Guided Exercise: Externalize the Configuration of Applications

Deploy a web server that takes configuration files from a configuration map.

Outcomes

- Create a web application deployment.
- Expose the web application deployment to external access.
- Create a configuration map from two files.
- Mount the configuration map in the web application deployment.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise. This command ensures that the cluster is accessible and that all resources are available for this exercise.

```
[student@workstation ~]$ lab start storage-configs
```

Instructions

1. Log in to the OpenShift web console as the `developer` user and switch to the **Administrator** perspective.

   Open a web browser and go to `https://console-openshift-console.apps.ocp4.example.com`

   Click **Red Hat Identity Management** and log in as the `developer` user with `developer` as the password. If the **Welcome to the Developer Perspective** message is displayed, then click **Skip tour**.

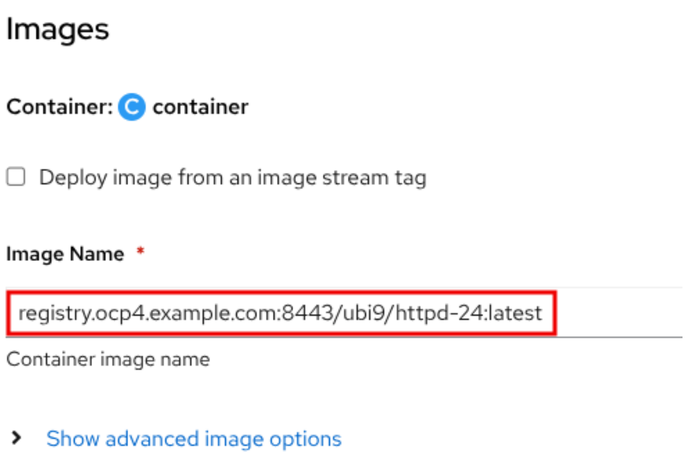
   From the perspective switcher, select **Administrator**.

2. Create a web application deployment named `webconfig` from the web console. Use the `registry.ocp4.example.com:8443/ubi9/httpd-24:latest` container image.

   Go to **Workloads → Deployments**.

   Set the project to `storage-configs` and click **Create Deployment**.

   Use `webconfig` for the deployment name and change the image name to `registry.ocp4.example.com:8443/ubi9/httpd-24:latest`.


Images
Container: C container
Deploy image from an image stream tag
Image Name *
Container image name
Show advanced image options

registry.ocp4.example.com:8443/ubi9/httpd-24:latest

   Click **Create**. Wait for the blue circle to indicate that three pods are running.

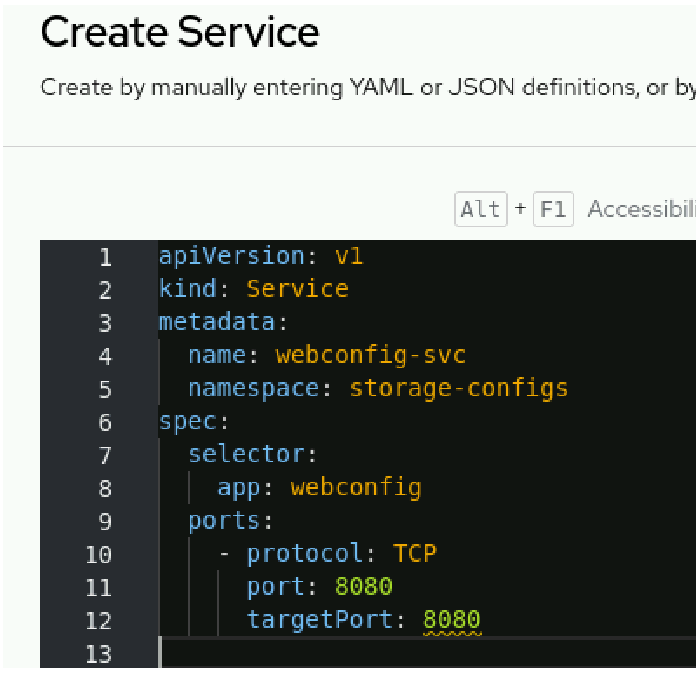
3. Expose the web application to external access from the web console. Create a service and a route for the web application. Use the values from the following tables to create the service and the route.

Service field	Service value
Service name	`webconfig-svc`
App selector	`webconfig`
Port number	`8080`
Target port	`8080`

Route field	Route value
Route name	`webconfig-rt`
Service name	`webconfig-svc`
Target port	`8080`

Go to **Networking** → **Services** and click **Create Service**.

Update the service values from the service table.


Create Service

Create by manually entering YAML or JSON definitions, or by

```
apiVersion: v1
kind: Service
metadata:
  name: webconfig-svc
  namespace: storage-configs
spec:
  selector:
    app: webconfig
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

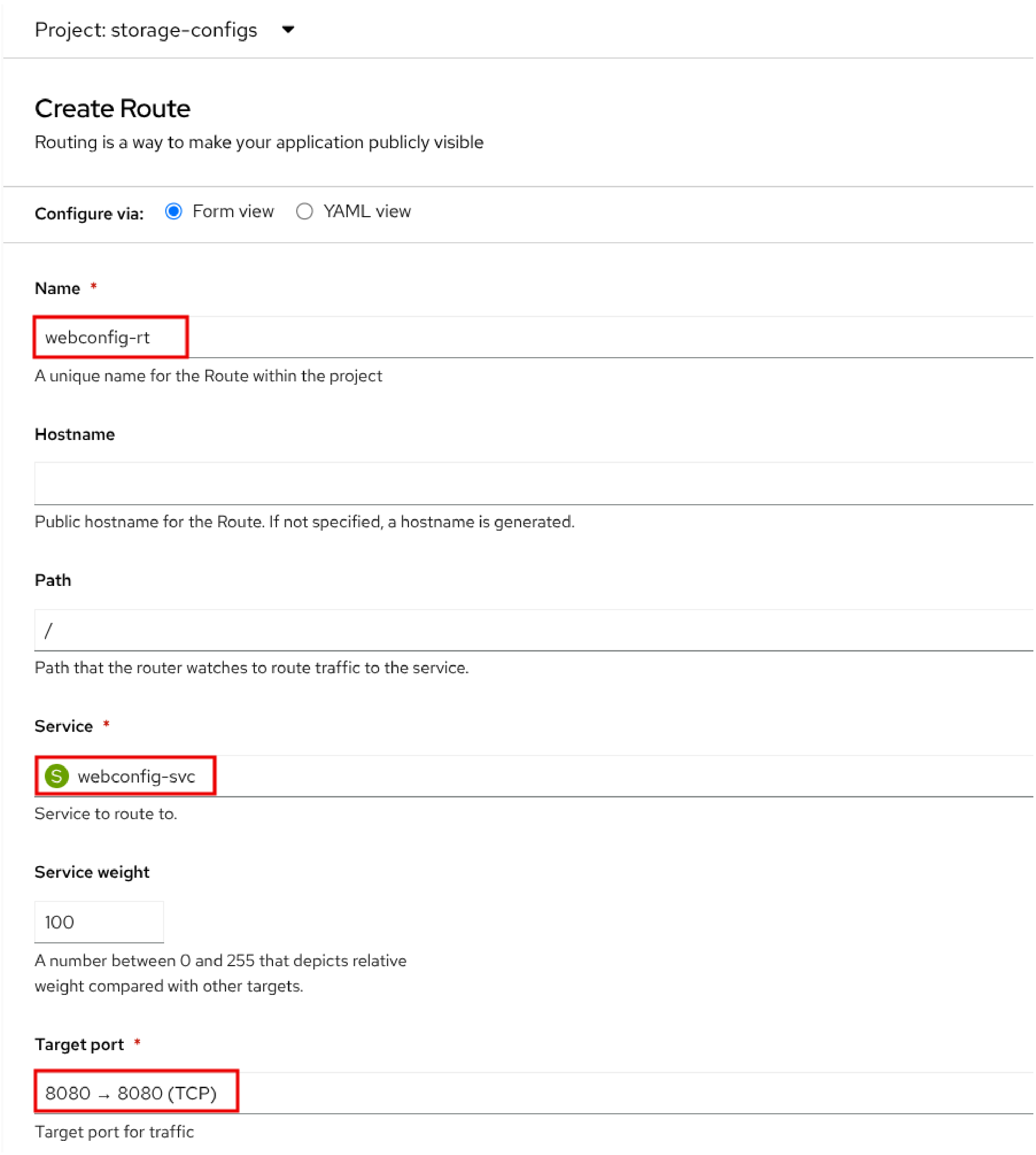
Click **Create**.

Alternatively, you can create the service from the CLI by using the following command:

```
[student@workstation ~]$ oc expose deploy/webconfig \
  --namespace storage-configs --name webconfig-svc \
  --selector app=webconfig \
  --target-port 8080 --port 8080
```

Go to **Networking** → **Routes** and click **Create Route**.

Create the route by using the values from the route table:


Project: storage-configs

Create Route

Routing is a way to make your application publicly visible

Configure via: [✓ Form view] YAML view

Name *

webconfig-rt

A unique name for the Route within the project

Hostname

Public hostname for the Route. If not specified, a hostname is generated.

Path

/

Path that the router watches to route traffic to the service.

Service *

webconfig-svc

Service to route to.

Service weight

100

A number between 0 and 255 that depicts relative weight compared with other targets.

Target port *

8080 → 8080 (TCP)

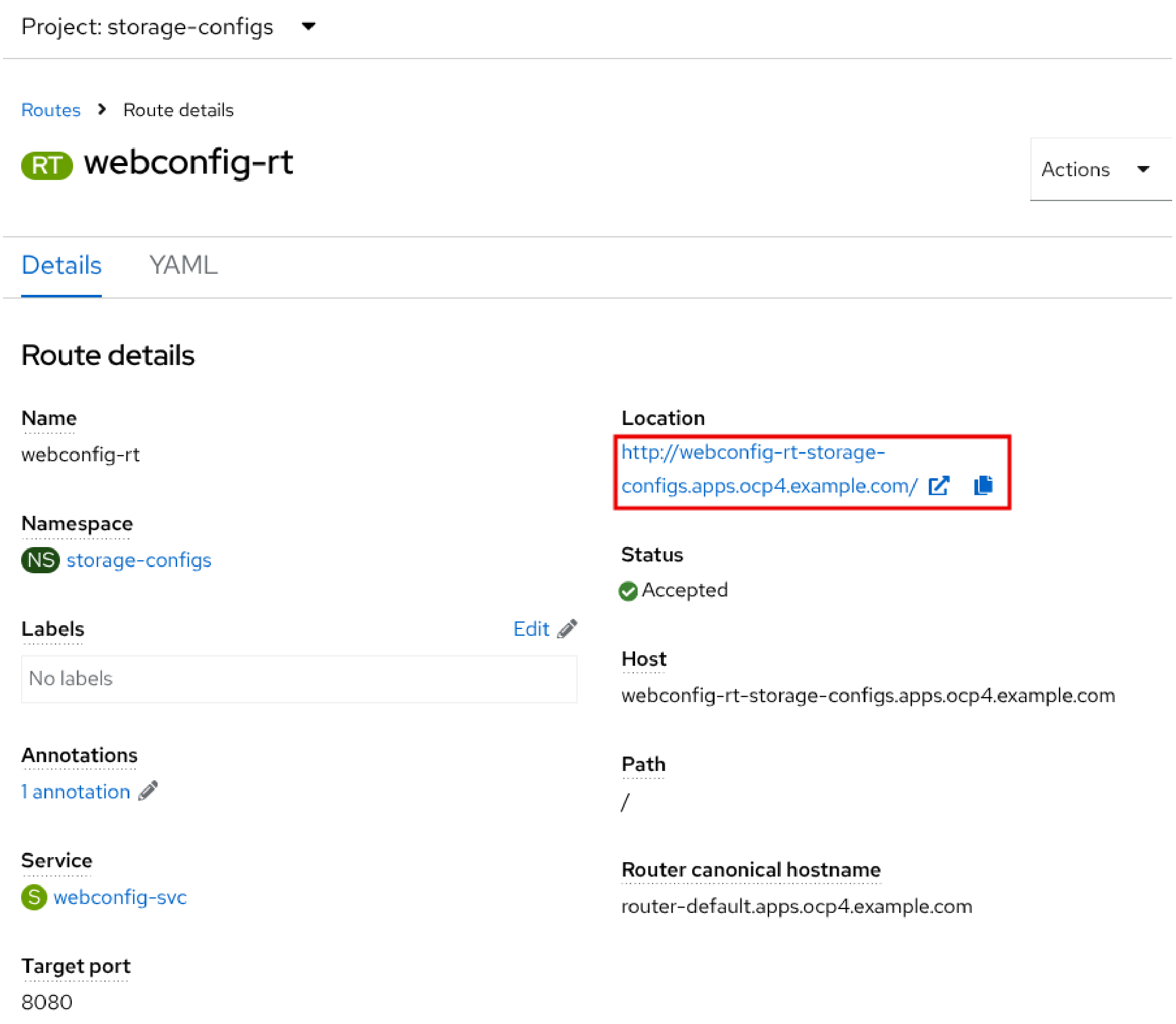
Target port for traffic

Click **Create**.

Alternatively, you can create the route from the CLI by using the following command:

```
[student@workstation ~]$ oc expose svc/webconfig-svc \
  --namespace storage-configs --name webconfig-rt
```

On the **Details** page of the route, click the **Location** link.



RT webconfig-rt

Route details

Name
webconfig-rt

Namespace
NS storage-configs

Labels
No labels

Annotations
1 annotation

Service
S webconfig-svc

Target port
8080

Location
http://webconfig-rt-storage-configs.apps.ocp4.example.com/

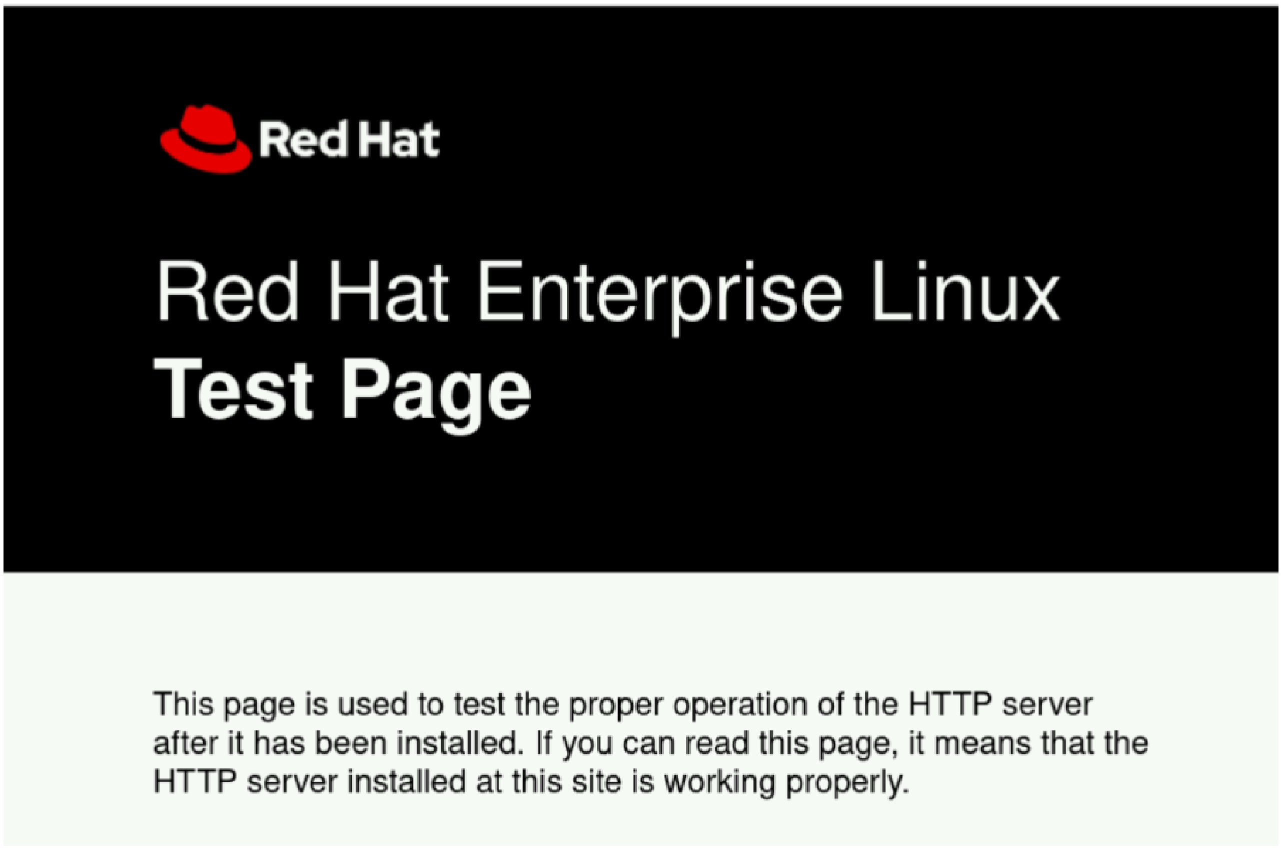
Status
Accepted

Host
webconfig-rt-storage-configs.apps.ocp4.example.com

Path
/

Router canonical hostname
router-default.apps.ocp4.example.com

The route link opens a browser tab that displays the default test page of the web server. Do not close the browser tab, because you use it in another step.



Red Hat Enterprise Linux
Test Page

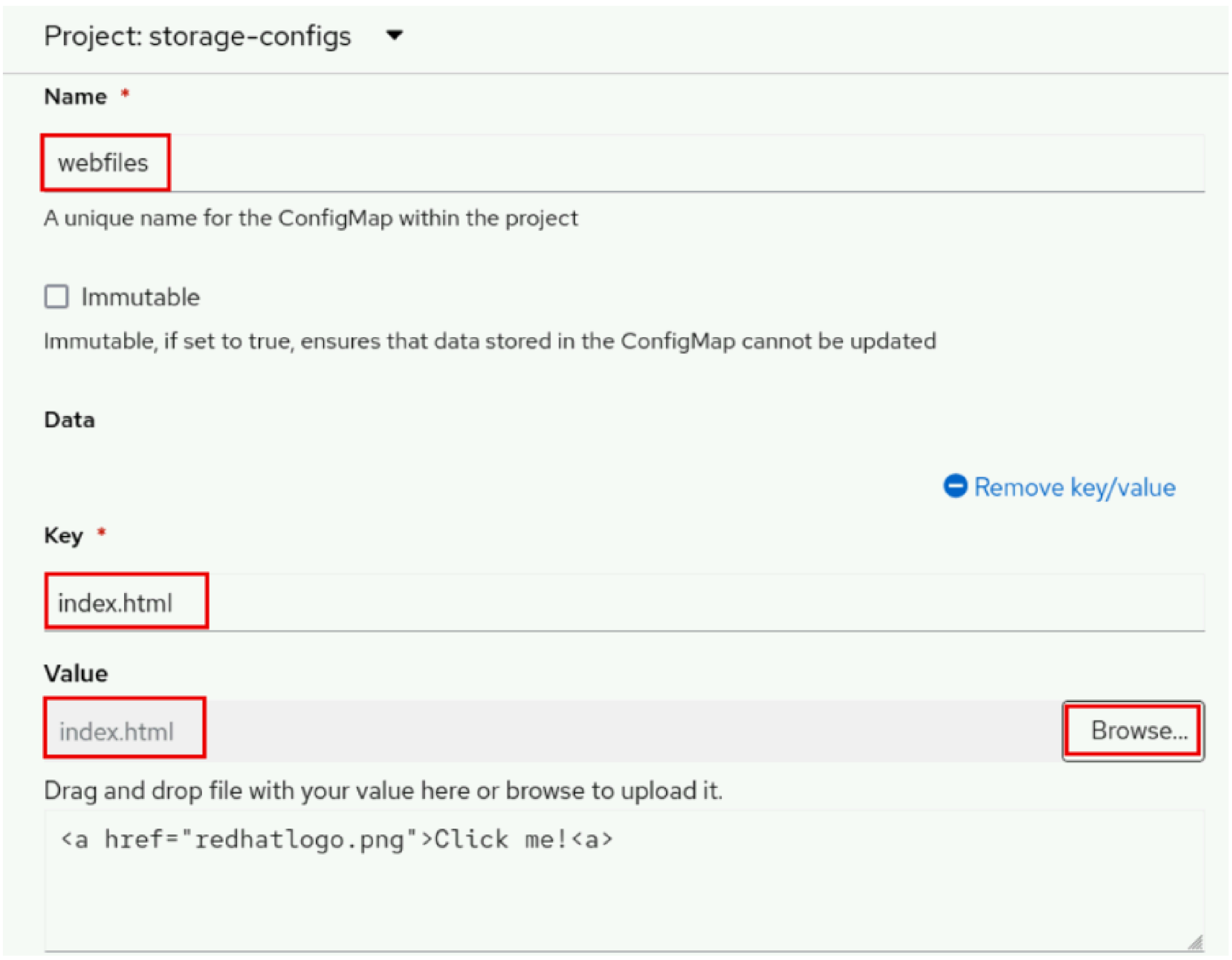
This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

4. Create a configuration map that contains the files for the web application by using the web console.

   Go to **Workloads** → **ConfigMaps** and click **Create ConfigMap**.

   Set the name of the configuration map to `webfiles`.

   Add a data key. In the **Data** section, define the `index.html` name as the **Key** value, and click **Browse** in the **Value** field to select the `/home/student/DO180/labs/storage-configs/index.html` file.


Project: storage-configs

Name *

webfiles

A unique name for the ConfigMap within the project

Immutable

Immutable, if set to true, ensures that data stored in the ConfigMap cannot be updated

Data

Remove key/value

Key *

index.html

Value

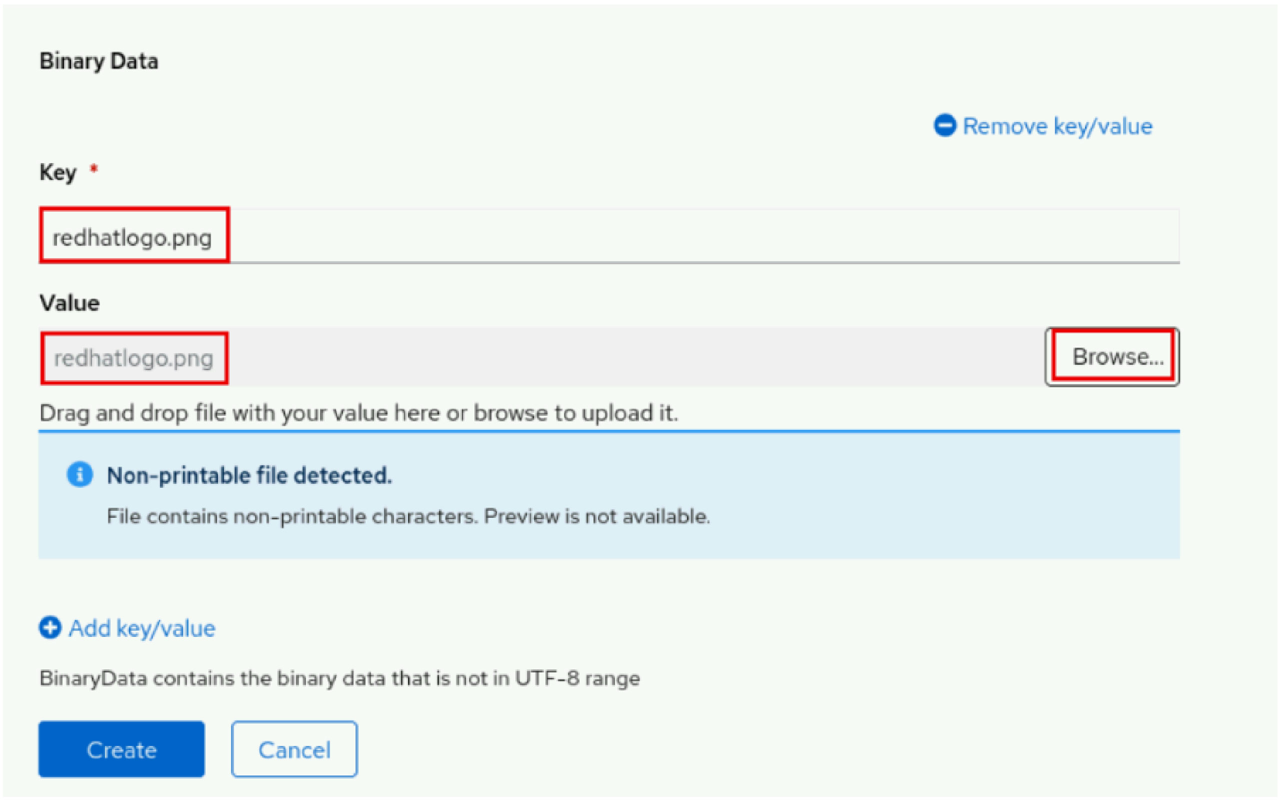
index.html Browse...

Drag and drop file with your value here or browse to upload it.

```
<a href="redhatlogo.png">Click me!<a>
```

Add a binary data key. In the **Binary Data** section, click **Add key/value**.

Define the `redhatlogo.png` name as the **Key** value, and click **Browse** in the **Value** field to select the `/home/student/DO180/labs/storage-configs/redhatlogo.png` file.


Binary Data

Remove key/value

Key *

redhatlogo.png

Value

redhatlogo.png Browse...

Drag and drop file with your value here or browse to upload it.

Non-printable file detected.

File contains non-printable characters. Preview is not available.

Add key/value

BinaryData contains the binary data that is not in UTF-8 range

Create Cancel

Click **Create**.

5. Mount the `webfiles` configuration map as a volume in the `webconfig` deployment by using the command line.

   In a terminal window, log in to the RHOCP cluster as the `developer` user with `developer` as the password.

```
[student@workstation ~]$ oc login -u developer -p developer \
  https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

   Select the `storage-configs` project.

```
[student@workstation ~]$ oc project storage-configs
Now using project "storage-configs" on server "https://api.ocp4.example.com:6443".
```

   Mount the `webfiles` configuration map as a volume.

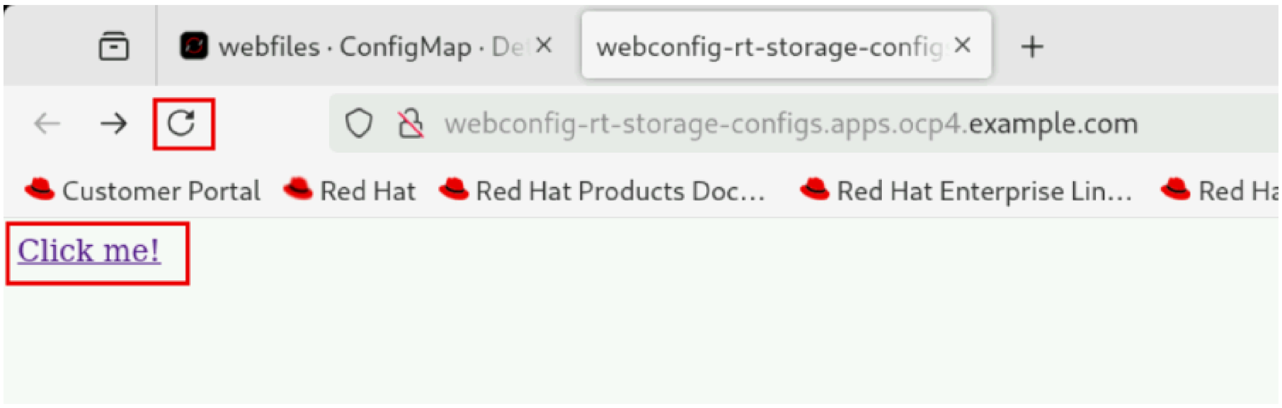
```
[student@workstation ~]$ oc set volume deployment/webconfig \
  --add --type configmap --configmap-name webfiles \
  --name webfiles-vol --mount-path /var/www/html/
deployment.apps/webconfig volume updated
```

   Verify that the deployment has three available replicas.

```
[student@workstation ~]$ oc get deployment
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
webconfig    3/3     3            3           3h20m
```

6. Verify that the web application shows the content from the configuration map.

   Switch to the web browser, and go to the web server tab. Click the reload icon in the web browser.


webfiles · ConfigMap · De
webconfig-rt-storage-config
webconfig-rt-storage-configs.apps.ocp4.example.com
Customer Portal
Red Hat
Red Hat Products Doc...
Red Hat Enterprise Lin...
Red Ha
Click me!

   Click the **Click me!** link that the web page displays. The web page shows the Red Hat logo.

Finish

On the `workstation` machine, use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish storage-configs
```