Monitor an OpenShift Cluster

Objectives

- Navigate the Events, Compute, and Observe panels of the OpenShift web console to assess the overall state of a cluster.

Overview of Nodes, Machines, and Machine Configurations

In Kubernetes, a *node* is any single system in the cluster where pods can run. These systems are any of the bare metal, virtual, or cloud computers that are members of the cluster. Nodes run the necessary services to communicate within the cluster, and receive control plane operational requests. When you deploy a pod, an available node is tasked with satisfying the request.

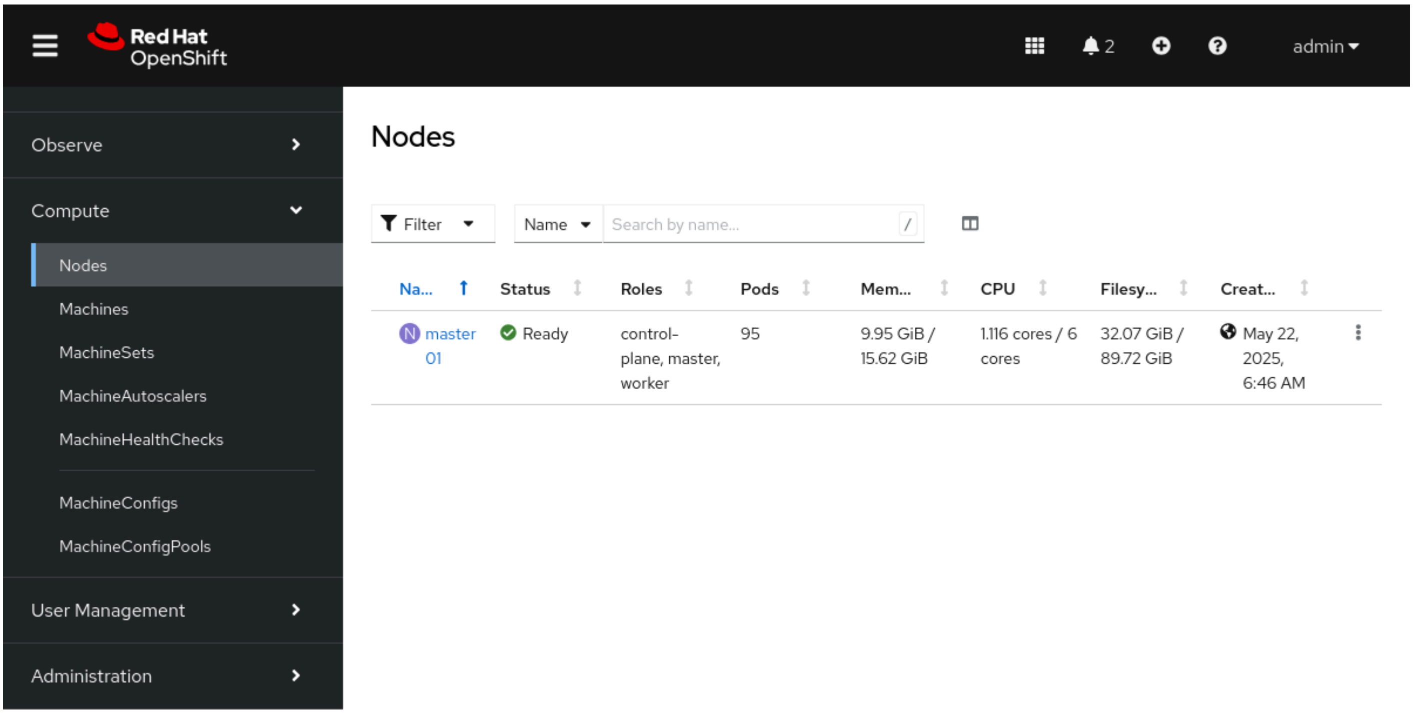
Whereas the *node* and *machine* terms are often interchangeable, Red Hat OpenShift Container Platform (RHOCP) uses the *machine* term more specifically. In OpenShift, a *machine* is the resource that describes a cluster node. Using a machine resource is particularly valuable when using public cloud providers to provision infrastructure.

A `MachineConfig` resource defines the initial state and any changes to files, services, operating system updates, and critical OpenShift service versions for the `kubelet` and `cri-o` services. OpenShift relies on the Machine Config Operator (MCO) to maintain the operating systems and configuration of the cluster machines. The MCO is a cluster-level operator that ensures the correct configuration of each machine. This operator also performs routine administrative tasks, such as system updates. This operator uses the machine definitions in a `MachineConfig` resource to continually validate and remediate the state of cluster machines to the intended state. After a `MachineConfig` change, the MCO orchestrates the execution of the changes for all affected nodes.

> **NOTE**
>
> The orchestration of `MachineConfig` changes through the MCO is prioritized alphabetically by zone, by using the `topology.kubernetes.io/zone` node label.

Identifying Errors from Nodes

Administrators routinely view the logs and connect to the nodes in the cluster by using a terminal. This technique is necessary to manage a cluster and to remediate issues that arise. From the web console, go to **Compute → Nodes** to view the list of all nodes in the cluster.

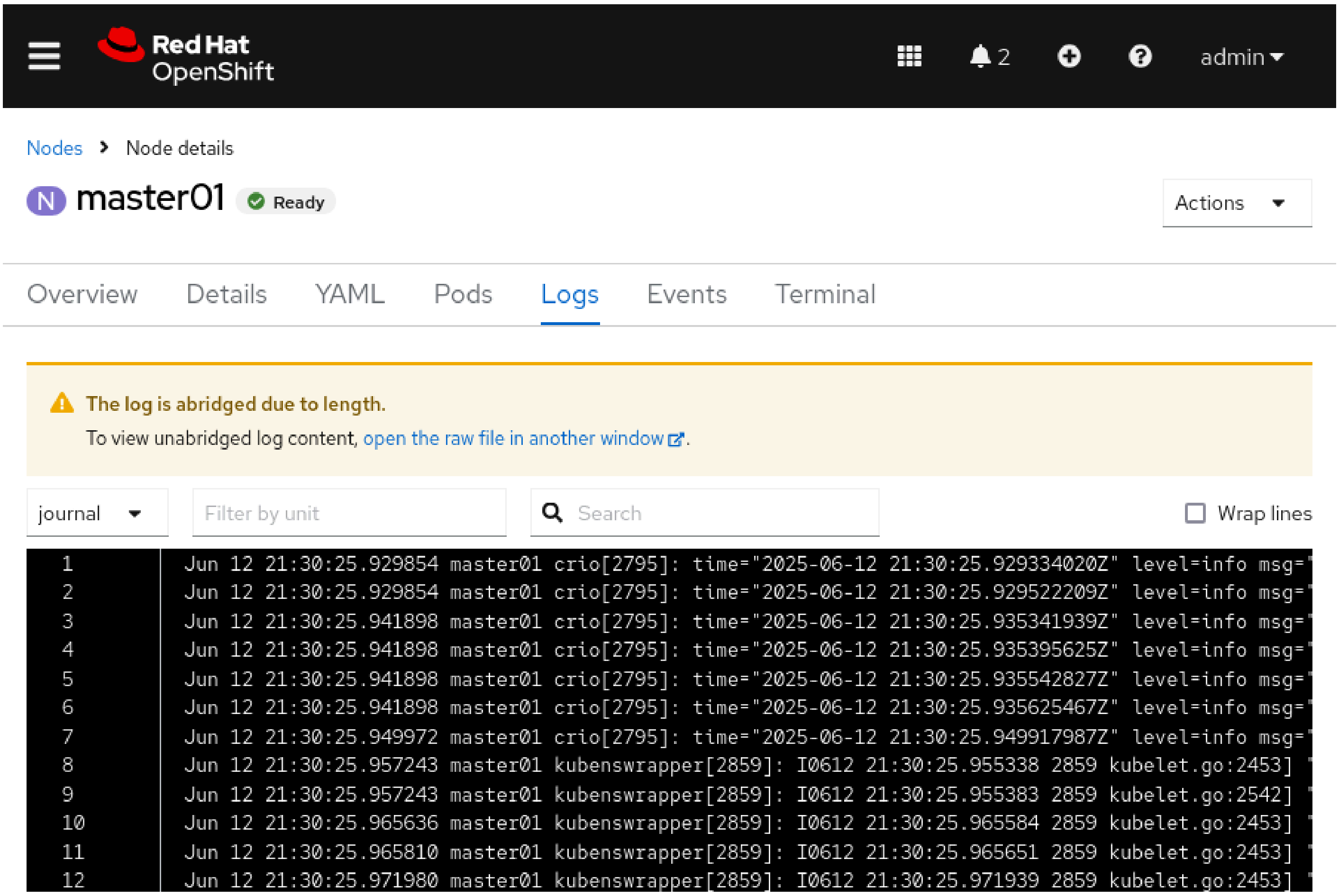


Nodes

Na...	Status	Roles	Pods	Mem...	CPU	Filesy...	Creat...
N master01	Ready	control-plane, master, worker	95	9.95 GiB / 15.62 GiB	1.116 cores / 6 cores	32.07 GiB / 89.72 GiB	May 22, 2025, 6:46 AM

Figure 1.40: Node list in the web console

Click a node's name to go to the overview page for the node. On the node overview page, you can view the node logs or connect to the node by using the terminal.



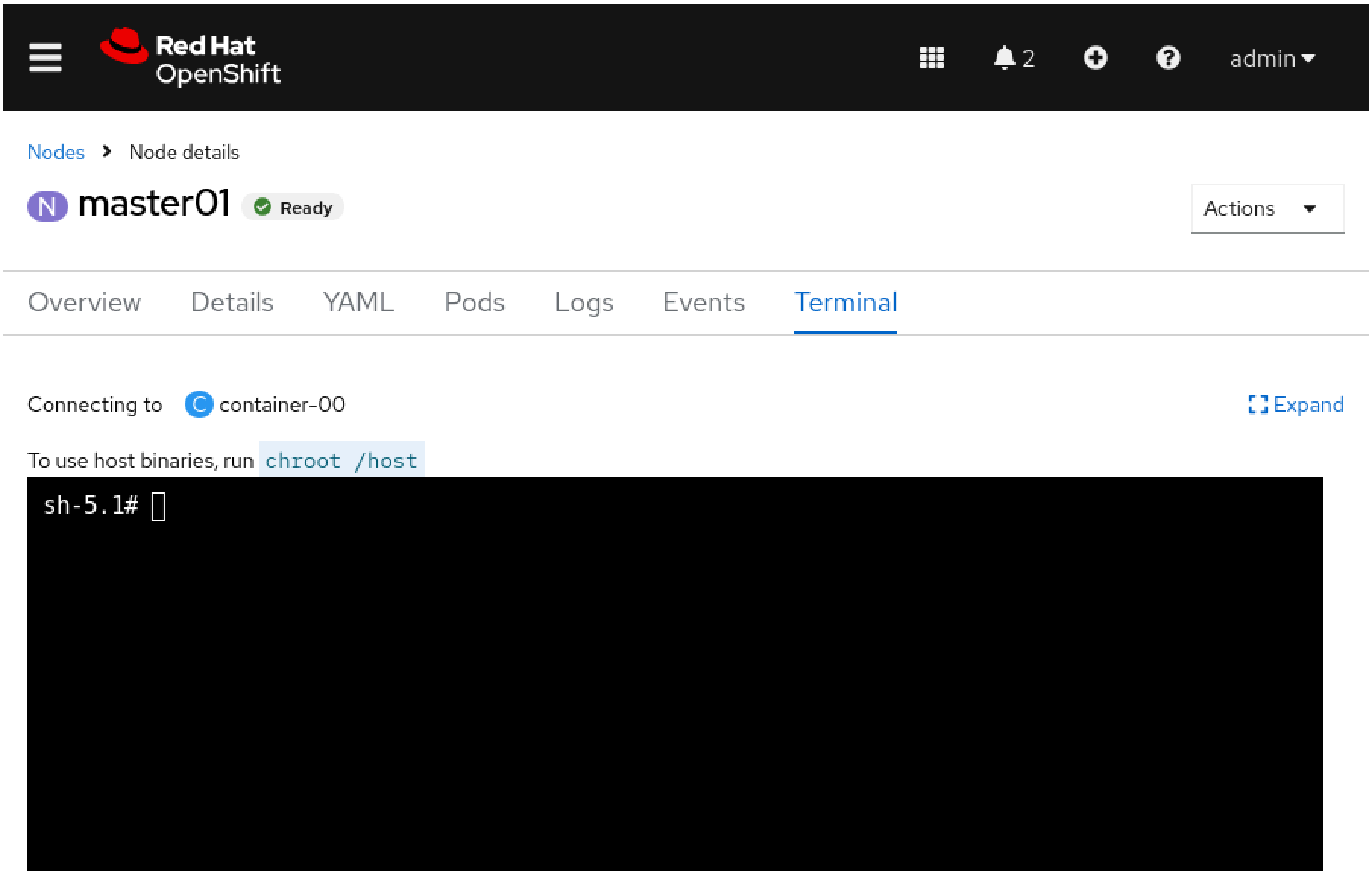
The log is abridged due to length.

To view unabridged log content, open the raw file in another window.

```
Jun 12 21:30:25.929854 master01 crio[2795]: time="2025-06-12 21:30:25.929334020Z" level=info msg="
Jun 12 21:30:25.929854 master01 crio[2795]: time="2025-06-12 21:30:25.929522209Z" level=info msg="
Jun 12 21:30:25.941898 master01 crio[2795]: time="2025-06-12 21:30:25.935341939Z" level=info msg="
Jun 12 21:30:25.941898 master01 crio[2795]: time="2025-06-12 21:30:25.935395625Z" level=info msg="
Jun 12 21:30:25.941898 master01 crio[2795]: time="2025-06-12 21:30:25.935542827Z" level=info msg="
Jun 12 21:30:25.941898 master01 crio[2795]: time="2025-06-12 21:30:25.935625467Z" level=info msg="
Jun 12 21:30:25.949972 master01 crio[2795]: time="2025-06-12 21:30:25.949917987Z" level=info msg="
Jun 12 21:30:25.957243 master01 kubenswrapper[2859]: I0612 21:30:25.955338 2859 kubelet.go:2453] "
Jun 12 21:30:25.957243 master01 kubenswrapper[2859]: I0612 21:30:25.955383 2859 kubelet.go:2542] "
Jun 12 21:30:25.965636 master01 kubenswrapper[2859]: I0612 21:30:25.965584 2859 kubelet.go:2453] "
Jun 12 21:30:25.965810 master01 kubenswrapper[2859]: I0612 21:30:25.965651 2859 kubelet.go:2453] "
Jun 12 21:30:25.971980 master01 kubenswrapper[2859]: I0612 21:30:25.971939 2859 kubelet.go:2453] "
```

Figure 1.41: Logs page in the web console

From the previous page in the web console, view the node logs and investigate the system information to aid troubleshooting and remediation for node issues.


Red Hat OpenShift
2
admin
Nodes > Node details
N master01 Ready
Actions
Overview Details YAML Pods Logs Events Terminal
Connecting to C container-00
Expand

To use host binaries, run `chroot /host`

```
sh-5.1#
```

Figure 1.42: Terminal shell in the web console

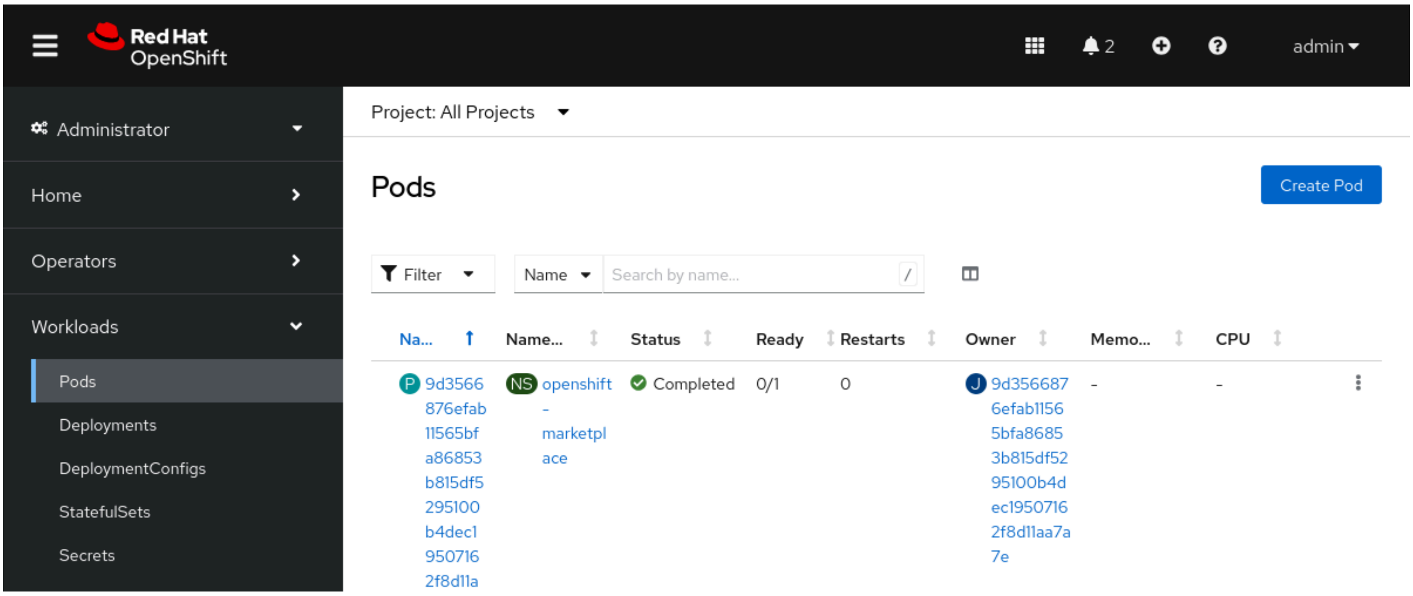
The preceding page shows the web console terminal that is connected to the cluster node. From this tab, you can access the debug pod and use the commands from the host binaries to view the status of the node's services. An OpenShift node debug pod is an interface to a container that runs on the node.

Although making changes directly on the cluster node from the terminal is not recommended, it is common practice to connect to the cluster node for diagnostic investigation and remediation. From this terminal, you can use the same binaries that are available within the cluster node itself.

Additionally, the tabs on the node overview page show metrics, events, and the node's YAML definition file.

Accessing Pod Logs

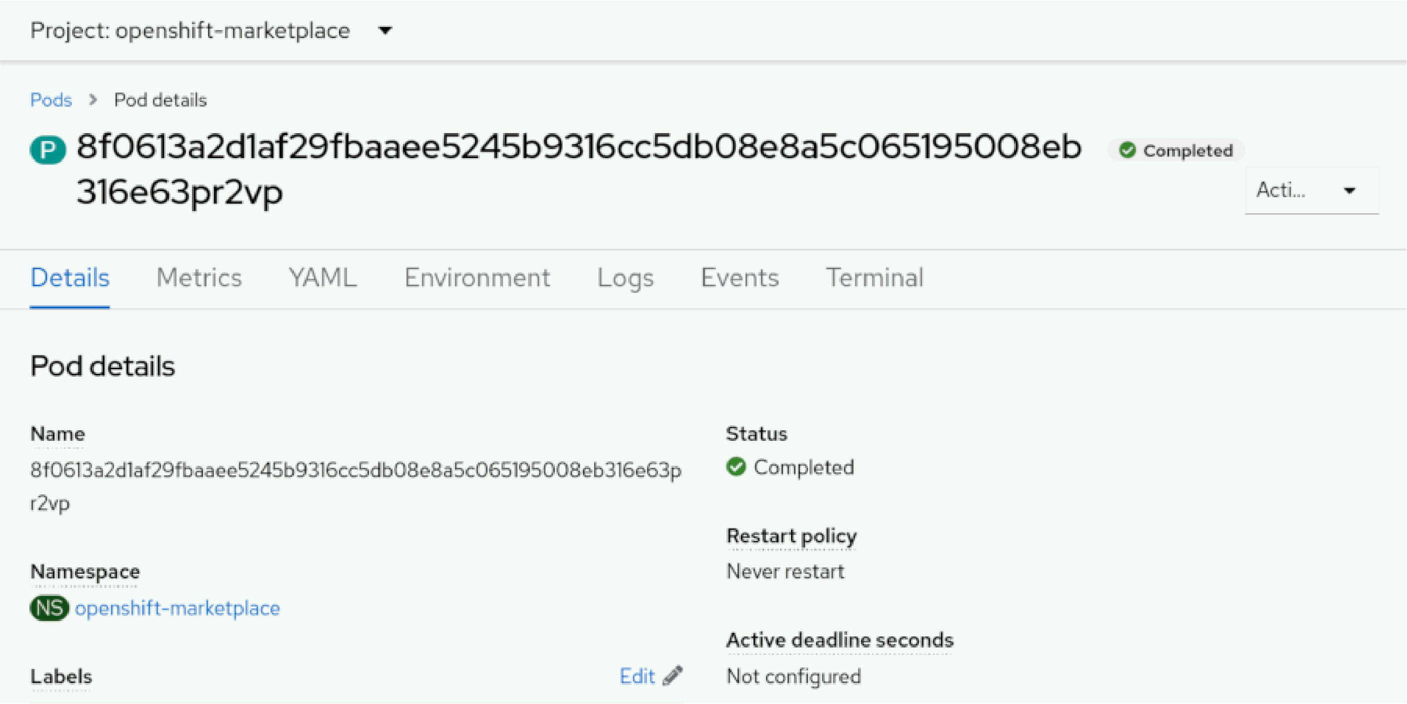
Administrators often peruse pod logs to assess the health of a deployed pod or to troubleshoot pod deployment issues. Go to the **Workloads** → **Pods** page to view the list of all pods in the cluster.



Na...	Name...	Status	Ready	Restarts	Owner	Memo...	CPU
P 9d3566876efab11565bfa86853b815df5295100b4dec1950716 2f8d11a	NS openshift-marketplace	Completed	0/1	0	J 9d3566876efab11565bfa86853b815df5295100b4dec1950716 2f8d11aa7a7e	-	-

Figure 1.43: Pods page in the web console

You can filter and order pods by project and by other fields. To view the pod details page, click a pod name in the list.



P 8f0613a2d1af29fbaaee5245b9316cc5db08e8a5c065195008eb316e63pr2vp Completed

Pod details

Name
8f0613a2d1af29fbaaee5245b9316cc5db08e8a5c065195008eb316e63pr2vp

Namespace
NS openshift-marketplace

Labels

Status
Completed

Restart policy
Never restart

Active deadline seconds
Not configured

Figure 1.44: Pod details page

The pod details page contains links to pod metrics, environment variables, logs, events, a terminal, and the pod's YAML definition. The pod logs are available on the **Pods** → **Logs** page and provide information about the pod status. The **Pods** → **Terminal** page opens a shell connection to the pod for inspection and issue remediation. Although it is not recommended to alter a running pod, the terminal is useful for diagnosing and remediating pod issues. To fix a pod, update the pod configuration to reflect the necessary changes, and redeploy the pod.

Red Hat OpenShift Container Platform Metrics and Alerts

In an RHOCP cluster, HTTP service endpoints provide data metrics that are collected to provide information for monitoring cluster and application performance. These metrics are authored at the application level for each service by using the client libraries that are provided by Prometheus, an open source monitoring and alerting toolkit. Metrics data is available from the service `/metrics` endpoint. You can use the data for creating monitors to alert based on degradation of the service. Monitors are processes that continuously assess the value for a specific metric and provide alerts that are based on a predefined condition, to signal a degradation in the service or a performance issue. Authoring a `ServiceMonitor` resource defines how a specific service uses the metrics to define a monitor and the alerting values. The same approach is available for monitoring pods by defining a `PodMonitor` resource that uses the metrics that are gathered from the pod.

Depending on the monitor definitions, alerting is then available based on the metric that is polled and the defined success criteria. The monitor continuously compares the gathered metric, and creates an alert when the success criteria are no longer met. As an example, a web service monitor polls on the listening port, port 80, and alerts only if the response from that port becomes invalid.

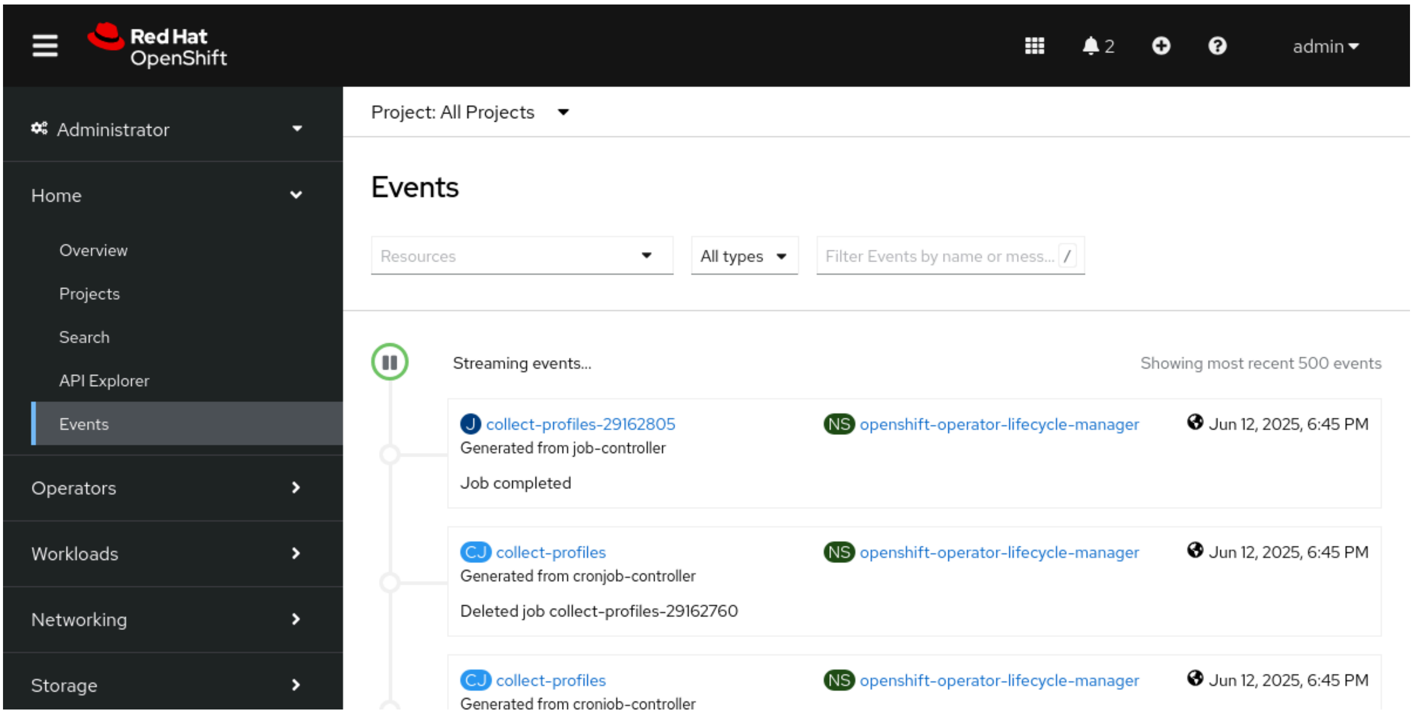
From the web console, go to **Observe** → **Metrics** to visualize gathered metrics by using a Grafana-based data query utility. On this page, users can submit queries to build data graphs and dashboards, which administrators can view to gather valuable statistics for the cluster and applications.

For configured monitors, visit **Observe** → **Alerting** to view firing alerts, and filter on the alert severity to view those alerts that need remediation. Alerting data is a key component to help administrators to deliver cluster and application accessibility and functions.

Kubernetes Events

Administrators are typically familiar with the contents of log files for services, whereas logs tend to be highly detailed and granular. *Events* provide a high-level abstraction to log files and to provide information about more significant changes. Events are useful in understanding the performance and behavior of the cluster, nodes, projects, or pods, at a glance. Events provide details to understand general performance and to highlight meaningful issues. Logs provide a deeper level of detail for remediating specific issues.

The **Home** → **Events** page shows the events for all projects or for a specific project. You can further filter and search events.



J collect-profiles-29162805 NS openshift-operator-lifecycle-manager Jun 12, 2025, 6:45 PM
Generated from job-controller

Job completed

CJ collect-profiles NS openshift-operator-lifecycle-manager Jun 12, 2025, 6:45 PM
Generated from cronjob-controller

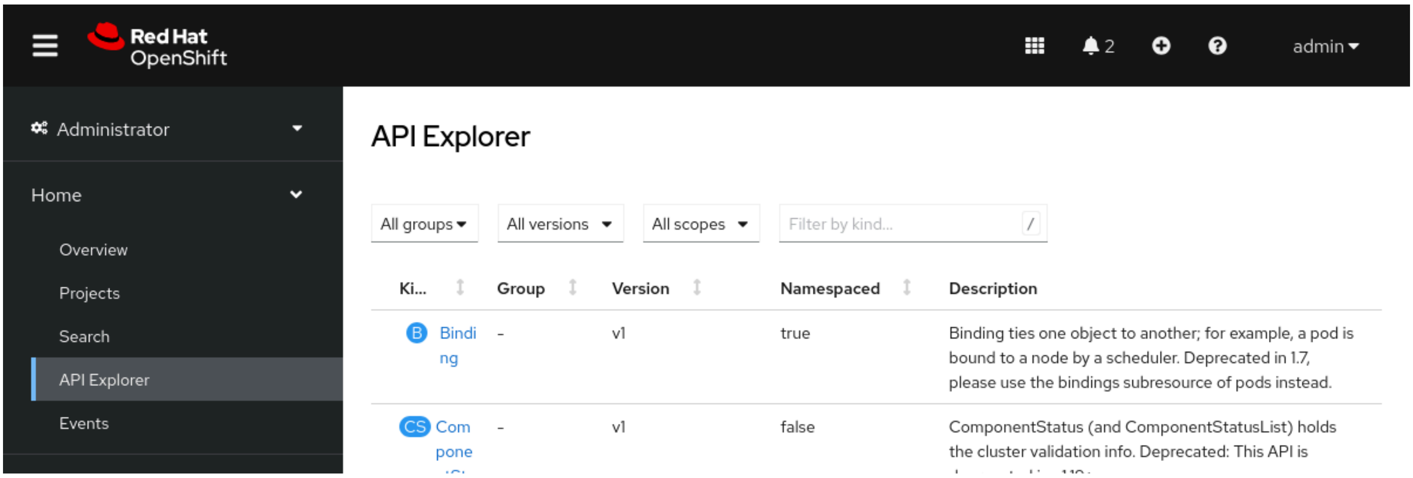
Deleted job collect-profiles-29162760

CJ collect-profiles NS openshift-operator-lifecycle-manager Jun 12, 2025, 6:45 PM
Generated from cronjob-controller

Figure 1.45: RHOCP Events console

Red Hat OpenShift Container Platform API Explorer

Starting from version 4, RHOCP includes the API Explorer feature, for users to view the catalog of Kubernetes resource types that are available within the cluster. By navigating to **Home** → **API Explorer**, you can view and explore the details for resources. Such details include the description, schema, and other metadata for the resource. This feature is helpful for all users, and especially for new administrators.



Ki...	Group	Version	Namespaced	Description
B Binding	-	v1	true	Binding ties one object to another; for example, a pod is bound to a node by a scheduler. Deprecated in 1.7, please use the bindings subresource of pods instead.
CS Compone	-	v1	false	ComponentStatus (and ComponentStatusList) holds the cluster validation info. Deprecated: This API is

Figure 1.46: The RHOCP API Explorer

REFERENCES

For more information about Red Hat OpenShift Container Platform machines, refer to the *Overview of Machine Management* chapter in the Red Hat OpenShift Container Platform 4.18 *Machine Management* documentation at https://docs.redhat.com/en/documentation/openshift_container_platform/4.18/html-single/machine_management/index#overview-of-machine-management

The API Explorer

For more information about the `topology.kubernetes.io/zone` label, refer to Well-known Labels, Annotations and Taints