que es debido porque al poner el FQDN, en vez de poner el dominio pusimos "master". Dejamos capturas de la información de la entidad certificadora que nos aparece en la web.

Page Info – https://master/

General

Media

Permissions

Security

### Web Site Identity

Web site: master

Owner: This web site does not supply ownership information.

Verified by: UAB

Expires on: 10 November 2022

View Certificate

### Privacy & History

Have I visited this web site before today? Yes, 24 times

Is this web site storing information on my computer? No

Have I saved any passwords for this web site? No

Clear Cookies and Site Data

View Saved Passwords

### Technical Details

Connection Encrypted (TLS\_AES\_128\_GCM\_SHA256, 128 bit keys, TLS 1.3)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorised people to view information travelling between computers. It is therefore unlikely that anyone read this page as it travelled across the network.

Help

Certificate for UAB

Firefox

about:certificate?cert=MIIDFTCCAmWgAwIBAgIIIDJ7Rb%2BxDnbcwDQYJKoZIhvcNAQELBQAwbzEl

UAB

Subject Name

Country ES

State/Province/County UAB

Locality UAB

Organisation UAB

Organisational Unit UAB

Common Name UAB

Email Address UAB@uab.cat

Issuer Name

Country ES

State/Province/County UAB

Locality UAB

Organisation UAB

Organisational Unit UAB

Common Name UAB

Email Address UAB@uab.cat

Validity

Not Before 01/10/2021, 21:30:00 (Central European Standard Time)

Not After 10/11/2022, 20:30:00 (Central European Standard Time)

Public Key Info

Algorithm RSA

Key Size 2048

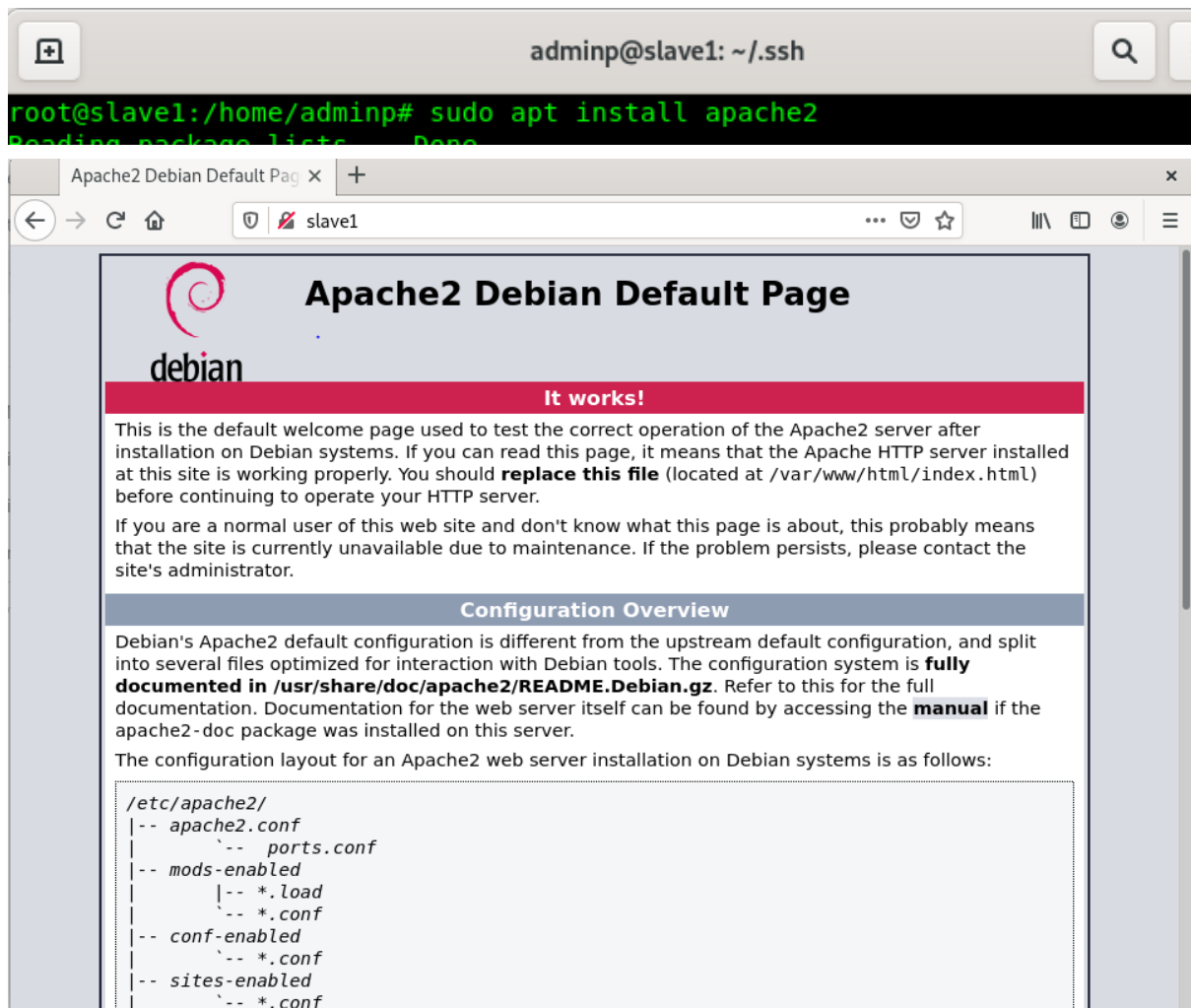
Exponent 65537

Modulus A2:C3:08:60:4B:A3:26:7E:11:6D:F3:E9:64:C4:28:7C:3D:B4:BE:44:01:B0:97:84:66:52:B9:CD:DC:F4:ED:CA:73:...

Miscellaneous

## 5. DNAT (layer 3) i Proxy reverse (layer 7)

Primero de todo deberemos instalar apache2 en la máquina B, una vez instalado modificamos la página web para diferenciarla de A.



Ahora vamos a añadir una regla de PREROUTING para realizar DNAT sobre los paquetes que llegan a la máquina master en el puerto 80. De esta manera en lugar de ser atendidos por A serán atendidos por el servicio de apache de B.

```
root@master:/etc/apache2/sites-available# iptables -t nat -A PREROUTING -p tcp --dport 80 -jDNAT --to 172.16.1.2:80
```

Ahora vamos a crear una proxy\_reverse, para que los paquetes antes de llegar al puerto 80 de la máquina A se vayan contra la máquina B. Primero de todo habilitaos el `mod_proxy_http`.

```

Loaded Modules:
  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
  unixd_module (static)
  access_compat_module (shared)
  alias_module (shared)
  auth_basic_module (shared)
  authn_core_module (shared)
  authn_file_module (shared)
  authz_core_module (shared)
  authz_host_module (shared)
  authz_user_module (shared)
  autoindex_module (shared)
  cgi_module (shared)
  deflate_module (shared)
  dir_module (shared)
  env_module (shared)
  filter_module (shared)
  mime_module (shared)
  mpm_prefork_module (shared)
  negotiation_module (shared)
  php7_module (shared)
  proxy_module (shared)
  proxy_http_module (shared)
  python_module (shared)
  reqtimeout_module (shared)
  setenvif_module (shared)
  socache_shmcb_module (shared)
  ssl_module (shared)
  status_module (shared)
root@master:/etc/apache2/sites-available#

```

Vamos a crear un site para hacer de proxy con la siguiente configuración, es importante poner a false la opción ProxyRequest .

```

adminp@master: ~
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName reverse.gax.org

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

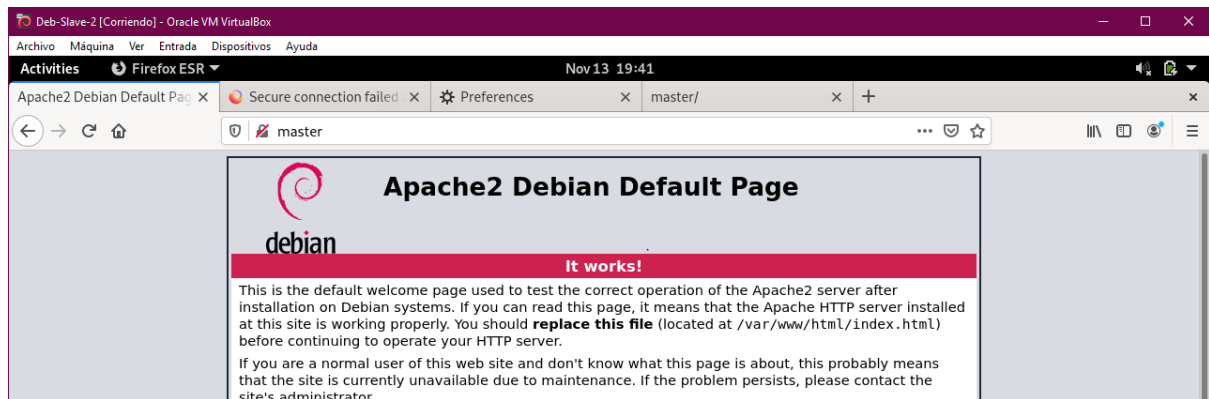
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPass / http://172.16.1.2//
    ProxyPassReverse / http://172.16.1.2//
    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
"reverse.conf" 34L, 1444B

```

Tal como vemos accedemos al master pero se nos muestra la pagina web de la maquina B. Esto es debido a que el proxy reverse. Antes de que las peticiones entren en el puerto 80 se envían a la máquina B.





## 6. Rendiment

Para probar el rendimiento de nuestra web hemos ejecutado el comando `ab -n 10000 -c 100 http://master:8080`, dónde el parámetro `-n` indica el número de peticiones y `-c` el número de conexiones concurrentes. Podemos observar que nuestra página web es robusta ya que ha servido el 100% de las peticiones, con un tiempo de respuesta más que aceptable de 0,32ms. El ratio de transferencia ha sido de 2,3mb/s.

```
Server Software:      Apache/2.4.51
Server Hostname:      localhost
Server Port:          80

Document Path:        /
Document Length:      607 bytes

Concurrency Level:    100
Time taken for tests:  3.261 seconds
Complete requests:    10000
Failed requests:       0
Non-2xx responses:    10000
Total transferred:    7990000 bytes
HTML transferred:     6070000 bytes
Requests per second:  3066.69 [#/sec] (mean)
Time per request:     32.608 [ms] (mean)
Time per request:     0.326 [ms] (mean, across all concurrent requests)
Transfer rate:        2392.85 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      1   2.4      1      33
Processing:      8     29  12.7     25     289
Waiting:        7     28  11.5     24      86
Total:         17     30  13.1     27     290

Percentage of the requests served within a certain time (ms)
 50%    27
 66%    30
 75%    34
 80%    37
 90%    49
 95%    56
 98%    69
 99%    76
100%   290 (longest request)
root@master:/etc/apache2/sites-available#
```