Openssl

Générer un certificat auto-signé (avec SAN)

Générer une autorité de certification

```
openssl genrsa -out nehemiebarkia.fr.key 4096
```

Générer une demande de certification

```
openssl req -new -key nehemiebarkia.fr.key -out nehemiebarkia.fr.csr
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Occitanie
Locality Name (eg, city) []:Toulouse
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MONENTREPRISE
Organizational Unit Name (eg, section) []:MONSERVICE
Common Name (e.g. server FQDN or YOUR name) []:nehemiebarkia.fr
Email Address []:contact@nehemiebarkia.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Générer le fichier SAN

```
touch v3.ext
```

```
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints       = CA:TRUE
keyUsage               = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement,
keyCertSign
subjectAltName         = DNS:nehemiebarkia.fr, DNS:*.nehemiebarkia.fr, IP:40.40.40.40
issuerAltName          = issuer:copy
```

Générer le certificat avec une validité de 10 000 jours.

```
openssl x509 -req -days 10000 -in nehemiebarkia.fr.csr -signkey nehemiebarkia.fr.key -out nehemiebarkia.fr.crt -
sha256 -extfile v3.ext
```

Afficher le certificat

```
openssl x509 -text -in nehemiebarkia.fr.crt -noout
```

Optionnel :

Convertir La key en p8 :

```
openssl pkcs8 -in nehemiebarkia.fr.key -topk8 -out nehemiebarkia.fr.key.p8 -nocrypt
```

Générer un certificat proprement

Générer l'autorité de certification

```
openssl genrsa -out CA.key 4096
openssl req -x509 -new -nodes -key CA.key -sha256 -days 3650 -out CA.crt
```

Générer le certificat

```
# Générer une clée
openssl genrsa -out CERT.key 4096
# Générer la demande
openssl req -newkey rsa:4096 -key CERT.key -out CERT.csr -config config.ext
# Voir la demande
openssl req -in CERT.csr -noout -text
# Approuver la demande
openssl x509 -req -days 10000 -in CERT.csr -CA CA.crt -CAkey CA.key -out CERT.crt -sha256 -extfile config.ext
# Transformer
openssl pkcs8 -in CERT.key -topk8 -out CERT.key.p8 -nocrypt
# Afficher le certif
openssl x509 -text -in CERT.crt -noout
```

config.ext

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req_ext
prompt = no

[req_distinguished_name]
C = FR
ST = Occitanie
L = Toulouse
O = X
OU = X
CN = monsite.fr

[v3_req_ext]
subjectAltName = @alt_names

[alt_names]
IP.1 = 1.1.1.1
DNS.1 = *.monsite.fr
```

Etablir une connexion ssl pour tester les certificats :

```
openssl s_client -connect host:port -CAfile chemin_vers_certificat_public_de_la_ca.pem
```

Récupérer le certificat publique automatiquement (en pem) :

```
echo | openssl s_client -showcerts -connect google.com:443 2>/dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > chaine_complete.crt
```

Déplacer le certificat dans les dossier d'autorités de certificiations locales :

```
sudo mv chaine_complete.crt /usr/local/share/ca-certificates/mon_super_cert.crt
```

Actualiser la liste des certifs :

```
sudo update-ca-certificates
```

Revision #18
Created 17 January 2023 14:35:21 by Nehemie
Updated 7 April 2026 11:55:12 by Nehemie