인증서 및 키 기술 노트

- 자체 서명된 루트 CA 생성, on page 1
- 자체 서명 루트 CA에서 서명한 인증서 생성, on page 1
- 루트 CA에 의해 서명된 중간 CA 생성 , on page 2
- 중간 CA를 사용하여 서명된 앱 인증서, on page 2
- 호스트에서 루트 CA를 신뢰할 수 있는 CA로 설치, on page 2

자체 서명된 루트 CA 생성

자체 서명된 루트 CA(인증 기관)를 생성합니다.

```
openssl genrsa -out myca.key 2048
# password protect key: openssl genrsa -out myca.key -des3 2048
openssl req -x509 -new -key myca.key -sha384 -days 1825 -out myca.crt \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=SecurityOU/CN=rootca.myorg.com/emailAddress=rootca@myorg.com"
```

이 루트 CA는 사용자(클라이언트) 머신에 신뢰할 수 있는 루트 CA로 설치해야 합니다.

Note **MacOS**를 사용하여 셀프 서명한 인증서를 생성하면 정방향 및 역방향 프록시 시나리오에 사용할 수 있는 적절한 인증서가 생성되지 않습니다. 인증서에서는 *Is CA* 옵션이 *True*로 설정되어 있어야 하며, MacOS를 사용하여 생성된 인증서는 그렇지 않습니다. 셀프 서명 인증서는 멀티 클라우드 방어 UI(Certificates(인증서)(Certificates(인증서)(Certificates)) - Create(생성(Create)) - Generate(생성) 내에서 또는 **Linux**를 사용하여 생성하는 것이 좋습니다.

자체 서명 루트 CA에서 서명한 인증서 생성

위의 루트 인증 기관(CA)에서 서명한 인증서를 생성합니다. 이 인증서는 애플리케이션에서 사용할 수 있습니다.

```
openssl genrsa -out app1.key 2048
# password protect key: openssl genrsa -out -des3 app1.key 2048
```

```
openssl req -new -key app1.key -out app1.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=AppOU/CN=app1.myorg.com/emailAddress=app1@myorg.com"
openssl x509 -req -in app1.csr -CA myca.crt -CAkey myca.key -out app1.crt -sha384\
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "
[SAN]\nbasicConstraints=CA:false\nsubjectAltName=DNS:app1.myorg.com,DNS:app1-
1.myorg.com,IP:192.168.10.21,IP:192.168.10.22")
```

루트 CA에 의해 서명된 중간 CA 생성

루트 CA(인증 기관)를 사용하여 앱 인증서에 서명하지 않으려면 루트 CA가 서명한 중간 CA를 생성한 다음 중간 CA를 사용하여 앱 인증서에 서명합니다. 앱 인증서에 중간 인증서를 추가합니다. 이 시점에서 앱 인증서에 체인으로 연결된 2개의 인증서가 있습니다.

```
openssl genrsa -out interca.key 2048
# password protect key: openssl genrsa -out -des3 interca.key 2048
openssl req -new -key interca.key -out interca.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=InterSecurityOU/CN=interca1.myorg.com/emailAddress=interca1@myorg.com"
openssl x509 -req -in interca.csr -CA myca.crt -CAkey myca.key -out interca.crt - sha384 \
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "[SAN]\nbasicConstraints=CA:true")
```

중간 CA를 사용하여 서명된 앱 인증서

```
openssl genrsa -out app1.key 2048
# password protect key: openssl genrsa -out -des3 app1.key 2048
openssl req -new -key app1.key -out app1.csr \
  -subj "/C=US/ST=CA/L=Santa
Clara/O=MyOrg/OU=AppOU/CN=app1.myorg.com/emailAddress=app1@myorg.com"
openssl x509 -req -in app1.csr -CA interca.crt -CAkey interca.key -out app1.crt - sha384 \
  -days 365 -CAcreateserial -extensions SAN \
  -extfile <(printf "
[SAN]\nbasicConstraints=CA:false\nsubjectAltName=DNS:app1.myorg.com,DNS:app1-
1.myorg.com,IP:192.168.10.21,IP:192.168.10.22")
```

파일에 app1.crt 및 interca.crt를 추가하여 결합된 인증서를 만들고 애플리케이션에서 결합된 인증서를 사용합니다. 루트 CA는 클라이언트 머신에 신뢰할 수 있는 루트 CA로 설치해야 합니다.

호스트에서 루트 CA를 신뢰할 수 있는 CA로 설치

OS	명령
Ubuntu	인증서 파일을 `/usr/local/share/ca-certificates`로 복사하고 `sudo update-ca- certificates` 명령을 실행합니다.
CentOS	인증서 파일을 `/etc/pki/ca-trust/source/anchors`로 복사하고 `sudo update-ca- trust extract` 명령을 실행합니다.

OS	명령
Windows	파일을 더블 클릭하고 신뢰할 수 있는 루트에 인증서를 추가하거나 `certutil -addstore "Root" <crt-file>` 명령을 실행합니다.