SSL Installation Guide - Ubuntu/Redhat

Quest Data Quality V3.1.3

Introduction

This comprehensive guide delivers a detailed, step-by-step walkthrough for installing SSL certificates on a Linux Ubuntu/Red Hat system. It covers everything from preparing your environment and obtaining your SSL certificate to configuring your server and verifying the installation, ensuring a secure and encrypted connection for your applications.

Import SSL Certificate

Step 1: Collect SSL certificate - Obtain a valid SSL certificate from your domain administrator.

Step 2: Copy the SSL certificate to the server -

1. Access your Ubuntu/Red Hat machine via Putty or by any means
2. Run the following commands on your terminal:

Ubuntu

```
None
#Create a folder ssl
sudo mkdir /etc/apache2/ssl

#Navigate to the directory
cd /etc/apache2/ssl
```

Redhat

```
None
#Navigate to the directory
cd /etc/ssl
```

Step 3: Copy SSL certificate to server - Copy and paste the following files into the ssl directory:

1. SSL crt
2. SSL Gd bundle
3. Machine private key

```
[brain@ip-172-31-66-231 ssl]$ ls
cert.pem  certs  ct_log_list.cnf  dqlabsai-net-private.key  openssl.cnf  STAR_dqlabsai_net.ca-bundle  STAR_dqlabsai_net.crt
[brain@ip-172-31-66-231 ssl]$
```

SSL Configuration Setup

Step 1: Open the `dqlabs.conf` file using the below command:

Ubuntu

```
None
sudo vi /etc/apache2/sites-enabled/dqlabs.conf
```

Redhat

```
None
sudo vi /etc/httpd/conf.d/dqlabs.conf
```

Step 2: Go to Insert mode by clicking “I”. Inside the file, locate the virtual host configuration and then change the port from `80` to `443` and add the DNS name in the server name section

```
<VirtualHost *:443>
        # 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 ubuntuenv.dqlabsai.net

        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
        <Directory /var/www/html>
 RewriteEngine On
  RewriteBase /
  RewriteRule ^index\.html$ - [L]
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-l
  RewriteRule . /index.html [L]
</Directory>
```

Step 3: Replace the placeholders in the lines below, copy and paste them before the line `</VirtualHost>` (Refer to the image given below) to specify the SSL key path in the correct certificate section of the configuration.

```
None
#Replace the placeholders with your file names
SSLEngine on
ProxyRequests Off
SSLProxyEngine on
SSLCertificateFile /etc/apache2/ssl/<cert>.crt
SSLCertificateKeyFile /etc/apache2/ssl/<private>.key
SSLCACertificateFile /etc/apache2/ssl/<gd>.crt
```

```
</Directory>

         ProxyPass        /socket.io/ http://localhost:8000/socket.io/
            ProxyPassReverse /socket.io/ http://localhost:8000/socket.io/


        ErrorLog /var/log/httpd/error.log
        CustomLog /var/log/httpd//access.log combined

               ProxyPass /api http://localhost:8000/api
            ProxyPassReverse /api http://localhost:8000/api
            ProxyPass /admin http://localhost:8000/admin
            ProxyPassReverse /admin http://localhost:8000/admin
            ProxyPass /help http://localhost:8000/help
            ProxyPassReverse /help http://localhost:8000/help
            ProxyPass /static/drf-yasg http://localhost:8000/static/drf-yasg
            ProxyPassReverse /static/drf-yasg http://localhost:8000/static/drf-yasg
             ProxyPass /logs http://localhost:8000/logs connectiontimeout=900 timeout=900
            ProxyPassReverse /logs http://localhost:8000/logs
             ProxyPass /logs http://localhost:8000/logs connectiontimeout=900 timeout=900
            ProxyPassReverse /logs http://localhost:8000/logs
            ProxyPass /media http://localhost:8000/media
            ProxyPassReverse /media http://localhost:8000/media

SSLEngine on
ProxyRequests Off
SSLProxyEngine on
SSLCertificateFile  /etc/ssl/STAR_dqlabsai_net.crt
SSLCertificateKeyFile /etc/ssl/dqlabsai-net-private.key
SSLCACertificateFile  /etc/ssl/STAR_dqlabsai_net.ca-bundle

</VirtualHost>
```

Step 4: Go to command mode by pressing the escape key. Save and exit the editor using `:wq!` and press Enter

Step 5: Navigate to the httpd.conf file (Only for RedHat, Skip this step for Ubuntu)

Redhat

```
None
sudo vi /etc/httpd/conf/httpd.conf
```

Click “`I`” to enter insert mode and modify the listener port from 80 to 443. Click Esc key and then `:wq`! To save and exit

```
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts.  See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 443

#
# Dynamic Shared Object (DSO) Support
```

Step 6: Restart the server using the command:

Ubuntu

```
None
sudo service apache2 restart
```

Redhat

```
None
sudo service httpd restart
```

Configuration Setup

Note: This step can be skipped if the application is installed using https + DNS.

Step 1: Download the DNS_upgrade_script into the DQLabs Server under the same user with which DQLabs was installed

```
None
sudo wget https://s3.us-east-1.amazonaws.com/erwin-2.0/code/linux/Erwin-Dns-upgrade-script.sh
```

Step 2: Open the script file

```
None
sudo vi Erwin-Dns-upgrade-script.sh
```

Step 3: Update the variables

```
None
#Provide the existing IP address used to access the DQLabs application
OLD_URL="https://18.204.207.114"

#Provide the DNS name that needs to be updated
NEW_URL="https://appenv.domain.net"

#Provide the current port used to run the application
PORT=80

#Provide the destination directory of the application
APP_DIR="/home/user/App"
```

```
#!/bin/bash
set -e  # Exit on any error

# ========================================
# Variables
# ========================================
JS_DIR="/var/www/html/js"
OLD_URL="http://145.132.105.55"
NEW_URL="https://ubuntuenv.dqlabsai.net"
PORT=80
APP_DIR="/home/dqlabs/app"
SERVER_ENV_FILE_PATH="$APP_DIR/DQLabs-Server/src"
AIRFLOW_ENV_FILE_PATH="$APP_DIR/DQLabs-Airflow/infra/airflow/dags/dqlabs"
```

Step 4: Save the file and exit

```
None
:wq!
```

Step 5: Grant permission to execute the script

```
None
sudo chmod 777 Erwin-Dns-upgrade-script.sh
```

Step 6: Execute the script

```
None
./Erwin-Dns-upgrade-script.sh
```

Once the script is executed successfully, access the DNS in your browser to confirm if the SSL configuration is working correctly.

Following these steps will ensure that SSL support is enabled for the DQLabs application.