Starter Labs (Python)

WORKSHOP MODULES

Application Health

Background: Readiness and Liveness Probes

As we have seen before in the UI via warnings, there is a concept of application health checks in OpenShift. These come in two flavors:

- Readiness probe
- Liveness probe

From the Application Health section of the documentation, we see the definitions:

Liveness Probe

> A liveness probe checks if the container in which it is configured is still running. If the liveness probe fails, the kubelet kills the container, which will be subjected to its restart policy. Set a liveness check by configuring the `template.spec.containers.livenessprobe` stanza of a pod configuration.

Readiness Probe

> A readiness probe determines if a container is ready to service requests. If the readiness probe fails a container, the endpoints controller ensures the container has its IP address removed from the endpoints of all services. A readiness probe can be used to signal to the endpoints controller that even though a container is running, it should not receive any traffic from a proxy. Set a readiness check by configuring the `template.spec.containers.readinessprobe` stanza of a pod configuration.

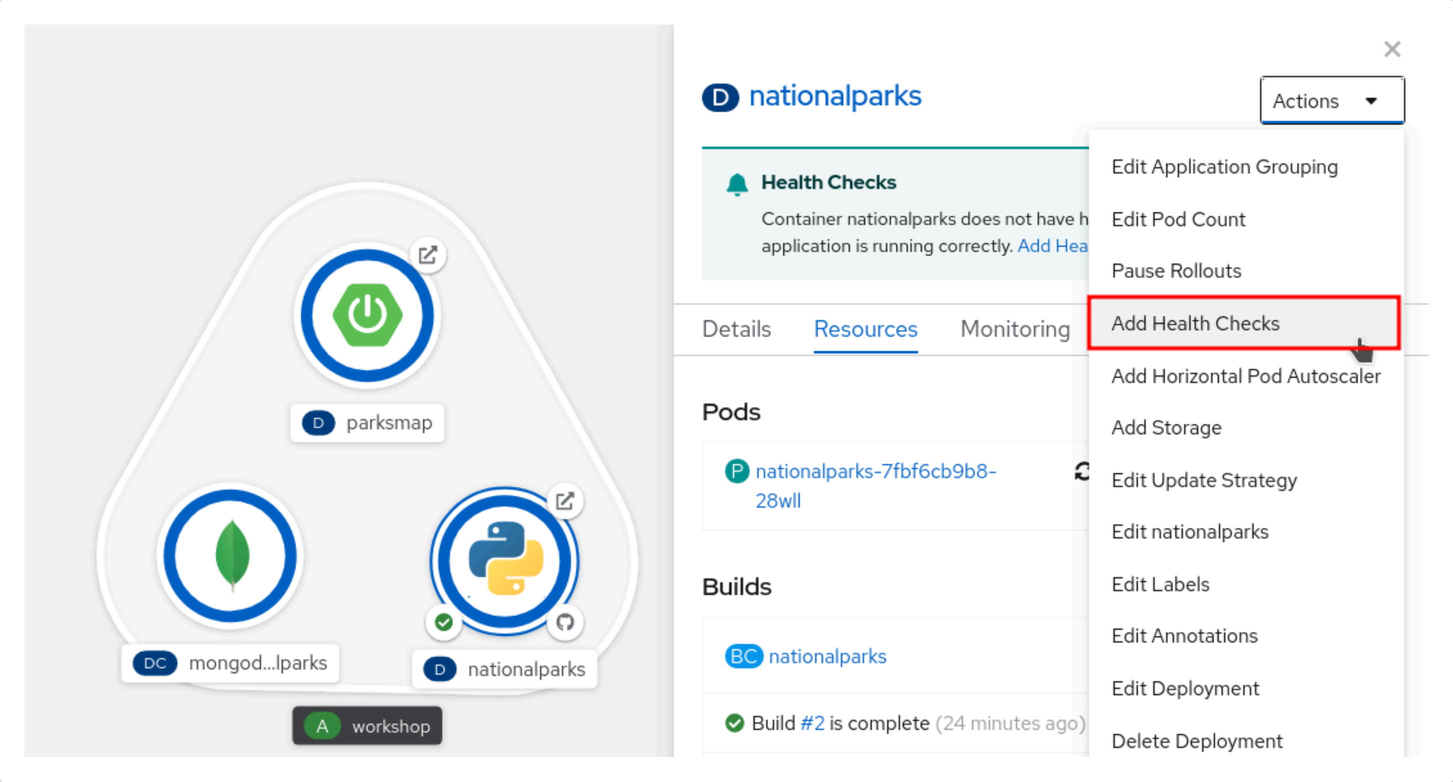
It sounds complicated, but it really isn’t. We will use the web console to add these probes to our `nationalparks` application.

Exercise: Add Health Checks

As we are going to be implementing a realistic CI/CD pipeline, we will be doing some testing of the "development" version of the application. However, in order to test the app, it must be ready. This is where OpenShift’s application health features come in very handy.

We are going to add both a readiness and liveness probe to the existing `nationalparks` deployment. This will ensure that OpenShift does not add any instances to the service until they pass the readiness checks, and will ensure that unhealthy instances are restarted (if they fail the liveness checks).

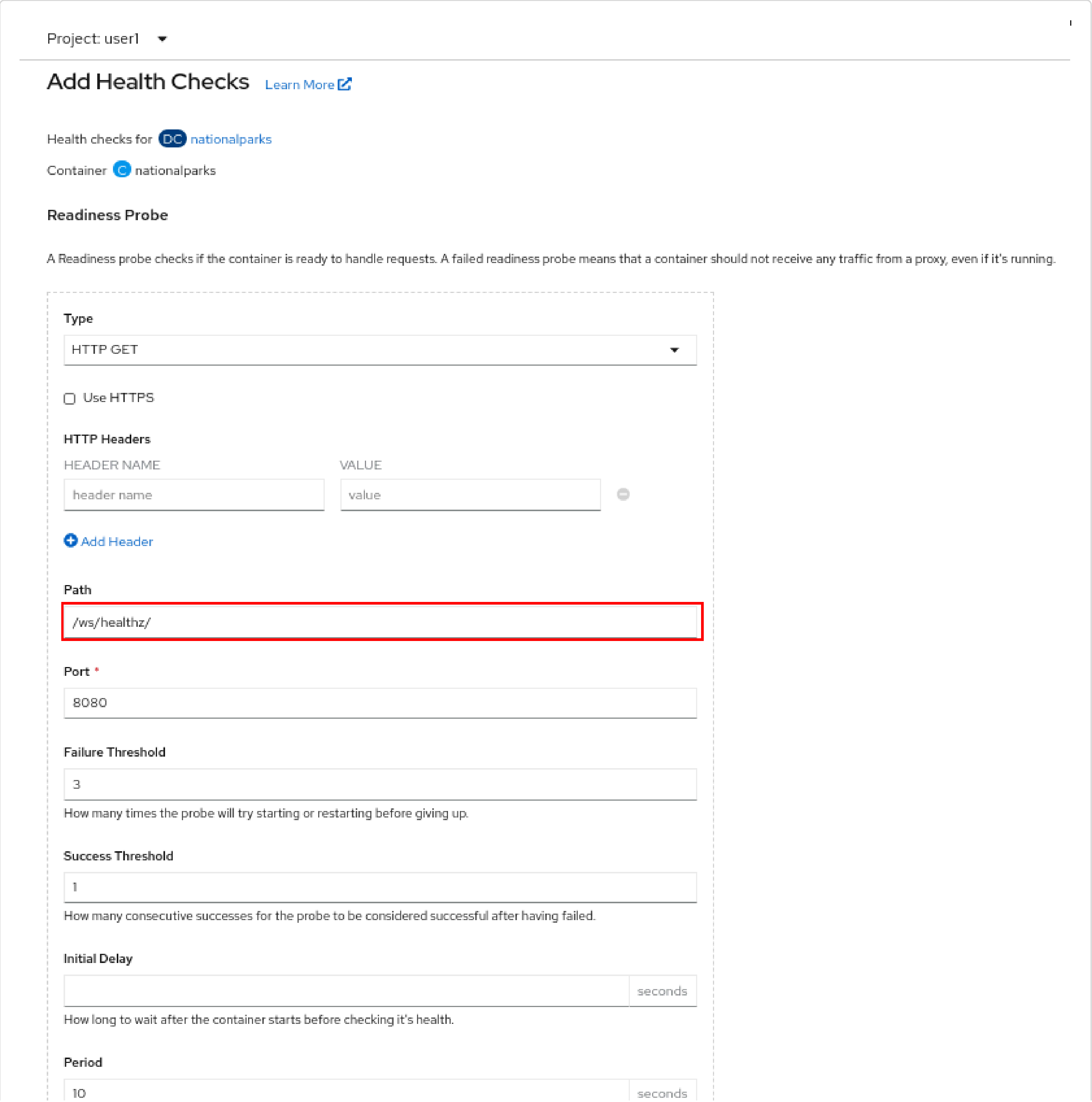
From the **Topology** view, click `nationalparks`. On the side panel, click the **Actions** dropdown menu and the select **Add Health Checks**.



Click to **Add Readiness Probe** and add in **Path** field:

```
/ws/healthz/
```

Leave all default settings like **Port** 8080 and **Type** HTTP GET. Click the little bottom-right confirmation gray tick to confirm:



Project: user1 ▾

Add Health Checks

Learn More

Health checks for DC nationalparks

Container C nationalparks

Readiness Probe

A Readiness probe checks if the container is ready to handle requests. A failed readiness probe means that a container should not receive any traffic from a proxy, even if it's running.

Type

HTTP GET ▾

□ Use HTTPS

HTTP Headers

HEADER NAME	VALUE
header name	value

Path

/ws/healthz/

Port *

8080

Failure Threshold

3

How many times the probe will try starting or restarting before giving up.

Success Threshold

1

How many consecutive successes for the probe to be considered successful after having failed.

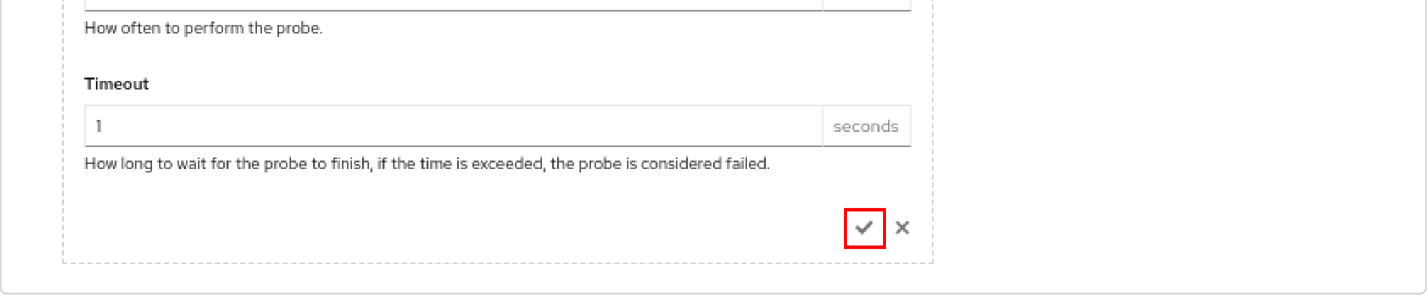
Initial Delay

seconds

How long to wait after the container starts before checking it's health.

Period

10 seconds



How often to perform the probe.

Timeout

1 seconds

How long to wait for the probe to finish, if the time is exceeded, the probe is considered failed.

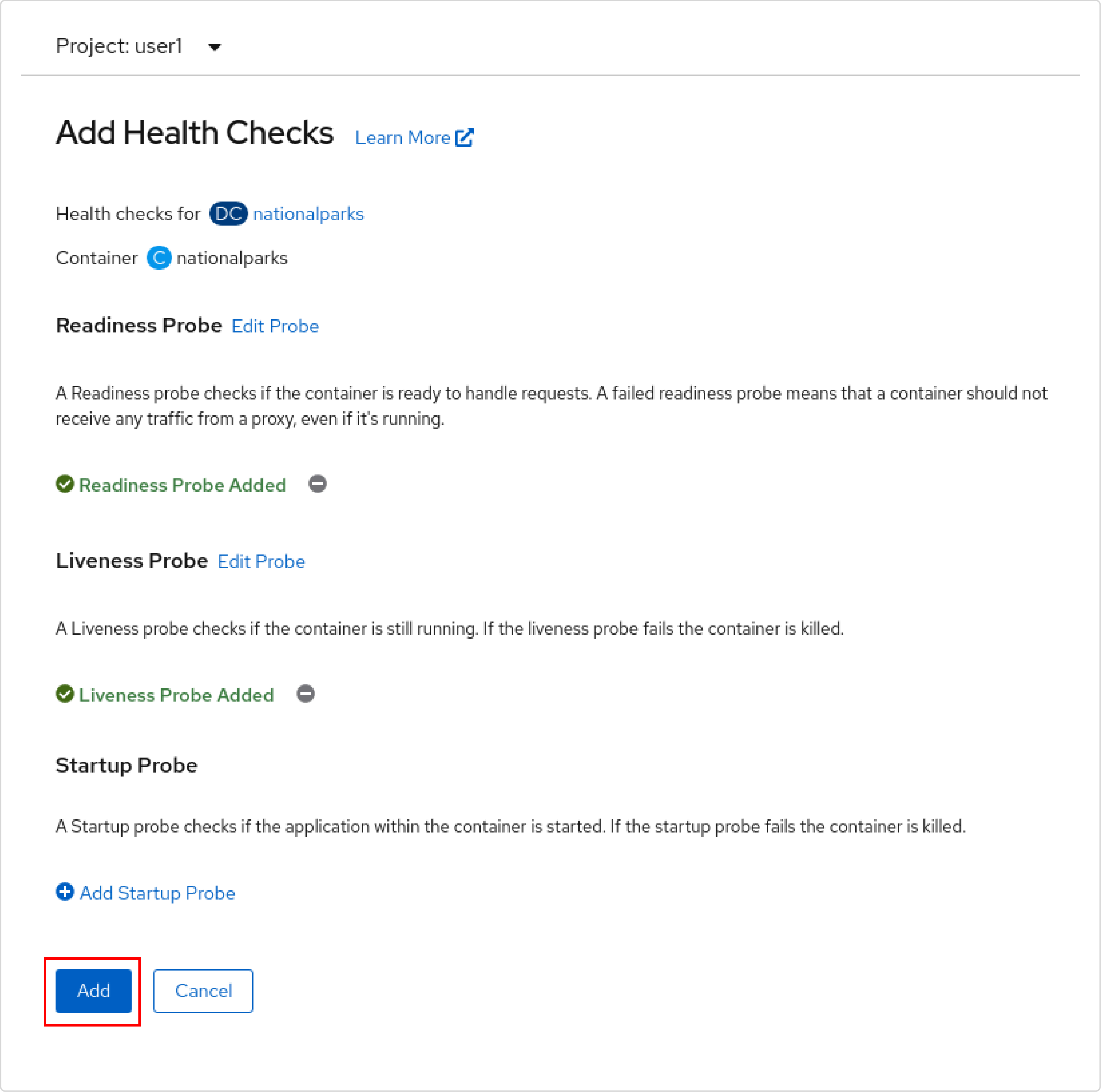
Repeat the same procedure for Liveness Probe, click to **Add Liveness Probe** and add in **Path** field:

```
/ws/healthz/
```

Leave all default settings like **Port** 8080 and **Type** HTTP GET. Click the little bottom-right confirmation gray tick to confirm.

> Note the trailing slash in the URL

Finally confirm all new changes clicking to **Add**:



Add Health Checks Learn More

Health checks for DC nationalparks

Container C nationalparks

Readiness Probe Edit Probe

A Readiness probe checks if the container is ready to handle requests. A failed readiness probe means that a container should not receive any traffic from a proxy, even if it's running.

✓ Readiness Probe Added ⊖

Liveness Probe Edit Probe

A Liveness probe checks if the container is still running. If the liveness probe fails the container is killed.

✓ Liveness Probe Added ⊖

Startup Probe

A Startup probe checks if the application within the container is started. If the startup probe fails the container is killed.

⊕ Add Startup Probe

You will notice that these changes caused a new deployment — they counted as a configuration change.

Continue