Modern Cloud-native Java runtimes performance monitoring on Red Hat Openshift

WORKSHOP MODULES

Monitoring the Applications

Monitoring

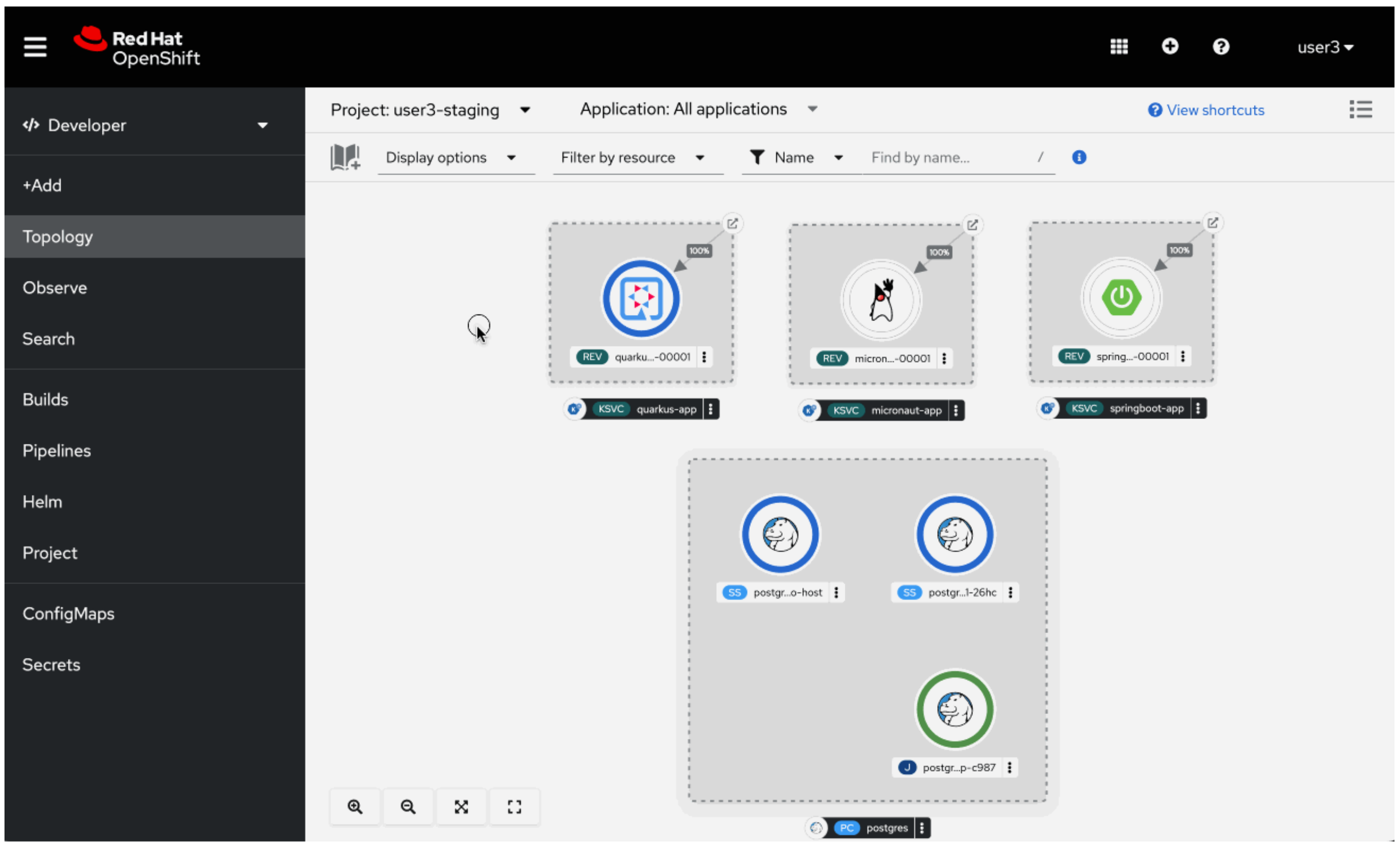
One of the first things you usually want to do once your application is deployed is to configure monitoring. For this, we'll be using the Monitoring capability built in Openshift. This feature comes with a pre-set of ready to use dashboards for monitoring application's workload metrics right from the Openshift Console.

We'll be also using Grafana (https://grafan.com) to visualize metrics provided out-of-the-box by the Openshift Monitoring Stack.

> The OpenShift Monitoring stack is based on the **Prometheus** (https://prometheus.io/) open source project and its wider ecosystem. To learn more about the Openshift Monitoring stack see About OpenShift Container Platform monitoring (https://docs.openshift.com/container-platform/4.12/monitoring/monitoring-overview.html).
>
> **Grafana** is an open platform for beautiful analytics and monitoring. For more information please visit the Grafana website (https://grafana.com/oss/).
>
> Red Hat provides an Openshift Operator to install and manage Grafana instances on your cluster. See Grafana Operator (https://operatorhub.io/operator/grafana-operator) in the Operator Hub for more details.

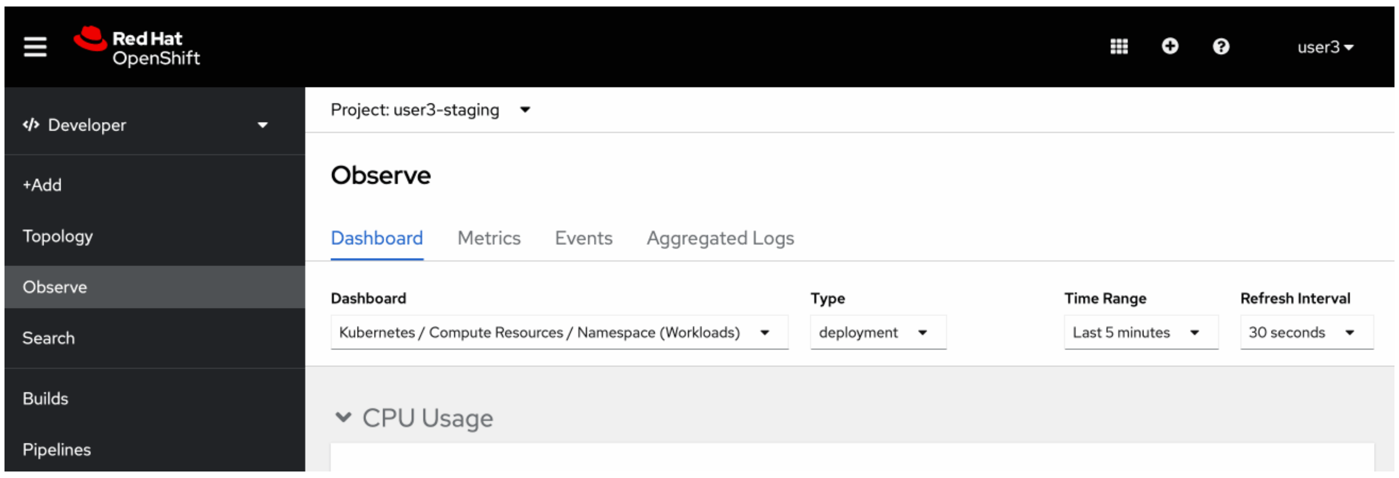
From the Openshift Console, Developer Perspective navigate to the Observe view (https://console-openshift-console.apps.cluster-s9gvd.s9gvd.sandbox1869.opentlc.com/dev-monitoring/ns/user18-staging). From there, select

- **Dashboard**: `Kubernetes / Compute Resources / Namespaces (Workloads)` ; and
- **Type** `Deployment` .



To see more specific data you can drill into the Metric Graph by clicking the Inspect link (top right of the graph). This will open the metric view where you can see the Query (PromoQL) (https://prometheus.io/docs/prometheus/latest/querying/basics/) that brings this metric data point from the integrated **Prometheus Data Source**.

For instance, the Graph bellow shows the CPU usage for the `quarkus-app` for the last 5min.



Dashboard: Kubernetes / Compute Resources / Namespace (Workloads)
Type: deployment
Time Range: Last 5 minutes
Refresh Interval: 30 seconds