ROUTE Types

Expose an application securely outside the cluster using OpenShift Routes with TLS termination.

Namespace Creation

```
oc new-project route-tls-lab
```

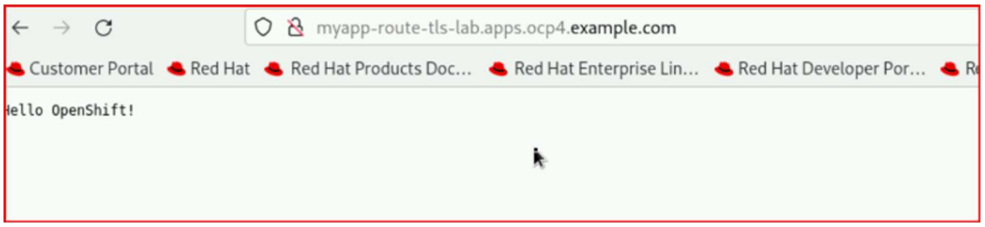
Application Sample

You can use any simple web server. We'll use a public "hello" container.

```
oc new-app --name=myapp quay.io/openshift-examples/hello-front
```

```
student@workstation ~]$ oc get all
arning: apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavaila
le in v4.10000+
AME                          READY   STATUS    RESTARTS   AGE
od/myapp-8669f948bf-wzd6v    1/1     Running   0          53s

AME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)             AG

ervice/myapp    ClusterIP   172.30.97.184   <none>        8080/TCP,8888/TCP   54


AME                     READY   UP-TO-DATE   AVAILABLE   AGE
eployment.apps/myapp    1/1     1            1           54s

AME                                DESIRED   CURRENT   READY   AGE
eplicaset.apps/myapp-67bfdcb8c6    0         0         0       54s
eplicaset.apps/myapp-8669f948bf    1         1         1       53s

AME                                     IMAGE REPOSITORY
                              TAGS      UPDATED
magestream.image.openshift.io/myapp     image-registry.openshift-image-registry.s
c:5000/route-tls-lab/myapp    latest    54 seconds ago
```

```
curl http://$(oc get route myapp -o jsonpath='{.spec.host}')
```



CA -Certificate Creation

```
openssl req -x509 -newkey rsa:2048 -nodes \
  -keyout tls.key -out tls.crt \
  -subj "/CN=myapp.example.com" -days 365
```

Secret Creation

```
oc create secret tls myapp-tls \
  --cert=tls.crt \
  --key=tls.key
```

Edge Route

TLS terminates at the OpenShift router. Backend receives HTTP.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: myapp-edge
spec:
  host: myapp.example.com
  to:
    kind: Service
    name: myapp
  port:
    targetPort: 8080
  tls:
    termination: edge
    insecureEdgeTerminationPolicy: Redirect
```

Apply:

```
oc apply -f route-edge.yaml
```

Test:

```
curl -k https://myapp.example.com
```

Passthrough Route:

Router does NOT decrypt. The backend must handle TLS.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: myapp-passthrough
spec:
  host: myapp.example.com
  to:
    kind: Service
    name: myapp
  tls:
    termination: passthrough
  port:
    targetPort: https
```

Apply:

```
oc apply -f route-passthrough.yaml
```

Test:

```
curl -k --resolve myapp.example.com:443:<router-ip>
https://myapp.example.com
```

Reencrypt Route

Router decrypts → re-encrypts → backend.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: myapp-reencrypt
spec:
  host: myapp.example.com
  to:
    kind: Service
    name: myapp
  tls:
    termination: reencrypt
    destinationCACertificate: |
      -----BEGIN CERTIFICATE-----
      (paste backend CA cert)
      -----END CERTIFICATE-----
    insecureEdgeTerminationPolicy: Redirect
```

Apply:

```
oc apply -f route-reencrypt.yaml
```

```
[student@workstation ~]$ vim route-reencrypt.yaml
[student@workstation ~]$ oc apply -f route-reencrypt.yaml
oute.route.openshift.io/myapp-reencrypt created
```
