Lesson 4 : Monitor an OpenShift Cluster

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

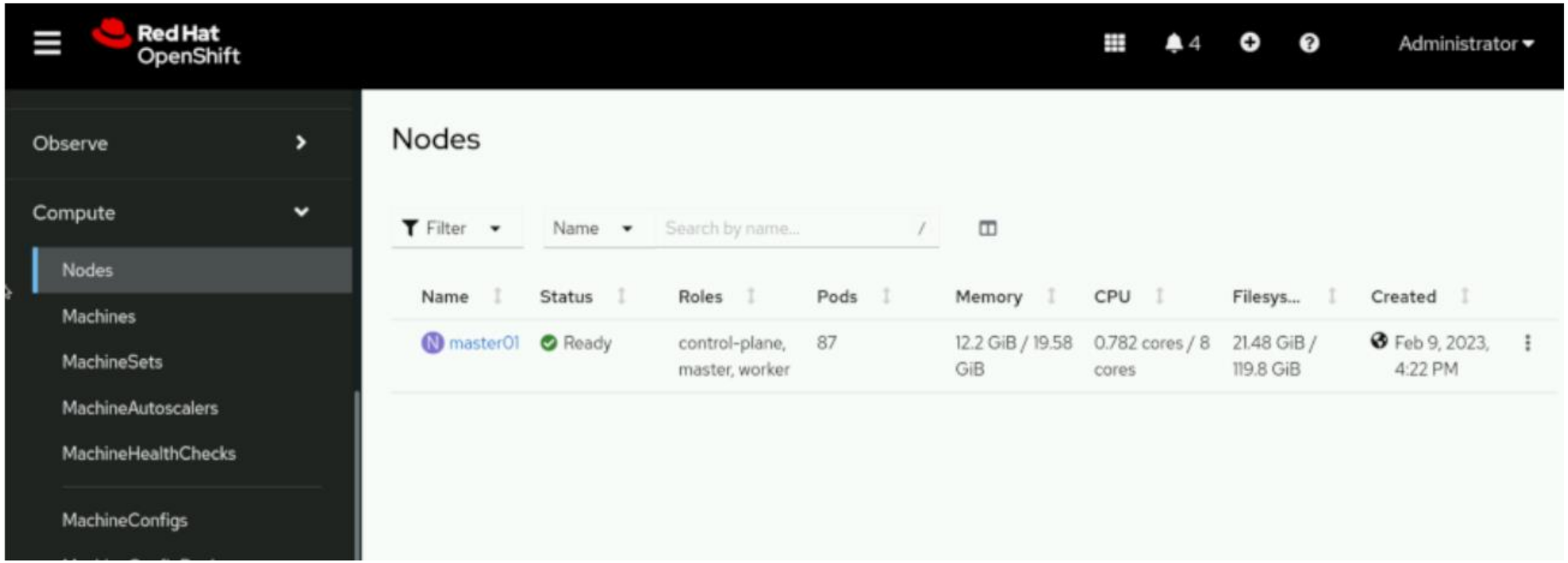
Overview of Nodes, Machines, and Machine Configurations

- Node (in Kubernetes):
  - must be part of cluster
  - single system where pods can run
  - can be bare metal, virtual machine, cloud instances
  - uses kubelet agent to communicate with master node / control plane
- Machine (in OpenShift)
  - to describe cluster node
- MachineConfig
  - defines initial state; including any changes to files, services, OS updates and critical OpenShift service version; particularly : kubelet and cri-o

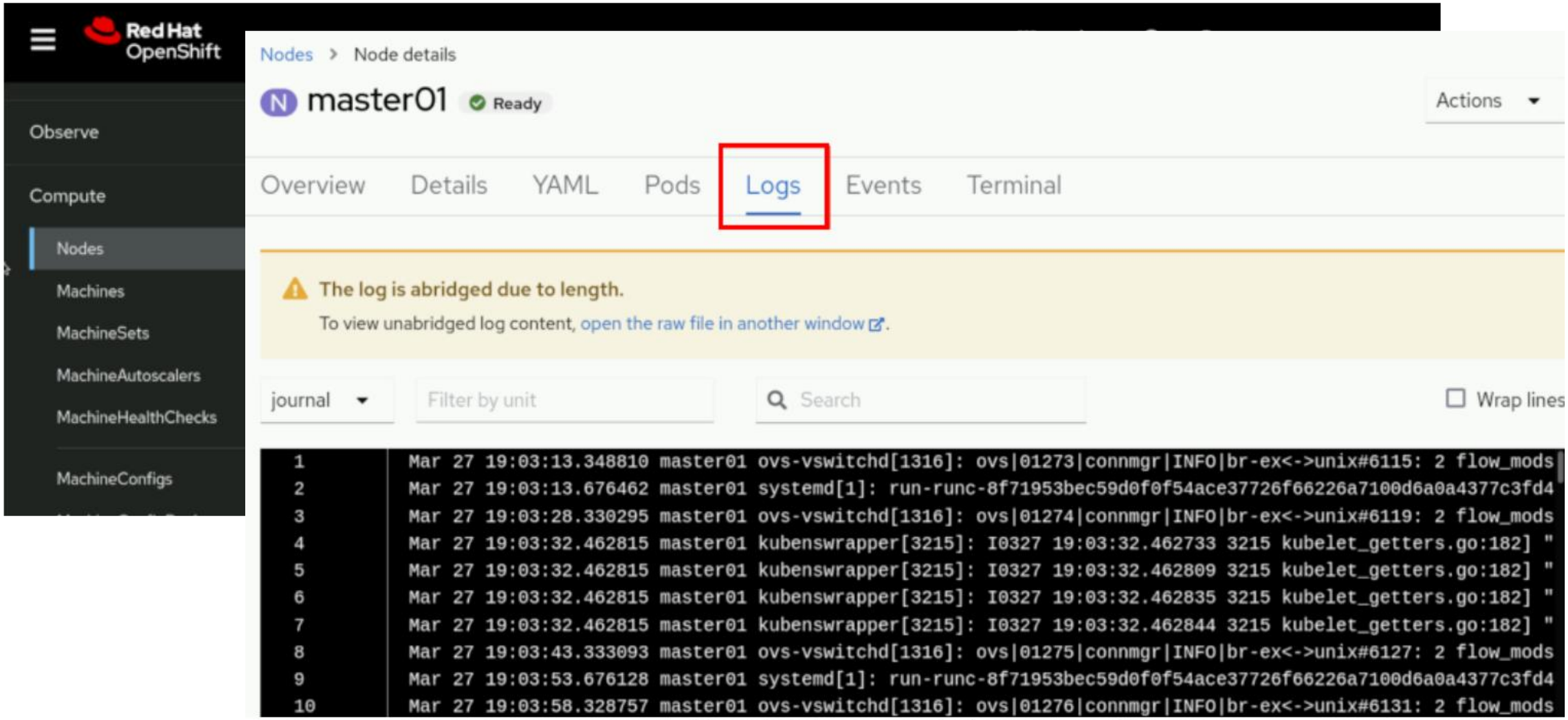
Machine Config Operator (MCO)

- is cluster-level operator
- maintain the OS and configuration of cluster machines
  - system updates, system state
- ensure correct configuration for each machine
- Uses **MachineConfig** resource to continually **validate** and **remediate** state of cluster machine to intended state
- If **MachineConfig** change, MCO orchestrates execution of changes across all cluster machines
- Prioritized alphabetically by zone (in cloud environment)

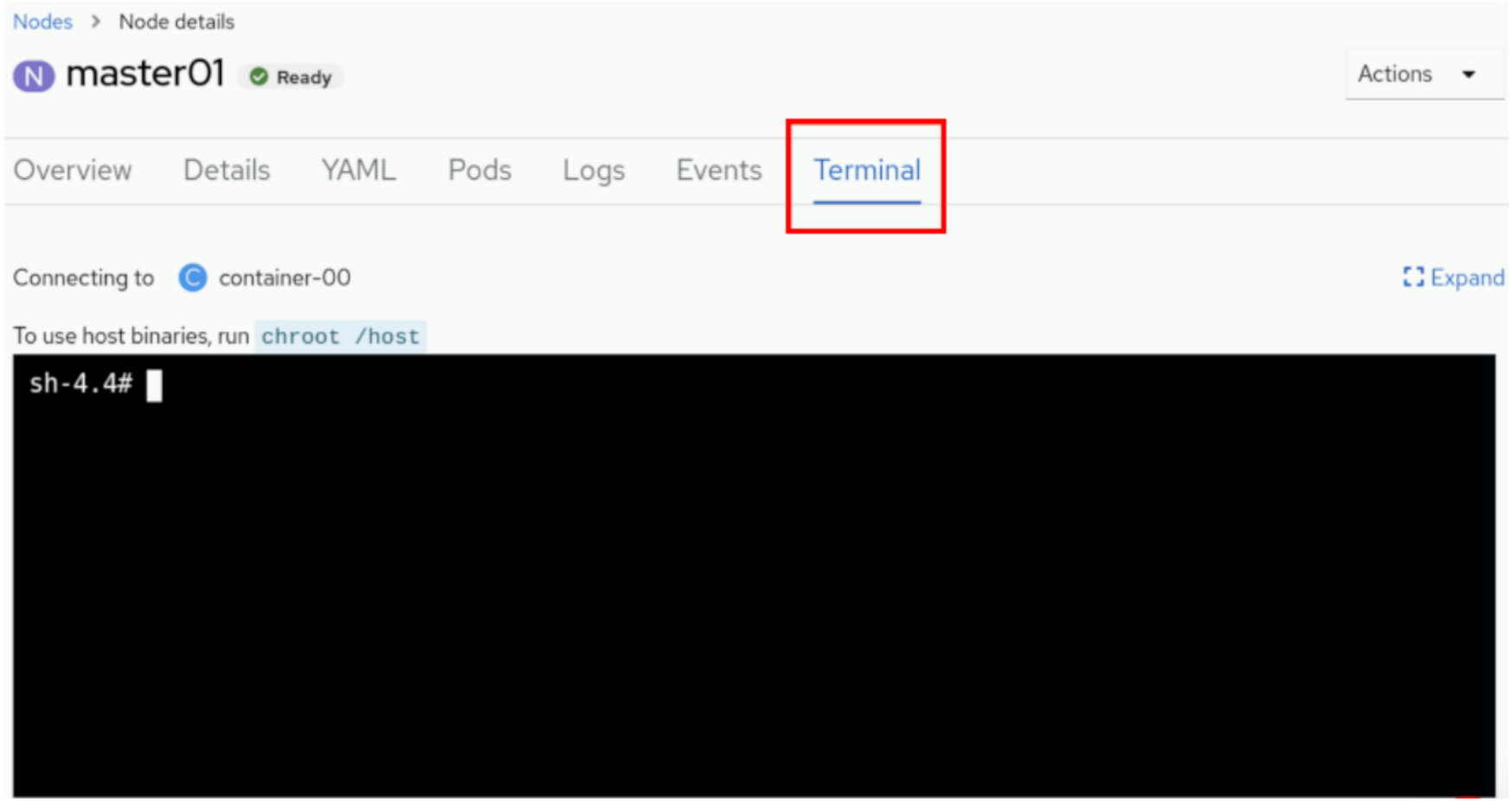
Identify Errors from nodes



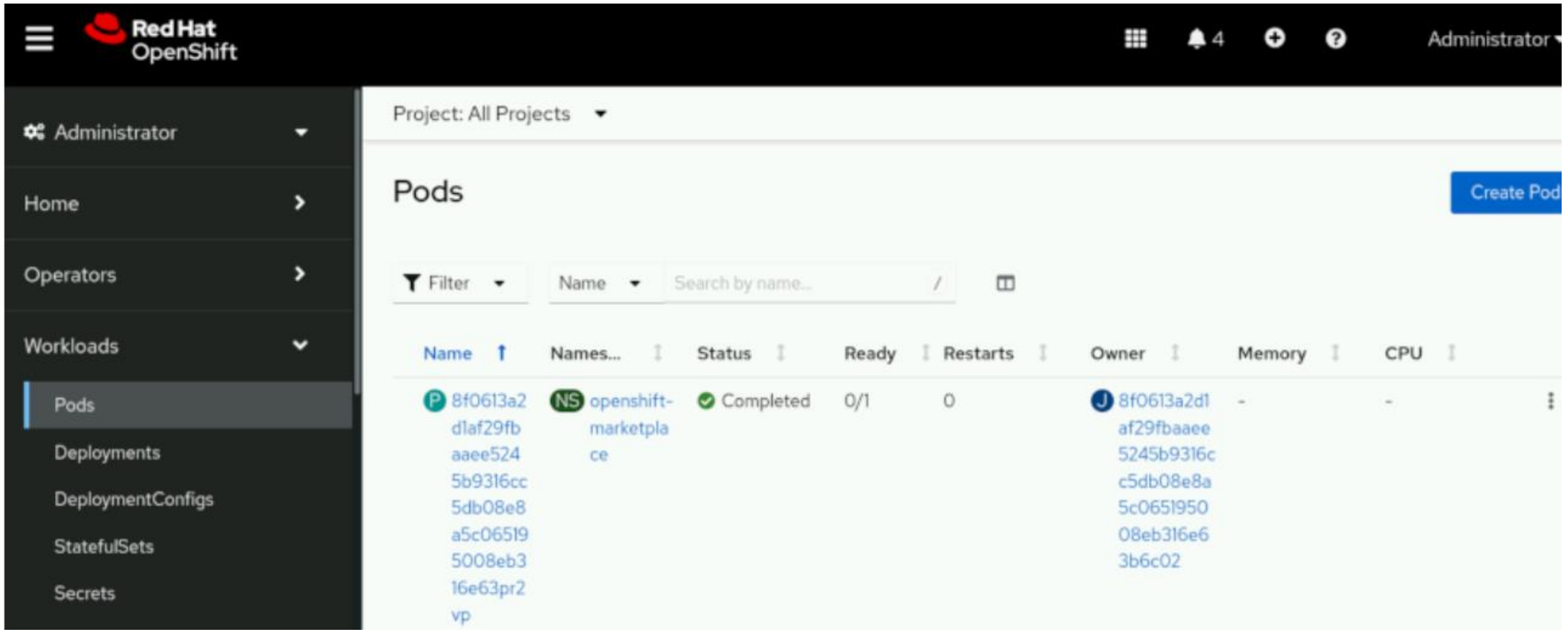
Identify Errors from nodes



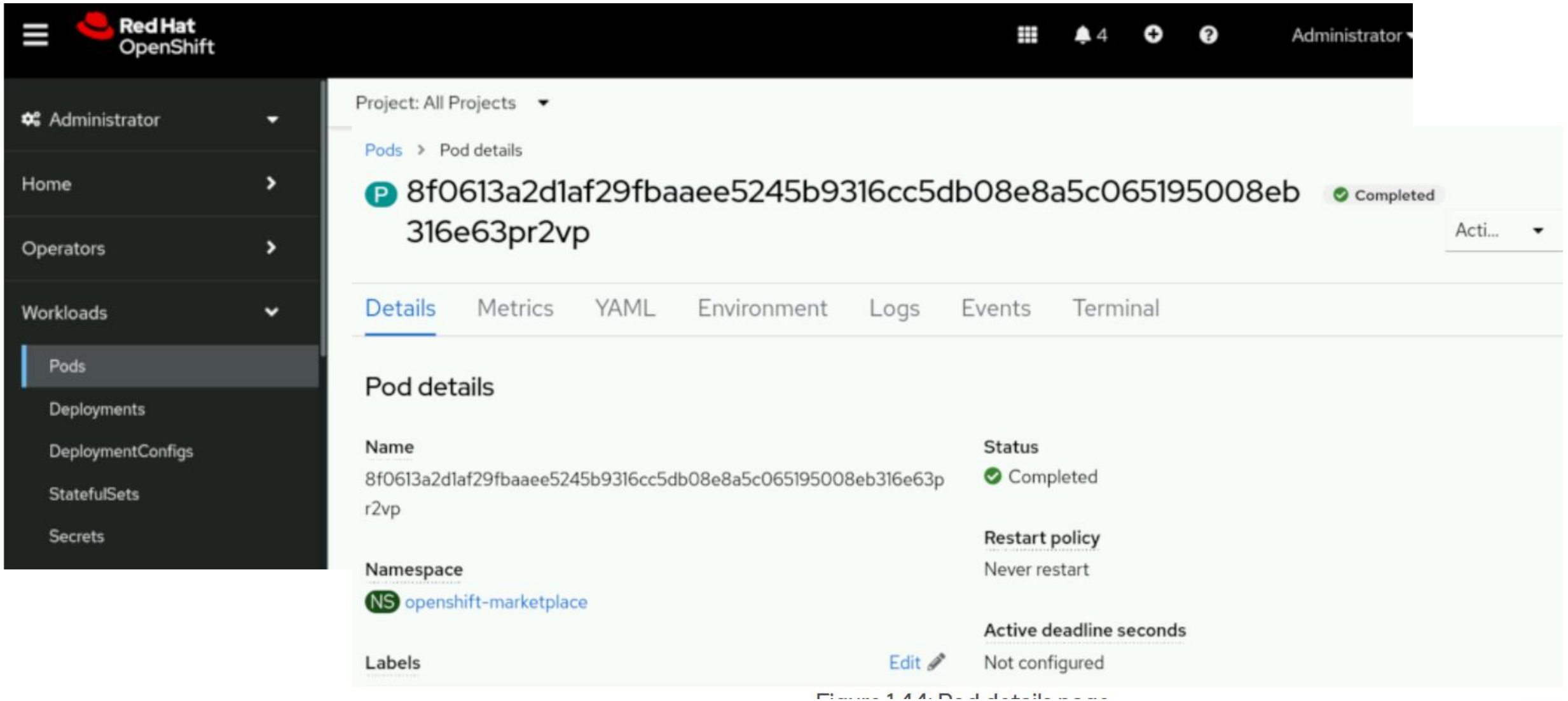
Further investigate thru terminal



Accessing Pod Logs



Accessing Pod Logs



Red Hat OpenShift Container Platform Metrics and Alerts

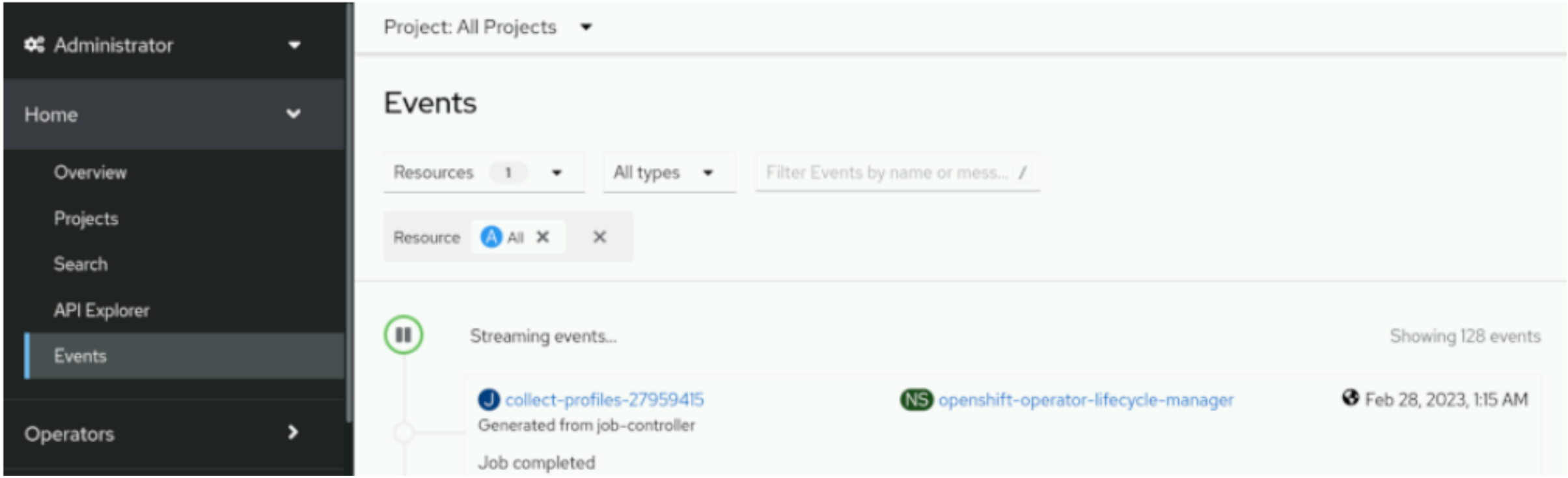
- Metrics pertaining cluster performance and application performance
- Prometheus
  - open-source monitoring and alerting toolkit
  - provide client libraries accessing application level's metrics
- Alert based on degradation of service
  - <= storage space
  - >= CPU utilization
- PodMonitor resource
  - Monitor and generate alerts based on information from pods
- ServiceMonitor resource
  - Monitor and generate alerts based on information from services

Kubernetes Events

- Application logs tends to
  - be highly detailed and granular
  - deeper level of details for remediating specific issues
- Events are meaningful in
  - provide high-level abstraction, significant changes
  - understand general performance
  - understand behaviour of nodes, projects, pods at glance
  - highlight meaningful issues

Events console page

Access thru **Home** → **Events** page

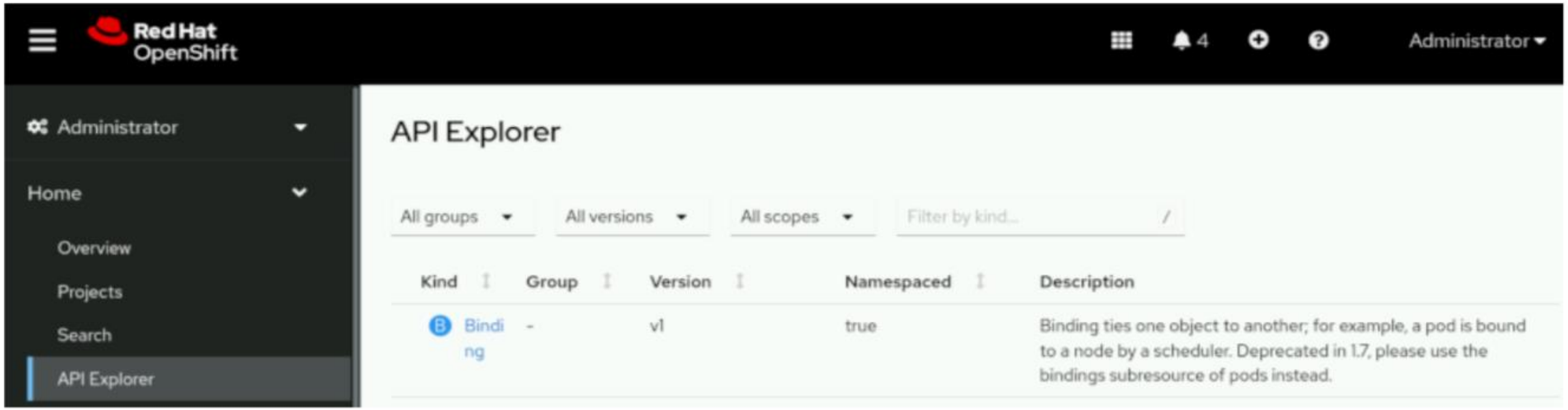


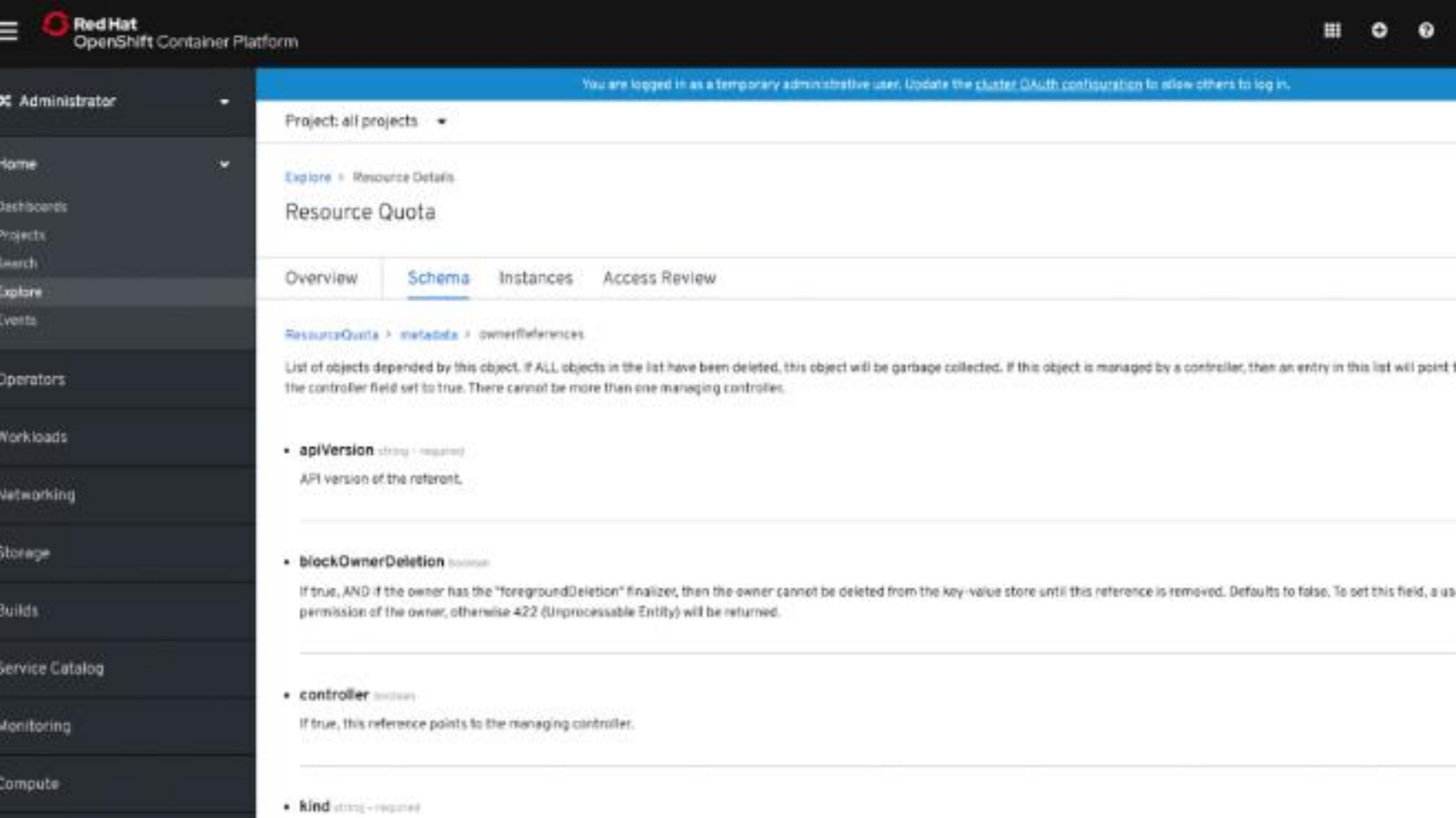
API Explorer

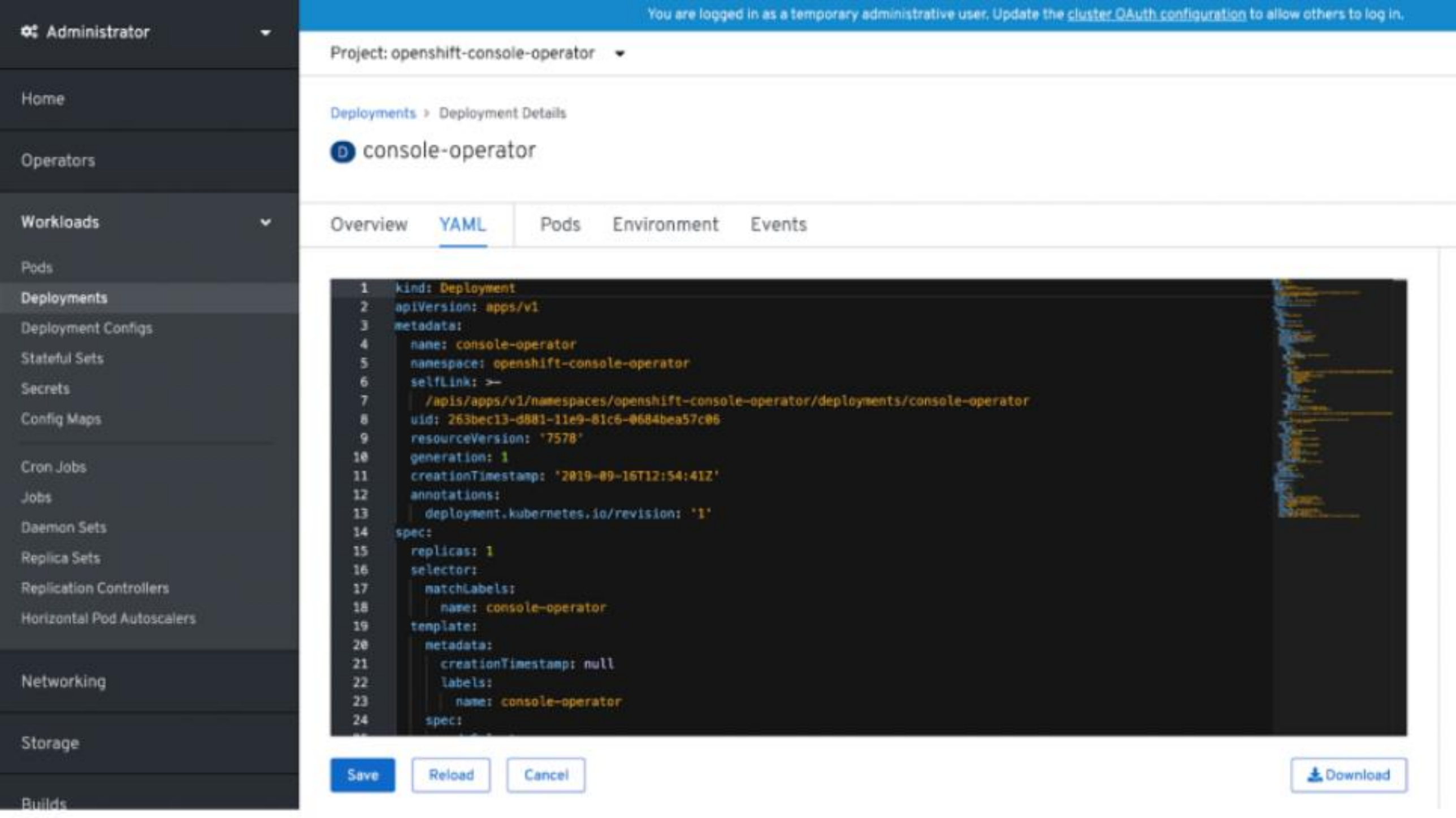
- Start learning more about the K8s resources

- Elobarated definition, metadata

- Access review (permissions)




Red Hat
OpenShift Container Platform
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.
Administrator
Project: all projects
Home
Dashboards
Projects
Search
Explore
Events
Operators
Workloads
Networking
Storage
Builds
Service Catalog
Monitoring
Compute
Explore > Resource Details
Resource Quota
Overview
Schema
Instances
Access Review
ResourceQuota > metadata > ownerReferences
List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point
the controller field set to true. There cannot be more than one managing controller.
apiVersion
API version of the referent.
blockOwnerDeletion
If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. Defaults to false. To set this field, a
permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
controller
If true, this reference points to the managing controller.
kind


Administrator
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.
Project: openshift-console-operator
Home
Operators
Workloads
Pods
Deployments
Deployment Configs
Stateful Sets
Secrets
Config Maps
Cron Jobs
Jobs
Daemon Sets
Replica Sets
Replication Controllers
Horizontal Pod Autoscalers
Networking
Storage
Builds
Deployments › Deployment Details
console-operator
Overview
YAML
Pods
Environment
Events
1 kind: Deployment
2 apiVersion: apps/v1
3 metadata:
4 name: console-operator
5 namespace: openshift-console-operator
6 selfLink: >-
7 /apis/apps/v1/namespaces/openshift-console-operator/deployments/console-operator
8 uid: 263bec13-d881-11e9-81c6-0684bea57c06
9 resourceVersion: '7578'
10 generation: 1
11 creationTimestamp: '2019-09-16T12:54:41Z'
12 annotations:
13 deployment.kubernetes.io/revision: '1'
14 spec:
15 replicas: 1
16 selector:
17 matchLabels:
18 name: console-operator
19 template:
20 metadata:
21 creationTimestamp: null
22 labels:
23 name: console-operator
24 spec:
Save
Reload
Cancel
Download

Guided Exercise: Monitor an OpenShift Cluster

Explore and show the monitoring features and components.

Explore the Overview page to inspect the cluster status.

Use a terminal connection to the master01 node to view the crio and kubelet services.

Explore the Monitoring page, alert rule configurations, and the etcd service dashboard.

Explore the events page, and filter events by resource name, type, and message.

Lab: Introduction to Kubernetes and OpenShift

You should be able to :

- Navigate the Red Hat OpenShift Container Platform web console to find various information items and configuration details.
- As the student user on the workstation machine, use the lab command to prepare your system for this exercise.
- This command ensures that the Red Hat OpenShift Container Platform is deployed and ready for the lab.

Chapter Summary

In this chapter, you learned:

- Containers are an isolated application runtime created with very little overhead.
- A container image packages an application with all of its dependencies, making it easier to run the application in different environments.
- Applications such as Podman create containers using features of the standard Linux kernel.
- Container image registries are the preferred mechanism for distributing container images to multiple users and hosts.
- OpenShift orchestrates applications composed of multiple containers using Kubernetes.
- Kubernetes manages load balancing, high availability, and persistent storage for containerized applications.
- OpenShift adds to Kubernetes multitenancy, security, ease of use, and continuous integration and continuous development features.
- OpenShift routes enable external access to containerized applications in a manageable way.