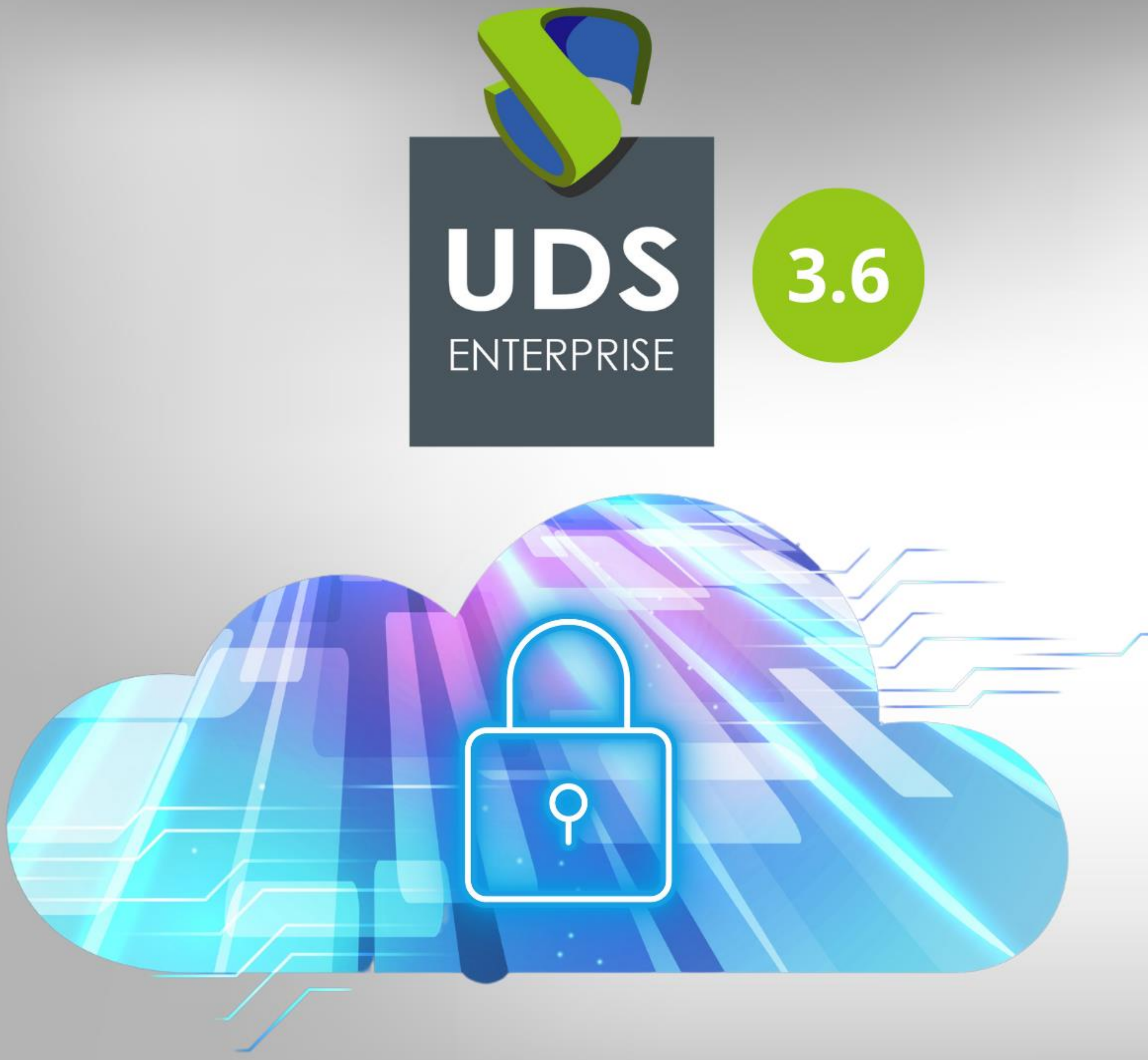
VIRTUAL
CABLE

Configuring Multi-domain Access in UDS Enterprise



Index

Introduction

UDS Enterprise 3.6 allows the use of different access domains to enter the same environment.

You must have available the different certificates of the domains that you are going to use. These certificates have to be in **PEM** format. Also you will need to have the server certificate file **(.crt, .pem**, etc ...) and the server key file (**.key, .pem**, etc ...).

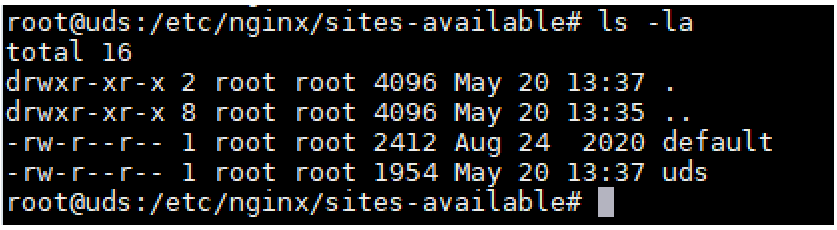
This document shows the tasks to be carried out on the UDS servers to enable all the access domains that are needed.

UDS servers configuration

Below is an example of a configuration with two domain names, each with its corresponding certificate.

Please carry out all the tasks described on the UDS-Server machine. In case of having a high availability environment with several UDS servers, these tasks must be carried out on all servers.

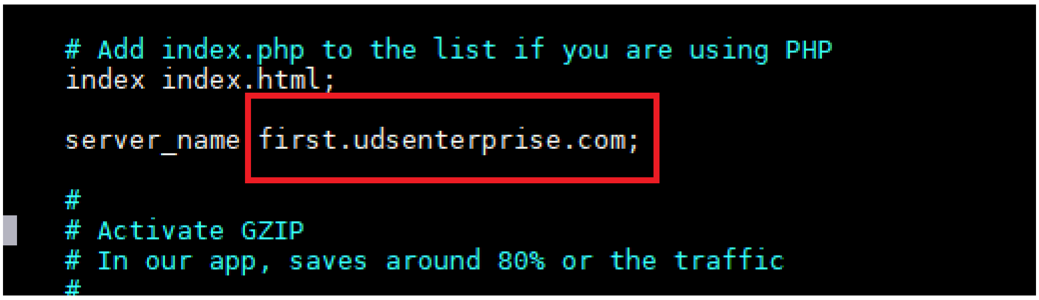
Access the path **/etc/nginx/sites-available/**



```
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#
```

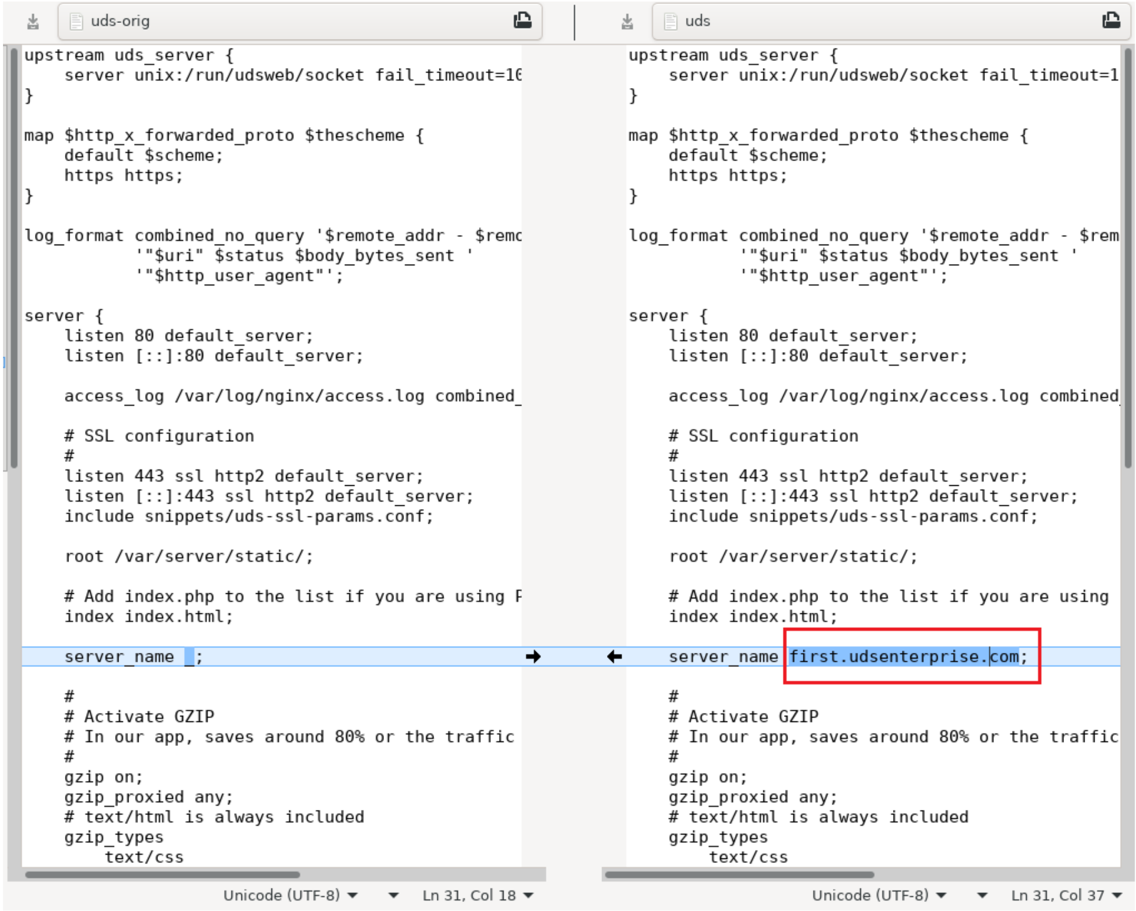
Edit the file: **uds**

Within this file, on line 30 approximately, you need to indicate the first access domain name in: **server_name** (in this example: **first.udsenterprise.com**):



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

    server_name first.udsenterprise.com;

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

Comparison with the original file:

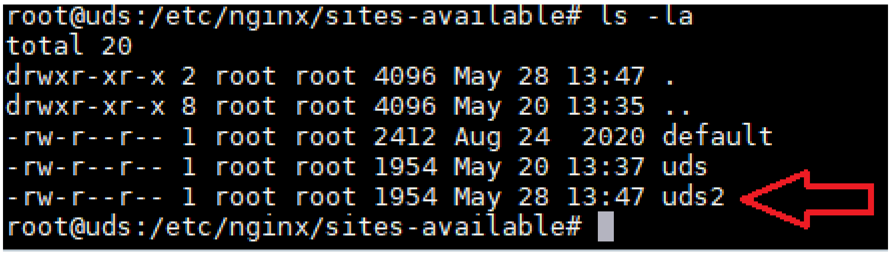


```
upstream uds_server {
    server unix:/run/udsweb/socket fail_timeout=16
}

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

log_format combined_no_query '$remote_addr - $remc
           '"$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 P
    index index.html;

    server_name  ;

    #
    # 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
```

```
upstream uds_server {
    server unix:/run/udsweb/socket fail_timeout=1
}

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

log_format combined_no_query '$remote_addr - $rem
           '"$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
    index index.html;

    server_name first.udsenterprise.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
```

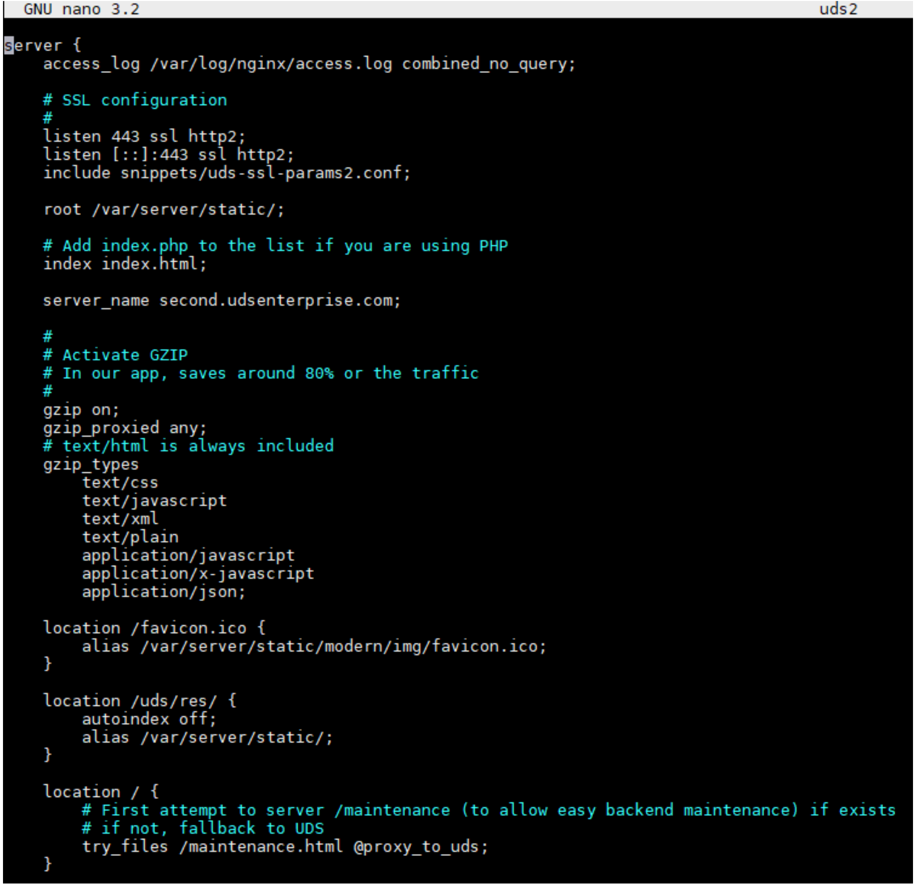
Now make a copy of this file (**uds**) and name it as “**uds2**”. This new file will help you to define the second access of the new name or domain.

Once the file is copied, you will have:



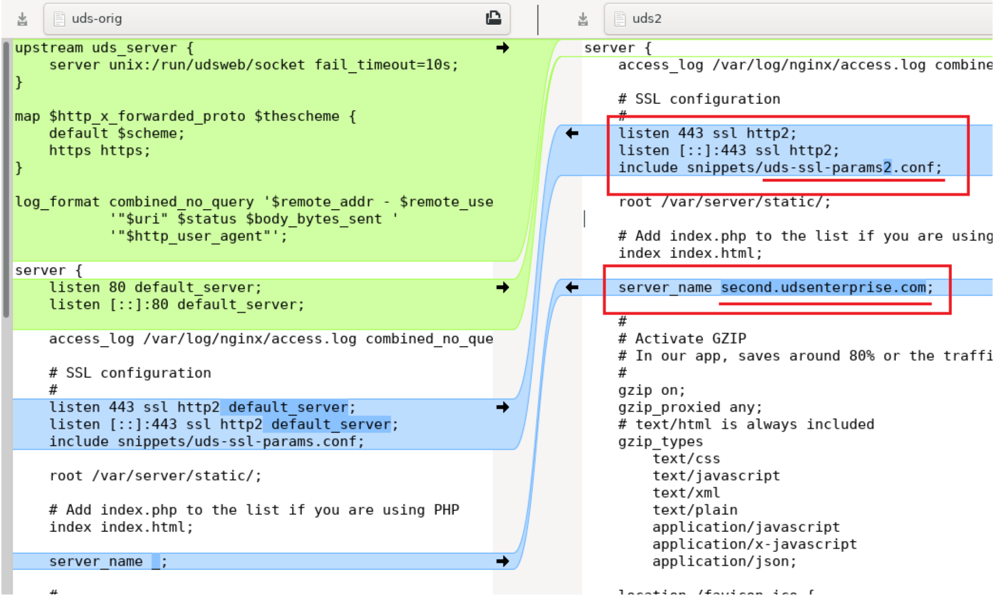
```
root@uds:/etc/nginx/sites-available# ls -la
total 20
drwxr-xr-x 2 root root 4096 May 28 13:47 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
-rw-r--r-- 1 root root 2412 Aug 24  2020 default
-rw-r--r-- 1 root root 1954 May 20 13:37 uds
-rw-r--r-- 1 root root 1954 May 28 13:47 uds2
root@uds:/etc/nginx/sites-available#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

Edit the copied file “uds2” and delete the upper part of the code up to the start of **server {** and so leave the file as shown in the following captures:



```
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.udsenterprise.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;
    }
```

Next, a comparison is made of the original file (**uds-orig**) with the new file (**uds2**):



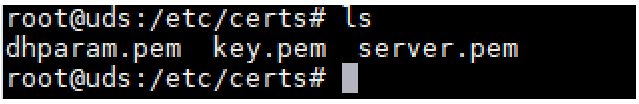
```
upstream uds_server {
    server unix:/run/udsweb/socket fail_timeout=10s;
}

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

log_format combined_no_query '$remote_addr - $remote_use
           '"$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_no_que

    # 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  ;

    #
```

```
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.udsenterprise.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 {
```

In addition to removing the code indicated in green in the comparison image, it is also necessary to make some changes:

- Remove **"default_server"** from the **"listen"**.
- In **"include snippets"**, create a new file (in this example: **uds-ssl-params2.conf**) it will be created in the following steps.
- In **"server_name"** indicate the second access domain name (in this example: **second.udsenterprise.com**).

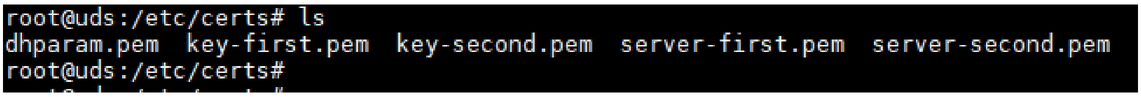
The next task that you will carry out will be the installation and configuration of the different certificates to be used for the different access domains. To do this, go to the path **/etc/certs**/



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

Here add the different certificates to use. It will be necessary to add the server certificate file and the key file for the different domains (all in **PEM** format).

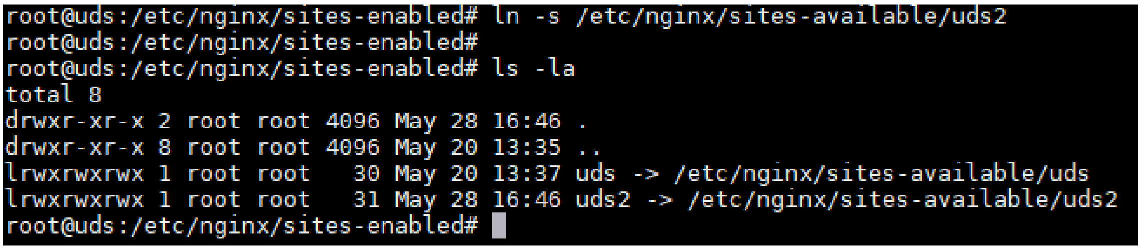
In this example the two certificates that are being configured will be added, being as follows,



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

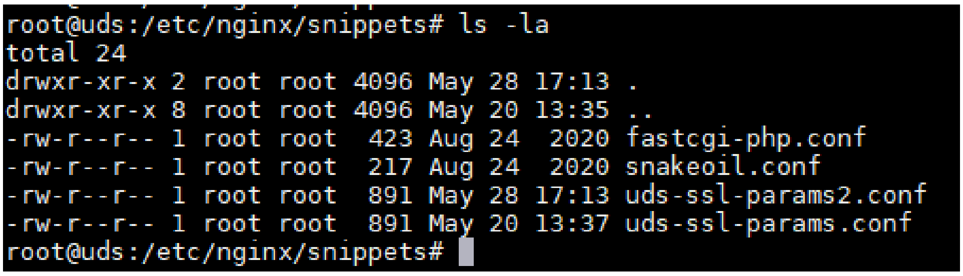
Now you can create a symbolic link for the previously created uds2 file. To do this, locate the path **/etc/nginx/sites-enabled** and execute the command:

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



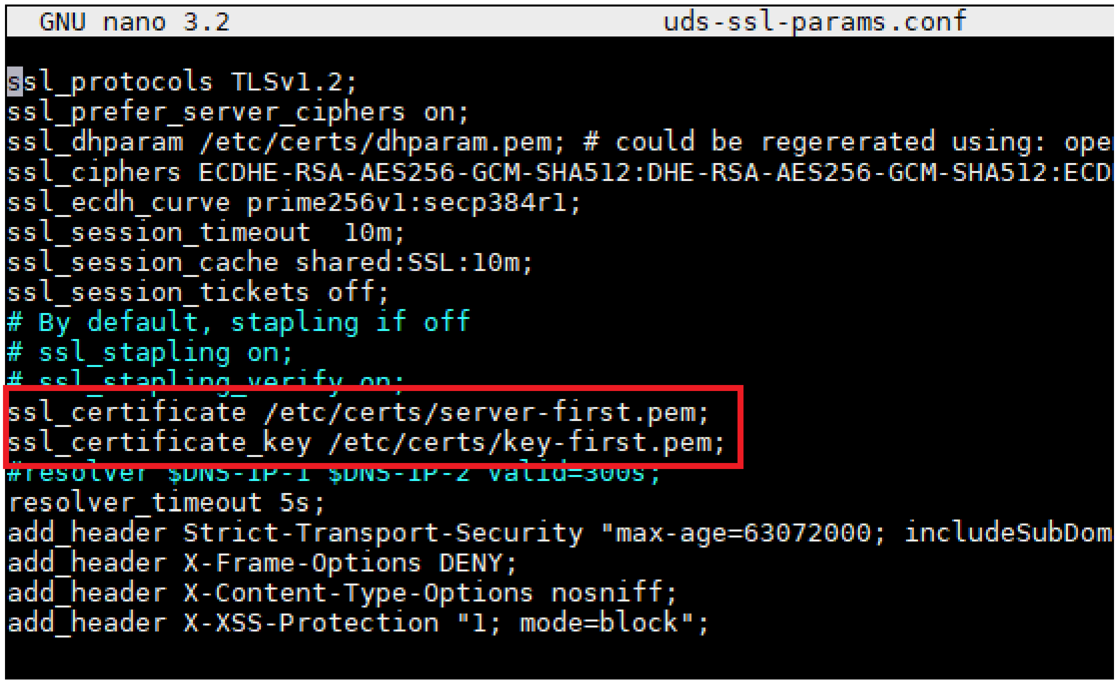
```
root@uds:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/uds2
root@uds:/etc/nginx/sites-enabled#
root@uds:/etc/nginx/sites-enabled# ls -la
total 8
drwxr-xr-x 2 root root 4096 May 28 16:46 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
lrwxrwxrwx 1 root root   30 May 20 13:37 uds -> /etc/nginx/sites-available/uds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

Finally, access the path **/etc/nginx/snippets** and duplicate the file **“uds-ssl-params.conf”**. Name the new file **"uds-ssl-params2.conf"**, so that it matches the name indicated in the file **"uds2"** (section **"include snippets"**), previously created and modified.



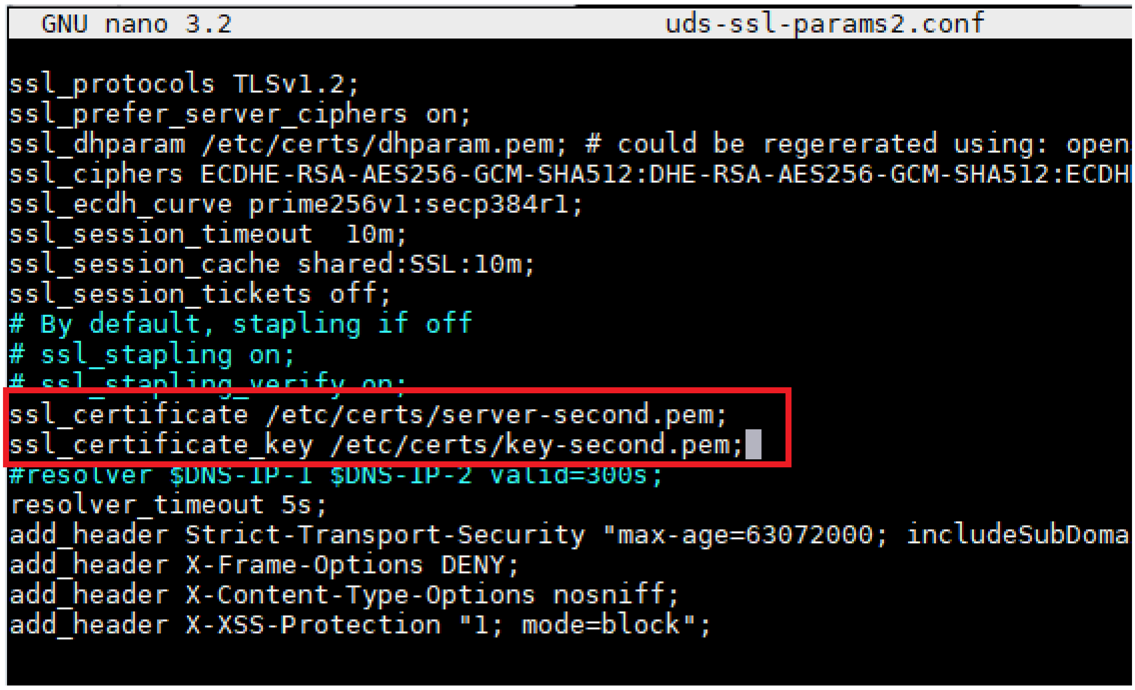
```
root@uds:/etc/nginx/snippets# ls -la
total 24
drwxr-xr-x 2 root root 4096 May 28 17:13 .
drwxr-xr-x 8 root root 4096 May 20 13:35 ..
-rw-r--r-- 1 root root  423 Aug 24  2020 fastcgi-php.conf
-rw-r--r-- 1 root root  217 Aug 24  2020 snakeoil.conf
-rw-r--r-- 1 root root  891 May 28 17:13 uds-ssl-params2.conf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

Start by editing the file **"uds-ssl-params.conf"**. Select the new name of the server certificate and key files:



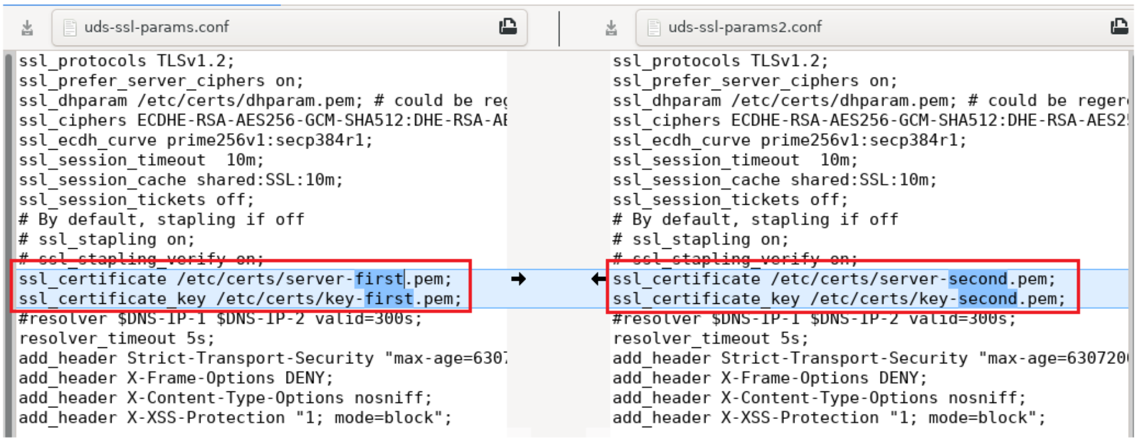
```
  GNU nano 3.2                                      uds-ssl-params.conf

ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be regererated 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";
```

Now edit the newly created file **"uds-ssl-params2.conf"** and indicate the path and name of the files of the second certificate:



```
  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 regererated 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";
```

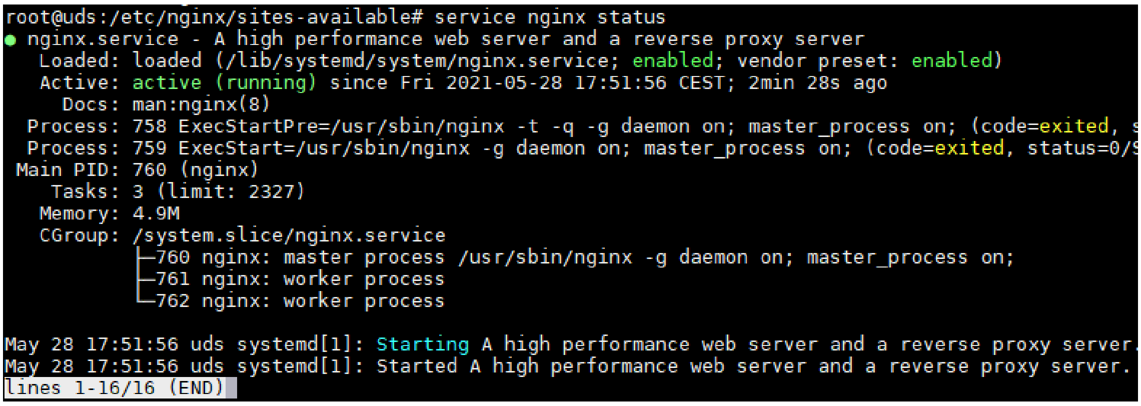
The final differences between the two files **“uds-ssl-params”** are shown below….



```
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/certs/dhparam.pem; # could be reg
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AE
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=6307
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 reger
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES2
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=630720
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

To apply all these changes, restart the server and confirm that the **"nginx"** service is correctly started:



```
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, s
  Process: 759 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/S
 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.
May 28 17:51:56 uds systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-16/16 (END)
```

Now, you can access through both URLs (https://first.udsenterprise.com or https://second.udsenterprise.com), check that the login portal is the same and that the certificate shown is the correct one for each access.

About Virtual Cable

Virtual Cable is a company specialized in the **digital transformation** of the **workplace**. The company develops, supports and markets UDS Enterprise. Its team of experts has designed **VDI** solutions tailored to **each sector** to provide a unique user experience fully adapted to the needs of each user profile. Virtual Cable professionals have **more than 30 years of experience** in IT and software development and more than 15 in virtualization technologies**. Millions of Windows and Linux virtual desktops with UDS Enterprise are deployed all over the world every day**.