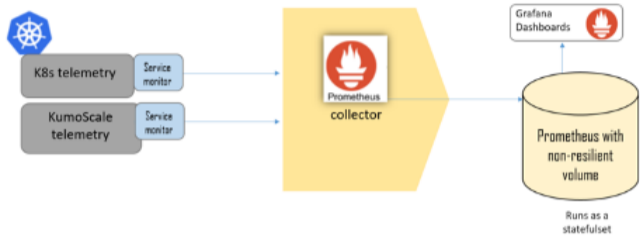


Telemetry

This section describes how to collect data and statistics in KumoScale.

KumoScale software collects data on the physical SSD and volume workloads. This data is stored in **Prometheus**, an open-source system monitoring and alerting toolkit. The volume capacity for storing the data is a configurable parameter that may be changed by the user. Open-source Grafana dashboards can be used to visualize telemetry data.



This section documents how to create and use the telemetry options supported in KumoScale. We will use custom resource files to show how the service is created and how data may be evaluated. For your orchestration environment, you may need to reference the relevant interface guide: [KumoScale REST API](#), [Cluster Manager CLI](#), [KumoScale Ansible](#), [KumoScale Kubernetes CSI Driver](#), or [KumoScale for OpenStack](#). This page covers:

- [Creating the Telemetry Service](#) using the telemetry custom resource file.
- [Telemetry Data](#) details the data returned by the service..

You may also want to review the following notes available on the KumoScale documentation site:

- [Grafana Dashboard Guide](#) for information on how to deploy and view reports using a Grafana dashboard deployed in a KumoScale cluster.
- [Automating Metrics Collection](#) for details on how to directly integrate metrics into your telemetry infrastructure.

Create the Telemetry Service

To deploy telemetry, KumoScale software provides a **Telemetry CRD**, a template of which is in **KumoScale_Operator/ks-config-operator/samples/kumoscale_v1_telemetry_cr.yaml**. Alternatively you can create the service using the appropriate interface

The table below shows the parameters supported in the Telemetry CRD.

Telemetry parameter	Description	Optional/Required
name	The telemetry configuration name; must comply with the Name field as defined in KumoScale KumoScale Field Types .	Required
ip	IP for the time series database to which telemetry data is pushed.	Required
port	Telemetry server port. The default value is 2003 .	Optional
intervalMin	The time interval (in seconds) between consecutive telemetry push events. Maximum is 86,400 (24 hours).	Optional
dataTypes	Volume telemetry (VOLUME;DRIVE), SSD telemetry (drive), or both (default).	Optional
pushState	Initial sending state: true - the telemetry is pushed. false - the telemetry is not pushed.	Optional
prefix	The prefix for the namespace in the time series database, from which the telemetry data structure is created. The default value is kumoscale .	Optional
telemetryTsdbType	Telemetry server type. Default is { GRAPHITE }.	Optional
transportType	Transport Type; either TCP_IP or UDP .	Optional

For example, to create a telemetry service called **telemetry1** using the CRD **myapp_telemetry_cr.yaml** (created from a copy of the KumoScale template in **kumoscale_telemetry_cr.yaml**).

1. Edit **myapp_telemetry_cr.yaml** , and save with required parameters **name (telemetry1)** and **ip**, and optional parameters **port**, **pushState**, and **dataTypes**:

```
apiVersion: kumoscale.kioxia.com/v1
kind: Telemetry
metadata:
  name: telemetry1
spec:
  telemetry:
    ip: 192.0.2.0
    port: 2003
    pushState: true
    dataTypes:
      - DRIVE
```

2. Create **telemetry1** with:

```
kubect1 create -f myapp_telemetry_cr.yaml
```

3. Verify the service is running with:

```
kubect1 get svc -A -o wide
```

The next sections describe the data provided by the telemetry service.

Telemetry Data

The following sections describe the information sent in the KumoScale software telemetry feed.

SSD Telemetry

SSD telemetry consists of SMART (Self-Monitoring, Analysis, and Reporting Technology) parameters listed in the table below:

Parameter	Description	Comments
Temperature	Device temperature	Reported in Kelvin
Available Spare	Margin rate (%)	
Percentage Used	Endurance in percentage	
Data Units Read	The number of data units the initiator has read from the controller	Reported in 4KB units
Data Units write	The number of data units the initiator has written to the controller	Reported in 4KB units
Host Read Commands	The number of read commands completed by the controller	Includes compare commands
Host Write Commands	The number of write commands completed by the controller	
ks_volume_ssd_used_capacity_bytes	The sum of the capacity of all sub-volumes that belongs to a volume on the same SSD.	Reported in bytes

Volume Telemetry

The volume-based data that is collected by KumoScale software consists of the following workload statistics:

Parameter	Description	Comments
IOs	Total commands issued to the volume	Read/Write
Bandwidth	Total bytes issued to the volume	Read/Write
IOPS	Read/Write, command size less than or equal to: 4KB/8KB/16KB/64KB/256KB/512KB/1MB/>1MB	Histogram
Latency	Total latency in µs	Read/Write
ks_volume_utilized_capacity_bytes	The number of written bytes on the volume.	
ks_volume_used_capacity_bytes	The allocated space the volume took. For: · Thick volumes, the capacity of the volume. · Thin volumes, the current allocated space of the volume.	

KumoScale Provisioner Telemetry Detail

The *Get Metrics* REST API command will present all Prometheus metrics that were collected from the KumoScale Provisioner. The available feeds are listed below followed by examples.

- [ks_node_used_capacity_bytes](#)
- [ks_node_free_capacity_bytes](#)
- [ks_node_state](#)
- [ks_connected_volumes_state](#)
- [ks_node_used_iops](#)
- [ks_node_used_bw_bytes_per_sec](#)
- [ks_node_free_iops](#)
- [ks_node_free_bw_bytes_per_sec](#)
- [ks_connected_replica_state](#)
- [ks_vol_capacity_bytes](#)

ks_node_used_capacity_bytes

Shows the capacity used for each storage node in bytes

Type: Gauge

Labels:

- node- Storage node persistent ID.

- name – Storage node name.

Example:

```
ks_node_used_capacity_bytes{node="00:0c:##:0f:##:##",name="ks-node6",} 2.01326592E8
```

ks_node_free_capacity_bytes

Shows the free capacity for each storage node in bytes

Type: gauge

Labels:

- node- Storage node persistent ID.
- name – Storage node name.

Example:

```
ks_node_free_capacity_bytes{node="00:0c:##:0f:##:##",name="ks-node6",} 2.19882192896E12
```

ks_node_state

Shows the state of each storage node

Type: gauge

State: either 1=Available or 2=Unavailable

Labels:

- node- Storage node persistent ID.
- name – Storage node name.
- rack- Storage node location Rack ID.
- zone- Storage node location zone ID.
- region- Storage node location region ID.

Example:

```
ks_node_state{node="00:0c:##:4b:##:##",name="ks-node7 rack="null",zone="null",region="null",}
```

ks_connected_volumes_state

Shows the state of connected volumes per volume. The state is considered Available (0) if at least one replica has a ReplicaState of Available (0). Reported only for published NVMe-oF volumes.

State: either 0=Available or 1=Unavailable

Type: gauge

Labels:

- `uuid` – For replicated volumes, equal to the UUID of the parent volume; otherwise, the volume `uuid`.
- `alias` - Volume's alias.
- `hostId` - The UUID of the initiator (host)
- `hostname` – The initiator (host) name.
- `nqn` - NQN of the initiator (host)
- `version` - Software version of the initiator's agent.

Example for a simple (non-replicated) volume:

```
ks_connected_volumes_state{uuid="#####-####-####-####-#####",alias="pvc-###",hostID="xxxxxxx-xxxx-xxxx-xxxxx"}
```

◀ ▶ 🔍

Example for a replicated volume:

```
ks_connected_volumes_state{uuid=" #####-##xx-####-xxxxx###x alias="pvc-xxxxxx",hostID=" xxxxxxxx-xxxx-xxxx-xxxx
```

◀ ▶ 🔍

ks_node_used_iops

Shows the IOPS used per storage node in IO per second

Type: gauge

Labels:

node- Storage node persistent ID

name - Storage node name.

Example:

```
ks_node_used_iops{node="00:0c:##:##:##:09",name="ks-node3",} 13670.0
```

ks_node_used_bw_bytes_per_sec

Shows the bandwidth used per storage node in bytes per second.

Type: gauge

Labels:

node- Storage node persistent ID

name - Storage node name.

Example:

```
ks_node_used_bw_bytes_per_sec{node="00:0c:##:##:##:##",name="ks-node3",} 1.7747392E8
```

ks_node_free_iops

Shows the IOPS free per Storage Node in IO per second

Type: gauge

Labels:

node- Storage node persistent ID

name - Storage node name.

Example:

ks_node_free_iops{node="00:##:##:0f:##:##",name="ks-node6",} 522900.0

ks_node_free_bw_bytes_per_sec

Shows the free bandwidth per Storage Node in bytes per second

Type gauge

Labels:

node- Storage node persistent ID

name - Storage node name.

Example:

```
ks_node_free_bw_bytes_per_sec{node="00:0c:##:0f:##:##",name="ks-node6",} 6.788481024E9
```

ks_connected_replica_state

Shows the state of connected replicas for each volume. Reported only for published NVMe-oF volumes.

Type: gauge

State: one of 0=Available, 1=Terminating, 2=Missing, 3=Unknown, 4=Synchronizing

Labels:

- uuid - UUID of the parent volume.
- alias- Alias of the parent volume
- repUUID- UUID of the replica
- hostId- UUID of the connected initiator (host) Connected host UUID
- hostname – the name of the connected initiator (host)
- Nqn- NQN of the connected initiator (host)

Example:

```
ks_connected_replica_state{uuid="f#####-####-####-####-#####8",alias="pvc-#####-####-####-####-#####",repUUID="f#####-####-####-####-#####8",hostId="00:0c:##:0f:##:##",hostname="ks-node6",Nqn="nqn.2016-06.io.spn.csi.storage.k8s.io"} 1.0
```

ks_vol_capacity_bytes

Shows the capacity of each volume in bytes.

- Type: gauge

Labels:

- uuid – For replicated volumes, equal to the UUID of the parent volume; otherwise, the volume uuid.
- alias - Volume's alias.
- numReplicas - Number of replicas
- tenantID - tenant ID (0 for the default Global Tenant)
- storageClassName – the name of the storage class or "unknown" if not provided
- provisioningType – One of THIN, THICK, SNAPSHOT/CLONE
- protocol – one of NVMeoF or Local
- repUUIDX - For replicated volumes, equal to the UUID of the X replica; otherwise, blank.
- nodeIDX - For replicated volumes, equal to the node ID (Persistent ID) of the X replica; otherwise, the node ID of the volume
- nodeNameX - - For replicated volumes, equal to the node name of the X replica; otherwise, the node ID of the volume

Example:

```
ks_vol_capacity_bytes{uuid="#####-####-####-####-#####",alias="pvc-#####-####-####-####-#####",nur
```

Volume Mapping Details

The available feeds are listed below followed by details on each:

- [ks_volume_ssd_used_capacity_bytes](#)
- [ks_volume_utilized_capacity_bytes](#)
- [ks_volume_used_capacity_bytes](#)

ks_volume_ssd_used_capacity_bytes

Shows the sum of the capacity in bytes of all sub-volumes that belong to a volume on the same SSD.

Type: gauge

- Labels:
- volID - For replicated volumes, equal to the UUID of the parent volume; otherwise, the volume uuid. .
- replicaID – UUID of the replica.
- nodeID – the storage node identifier
- driveID – the SSD identifier.
- nodeName – the name of the storage node

Example:

```
ks_volume_ssd_used_capacity_bytes{volID="#####-####-####-####-#####",replicaID="#####-####-####-####-179-
```

ks_volume_utilized_capacity_bytes

Shows the number of bytes **written** on the volume.

Type: gauge

Labels:

- volID- For replicated volumes, equal to the UUID of the parent volume; otherwise, the volume uuid. .
- replicaID –UUID of the replica
- nodeID – the storage node identifier.
- nodeName – the storage node name

ks_volume_used_capacity_bytes

For thick volume, shows the volume’s capacity in bytes. For thin volumes, snapshots, or snapshot volumes, shows the reserved space in bytes.

Type: gauge

Labels:

- volID - For replicated volumes, equal to the UUID of the parent volume; otherwise, the volume uuid. .
- replicaID –UUID of the replica
- nodeID – the storage node identifier
- nodeName – the storage node name

KumoScale Storage Nodes Telemetry Detail

The *Get Prometheus Metrics* REST API command will return the Prometheus metrics from the storage node.

The available feeds are listed below followed by details on each:

- [totalBytesRead](#)
- [Data_Units_Read](#)
- [totalWriteLatency](#)
- [totalReadCommands](#)
- [IOWriteCnt](#)
- [Media_and_Data_Integrity_Errors](#)
- [totalWriteCommands](#)
- [Percentage_Used](#)
- [Available_Spare](#)
- [Data_Units_Written](#)
- [Latency](#)
- [Host_Read_Commands](#)
- [IOReadCnt](#)
- [totalBytesWrite](#)
- [totalReadLatency](#)

- [Host Write Commands](#)
- [Composite Temperature](#)

totalBytesRead

Shows the total bytes read per volume

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalBytesRead{DRIVE=" VMWare_NVME-0000",VOLUME="#####-e45e-4bcb-a03e-#####",} 3.4318336E
```

Data_Units_Read

Shows the number of 4k, 512 byte SSD data units read from the controller. For the NVM command set, logical blocks read as part of Compare and Read operations are included in this value.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Data_Units_Read{DRIVE="VMWare_NVME-0000",VOLUME=" be3f5f29-e45e-4bcb-a03e-56578921a19e ",} 0.0
```

totalWriteLatency

Shows the total write latency per volume in micro seconds.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalWriteLatency{DRIVE=" VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 89694.0
```

totalReadCommands

Shows the total number of reads per volume

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalReadCommands{DRIVE=" VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 520.0
```

IOWriteCnt

Shows the total number of write commands by size.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID
- SIZE- Sizes = {1e4K, 1e8K, 1e16K, 1e64K, 1e256K, 1e1MB,inf}. For example : 1e16K is the total read commands with 8K < Size ≤ 16K.

Example:

```
IOWriteCnt{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",size="1e1M",} 17.0
```

Media_and_Data_Integrity_Errors

Shows SSD media and data integrity errors.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Media_and_Data_Integrity_Errors{DRIVE="VMWare_NVME-0000",VOLUME=" be3f5f29-e45e-4bcb-a03e-56578921a19e ",} 0.0
```

totalWriteCommands

The total write commands transferred in a Volume.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalWriteCommands{DRIVE=" VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 339.0
```

Percentage_Used

The SSD Percentage used. The SSD Endurance normalizes the value from (0-255) to (0-100) (i.e. x100/255).

Type: gauge

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Percentage_Used{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 0.0
```

Available_Spare

Shows the SSD available spare by margin rate, a percentage (%).

Type: gauge

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Available_Spare{DRIVE="VMWare_NVME-0000",VOLUME=" be3f5f29-e45e-4bcb-a03e-56578921a19e ",} 1.0
```

Data_Units_Written

Shows the SSD Data Units Written in 4K units. For the NVM command set, logical blocks written as part of Write operations are included in this value. Write Uncorrectable commands do not impact this value.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Data_Units_Written{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 0.0
```

Latency

Shows the latency of commands of specific size and type for a quintile measurement. For example, latency with percentile=50, size=4k. rw=“rd” – is the latency of 4K read commands for 50%.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID
- percentile = {Avg, 50, 90, 95, 99}
- size {4K, 8K, 16K, 64K, 256K}
- rw (“rd”, “wr”)

Example:

```
Latency{DRIVE="",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",percentile="99",size="64K",rw="rd",} 0.0
```

Host_Read_Commands

Shows SSD Host Read Commands. Contains the number of read commands completed by the controller. For the NVM command set, this is the number of Compare and Read commands.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Host_Read_Commands{DRIVE="VMWare_NVME-0000",VOLUME=" be3f5f29-e45e-4bcb-a03e-56578921a19e ",} 0.0
```

IOReadCnt

Shows the total read commands by size.

Type: counter

Labels:

- DRIVE – SSD Persistent Id
- VOLUME – Volume UUID
- Size – Sizes = {le4K, le8K, le16K, le64K, le256K, le1MB,inf}
 - le4K - total read commands with $0 < \text{Size} \leq 4\text{K}$
 - le8K - total read commands with $4 < \text{Size} \leq 8\text{K}$
 - le16K - total read commands with $8\text{K} < \text{Size} \leq 16\text{K}$
 - le64K - total read commands with $16\text{k} < \text{Size} \leq 64\text{K}$
 - le64K - total read commands with $64\text{k} < \text{Size} \leq 256\text{K}$
 - le1MB - total read commands with $256\text{k} < \text{Size} \leq 1\text{MB}$
- inf - total read commands with size > 1MB.

Example:

```
IOReadCnt{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",size="le1M",} 331.0
```

totalBytesWrite

Shows the total bytes written per volume

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalBytesWrite{DRIVE=" VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 9416704.0
```

totalReadLatency

Shows the total read latency per volume in micro seconds

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
totalReadLatency{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 1105927.0
```

Host_Write_Commands

Shows the number of SSD write commands completed by the controller. For the NVM command set, this is the number of write commands.

Type: counter

Labels:

- DRIVE – SSD Persistent Id. Note that when using a virtual SSD, the SSD persistent id will not appear in a Serial Number format.
- VOLUME – Volume UUID

Example:

```
Host_Write_Commands{DRIVE="VMWare_NVME-0000",VOLUME="be3f5f29-e45e-4bcb-a03e-56578921a19e",} 0.0
```

Composite_Temperature

Shows the device temperature, SSD Composite Temperature, in Celsius.

Type: gauge

Example:

```
Composite_Temperature{DRIVE="VMWare_NVME-0000",VOLUME=" be3f5f29-e45e-4bcb-a03e-56578921a19e",} 11759.0
```

Additional Statistics and Information

See the section [Logging, Monitoring, and Alerting](#) for information on other tools available such as Prometheus and Grafana.

The KumoScale REST APIs and Cluster Manager CLI support additional commands for retrieving information on:

- Available Network Interface Cards (NICs)
- CPU and memory utilization
- SSD status
- System alerts
- Network and storage performance statistics
- Ongoing sessions
- Allocated volumes

See the [KumoScale REST API](#) and [KumoScale Cluster Manager CLI](#) documentation for more information.

Next: [Syslog](#)

