Enable TLS termination on the ingress side

Step 1: Generate a Self-Signed Certificate**

We'll use the `openssl` command to create a self-signed certificate and private key. This command will generate two files: `tls.crt` and `tls.key`.

Run the following commands on your local machine:

```
# Create a self-signed certificate and key with OpenSSL
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=example.com/O=example.com"
```

- **Explanation**:
  - `-x509`: Specifies that the certificate should be self-signed.
  - `-nodes`: Prevents OpenSSL from encrypting the private key.
  - `-days 365`: Sets the certificate's validity period to 365 days.
  - `-newkey rsa:2048`: Creates a new private key using RSA with a 2048-bit key length.
  - `-keyout tls.key`: Specifies the output file for the private key.
  - `-out tls.crt`: Specifies the output file for the certificate.
  - `-subj "/CN=example.com/O=example.com"`: Sets the subject name (`CN`) and organization (`O`) for the certificate.

This will create two files in your current directory:

- `tls.crt`: The certificate file.
- `tls.key`: The private key file.

Step 2: Create a Kubernetes Secret

Next, we'll create a Kubernetes Secret to store the self-signed certificate and private key. The Secret can then be referenced by the Ingress object.

Run the following command:

```
kubectl create secret tls example-tls-secret --cert=tls.crt --key=tls.key
```

- **Explanation**:
  - `tls`: Specifies that the secret type is TLS.
  - `example-tls-secret`: The name of the Secret.
  - `--cert=tls.crt`: Path to the certificate file.
  - `--key=tls.key`: Path to the private key file.

Step 3: Update the Ingress Object to Use TLS

We'll modify the existing Ingress object to enable TLS termination by referencing the newly created Secret. The Ingress object will now listen on both HTTP (`port 80`) and HTTPS (`port 443`).

Updated Ingress Configuration (`example-ingress.yaml`):

Add the following to the file

```yaml
# example-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
spec:
  ingressClassName: nginx
# add this part
  tls:
  - hosts:
    - www.example.com
    - api.example.com
    secretName: example-tls-secret
# till here and continue with the rest of the file as it is
```

- **Explanation**:
    - **`tls` Section**: Defines the hosts (`www.example.com` and `api.example.com`) that should use the TLS termination and references the Secret (`example-tls-secret`) created in Step 2.
    - **`secretName`**: Name of the TLS Secret that contains the certificate and key.

Step 4: Apply the Updated Ingress Configuration

Run the following command to update your Ingress resource:

```
kubectl apply -f example-ingress.yaml
```

Test HTTP to HTTPS Redirection (Optional):

If you want to ensure that all HTTP traffic is redirected to HTTPS, add the following annotation to your Ingress object:

```yaml
metadata:
  annotations:
    nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
```

This will force all HTTP traffic to redirect to HTTPS.