



## Introducción

UDS Enterprise 3.5 permite utilizar diferentes dominios de acceso para acceder a un mismo entorno.

Deberemos tener disponibles previamente los distintos certificados de los dominios que vamos a utilizar. Estos certificados tienen que estar en formato **PEM**. También deberemos tener el fichero del certificado del servidor (**.crt**, **.pem**, etc...) y del fichero de claves del servidor (**.key**, **.pem**, etc...).

En este documento se muestran las tareas que hay que realizar en los servidores UDS para habilitar todos los dominios de acceso que necesitemos.

## Configuración de los servidores UDS

A continuación se muestra un ejemplo de configuración con dos nombres de dominio, cada uno con su correspondiente certificado.

Todas las tareas descritas las realizaremos en la máquina **UDS-Server**. En caso de tener un entorno en alta disponibilidad con varios servidores UDS, hay que realizar estas tareas en todos los servidores.

Accedemos a la ruta **/etc/nginx/sites-available/**

```
root@uds:/etc/nginx/sites-available# ls -la
total 16
drwxr-xr-x 2 root root 4096 May 20 13:37 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
-rw-r--r-- 1 root root 2412 Aug 24 2020 default
-rw-r--r-- 1 root root 1954 May 20 13:37 uds
root@uds:/etc/nginx/sites-available#
```

Editamos el fichero: **uds**

Dentro de este fichero, en la línea 30 aproximadamente, indicaremos el primer nombre de dominio de acceso en: **server\_name** (en este ejemplo: **first.udsenderprise.com**):

```
# Add index.php to the list if you are using PHP
index index.html;

server_name first.udsenderprise.com;

#
# Activate GZIP
# In our app, saves around 80% or the traffic
#
```



Comparación con el fichero original:

```
uds-orig | uds
upstream uds_server {
  server unix:/run/udsweb/socket fail_timeout=10
}
map $http_x_forwarded_proto $thescheme {
  default $scheme;
  https https;
}
log_format combined_no_query '$remote_addr - $remote_user [$time_local] "$uri" $status $body_bytes_sent "$http_user_agent"';
server {
  listen 80 default_server;
  listen [::]:80 default_server;
  access_log /var/log/nginx/access.log combined;
  # SSL configuration
  #
  listen 443 ssl http2 default_server;
  listen [::]:443 ssl http2 default_server;
  include snippets/uds-ssl-params.conf;
  root /var/server/static/;
  # Add index.php to the list if you are using PHP
  index index.html;
  server_name ;
  #
  # Activate GZIP
  # In our app, saves around 80% of the traffic
  #
  gzip on;
  gzip_proxied any;
  # text/html is always included
  gzip_types
    text/css
}
upstream uds_server {
  server unix:/run/udsweb/socket fail_timeout=10
}
map $http_x_forwarded_proto $thescheme {
  default $scheme;
  https https;
}
log_format combined_no_query '$remote_addr - $remote_user [$time_local] "$uri" $status $body_bytes_sent "$http_user_agent"';
server {
  listen 80 default_server;
  listen [::]:80 default_server;
  access_log /var/log/nginx/access.log combined;
  # SSL configuration
  #
  listen 443 ssl http2 default_server;
  listen [::]:443 ssl http2 default_server;
  include snippets/uds-ssl-params.conf;
  root /var/server/static/;
  # Add index.php to the list if you are using PHP
  index index.html;
  server_name first.udsenderprise.com;
  #
  # Activate GZIP
  # In our app, saves around 80% of the traffic
  #
  gzip on;
  gzip_proxied any;
  # text/html is always included
  gzip_types
    text/css
}
```

Ahora realizaremos una copia de este fichero (**uds**) y lo nombraremos como “**uds2**”. Este nuevo fichero nos servirá para definir el segundo acceso del nuevo nombre o dominio.

Una vez copiado el fichero, tendremos:

```
root@uds:/etc/nginx/sites-available# ls -la
total 20
drwxr-xr-x 2 root root 4096 May 28 13:47 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
-rw-r--r-- 1 root root 2412 Aug 24 2020 default
-rw-r--r-- 1 root root 1954 May 20 13:37 uds
-rw-r--r-- 1 root root 1954 May 28 13:47 uds2
root@uds:/etc/nginx/sites-available#
```



# UDS Enterprise 3.5

## Configurar multidominios de acceso

www.udsenderprise.com

Editamos el fichero copiado "uds2" y eliminamos la parte superior del código hasta el inicio de **server{** y así dejar el fichero como se muestra en las siguientes capturas:

```
GNU nano 3.2 uds2
server {
    access_log /var/log/nginx/access.log combined_no_query;

    # SSL configuration
    #
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    include snippets/uds-ssl-params2.conf;

    root /var/server/static;

    # Add index.php to the list if you are using PHP
    index index.html;

    server_name second.udsenderprise.com;

    #
    # Activate GZIP
    # In our app, saves around 80% or the traffic
    #
    gzip on;
    gzip_proxied any;
    # text/html is always included
    gzip_types
        text/css
        text/javascript
        text/xml
        text/plain
        application/javascript
        application/x-javascript
        application/json;

    location /favicon.ico {
        alias /var/server/static/modern/img/favicon.ico;
    }

    location /uds/res/ {
        autoindex off;
        alias /var/server/static;
    }

    location / {
        # First attempt to server /maintenance (to allow easy backend maintenance) if exists
        # if not, fallback to UDS
        try_files /maintenance.html @proxy_to_uds;
    }
}
```

A continuación, se realiza una comparación del fichero original (**uds-orig**) con el nuevo fichero (**uds2**):

```
uds-orig
upstream uds_server {
    server unix:/run/udsweb/socket fail_timeout=10s;
}

map $http_x_forwarded_proto $thescheme {
    default $scheme;
    https https;
}

log_format combined_no_query '$remote_addr - $remote_user [$time_local] "$uri" $status $body_bytes_sent "$http_referer" "$http_user_agent" "$http_x_forwarded_for"';

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    access_log /var/log/nginx/access.log combined_no_query;

    # SSL configuration
    #
    listen 443 ssl http2 default_server;
    listen [::]:443 ssl http2 default_server;
    include snippets/uds-ssl-params.conf;

    root /var/server/static;

    # Add index.php to the list if you are using PHP
    index index.html;

    server_name ;
}

uds2
server {
    access_log /var/log/nginx/access.log combine

    # SSL configuration
    #
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    include snippets/uds-ssl-params2.conf;

    root /var/server/static;

    # Add index.php to the list if you are using
    index index.html;

    server_name second.udsenderprise.com;

    #
    # Activate GZIP
    # In our app, saves around 80% or the traffi
    #
    gzip on;
    gzip_proxied any;
    # text/html is always included
    gzip_types
        Text/css
        text/javascript
        text/xml
        text/plain
        application/javascript
        application/x-javascript
        application/json;

    location /favicon.ico {

```



Además de eliminar el código indicado en verde en la imagen de comparación, también es necesario realizar algunos cambios:

- Eliminamos “**default\_server**” de los “**listen**”.
- En “**include snippets**”, indicamos un nuevo fichero (en este ejemplo: **uds-ssl-params2.conf**) que crearemos en los siguientes pasos.
- En “**server\_name**” indicaremos el segundo nombre de dominio de acceso (en este ejemplo: **second.udsenderprise.com**).

La siguiente tarea que realizaremos será la instalación y configuración de los diferentes certificados a utilizar para los distintos dominios de acceso. Para ello, nos dirigimos a la ruta **/etc/certs/**

```
root@uds:/etc/certs# ls
dhparam.pem key.pem server.pem
root@uds:/etc/certs#
```

Aquí añadiremos los diferentes certificados a utilizar. Será necesario añadir el fichero del certificado del servidor y el fichero de claves para los diferentes dominios (todos en formato **PEM**).

En este ejemplo, añadiremos los certificados para los dos accesos que estamos configurando, quedando de la siguiente manera:

```
root@uds:/etc/certs# ls
dhparam.pem key-first.pem key-second.pem server-first.pem server-second.pem
root@uds:/etc/certs#
```

Ahora crearemos un link simbólico para el fichero **uds2** previamente creado. Para ello, nos situaremos la ruta **/etc/nginx/sites-enabled** y ejecutaremos el comando:

```
ln -s /etc/nginx/sites-available/uds2
```

```
root@uds:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/uds2
root@uds:/etc/nginx/sites-enabled#
root@uds:/etc/nginx/sites-enabled# ls -la
total 8
drwxr-xr-x 2 root root 4096 May 28 16:46 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
lrwxrwxrwx 1 root root 30 May 20 13:37 uds -> /etc/nginx/sites-available/uds
lrwxrwxrwx 1 root root 31 May 28 16:46 uds2 -> /etc/nginx/sites-available/uds2
root@uds:/etc/nginx/sites-enabled#
```

Por último, accedemos a la ruta **/etc/nginx/snippets** y duplicamos el fichero “**uds-ssl-params.conf**”. Ponemos al nuevo fichero el nombre “**uds-ssl-params2.conf**”, de manera que coincida con el nombre indicado en el fichero “**uds2**” (apartado “**include snippets**”), anteriormente creado y modificado.

```
root@uds:/etc/nginx/snippets# ls -la
total 24
drwxr-xr-x 2 root root 4096 May 28 17:13 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
-rw-r--r-- 1 root root 423 Aug 24 2020 fastcgi-php.conf
-rw-r--r-- 1 root root 217 Aug 24 2020 snakeoil.conf
-rw-r--r-- 1 root root 891 May 28 17:13 uds-ssl-params2.conf
-rw-r--r-- 1 root root 891 May 20 13:37 uds-ssl-params.conf
root@uds:/etc/nginx/snippets#
```



Empezamos editando el fichero “**uds-ssl-params.conf**”. Indicaremos el nuevo nombre de los ficheros de certificado de servidor y de claves:

```
GNU nano 3.2 uds-ssl-params.conf
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be regenerated using: ope
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECD
ssl_ecdh_curve prime256v1:secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
# By default, stapling if off
# ssl_stapling on;
# ssl_stapling_verify on;
ssl_certificate /etc/certs/server-first.pem;
ssl_certificate_key /etc/certs/key-first.pem;
#resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=63072000; includeSubDom
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

Ahora editamos el fichero recién creado “**uds-ssl-params2.conf**” e indicamos la ruta y nombre de los ficheros del segundo certificado:

```
GNU nano 3.2 uds-ssl-params2.conf
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be regenerated using: open
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDH
ssl_ecdh_curve prime256v1:secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
# By default, stapling if off
# ssl_stapling on;
# ssl_stapling_verify on;
ssl_certificate /etc/certs/server-second.pem;
ssl_certificate_key /etc/certs/key-second.pem;
#resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=63072000; includeSubDoma
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```



A continuación se muestran las diferencias finales entre los dos ficheros “uds-ssl-params”....

```
uds-ssl-params.conf | uds-ssl-params2.conf
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be regenerated
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512
ssl_ecdh_curve prime256v1:secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
# By default, stapling is off
# ssl_stapling on;
# ssl_stapling_verify on;
ssl_certificate /etc/certs/server-first.pem;
ssl_certificate_key /etc/certs/key-first.pem;
#resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=6307200";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";

ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be regenerated
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512
ssl_ecdh_curve prime256v1:secp384r1;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
# By default, stapling is off
# ssl_stapling on;
# ssl_stapling_verify on;
ssl_certificate /etc/certs/server-second.pem;
ssl_certificate_key /etc/certs/key-second.pem;
#resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=6307200";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

Para aplicar todos estos cambios, reiniciamos el servidor y confirmamos que el servicio “nginx” está correctamente iniciado:

```
root@uds:/etc/nginx/sites-available# service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-05-28 17:51:56 CEST; 2min 28s ago
     Docs: man:nginx(8)
  Process: 758 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 759 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 760 (nginx)
    Tasks: 3 (limit: 2327)
   Memory: 4.9M
    CGroup: /system.slice/nginx.service
            └─760 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─761 nginx: worker process
                 └─762 nginx: worker process

May 28 17:51:56 uds systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.
May 28 17:51:56 uds systemd[1]: Started A high performance web server and a reverse proxy server: nginx.
lines 1-16/16 (END)
```

Ahora, cuando accedamos a través de ambas URL (<https://first.udsenderprise.com> o <https://second.udsenderprise.com>), comprobaremos que el portal de login es el mismo y que el certificado mostrado es el correcto para cada acceso.



## Sobre Virtual Cable

Virtual Cable desarrolla y comercializa UDS Enterprise mediante un modelo de suscripción según el número de usuarios, incluyendo soporte y actualizaciones.

Además, Virtual Cable ofrece servicios profesionales para instalar y configurar UDS Enterprise.

Para más información, visite [www.udsenderprise.com](http://www.udsenderprise.com) o envíenos un email a [info@udsenderprise](mailto:info@udsenderprise).