In this lesson, you will learn how to access and understand the available performance information within the Developer Sandbox.

What you need to know

In order to get full benefit from this lesson you need to:

- Understand how Kubernetes is the foundational technology for OpenShift and the Developer Sandbox.
- Understand what a Kubernetes pod is at the architectural and operational levels.
- Know how to access the Developer Sandbox via a browser.
- Know how to use the Topology view within the web console to inspect applications and components running in the Developer Sandbox.

What you will learn

In this lesson you will:

- Install a custom application from the command line that exerts pressure on different resources within the Developer Sandbox.
- Access a variety of metrics and graphs that describe performance information within the Developer Sandbox that's relevant to your application.

Being able to quantitatively observe an application running in OpenShift's performance behavior is important for both system administrations and developers. An application that hogs resources can significantly affect the end-user experience.

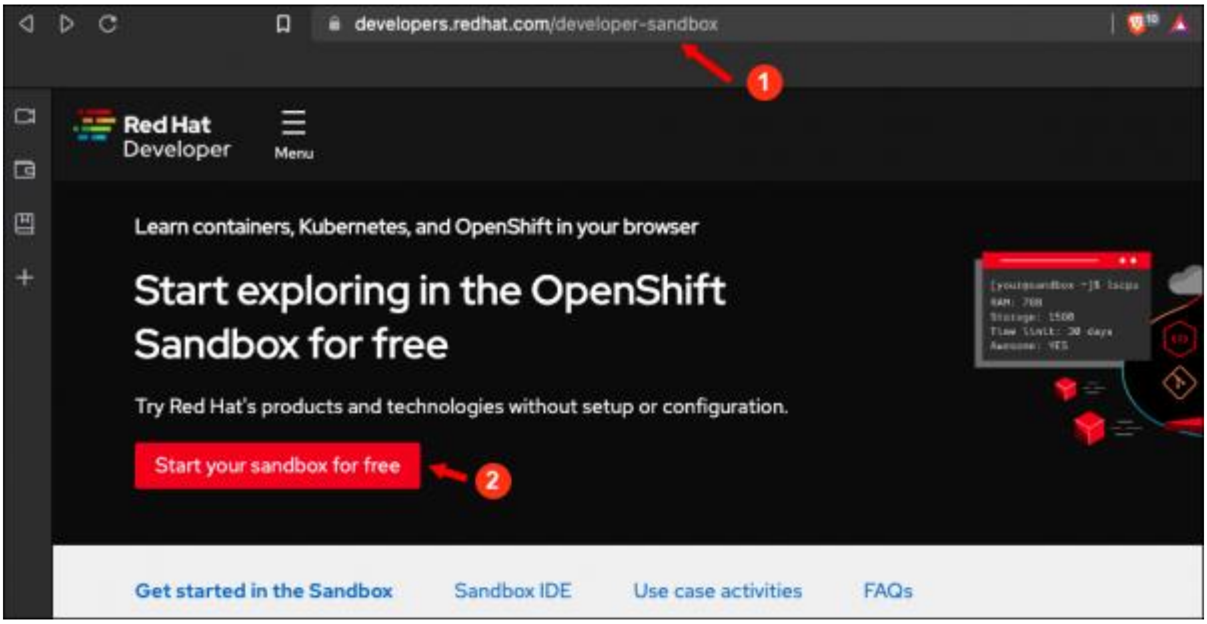
Fortunately, the OpenShift web console publishes a good deal of performance statistics that are available right out of the box. All you need to do to access the information is to know where to look within the user interface (UI). The objective of this lesson is to show you one way to get to that information within the UI.

First, you will use the web console to install a demonstration application that hogs OpenShift resources. Then you will learn how to view information about the application's performance using basic OpenShift reporting features.

Access the OpenShift web console

The following steps will show you how to access the OpenShift web console in the Developer Sandbox. You will do the work of installing the resource pressurizer application using the web console:

1. In your web browser, navigate here and select **Start your sandbox for free** (Figure 1).



developers.redhat.com/developer-sandbox

Learn containers, Kubernetes, and OpenShift in your browser

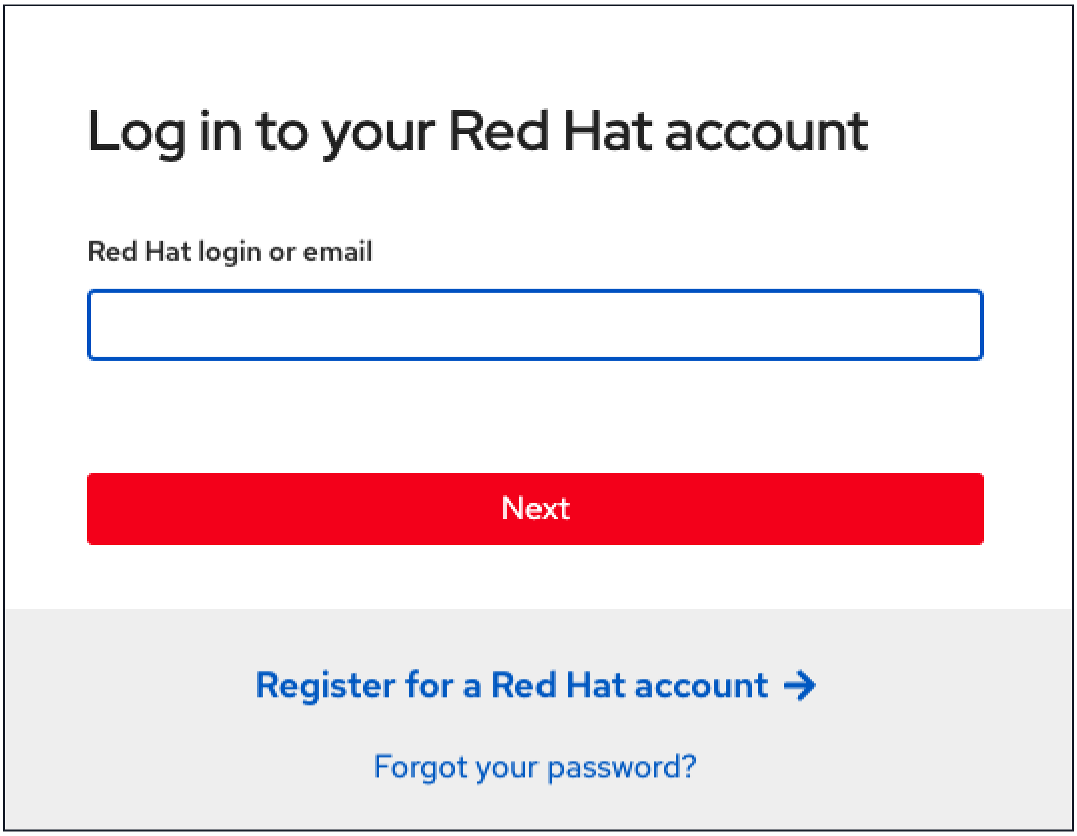
Start exploring in the OpenShift Sandbox for free

Try Red Hat's products and technologies without setup or configuration.

```
[yoursandbox ~]$ lscpu
RAM: 7GB
Storage: 15GB
Time limit: 30 days
Awesome: YES
```

Figure 1: The entry point for access to the Developer Sandbox.

2. Provide the username and password for your Red Hat account (Figure 2).



Log in to your Red Hat account

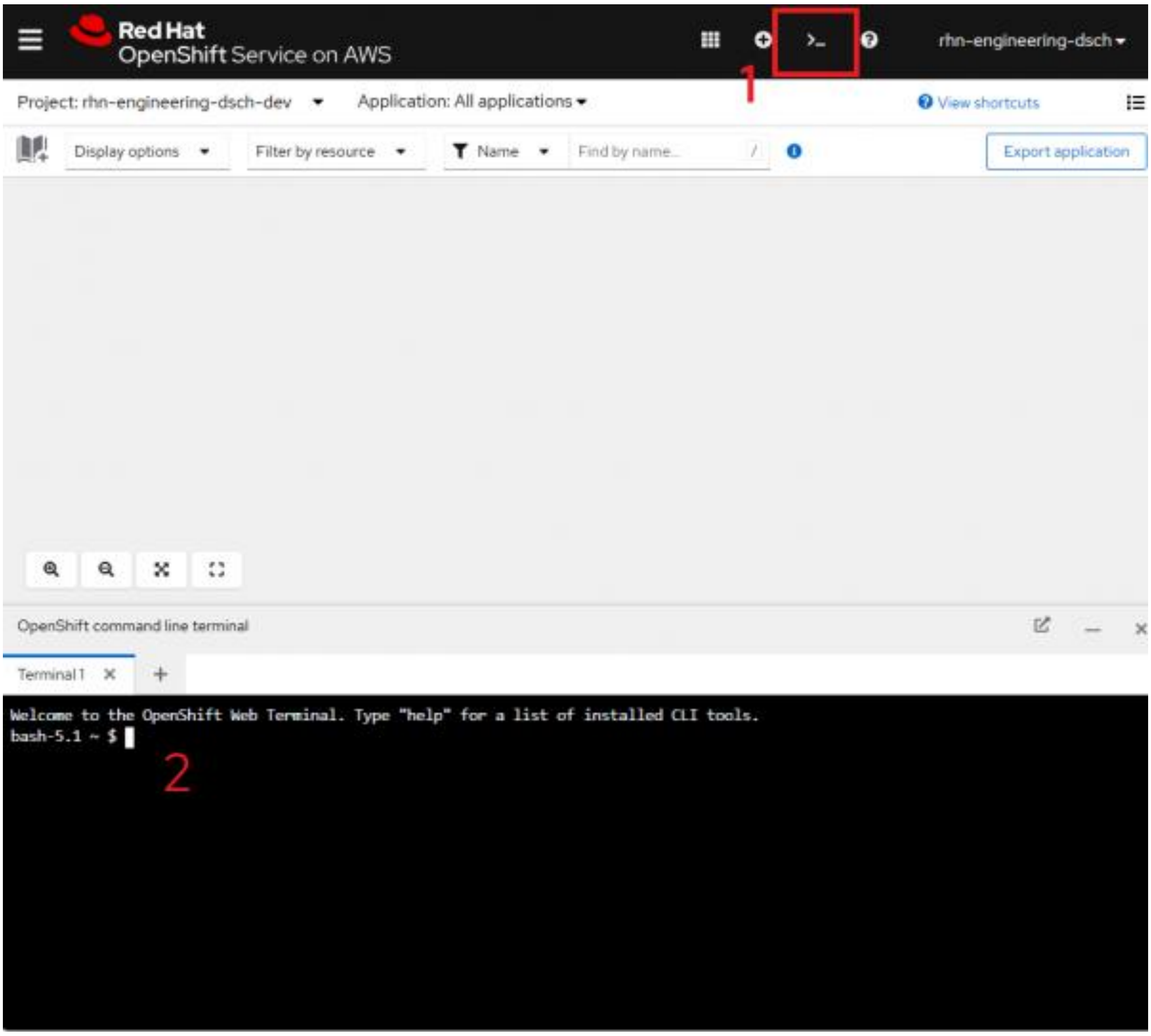
Red Hat login or email

Figure 2: Log into the OpenShift web console with the username and password associated with your account on Red Hat.

Install the demonstration application

The following steps will show you how to install the Node.js demonstration application into the Developer Sandbox. You will install from the command line, using the web console's Web Terminal feature.

1. Click the Web Terminal icon in the upper right part of the web console (Figure 3, Callout 1). The web terminal will open in the bottom part of the screen (Figure 3, Callout 2).



```
Welcome to the OpenShift Web Terminal. Type "help" for a list of installed CLI tools.
bash-5.1 ~ $
```

Figure 3: Open the web terminal.

2. Type the following command at the command prompt, then press Enter:

```
oc create -f https://raw.githubusercontent.com/redhat-developer-demos/resource_pressurizer/refs/heads/main/rpjob.yaml
```

Copy snippet

This will launch a Kubernetes job in your cluster. We're using a job because after it's finished, it stops running. If you simply launch an application that self-terminates, Kubernetes will see that the pod has stopped and will replace it with a new, running, pod. We want to be good citizens of Developer Sandbox and limit our Resource Pressurizer application, so a job is used. The job will spend 90 seconds using both network and CPU resources. After 90 seconds, the network traffic will stop, but the CPU (and file system) usage will continue for another 90 seconds.

3. To continue, choose the Observe option from the menu on the left (Figure 4).

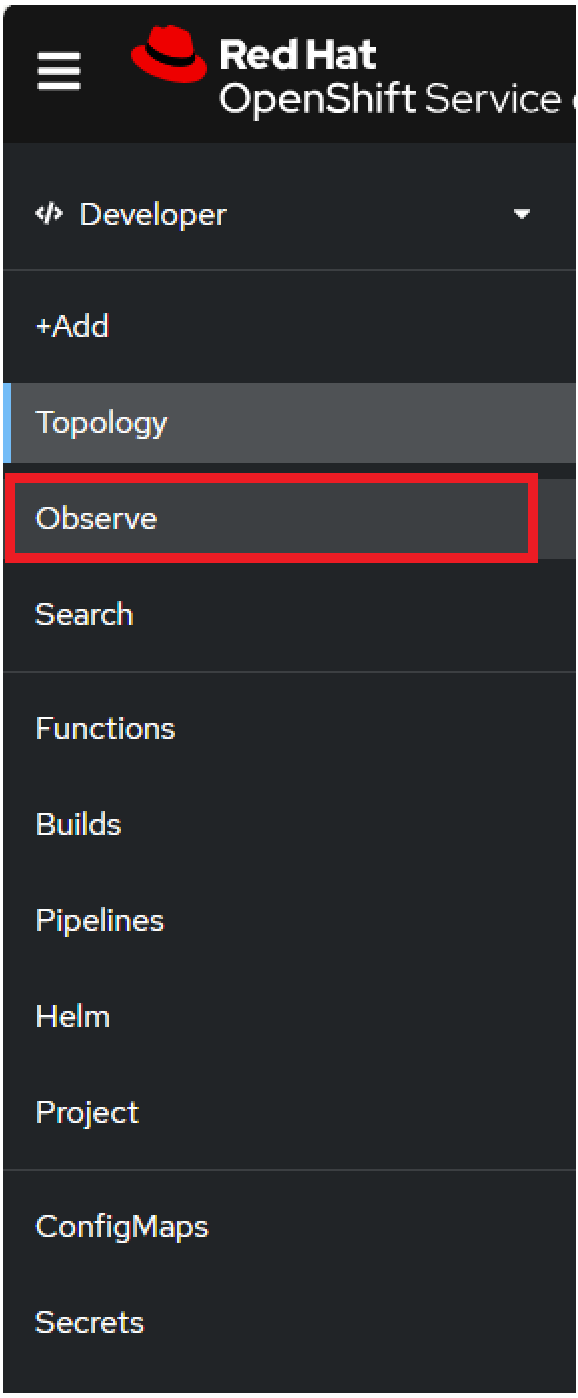
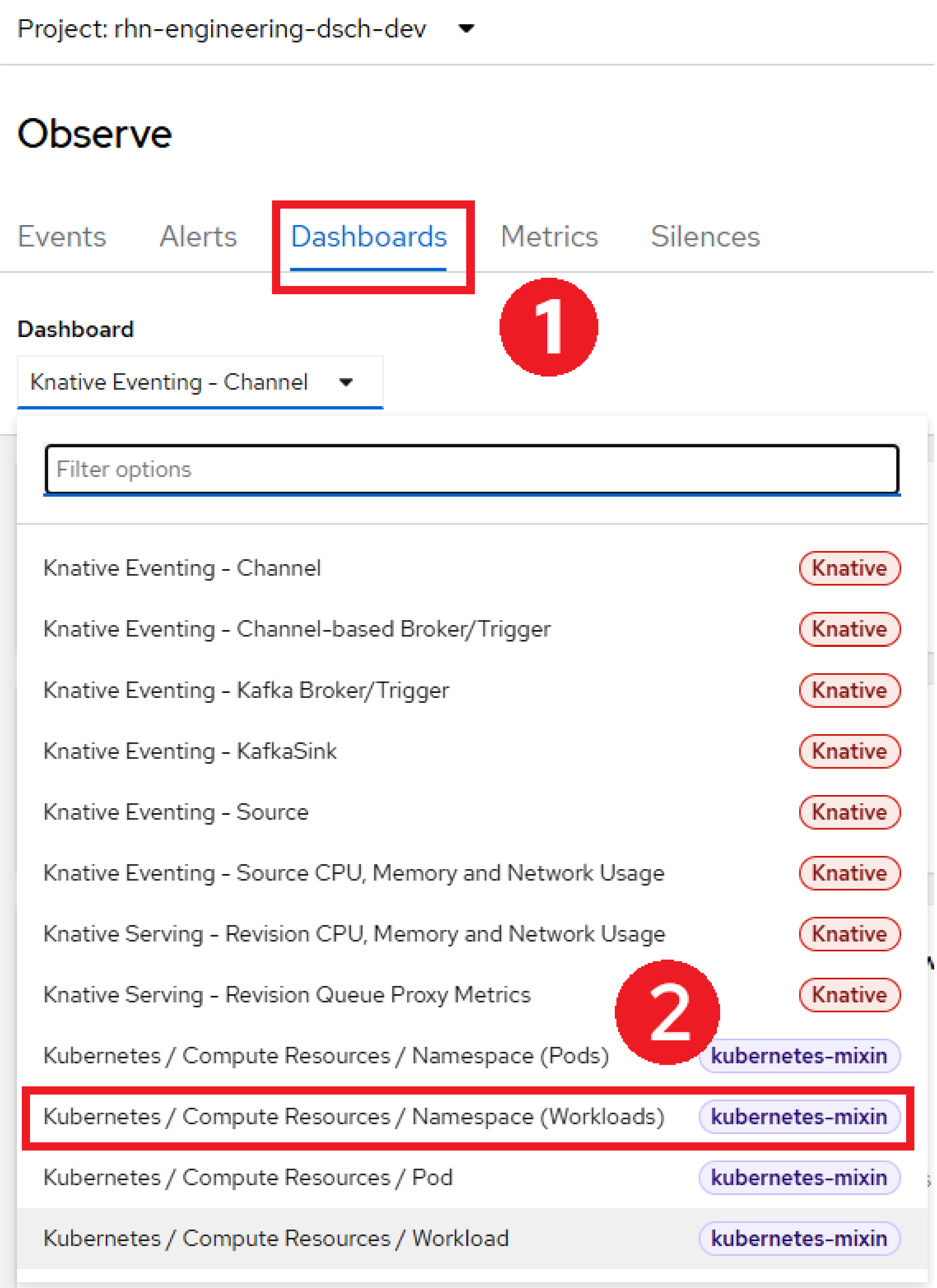


Figure 4: Observe your cluster from the web console.

4. Select the Dashboard tab (Figure 5, Callout 1), then choose the "Kubernetes / Compute Resources / Namespace (Workloads)" option (Figure 5, Callout 2).


Project: rhn-engineering-dsch-dev

Observe

Events Alerts Dashboards Metrics Silences

Dashboard

Knative Eventing - Channel

Filter options

Knative Eventing - Channel	Knative
Knative Eventing - Channel-based Broker/Trigger	Knative
Knative Eventing - Kafka Broker/Trigger	Knative
Knative Eventing - KafkaSink	Knative
Knative Eventing - Source	Knative
Knative Eventing - Source CPU, Memory and Network Usage	Knative
Knative Serving - Revision CPU, Memory and Network Usage	Knative
Knative Serving - Revision Queue Proxy Metrics	Knative
Kubernetes / Compute Resources / Namespace (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Workloads)	kubernetes-mixin
Kubernetes / Compute Resources / Pod	kubernetes-mixin
Kubernetes / Compute Resources / Workload	kubernetes-mixin

Figure 5: Select what you want to observe.

5. After 30 seconds, you will begin to see metric data appear (Figure 6). Continue to watch as the network traffic eventually falls to zero and the CPU usage continues. You may wish to select different viewing options to get a better feel of which metrics are available.

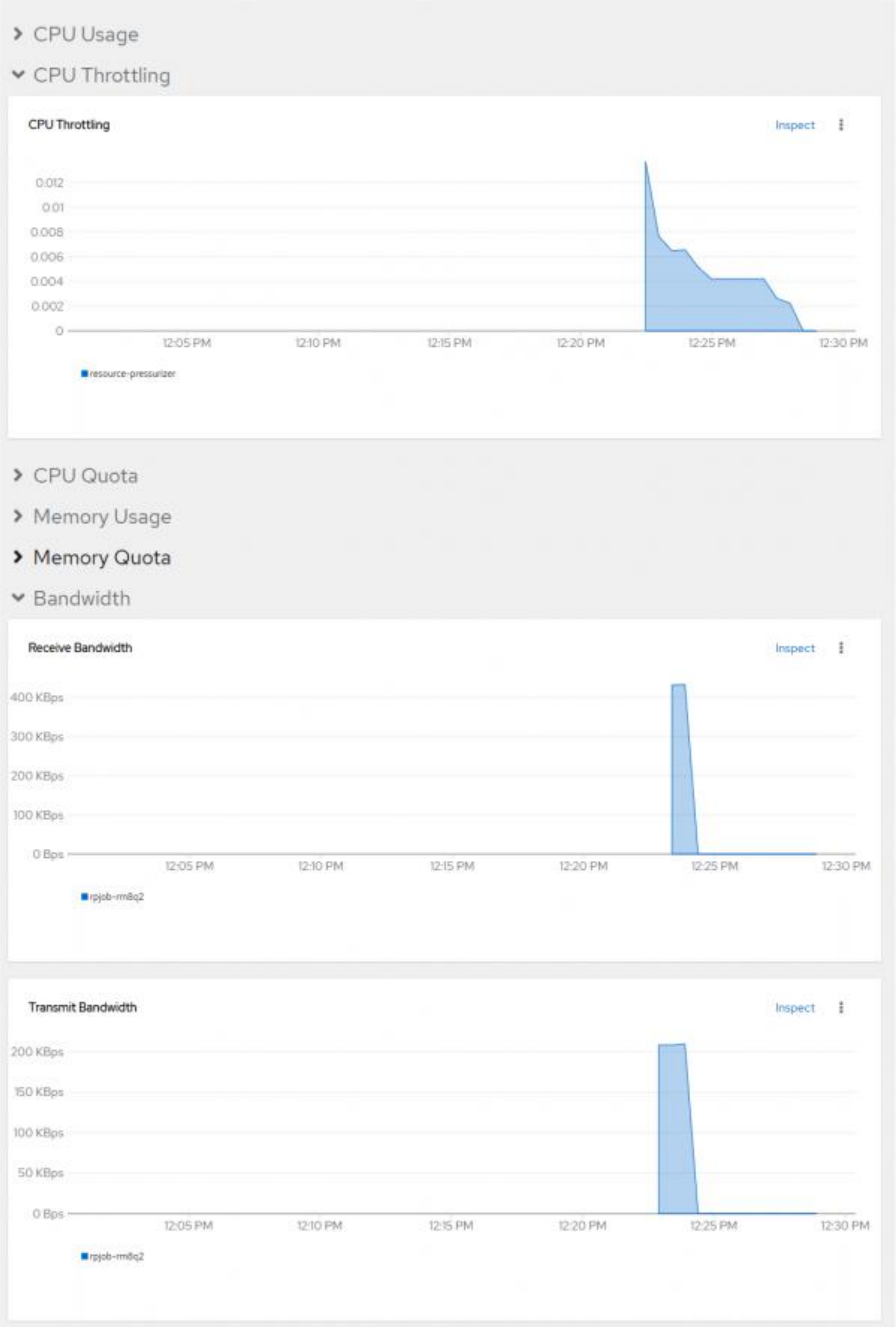


Figure 6: OpenShift provides several metrics.

Summary

In this lesson, you learned how to use the web console to access the graphs and tables that display the performance metrics for a custom application.

Once in the details page, you then learned how to see the graphs and tables associate