Kubernetes

Cloud Insights

NetApp
August 08, 2024

Table of Contents

Kubernetes

Kubernetes Cluster Overview

The Cloud Insights Kubernetes Explorer is a powerful tool for displaying the overall health and usage of your Kubernetes clusters and allows you to easily drill down into areas of investigation.

Clicking on **Dashboards > Kubernetes Explorer** opens the Kubernetes Cluster list page. This overview page contains table of the Kubernetes clusters in your environment.

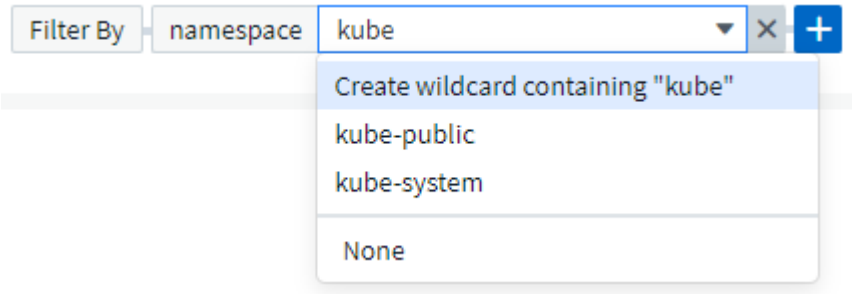
Cluster list

The cluster list displays the following information for each cluster in your environment:

- Cluster **Name**. Clicking on a cluster name will open the **detail page** for that cluster.
- **Saturation** percentages. Overall Saturation is the highest of CPU, Memory, or Storage Saturation.
- Number of **Nodes** in the cluster. Clicking this number will open the Node list page.
- Number of **Pods** in the cluster. Clicking this number will open the Pod list page.
- Number of **Namespaces** in the cluster. Clicking this number will open the Namespace list page.
- Number of **Workloads** in the cluster. Clicking this number will open the Workload list page.

Refining the Filter

When you are filtering, as you begin typing you are presented with the option to create a **wildcard filter** based on the current text. Selecting this option will return all results that match the wildcard expression. You can also create **expressions** using NOT or AND, or you can select the "None" option to filter for null values in the field.



Filters based on wildcards or expressions (e.g. NOT, AND, "None", etc.) display in dark blue in the filter field. Items that you select directly from the list are displayed in light blue.

Kubernetes filters are contextual, meaning for example that if you are on a specific node page, the pod_name filter only lists pods related to that node. Moreover, if you apply a filter for a specific namespace, then the pod_name filter will list only pods on that node *and* in that namespace.

Note that Wildcard and Expression filtering works with text or lists but not with numerics, dates or booleans.

Before Installing or Upgrading the NetApp Kubernetes Monitoring Operator

Read this information before installing or upgrading the Kubernetes Monitoring Operator.

Component	Requirement
Kubernetes version	Kubernetes v1.20 and above.
Kubernetes distributions	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Linux OS	Cloud Insights does not support nodes that are running with Arm64 architecture. Network monitoring: must be running Linux kernel version 4.18.0 or above. Photon OS is not supported.
Labels	Cloud Insights supports monitoring of Kubernetes nodes that are running Linux, by specifying a Kubernetes node selector that looks for the following Kubernetes labels on these platforms: Kubernetes v1.20 and above: Kubernetes.io/os = linux Rancher + cattle.io as orchestration/Kubernetes platform: cattle.io/os = linux
Commands	The curl and kubectl commands must be available.; for best results, add these commands to the PATH.
Connectivity	kubectl cli is configured to communicate with the target K8s cluster, and have Internet connectivity to your Cloud Insights environment. If you are behind a proxy during installation, follow the instructions in the Configuring Proxy Support section of the Operator installation. For accurate audit and data reporting, synchronize the time on the Agent machine using Network Time Protocol (NTP) or Simple Network Time Protocol (SNTP).

Component	Requirement
Other	If you are running on OpenShift 4.6 or higher, you must follow the OpenShift Instructions in addition to ensuring these pre-requisites are met.
API Token	If you are re-deploying the Operator (i.e. you are updating or replacing it), there is no need to create a new API token; you can re-use the previous token.

Important Things to Note Before You Start

If you are running with a proxy, have a custom repository, or are using OpenShift, read the following sections carefully.

Also read about Permissions.

Configuring Proxy Support

There are two places where you may use a proxy in your environment in order to install the NetApp Kubernetes Monitoring Operator. These may be the same or separate proxy systems:

- Proxy needed during execution of the installation code snippet (using "curl") to connect the system where the snippet is executed to your Cloud Insights environment
- Proxy needed by the target Kubernetes cluster to communicate with your Cloud Insights environment

If you use a proxy for either or both of these, to install the NetApp Kubernetes Operating Monitor you must first ensure that your proxy is configured to allow good communication to your Cloud Insights environment. For example, from the servers/VMs from which you wish to install the Operator, you need to be able to access Cloud Insights and be able to download binaries from Cloud Insights.

For the proxy used to install the NetApp Kubernetes Operating Monitor, before installing the Operator, set the *http_proxy/https_proxy* environment variables. For some proxy environments, you may also need to set the *no_proxy environment* variable.

To set the variable(s), perform the following steps on your system **before** installing the NetApp Kubernetes Monitoring Operator:

1. Set the *https_proxy* and/or *http_proxy* environment variable(s) for the current user:
   a. If the proxy being setup does not have Authentication (username/password), run the following command:

```
export https_proxy=<proxy_server>:<proxy_port>
```

   b. If the proxy being setup does have Authentication (username/password), run this command:

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_po
rt>
```

For the proxy used for your Kubernetes cluster to communicate with your Cloud Insights environment, install the NetApp Kubernetes Monitoring Operator after reading all of these instructions.

Configure the proxy section of AgentConfiguration in operator-config.yaml before deploying the NetApp Kubernetes Monitoring Operator.

```
agent:
  ...
  proxy:
    server: <server for proxy>
    port: <port for proxy>
    username: <username for proxy>
    password: <password for proxy>

    # In the noproxy section, enter a comma-separated list of
    # IP addresses and/or resolvable hostnames that should bypass
    # the proxy
    noproxy: <comma separated list>

    isTelegrafProxyEnabled: true
    isFluentbitProxyEnabled: <true or false> # true if Events Log enabled
    isCollectorsProxyEnabled: <true or false> # true if Network
Performance and Map enabled
    isAuProxyEnabled: <true or false> # true if AU enabled
  ...
...
```

Using a custom or private docker repository

By default, the NetApp Kubernetes Monitoring Operator will pull container images from the Cloud Insights repository. If you have a Kubernetes cluster used as the target for monitoring, and that cluster is configured to only pull container images from a custom or private Docker repository or container registry, you must configure access to the containers needed by the NetApp Kubernetes Monitoring Operator.

Run the “Image Pull Snippet” from the NetApp Monitoring Operator install tile. This command will log into the Cloud Insights repository, pull all image dependencies for the operator, and log out of the Cloud Insights repository. When prompted, enter the provided repository temporary password. This command downloads all images used by the operator, including for optional features. See below for which features these images are used for.

Core Operator Functionality and Kubernetes Monitoring

- netapp-monitoring
- kube-rbac-proxy
- kube-state-metrics
- telegraf
- distroless-root-user

Events Log

- fluent-bit
- kubernetes-event-exporter

Network Performance and Map

- ci-net-observer

Push the operator docker image to your private/local/enterprise docker repository according to your corporate policies. Ensure that the image tags and directory paths to these images in your repository are consistent with those in the Cloud Insights repository.

Edit the monitoring-operator deployment in operator-deployment.yaml, and modify all image references to use your private Docker repository.

```
image: <docker repo of the enterprise/corp docker repo>/kube-rbac-
proxy:<kube-rbac-proxy version>
image: <docker repo of the enterprise/corp docker repo>/netapp-
monitoring:<version>
```

Edit the AgentConfiguration in operator-config.yaml to reflect the new docker repo location. Create a new imagePullSecret for your private repository, for more details see *https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/*

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation for
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository[using a custom or private docker repository].
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift Instructions

If you are running on OpenShift 4.6 or higher, you must edit the AgentConfiguration in *operator-config.yaml* to enable the *runPrivileged* setting:

```
# Set runPrivileged to true SELinux is enabled on your kubernetes nodes
runPrivileged: true
```

Openshift may implement an added level of security that may block access to some Kubernetes components.

Permissions

If the cluster you are monitoring contains Custom Resources which do not have a ClusterRole which aggregates to view, you will need to manually grant the operator access to these resources to monitor them with Event Logs.

1. Edit *operator-additional-permissions.yaml* before installing, or after installing edit the resource *ClusterRole/<namespace>-additional-permissions*
2. Create a new rule for the desired apiGroups and resources with the verbs ["get", "watch", "list"]. See https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Apply your changes to the cluster

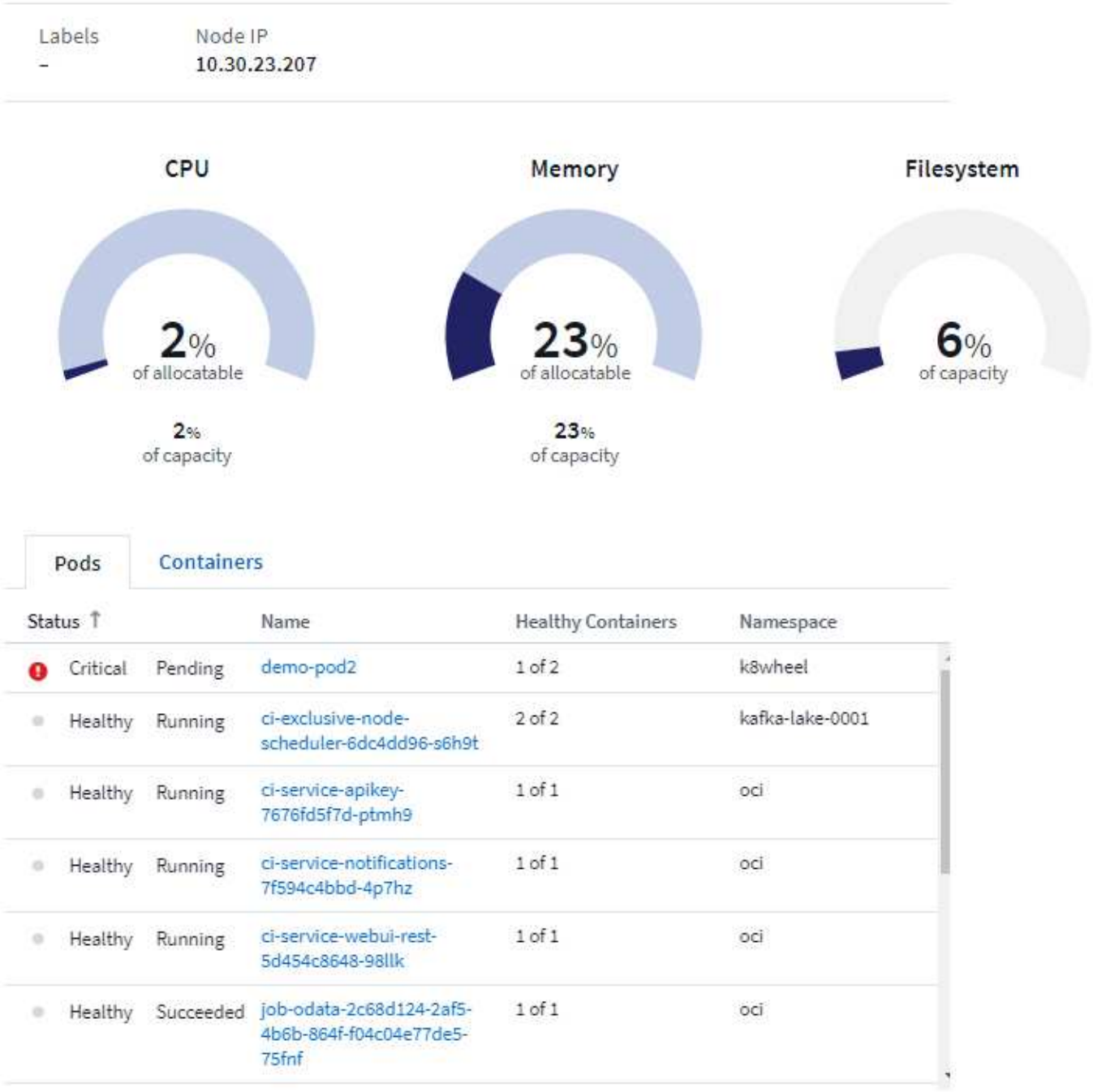
kube-state-metrics Counters

Use the following links to access information for these kube state metrics counters:

1. ConfigMap Metrics
2. DaemonSet Metrics
3. Deployment Metrics
4. Ingress Metrics
5. Namespace Metrics
6. Node Metrics
7. Persistent Volume Metrics
8. Persistant Volume Claim Metrics
9. Pod Metrics
10. ReplicaSet metrics
11. Secret metrics
12. Service metrics
13. StatefulSet metrics

Detail Section

Each page of the Kubernetes Explorer displays a set of usage graphs that may include CPU, Memory, and Storage. Below these graphs are summaries and lists of the top objects in each category, with links to underlying details. For example, *Node* shows pods and containers, *Pod* shows PVCs and related objects and containers, etc. The following illustration shows an example of the detailed information on a *Node* page:

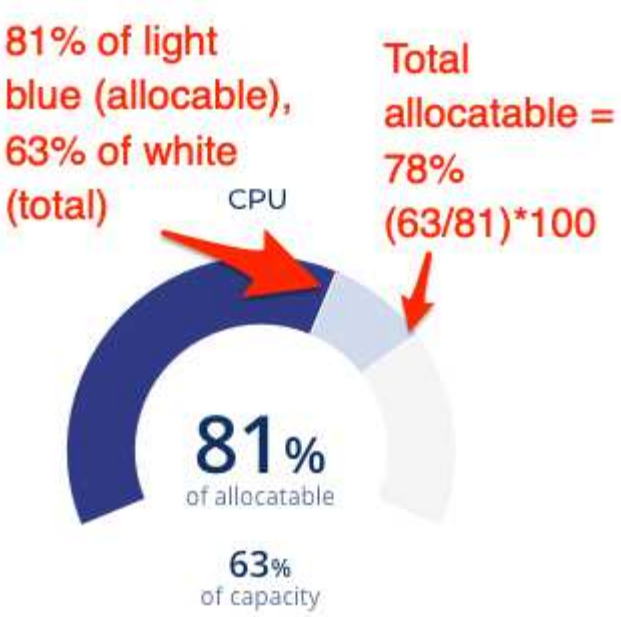


Resources experiencing alerts will show at the top of the lists. Click on the affected resource to drill into it for more detail.

Keep the following in mind when reading the gauges.

The dark blue band shows the amount used.

- When viewed against the *light blue band*, the dark blue shows used as the % of allocatable amount. This matches the "% of allocatable" value shown (81 in the example below).
- When viewed against the *white background*, the dark blue shows used as the % of total capacity. This matches the "% of capacity" value shown (63 in this example).



Details

Hovering over a circle displays a summary of information for that service.