**Deploy Spring Boot Web Application with External Configuration File**

**from OpenShift Web Console and oc**

**General discussion**

A Spring Boot web application is essentially a Java application. There are two ways to load configuration files into the application: an external configuration file from the file system and a configuration file packaged inside the jar/war file. The former approach is the best industry practice, since it allows the configuration being modified and re-deployed without re-deploying the application.

For a Java application to load an external configuration file from the file system, the configuration file needs to be physically located in the specified path and we need to tell JVM the file path via *JAVA\_OPTIONS* environment variable. For Cloud containers, this means we need to do two things: mount the file in the volume with the specified path and define the *JAVA\_OPTIONS* environment variable. To mount the file in the volume, we need to use CongMap. Since we have two configuration files for this example: *application.properties* for Spring Boot and *logback.xml* for logging configuration, we need to create two ConfigMap objects. After the Deployment Config object is created, we need mount both files to their respective volumes, by modifying the Deployment Config object.

The last thing I’d like to highlight is that the port 8082 is used intentionally, instead of the default 8080.

For Spring Boot application, this port is defined in the configuration file *application.properties*:

server.port=8082

By not using the default port 8080, we can easily find whether JVM is able to load and use *application.properties* or not.

**Create ConfigMap from oc (two ConfigMap objects as discuss abobe)**

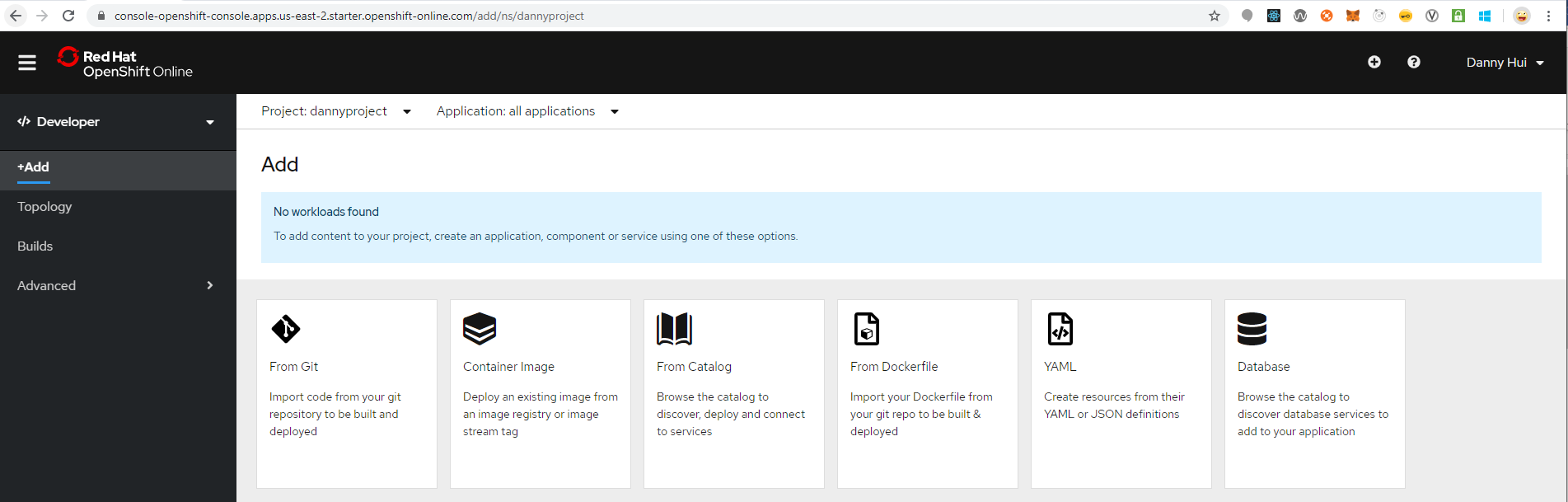
oc create configmap springbootstarter-w-config-application-properties --from-file=/opt/springbootstarter-w-config/config/application.properties

oc create configmap springbootstarter-w-config-logback-xml --from-file=/opt/springbootstarter-w-config/config/logback/logback.xml

**OpenShift Web Console**

Developer Perspective

+Add



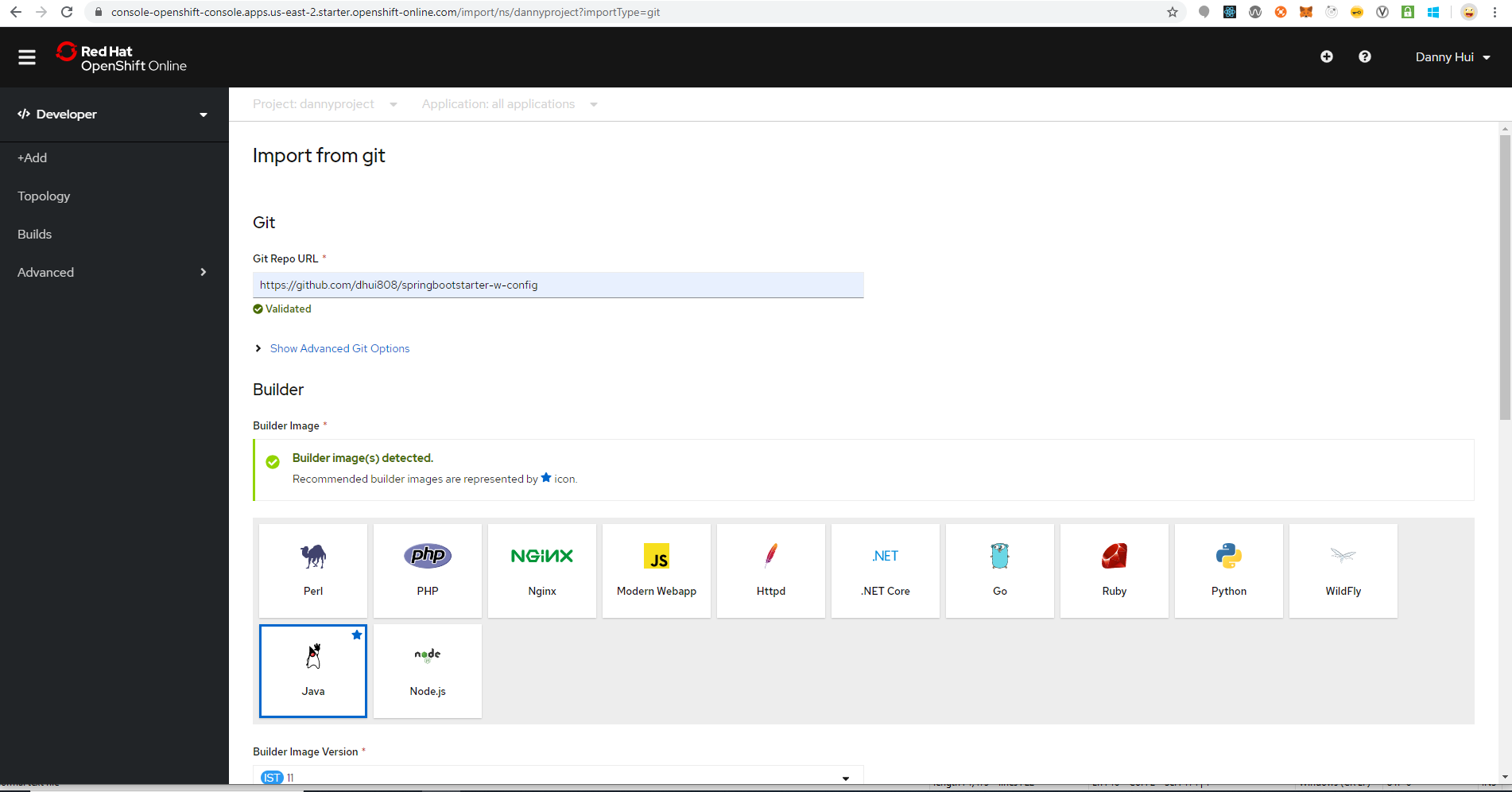
From Git

Git repo URL

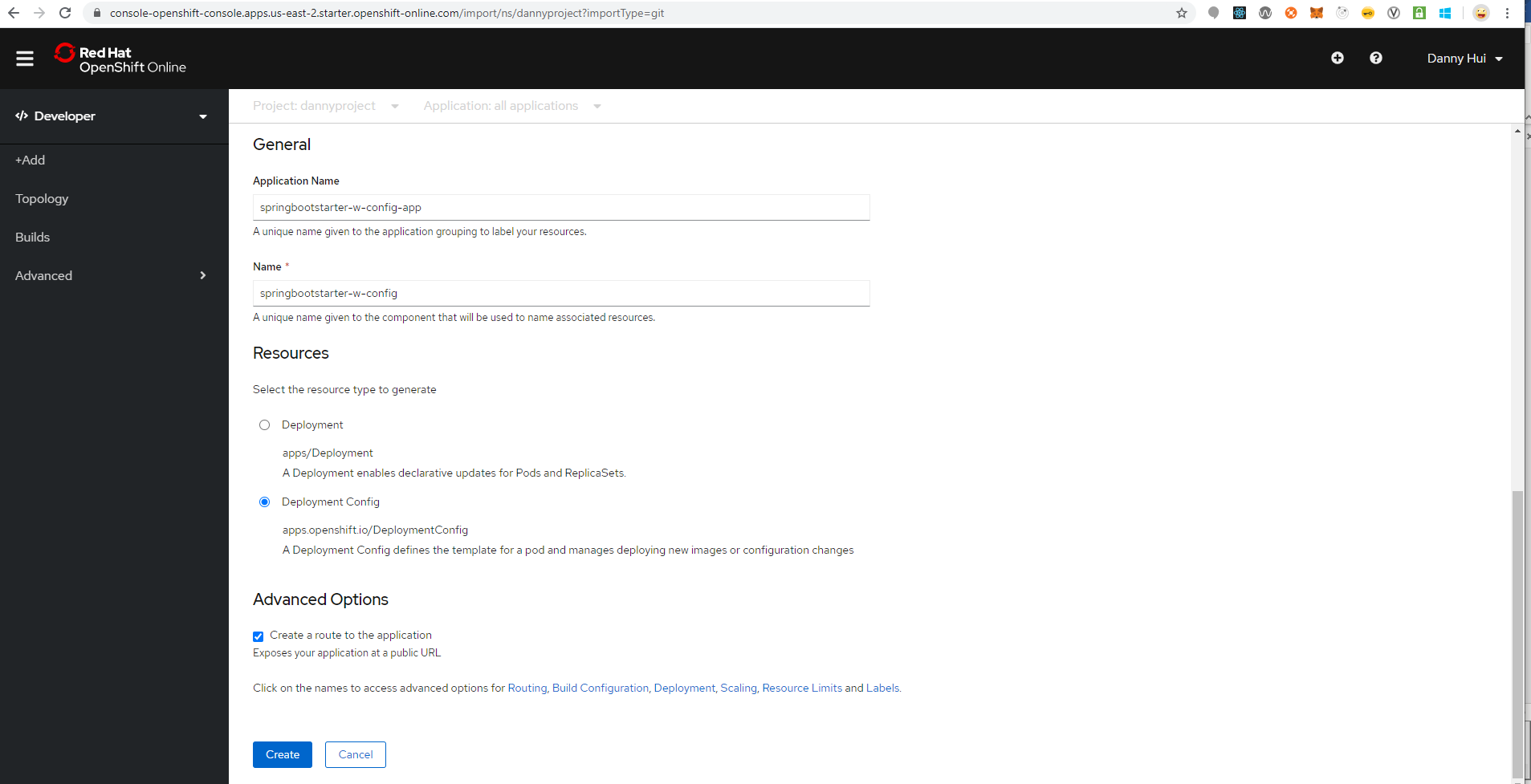
https://github.com/dhui808/springbootstarter-w-config

Builder Image

Java



Select Deployment Config (selecting Deployment also works)



Click Deployment link (above the buttons)

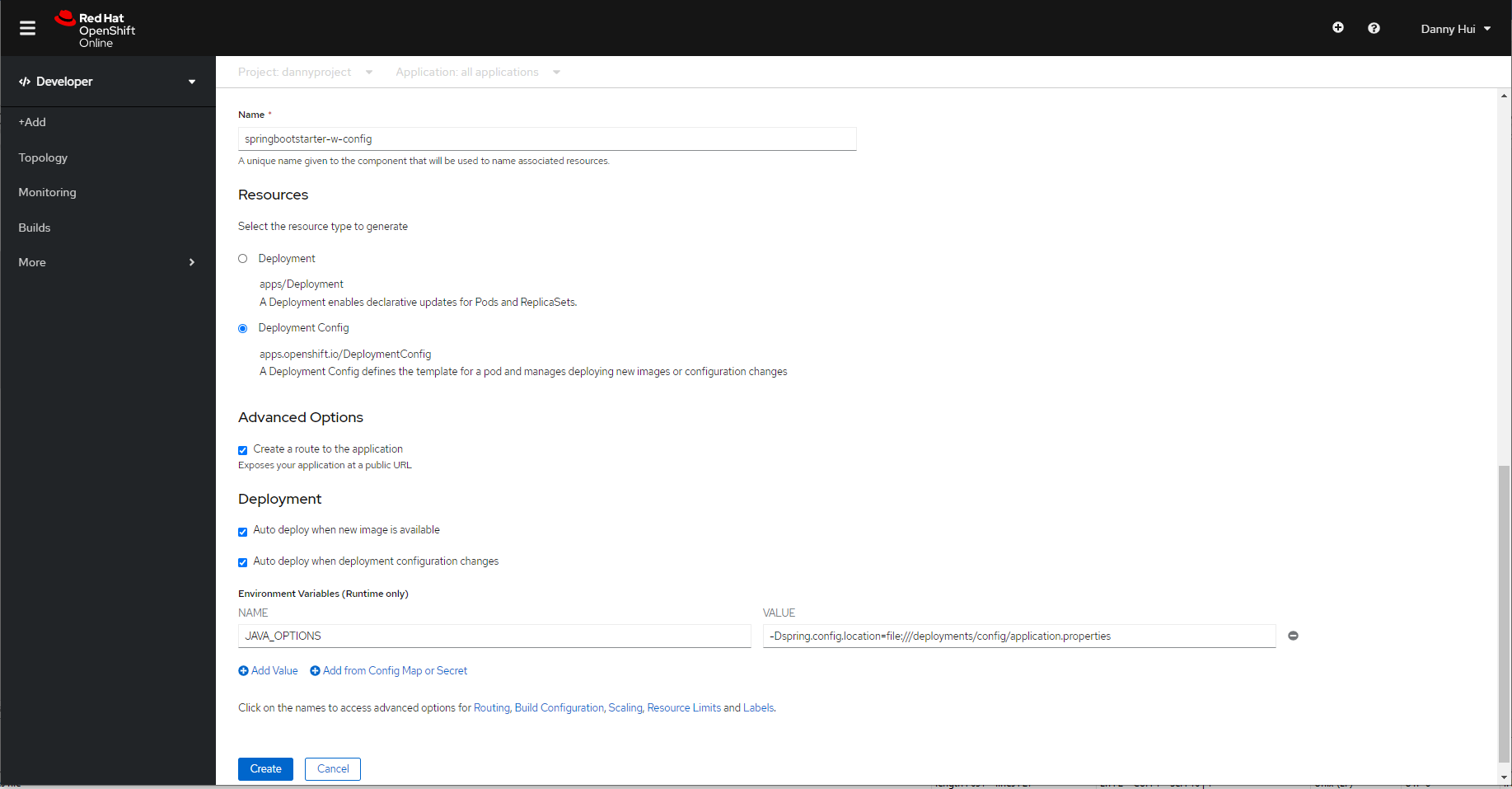
Environment Variables (Runtime only)

NAME

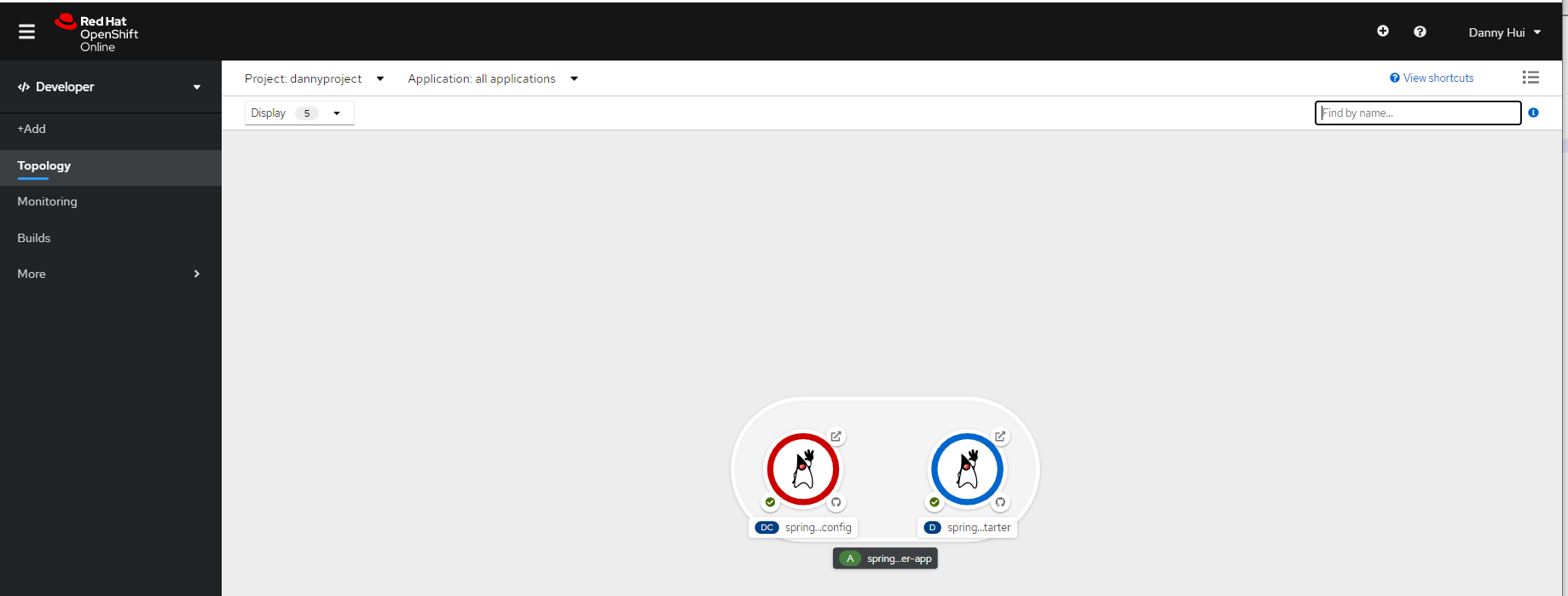
JAVA\_OPTIONS

VALUE

-Dspring.config.location=file:///deployments/config/application.properties



Create



**Add Volume and Volume Mount from oc**

(springbootstarter-w-config is the name of the newly created Deployment Config)

oc set volume dc/springbootstarter-w-config --add --name volume-config -t configmap --configmap-name springbootstarter-w-config-application-properties -m /deployments/config -o yaml

oc set volume dc/springbootstarter-w-config --add --name volume-logback-config -t configmap --configmap-name springbootstarter-w-config-logback-xml -m /opt/springbootstarter-w-config/config/logback --read-only=true -o yaml

If Deployment is selected instead of Deployment Config in the previous application creation step, we must use the corresponding oc commands below:

oc set volume deployment/springbootstarter-w-config --add --name volume-config -t configmap --configmap-name springbootstarter-w-config-application-properties -m /deployments/config -o yaml

oc set volume deployment/springbootstarter-w-config --add --name volume-logback-config -t configmap --configmap-name springbootstarter-w-config-logback-xml -m /opt/springbootstarter-w-config/config/logback --read-only=true -o yaml

Now we need to tell OpenShift Container that the Pod should use port 8082 instead of 8080.

**Administrator Perspective**

Networking – Services - springbootstarter-w-config - YAML

Under spec – ports

- name: 8082-tcp

protocol: TCP

port: 8082

targetPort: 8082

Save

Networking – Routes - springbootstarter-w-config - YAML

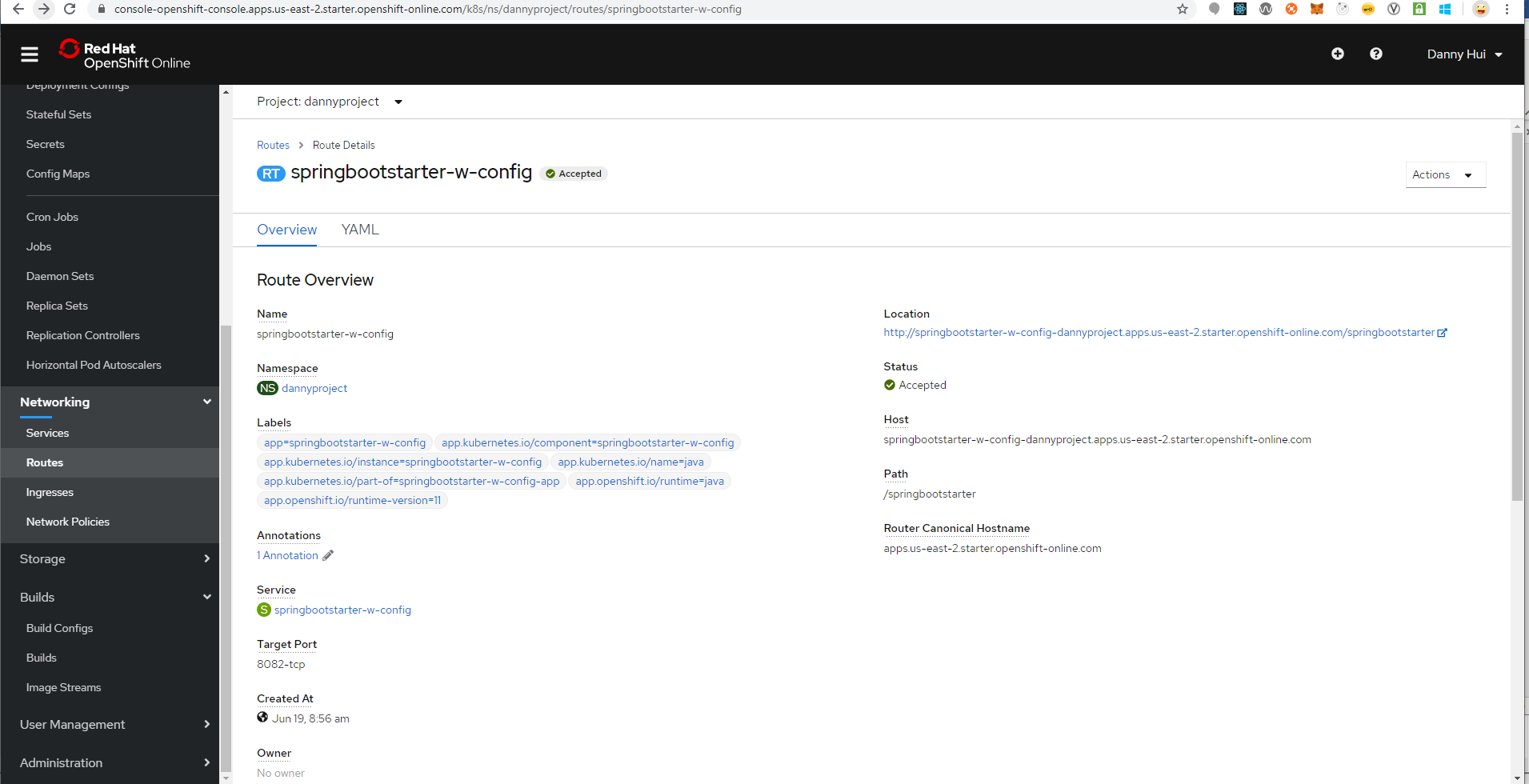
Under spec (actually we do not need to specify path)

port:

targetPort: 8082-tcp

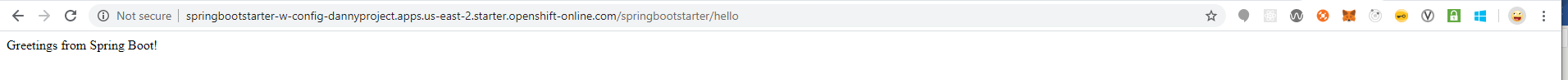
path: /springbootstarter

Save



**Browser**

<http://springbootstarter-w-config-dannyproject.apps.us-east-2.starter.openshift-online.com/springbootstarter/hello>



Voila!