

Pour activer TLS/SSL avec MongoDB et Docker, vous devez créer des certificats SSL et les configurer correctement dans votre conteneur MongoDB. Voici un guide étape par étape pour configurer TLS/SSL dans votre configuration Docker Compose.

Étape 1: Générer les certificats SSL

1- Créer un répertoire pour les certificats :

```
mkdir -p mongo-certs
```

```
cd mongo-certs
```

2- Générer une clé privée et un certificat auto-signé :

```
openssl req -newkey rsa:4096 -new -x509 -days 365 -nodes -out mongodb-cert.crt -keyout  
mongodb-cert.key
```

3- Combiner la clé et le certificat en un seul fichier :

```
cat mongodb-cert.key mongodb-cert.crt > mongodb.pem
```

Étape 2: Configurer MongoDB pour utiliser TLS/SSL

1. Mettre à jour le fichier Docker Compose pour inclure les certificats :

Assurez-vous que les certificats générés sont montés dans le conteneur MongoDB et que MongoDB est configuré pour utiliser TLS/SSL.

Container backend :

environment: -

MONGO_URI=mongodb://root:mongopass@mongo-container:27017/authentication?

authSource=admin&ssl=true – MONGO_CERT_PATH=/app/certs/mongodb-cert.crt

volumes:

```
- "/backend:/app"  
- "/mongo-certs:/app/certs"
```

container mongo:

```
volumes: - mongo-data:/data/db - ./mongo-certs:/etc/mongo-certs command: mongod --tlsMode  
requireTLS --tlsCertificateKeyFile /etc/mongo-certs/mongodb.pem --tlsCAFile  
/etc/mongo-certs/mongodb-cert.crt
```

Étape 3 : Mettre à jour votre application Node.js

2. Utilisez les variables d'environnement pour le chemin du certificat :

```
const mongoUri = process.env.MONGO_URI ||  
"mongodb://root:mongopass@localhost:27017/authentication?authSource=admin&ssl=true"; const  
certPath = process.env.MONGO_CERT_PATH || '/app/certs/mongodb-cert.crt'; const options =
```

```
{ useNewUrlParser: true, useUnifiedTopology: true, ssl: true, sslValidate: false, sslCA:  
fs.readFileSync(certPath), // Chemin vers le certificat CA };
```