**Enable Secure HTTPS Connection between Browser and OpenShift Application**

**General Discussion**

Enabling HTTPS connection between the browser and the OpenShift application is a simple task. Unfortunately, it is not clearly explained in OpenShift documentation. I kept getting an error when trying to do it for the first time. What happened is that I followed this link: <https://docs.openshift.com/container-platform/4.4/networking/routes/secured-routes.html>, which explains how to configure a secure route using a custom certificate. It does not tell you that you could just use the default certificate. For some unknown reasons, I noticed that the “Create” button is disabled when trying to create a secure route from the OpenShift web console. I thought that it was because the certificate/key/CA fields are blank, so I created a self-signed certificate. After populating the certificate and private key, the “Create” button is enabled. However, I got an error when clicking the “Create” button:

*admission webhook "validate.route.create" denied the request: Routes with custom-host prohibited on this cluster.*

This error message is quite misleading – I did not use a custom-host. I was using a custom certificate.

I still do not know the root cause of the above error. I guess that Red Hat does not developers to use the custom certificates under the 60-day free OpenShift trial program.

Anyway, I read the “oc create route” help document and tried to create a secure route without specifying certificate. It was successful. Next I tried creating a secure route using OpenShift web console and it was a success, too. Below is the summary of the steps for each approach, to create a HTTPS connection between browser and the OpenShift application.

**Approach 1: using oc command:**

In the previous article (OpenShift-Deployment-1-Simple-Spring-Boot-Application.pdf), we created the HTTP route named “springbootstarter”, which uses the service “springbootstarter”. Now let’s enable HTTPS connection by simply creating a route name springbootstarter1“, using the same service:

*oc create route edge springbootstarter1 --service=springbootstarter*

Pointing your browser at

<https://springbootstarter1-dannyproject.apps.us-east-2.starter.openshift-online.com/springbootstarter/hello>

Note the URL is different from the original HTTP one.

**Approach 2: creating a secure route from Web Console**

Networking – Routes – Create Route

Name: springbootstarter1

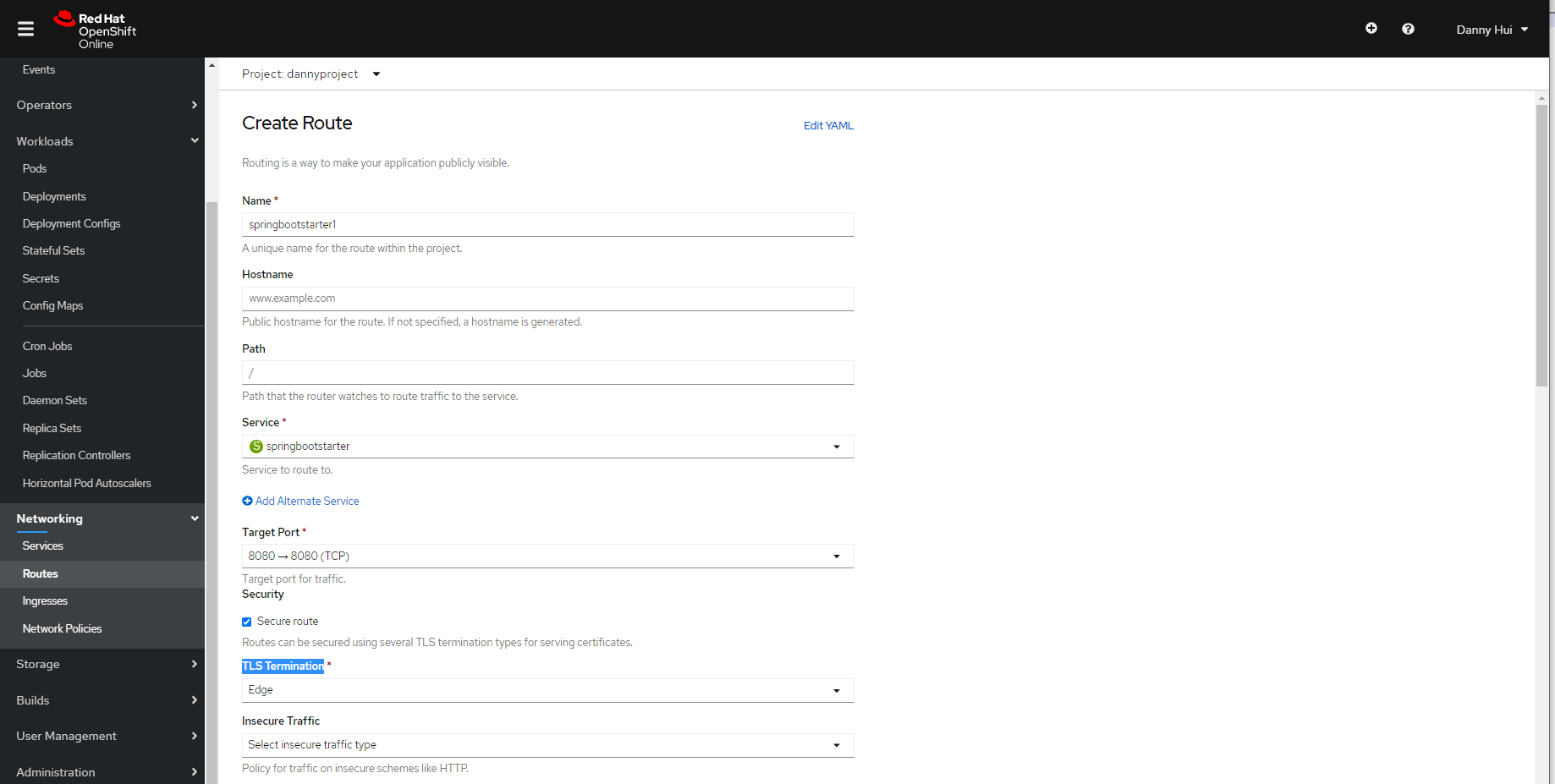
Service: springbootstarter

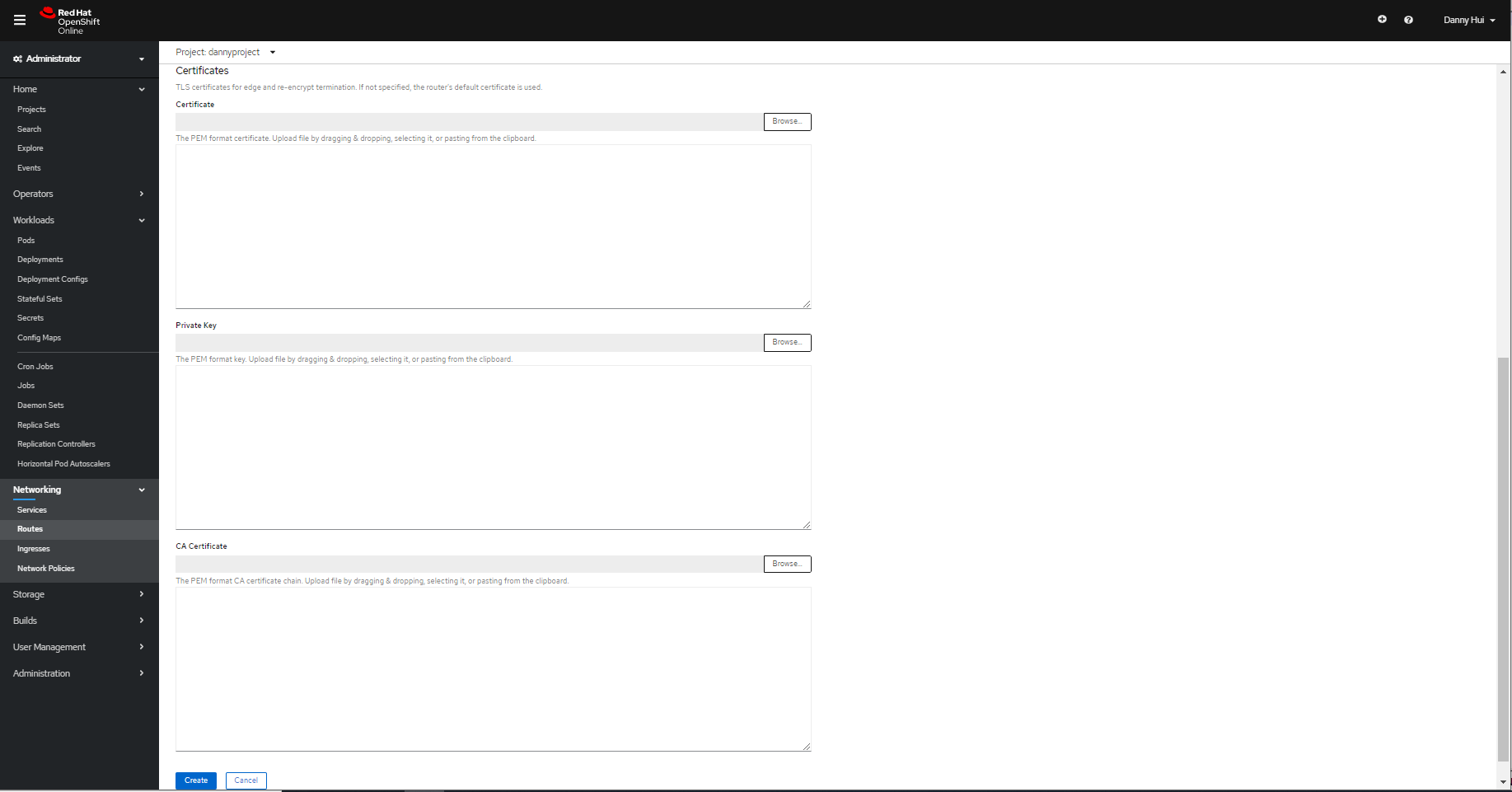
Target Port: 8080->8080-TCP

Check Secure route

TLS Termination: Edge

Leave all other fields blank





Create

**Approach 3: modifying non-secured route from Web Console**

Networking – Routes – [springbootstarter](https://console-openshift-console.apps.us-east-2.starter.openshift-online.com/k8s/ns/dannyproject/routes/springbootstarter) - YAML

Under spec, add

tls:

termination: edge

Save

Pointing browser at <https://springbootstarter-dannyproject.apps.us-east-2.starter.openshift-online.com/springbootstarter/hello>