Official CVE Feed

FEATURE STATE: `Kubernetes v1.27 [beta]`

This is a community maintained list of official CVEs announced by the Kubernetes Security Response Committee. See Kubernetes Security and Disclosure Information for more details.

The Kubernetes project publishes a programmatically accessible feed of published security issues in JSON feed and RSS feed formats. You can access it by executing the following commands:

- JSON feed
- RSS feed

Link to JSON format

```
curl -Lv https://k8s.io/docs/reference/issues-security/official-cve-feed/index.json
```

Link to RSS format

```
curl -Lv https://k8s.io/docs/reference/issues-security/official-cve-feed/feed.xml
```

Official Kubernetes CVE List (last updated: 14 Nov 2025 16:29:29 UTC)

CVE ID	Issue Summary	CVE GitHub Issue URL
CVE-2025-9708	Kubernetes C# Client: improper certificate validation in custom CA mode may lead to man-in-the-middle attacks	#134063
CVE-2025-7445	secrets-store-sync-controller discloses service account tokens in logs	#133897
CVE-2025-5187	Nodes can delete themselves by adding an OwnerReference	#133471
CVE-2025-7342	VM images built with Kubernetes Image Builder Nutanix or OVA providers use default credentials for Windows images if user did not override	#133115
CVE-2025-4563	Nodes can bypass dynamic resource allocation authorization checks	#132151
CVE-2025-1974	ingress-nginx admission controller RCE escalation	#131009
CVE-2025-1098	ingress-nginx controller configuration injection via unsanitized mirror annotations	#131008
CVE-2025-1097	ingress-nginx controller configuration injection via unsanitized auth-tls-match-cn annotation	#131007
CVE-2025-24514	ingress-nginx controller configuration injection via unsanitized auth-url annotation	#131006
CVE-2025-24513	ingress-nginx controller auth secret file path traversal vulnerability	#131005
CVE-2025-1767	GitRepo Volume Inadvertent Local Repository Access	#130786
CVE-2025-0426	Node Denial of Service via kubelet Checkpoint API	#130016
CVE-2024-9042	Command Injection affecting Windows nodes via nodes/*/logs/query API	#129654
CVE-2024-10220	Arbitrary command execution through gitRepo volume	#128885
CVE-2024-9594	VM images built with Image Builder with some providers use default credentials during builds	#128007
CVE-2024-9486	VM images built with Image Builder and Proxmox provider use default credentials	#128006
CVE-2024-7646	Ingress-nginx Annotation Validation Bypass	#126744
CVE-2024-7598	Network restriction bypass via race condition during namespace termination	#126587
CVE-2024-5321	Incorrect permissions on Windows containers logs	#126161
CVE-2024-3744	azure-file-csi-driver discloses service account tokens in logs	#124759
CVE-2024-3177	Bypassing mountable secrets policy imposed by the ServiceAccount admission plugin	#124336
CVE-2023-5528	Insufficient input sanitization in in-tree storage plugin leads to privilege escalation on Windows nodes	#121879
CVE-2023-5044	Code injection via nginx.ingress.kubernetes.io/permanent-redirect annotation	#126817
CVE-2023-5043	Ingress nginx annotation injection causes arbitrary command execution	#126816
CVE-2022-4886	ingress-nginx path sanitization can be bypassed	#126815
CVE-2023-3955	Insufficient input sanitization on Windows nodes leads to privilege escalation	#119595
CVE-2023-3893	Insufficient input sanitization on kubernetes-csi-proxy leads to privilege escalation	#119594
CVE-2023-3676	Insufficient input sanitization on Windows nodes leads to privilege escalation	#119339
CVE-2023-2431	Bypass of seccomp profile enforcement	#118690
CVE-2023-2728	Bypassing policies imposed by the ImagePolicyWebhook and bypassing mountable secrets policy imposed by the ServiceAccount admission plugin	#118640
CVE-2023-2727	Bypassing policies imposed by the ImagePolicyWebhook and bypassing mountable secrets policy imposed by the ServiceAccount admission plugin	#118640
CVE-2023-2878	secrets-store-csi-driver discloses service account tokens in logs	#118419
CVE-2022-3294	Node address isn't always verified when proxying	#113757
CVE-2022-3162	Unauthorized read of Custom Resources	#113756
CVE-2022-3172	Aggregated API server can cause clients to be redirected (SSRF)	#112513
CVE-2021-25749	\`runAsNonRoot\` logic bypass for Windows containers	#112192
CVE-2021-25748	Ingress-nginx \`path\` sanitization can be bypassed with newline character	#126814

CVE ID	Issue Summary	CVE GitHub Issue URL
CVE-2021-25746	Ingress-nginx directive injection via annotations	#126813
CVE-2021-25745	Ingress-nginx `path` can be pointed to service account token file	#126812
CVE-2021-25742	Ingress-nginx custom snippets allows retrieval of ingress-nginx serviceaccount token and secrets across all namespaces	#126811
CVE-2021-25741	Symlink Exchange Can Allow Host Filesystem Access	#104980
CVE-2020-8561	Webhook redirect in kube-apiserver	#104720
CVE-2021-25740	Endpoint & EndpointSlice permissions allow cross-Namespace forwarding	#103675
CVE-2021-25737	Holes in EndpointSlice Validation Enable Host Network Hijack	#102106
CVE-2020-8562	Bypass of Kubernetes API Server proxy TOCTOU	#101493
CVE-2021-3121	Processes may panic upon receipt of malicious protobuf messages	#101435
CVE-2021-25735	Validating Admission Webhook does not observe some previous fields	#100096
CVE-2020-8554	Man in the middle using LoadBalancer or ExternalIPs	#97076
CVE-2020-8566	Ceph RBD adminSecrets exposed in logs when loglevel >= 4	#95624
CVE-2020-8565	Incomplete fix for CVE-2019-11250 allows for token leak in logs when logLevel >= 9	#95623
CVE-2020-8564	Docker config secrets leaked when file is malformed and log level >= 4	#95622
CVE-2020-8563	Secret leaks in kube-controller-manager when using vSphere provider	#95621
CVE-2020-8557	Node disk DOS by writing to container /etc/hosts	#93032
CVE-2020-8559	Privilege escalation from compromised node to cluster	#92914
CVE-2020-8558	Node setting allows for neighboring hosts to bypass localhost boundary	#92315
CVE-2020-8555	Half-Blind SSRF in kube-controller-manager	#91542
CVE-2020-10749	IPv4 only clusters susceptible to MitM attacks via IPv6 rogue router advertisements	#91507
CVE-2019-11254	kube-apiserver Denial of Service vulnerability from malicious YAML payloads	#89535
CVE-2020-8552	apiserver DoS (oom)	#89378
CVE-2020-8551	Kubelet DoS via API	#89377
CVE-2020-8553	ingress-nginx auth-type basic annotation vulnerability	#126818
CVE-2019-11251	kubectl cp symlink vulnerability	#87773
CVE-2018-1002102	Unvalidated redirect	#85867
CVE-2019-11255	CSI volume snapshot, cloning and resizing features can result in unauthorized volume data access or mutation	#85233
CVE-2019-11253	Kubernetes API Server JSON/YAML parsing vulnerable to resource exhaustion attack	#83253
CVE-2019-11250	Bearer tokens are revealed in logs (audit finding TOB-K8S-001)	#81114
CVE-2019-11248	/debug/pprof exposed on kubelet's healthz port	#81023
CVE-2019-11249	Incomplete fixes for CVE-2019-1002101 and CVE-2019-11246, kubectl cp potential directory traversal	#80984
CVE-2019-11247	API server allows access to custom resources via wrong scope	#80983
CVE-2019-11245	container uid changes to root after first restart or if image is already pulled to the node	#78308
CVE-2019-11243	rest.AnonymousClientConfig() does not remove the serviceaccount credentials from config created by rest.InClusterConfig()	#76797
CVE-2019-11244	`kubectl --http-cache=<world-accessible dir>` creates world-writeable cached schema files	#76676
CVE-2019-1002100	json-patch requests can exhaust apiserver resources	#74534
CVE-2018-1002105	proxy request handling in kube-apiserver can leave vulnerable TCP connections	#71411
CVE-2018-1002101	smb mount security issue	#65750
CVE-2018-1002100	Kubectl copy doesn't check for paths outside of it's destination directory.	#61297
CVE-2017-1002102	atomic writer volume handling allows arbitrary file deletion in host filesystem	#60814
CVE-2017-1002101	subpath volume mount handling allows arbitrary file access in host filesystem	#60813
CVE-2017-1002100	Azure PV should be Private scope not Container scope	#47611

CVE ID	Issue Summary	CVE GitHub Issue URL
CVE-2017-1000056	PodSecurityPolicy admission plugin authorizes incorrectly	#43459

This feed is auto-refreshing with a noticeable but small lag (minutes to hours) from the time a CVE is announced to the time it is accessible in this feed.

The source of truth of this feed is a set of GitHub Issues, filtered by a controlled and restricted label `official-cve-feed`. The raw data is stored in a Google Cloud Bucket which is writable only by a small number of trusted members of the Community.

Kubernetes z-pages

Provides runtime diagnostics for Kubernetes components, offering insights into component runtime status and configuration flags.
FEATURE STATE: `Kubernetes v1.32 [alpha]`

Kubernetes core components can expose a suite of *z-endpoints* to make it easier for users to debug their cluster and its components. These endpoints are strictly to be used for human inspection to gain real time debugging information of a component binary. Avoid automated scraping of data returned by these endpoints; in Kubernetes 1.34 these are an **alpha** feature and the response format may change in future releases.

z-pages

Kubernetes v1.34 allows you to enable *z-pages* to help you troubleshoot problems with its core control plane components. These special debugging endpoints provide internal information about running components. For Kubernetes 1.34, components serve the following endpoints (when enabled):

- z-pages
  - statusz
  - flagz

statusz

Enabled using the `ComponentStatusz` feature gate, the `/statusz` endpoint displays high level information about the component such as its Kubernetes version, emulation version, start time and more.

The `/statusz` response from the API server is similar to:

```
kube-apiserver statusz
Warning: This endpoint is not meant to be machine parseable, has no formatting compatibility guarantees and is for debugging purpo

Started: Wed Oct 16 21:03:43 UTC 2024
Up: 0 hr 00 min 16 sec
Go version: go1.23.2
Binary version: 1.32.0-alpha.0.1484&#43;5eeac4f21a491b-dirty
Emulation version: 1.32.0-alpha.0.1484
```

flagz

Enabled using the `ComponentFlagz` feature gate, the `/flagz` endpoint shows you the command line arguments that were used to start a component.

The `/flagz` data for the API server looks something like:

```
kube-apiserver flags
Warning: This endpoint is not meant to be machine parseable, has no formatting compatibility guarantees and is for debugging purpo

advertise-address=192.168.8.2
contention-profiling=false
enable-priority-and-fairness=true
profiling=true
authorization-mode=[Node,RBAC]
authorization-webhook-cache-authorized-ttl=5m0s
authorization-webhook-cache-unauthorized-ttl=30s
authorization-webhook-version=v1beta1
default-watch-cache-size=100
```

CRI Pod & Container Metrics

Collection of Pod & Container metrics via the CRI.
FEATURE STATE: `Kubernetes v1.23 [alpha]`

The kubelet collects pod and container metrics via cAdvisor. As an alpha feature, Kubernetes lets you configure the collection of pod and container metrics via the Container Runtime Interface (CRI). You must enable the `PodAndContainerStatsFromCRI` feature gate and use a compatible CRI implementation (containerd >= 1.6.0, CRI-O >= 1.23.0) to use the CRI based collection mechanism.

CRI Pod & Container Metrics

With `PodAndContainerStatsFromCRI` enabled, the kubelet polls the underlying container runtime for pod and container stats instead of inspecting the host system directly using cAdvisor. The benefits of relying on the container runtime for this information as opposed to direct collection with cAdvisor include:

- Potential improved performance if the container runtime already collects this information during normal operations. In this case, the data can be re-used instead of being aggregated again by the kubelet.
- It further decouples the kubelet and the container runtime allowing collection of metrics for container runtimes that don't run processes directly on the host with kubelet where they are observable by cAdvisor (for example: container runtimes that use virtualization).

Kubernetes Component SLI Metrics

High-level indicators for measuring the reliability and performance of Kubernetes components.
FEATURE STATE: `Kubernetes v1.32 [stable]` (enabled by default: true)

By default, Kubernetes 1.34 publishes Service Level Indicator (SLI) metrics for each Kubernetes component binary. This metric endpoint is exposed on the serving HTTPS port of each component, at the path `/metrics/slis`. The `ComponentSLIs` feature gate defaults to enabled for each Kubernetes component as of v1.27.

SLI Metrics

With SLI metrics enabled, each Kubernetes component exposes two metrics, labeled per healthcheck:

- a gauge (which represents the current state of the healthcheck)
- a counter (which records the cumulative counts observed for each healthcheck state)

You can use the metric information to calculate per-component availability statistics. For example, the API server checks the health of etcd. You can work out and report how available or unavailable etcd has been - as reported by its client, the API server.

The prometheus gauge data looks like this:

```
# HELP kubernetes_healthcheck [ALPHA] This metric records the result of a single healthcheck.
# TYPE kubernetes_healthcheck gauge
kubernetes_healthcheck{name="autoregister-completion",type="healthz"} 1
kubernetes_healthcheck{name="autoregister-completion",type="readyz"} 1
kubernetes_healthcheck{name="etcd",type="healthz"} 1
kubernetes_healthcheck{name="etcd",type="readyz"} 1
kubernetes_healthcheck{name="etcd-readiness",type="readyz"} 1
kubernetes_healthcheck{name="informer-sync",type="readyz"} 1
kubernetes_healthcheck{name="log",type="healthz"} 1
kubernetes_healthcheck{name="log",type="readyz"} 1
kubernetes_healthcheck{name="ping",type="healthz"} 1
kubernetes_healthcheck{name="ping",type="readyz"} 1
```

While the counter data looks like this:

```
# HELP kubernetes_healthchecks_total [ALPHA] This metric records the results of all healthcheck.
# TYPE kubernetes_healthchecks_total counter
kubernetes_healthchecks_total{name="autoregister-completion",status="error",type="readyz"} 1
kubernetes_healthchecks_total{name="autoregister-completion",status="success",type="healthz"} 15
kubernetes_healthchecks_total{name="autoregister-completion",status="success",type="readyz"} 14
kubernetes_healthchecks_total{name="etcd",status="success",type="healthz"} 15
kubernetes_healthchecks_total{name="etcd",status="success",type="readyz"} 15
kubernetes_healthchecks_total{name="etcd-readiness",status="success",type="readyz"} 15
kubernetes_healthchecks_total{name="informer-sync",status="error",type="readyz"} 1
kubernetes_healthchecks_total{name="informer-sync",status="success",type="readyz"} 14
kubernetes_healthchecks_total{name="log",status="success",type="healthz"} 15
kubernetes_healthchecks_total{name="log",status="success",type="readyz"} 15
kubernetes_healthchecks_total{name="ping",status="success",type="healthz"} 15
kubernetes_healthchecks_total{name="ping",status="success",type="readyz"} 15
```

Using this data

The component SLIs metrics endpoint is intended to be scraped at a high frequency. Scraping at a high frequency means that you end up with greater granularity of the gauge's signal, which can be then used to calculate SLOs. The `/metrics/slis` endpoint provides the raw data necessary to calculate an availability SLO for the respective Kubernetes component.

Kubernetes Issues and Security

Kubernetes Issue Tracker

Kubernetes Security and Disclosure Information

Official CVE Feed

Understand Pressure Stall Information (PSI) Metrics

Detailed explanation of Pressure Stall Information (PSI) metrics and how to use them to identify resource pressure in Kubernetes.
FEATURE STATE: `Kubernetes v1.34 [beta]`

As a beta feature, Kubernetes lets you configure the kubelet to collect Linux kernel Pressure Stall Information (PSI) for CPU, memory, and I/O usage. The information is collected at node, pod and container level. This feature is enabled by default by setting the `KubeletPSI` feature gate.

PSI metrics are exposed through two different sources:

- The kubelet's Summary API, which provides PSI data at the node, pod, and container level.
- The `/metrics/cadvisor` endpoint on the kubelet, which exposes PSI metrics in the Prometheus format.

Requirements

Pressure Stall Information requires the following on your Linux nodes:

- The Linux kernel must be version **4.20 or newer**.
- The kernel must be compiled with the `CONFIG_PSI=y` option. Most modern distributions enable this by default. You can check your kernel's configuration by running `zgrep CONFIG_PSI /proc/config.gz`.
- Some Linux distributions may compile PSI into the kernel but disable it by default. If so, you need to enable it at boot time by adding the `psi=1` parameter to the kernel command line.
- The node must be using cgroup v2.

Understanding PSI Metrics

Pressure Stall Information (PSI) metrics are provided for three resources: CPU, memory, and I/O. They are categorized into two main types of pressure: `some` and `full`.

- **`some`**: This value indicates that some tasks (one or more) are stalled on a resource. For example, if some tasks are waiting for I/O, this metric will increase. This can be an early indicator of resource contention.
- **`full`**: This value indicates that *all* non-idle tasks are stalled on a resource simultaneously. This signifies a more severe resource shortage, where the entire system is unable to make progress.

Each pressure type provides four metrics: `avg10`, `avg60`, `avg300`, and `total`. The `avg` values represent the percentage of wall-clock time that tasks were stalled over 10-second, 60-second, and 3-minute moving averages. The `total` value is a cumulative counter in microseconds showing the total time tasks have been stalled.

Example Scenarios

You can use a simple Pod with a stress-testing tool to generate resource pressure and observe the PSI metrics. The following examples use the `agnhost` container image, which includes the `stress` tool.

Generating CPU Pressure

Create a Pod that generates CPU pressure using the `stress` utility. This workload will put a heavy load on one CPU core.

Create a file named `cpu-pressure-pod.yaml`:

```
apiVersion: v1
kind: Podmetadata:  name: cpu-pressure-podspec:  restartPolicy: Never  containers:  - name: cpu-stress    image: registry.k8s.io/e
```

Apply it to your cluster: `kubectl apply -f cpu-pressure-pod.yaml`

Observing CPU Pressure

After the Pod is running, you can observe the CPU pressure through either the Summary API or the Prometheus metrics endpoint.

Using the Summary API:

Watch the summary stats for your node. In a separate terminal, run:

```
# Replace <node-name> with the name of a node in your cluster
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/stats/summary" | jq '.pods[] | select(.podRef.name | contains("cpu-pressure-pod
```

You will see the `some` PSI metrics for CPU increase in the summary API output. The `avg10` value for `some` pressure should rise above zero, indicating that tasks are spending time stalled on the CPU.

Using the Prometheus metrics endpoint:

Query the `/metrics/cadvisor` endpoint to see the `container_pressure_cpu_waiting_seconds_total` metric.

```
# Replace <node-name> with the name of the node where the pod is running
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/metrics/cadvisor" | \
    grep 'container_pressure_cpu_waiting_seconds_total{container="cpu-stress"'
```

The output should show an increasing value, indicating that the container is spending time stalled waiting for CPU resources.

Cleanup

Clean up the Pod when you are finished:

```
kubectl delete pod cpu-pressure-pod
```

Generating Memory Pressure

This example creates a Pod that continuously writes to files in the container's writable layer, causing the kernel's page cache to grow and forcing memory reclamation, which generates pressure.

Create a file named `memory-pressure-pod.yaml`:

```
apiVersion: v1
kind: Podmetadata:  name: memory-pressure-podspec:  restartPolicy: Never  containers:  - name: memory-stress    image: registry.k8
```

Apply it to your cluster: `kubectl apply -f memory-pressure-pod.yaml`

Observing Memory Pressure

Using the Summary API:

In the summary output, you will observe an increase in the `full` PSI metrics for memory, indicating that the system is under significant memory pressure.

```
# Replace <node-name> with the name of a node in your cluster
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/stats/summary" | jq '.pods[] | select(.podRef.name | contains("memory-pressure-
```

Using the Prometheus metrics endpoint:

Query the `/metrics/cadvisor` endpoint to see the `container_pressure_memory_waiting_seconds_total` metric.

```
# Replace <node-name> with the name of the node where the pod is running
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/metrics/cadvisor" | \
    grep 'container_pressure_memory_waiting_seconds_total{container="memory-stress"'
```

In the output, you will observe an increasing value for the metric, indicating that the system is under significant memory pressure.

Cleanup

Clean up the Pod when you are finished:

```
kubectl delete pod memory-pressure-pod
```

Generating I/O Pressure

This Pod generates I/O pressure by repeatedly writing a file to disk and using `sync` to flush the data from memory, which creates I/O stalls.

Create a file named `io-pressure-pod.yaml`:

```
apiVersion: v1
kind: Podmetadata:  name: io-pressure-podspec:  restartPolicy: Never  containers:  - name: io-stress    image: registry.k8s.io/e2e
```

Apply this to your cluster: `kubectl apply -f io-pressure-pod.yaml`

Observing I/O Pressure

Using the Summary API:

You will see the `some` PSI metrics for I/O increase as the Pod continuously writes to disk.

```
# Replace <node-name> with the name of a node in your cluster
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/stats/summary" | jq '.pods[] | select(.podRef.name | contains("io-pressure-pod"
```

Using the Prometheus metrics endpoint:

Query the `/metrics/cadvisor` endpoint to see the `container_pressure_io_waiting_seconds_total` metric.

```
# Replace <node-name> with the name of the node where the pod is running
kubectl get --raw "/api/v1/nodes/<node-name>/proxy/metrics/cadvisor" | \
    grep 'container_pressure_io_waiting_seconds_total{container="io-stress"'
```

You will see the metric's value increase as the Pod continuously writes to disk.

Cleanup

Clean up the Pod when you are finished:

```
kubectl delete pod io-pressure-pod
```

What's next

The task pages for Troubleshooting Clusters discuss how to use a metrics pipeline that rely on these data.

Kubernetes Metrics Reference

Details of the metric data that Kubernetes components export.

Metrics (v1.34)

This page details the metrics that different Kubernetes components export. You can query the metrics endpoint for these components using an HTTP scrape, and fetch the current metrics data in Prometheus format.

List of Stable Kubernetes Metrics

Stable metrics observe strict API contracts and no labels can be added or removed from stable metrics during their lifetime.

apiserver_admission_controller_admission_duration_seconds
Admission controller latency histogram in seconds, identified by name and broken out for each operation and API resource and type (validate or admit).

- Stability Level:STABLE
- Type: Histogram
- Labels:nameoperationrejectedtype

apiserver_admission_step_admission_duration_seconds
Admission sub-step latency histogram in seconds, broken out for each operation and API resource and step type (validate or admit).

- Stability Level:STABLE
- Type: Histogram
- Labels:operationrejectedtype

apiserver_admission_webhook_admission_duration_seconds
Admission webhook latency histogram in seconds, identified by name and broken out for each operation and API resource and type (validate or admit).

- Stability Level:STABLE
- Type: Histogram
- Labels:nameoperationrejectedtype

apiserver_current_inflight_requests
Maximal number of currently used inflight request limit of this apiserver per request kind in last second.

- Stability Level:STABLE
- Type: Gauge
- Labels:request_kind

apiserver_longrunning_requests
Gauge of all active long-running apiserver requests broken out by verb, group, version, resource, scope and component. Not all requests are tracked this way.

- Stability Level:STABLE
- Type: Gauge
- Labels:componentgroupresourcescopesubresourceverbversion

apiserver_request_duration_seconds
Response latency distribution in seconds for each verb, dry run value, group, version, resource, subresource, scope and component.

- Stability Level:STABLE
- Type: Histogram
- Labels:componentdry_rungroupresourcescopesubresourceverbversion

apiserver_request_total
Counter of apiserver requests broken out for each verb, dry run value, group, version, resource, scope, component, and HTTP response code.

- Stability Level:STABLE
- Type: Counter
- Labels:codecomponentdry_rungroupresourcescopesubresourceverbversion

apiserver_requested_deprecated_apis
Gauge of deprecated APIs that have been requested, broken out by API group, version, resource, subresource, and removed_release.

- Stability Level:STABLE
- Type: Gauge
- Labels:groupremoved_releaseresourcesubresourceversion

apiserver_response_sizes
Response size distribution in bytes for each group, version, verb, resource, subresource, scope and component.

- Stability Level:STABLE
- Type: Histogram
- Labels:componentgroupresourcescopesubresourceverbversion

apiserver_storage_objects
[DEPRECATED, consider using apiserver_resource_objects instead] Number of stored objects at the time of last check split by kind. In case of a fetching error, the value will be -1.

- Stability Level:STABLE
- Type: Gauge
- Labels:resource
- Deprecated Versions:1.34.0

apiserver_storage_size_bytes
Size of the storage database file physically allocated in bytes.

- Stability Level:STABLE

- Type: Custom
- Labels:storage_cluster_id

container_cpu_usage_seconds_total
Cumulative cpu time consumed by the container in core-seconds

- Stability Level:STABLE
- Type: Custom
- Labels:containerpodnamespace

container_memory_working_set_bytes
Current working set of the container in bytes

- Stability Level:STABLE
- Type: Custom
- Labels:containerpodnamespace

container_start_time_seconds
Start time of the container since unix epoch in seconds

- Stability Level:STABLE
- Type: Custom
- Labels:containerpodnamespace

cronjob_controller_job_creation_skew_duration_seconds
Time between when a cronjob is scheduled to be run, and when the corresponding job is created

- Stability Level:STABLE
- Type: Histogram

job_controller_job_pods_finished_total
The number of finished Pods that are fully tracked

- Stability Level:STABLE
- Type: Counter
- Labels:completion_moderesult

job_controller_job_sync_duration_seconds
The time it took to sync a job

- Stability Level:STABLE
- Type: Histogram
- Labels:actioncompletion_moderesult

job_controller_job_syncs_total
The number of job syncs

- Stability Level:STABLE
- Type: Counter
- Labels:actioncompletion_moderesult

job_controller_jobs_finished_total
The number of finished jobs

- Stability Level:STABLE
- Type: Counter
- Labels:completion_modereasonresult

kube_pod_resource_limit
Resources limit for workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.

- Stability Level:STABLE
- Type: Custom
- Labels:namespacepodnodeschedulerpriorityresourceunit

kube_pod_resource_request
Resources requested by workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.

- Stability Level:STABLE
- Type: Custom
- Labels:namespacepodnodeschedulerpriorityresourceunit

kubernetes_healthcheck
This metric records the result of a single healthcheck.

- Stability Level:STABLE
- Type: Gauge
- Labels:nametype

kubernetes_healthchecks_total
This metric records the results of all healthcheck.

- Stability Level:STABLE
- Type: Counter
- Labels:namestatustype

node_collector_evictions_total
Number of Node evictions that happened since current instance of NodeController started.

- Stability Level:STABLE
- Type: Counter
- Labels:zone

node_cpu_usage_seconds_total
Cumulative cpu time consumed by the node in core-seconds

- Stability Level:STABLE
- Type: Custom

node_memory_working_set_bytes
Current working set of the node in bytes

- Stability Level:STABLE
- Type: Custom

pod_cpu_usage_seconds_total
Cumulative cpu time consumed by the pod in core-seconds

- Stability Level:STABLE
- Type: Custom
- Labels:podnamespace

pod_memory_working_set_bytes
Current working set of the pod in bytes

- Stability Level:STABLE
- Type: Custom
- Labels:podnamespace

resource_scrape_error
1 if there was an error while getting container metrics, 0 otherwise

- Stability Level:STABLE
- Type: Custom

scheduler_framework_extension_point_duration_seconds
Latency for running all plugins of a specific extension point.

- Stability Level:STABLE
- Type: Histogram
- Labels:extension_pointprofilestatus

scheduler_pending_pods
Number of pending pods, by the queue type. 'active' means number of pods in activeQ; 'backoff' means number of pods in backoffQ; 'unschedulable' means number of pods in unschedulablePods that the scheduler attempted to schedule and failed; 'gated' is the number of unschedulable pods that the scheduler never attempted to schedule because they are gated.

- Stability Level:STABLE
- Type: Gauge
- Labels:queue

scheduler_pod_scheduling_attempts
Number of attempts to successfully schedule a pod.

- Stability Level:STABLE
- Type: Histogram

scheduler_preemption_attempts_total
Total preemption attempts in the cluster till now

- Stability Level:STABLE
- Type: Counter

scheduler_preemption_victims
Number of selected preemption victims

- Stability Level:STABLE
- Type: Histogram

scheduler_queue_incoming_pods_total
Number of pods added to scheduling queues by event and queue type.

- Stability Level:STABLE
- Type: Counter
- Labels:eventqueue

scheduler_schedule_attempts_total
Number of attempts to schedule pods, by the result. 'unschedulable' means a pod could not be scheduled, while 'error' means an internal scheduler problem.

- Stability Level:STABLE
- Type: Counter
- Labels:profileresult

scheduler_scheduling_attempt_duration_seconds
Scheduling attempt latency in seconds (scheduling algorithm + binding)

- Stability Level:STABLE
- Type: Histogram
- Labels:profileresult

List of Beta Kubernetes Metrics

Beta metrics observe a looser API contract than its stable counterparts. No labels can be removed from beta metrics during their lifetime, however, labels can be added while the metric is in the beta stage. This offers the assurance that beta metrics will honor existing dashboards and alerts, while allowing for amendments in the future.

apiserver_authentication_config_controller_automatic_reload_last_timestamp_seconds
Timestamp of the last automatic reload of authentication configuration split by status and apiserver identity.

- Stability Level:BETA
- Type: Gauge
- Labels:apiserver_id_hashstatus

apiserver_authentication_config_controller_automatic_reloads_total
Total number of automatic reloads of authentication configuration split by status and apiserver identity.

- Stability Level:BETA
- Type: Counter
- Labels:apiserver_id_hashstatus

apiserver_authorization_config_controller_automatic_reload_last_timestamp_seconds
Timestamp of the last automatic reload of authorization configuration split by status and apiserver identity.

- Stability Level:BETA
- Type: Gauge
- Labels:apiserver_id_hashstatus

apiserver_authorization_config_controller_automatic_reloads_total
Total number of automatic reloads of authorization configuration split by status and apiserver identity.

- Stability Level:BETA
- Type: Counter
- Labels:apiserver_id_hashstatus

apiserver_cel_compilation_duration_seconds
CEL compilation time in seconds.

- Stability Level:BETA
- Type: Histogram

apiserver_cel_evaluation_duration_seconds
CEL evaluation time in seconds.

- Stability Level:BETA
- Type: Histogram

apiserver_flowcontrol_current_executing_requests
Number of requests in initial (for a WATCH) or any (for a non-WATCH) execution stage in the API Priority and Fairness subsystem

- Stability Level:BETA
- Type: Gauge
- Labels:flow_schemapriority_level

apiserver_flowcontrol_current_executing_seats
Concurrency (number of seats) occupied by the currently executing (initial stage for a WATCH, any stage otherwise) requests in the API Priority and Fairness subsystem

- Stability Level:BETA
- Type: Gauge
- Labels:flow_schemapriority_level

apiserver_flowcontrol_current_inqueue_requests
Number of requests currently pending in queues of the API Priority and Fairness subsystem

- Stability Level:BETA
- Type: Gauge
- Labels:flow_schemapriority_level

apiserver_flowcontrol_dispatched_requests_total

Number of requests executed by API Priority and Fairness subsystem

- Stability Level:BETA
- Type: Counter
- Labels:flow_schemapriority_level

apiserver_flowcontrol_nominal_limit_seats
Nominal number of execution seats configured for each priority level

- Stability Level:BETA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_rejected_requests_total
Number of requests rejected by API Priority and Fairness subsystem

- Stability Level:BETA
- Type: Counter
- Labels:flow_schemapriority_levelreason

apiserver_flowcontrol_request_wait_duration_seconds
Length of time a request spent waiting in its queue

- Stability Level:BETA
- Type: Histogram
- Labels:executeflow_schemapriority_level

apiserver_validating_admission_policy_check_duration_seconds
Validation admission latency for individual validation expressions in seconds, labeled by policy and further including binding and enforcement action taken.

- Stability Level:BETA
- Type: Histogram
- Labels:enforcement_actionerror_typepolicypolicy_binding

apiserver_validating_admission_policy_check_total
Validation admission policy check total, labeled by policy and further identified by binding and enforcement action taken.

- Stability Level:BETA
- Type: Counter
- Labels:enforcement_actionerror_typepolicypolicy_binding

apiserver_validation_declarative_validation_mismatch_total
Number of times declarative validation results differed from handwritten validation results for core types.

- Stability Level:BETA
- Type: Counter

apiserver_validation_declarative_validation_panic_total
Number of times declarative validation has panicked during validation.

- Stability Level:BETA
- Type: Counter

disabled_metrics_total
The count of disabled metrics.

- Stability Level:BETA
- Type: Counter

hidden_metrics_total
The count of hidden metrics.

- Stability Level:BETA
- Type: Counter

kubernetes_feature_enabled
This metric records the data about the stage and enablement of a k8s feature.

- Stability Level:BETA
- Type: Gauge
- Labels:namestage

prober_probe_total
Cumulative number of a liveness, readiness or startup probe for a container by result.

- Stability Level:BETA
- Type: Counter
- Labels:containernamespacepodpod_uidprobe_typeresult

registered_metrics_total
The count of registered metrics broken by stability level and deprecation version.

- Stability Level:BETA
- Type: Counter

- Labels:deprecated_versionstability_level

scheduler_pod_scheduling_sli_duration_seconds
E2e latency for a pod being scheduled, from the time the pod enters the scheduling queue and might involve multiple scheduling attempts.

- Stability Level:BETA
- Type: Histogram
- Labels:attempts

List of Alpha Kubernetes Metrics

Alpha metrics do not have any API guarantees. These metrics must be used at your own risk, subsequent versions of Kubernetes may remove these metrics altogether, or mutate the API in such a way that breaks existing dashboards and alerts.

aggregator_discovery_aggregation_count_total
Counter of number of times discovery was aggregated

- Stability Level:ALPHA
- Type: Counter

aggregator_openapi_v2_regeneration_count
Counter of OpenAPI v2 spec regeneration count broken down by causing APIService name and reason.

- Stability Level:ALPHA
- Type: Counter
- Labels:apiservicereason

aggregator_openapi_v2_regeneration_duration
Gauge of OpenAPI v2 spec regeneration duration in seconds.

- Stability Level:ALPHA
- Type: Gauge
- Labels:reason

aggregator_unavailable_apiservice
Gauge of APIServices which are marked as unavailable broken down by APIService name.

- Stability Level:ALPHA
- Type: Custom
- Labels:name

aggregator_unavailable_apiservice_total
Counter of APIServices which are marked as unavailable broken down by APIService name and reason.

- Stability Level:ALPHA
- Type: Counter
- Labels:namereason

apiextensions_apiserver_validation_ratcheting_seconds
Time for comparison of old to new for the purposes of CRDValidationRatcheting during an UPDATE in seconds.

- Stability Level:ALPHA
- Type: Histogram

apiextensions_openapi_v2_regeneration_count
Counter of OpenAPI v2 spec regeneration count broken down by causing CRD name and reason.

- Stability Level:ALPHA
- Type: Counter
- Labels:crdreason

apiextensions_openapi_v3_regeneration_count
Counter of OpenAPI v3 spec regeneration count broken down by group, version, causing CRD and reason.

- Stability Level:ALPHA
- Type: Counter
- Labels:crdgroupreasonversion

apiserver_admission_match_condition_evaluation_errors_total
Admission match condition evaluation errors count, identified by name of resource containing the match condition and broken out for each kind containing matchConditions (webhook or policy), operation and admission type (validate or admit).

- Stability Level:ALPHA
- Type: Counter
- Labels:kindnameoperationtype

apiserver_admission_match_condition_evaluation_seconds
Admission match condition evaluation time in seconds, identified by name and broken out for each kind containing matchConditions (webhook or policy), operation and type (validate or admit).

- Stability Level:ALPHA
- Type: Histogram
- Labels:kindnameoperationtype

apiserver_admission_match_condition_exclusions_total
Admission match condition evaluation exclusions count, identified by name of resource containing the match condition and broken out for each kind containing matchConditions (webhook or policy), operation and admission type (validate or admit).

- Stability Level:ALPHA
- Type: Counter
- Labels:kindnameoperationtype

apiserver_admission_step_admission_duration_seconds_summary
Admission sub-step latency summary in seconds, broken out for each operation and API resource and step type (validate or admit).

- Stability Level:ALPHA
- Type: Summary
- Labels:operationrejectedtype

apiserver_admission_webhook_fail_open_count
Admission webhook fail open count, identified by name and broken out for each admission type (validating or admit).

- Stability Level:ALPHA
- Type: Counter
- Labels:nametype

apiserver_admission_webhook_rejection_count
Admission webhook rejection count, identified by name and broken out for each admission type (validating or admit) and operation. Additional labels specify an error type (calling_webhook_error or apiserver_internal_error if an error occurred; no_error otherwise) and optionally a non-zero rejection code if the webhook rejects the request with an HTTP status code (honored by the apiserver when the code is greater or equal to 400). Codes greater than 600 are truncated to 600, to keep the metrics cardinality bounded.

- Stability Level:ALPHA
- Type: Counter
- Labels:error_typenameoperationrejection_codetype

apiserver_admission_webhook_request_total
Admission webhook request total, identified by name and broken out for each admission type (validating or admit) and operation. Additional labels specify whether the request was rejected or not and an HTTP status code. Codes greater than 600 are truncated to 600, to keep the metrics cardinality bounded.

- Stability Level:ALPHA
- Type: Counter
- Labels:codenameoperationrejectedtype

apiserver_audit_error_total
Counter of audit events that failed to be audited properly. Plugin identifies the plugin affected by the error.

- Stability Level:ALPHA
- Type: Counter
- Labels:plugin

apiserver_audit_event_total
Counter of audit events generated and sent to the audit backend.

- Stability Level:ALPHA
- Type: Counter

apiserver_audit_level_total
Counter of policy levels for audit events (1 per request).

- Stability Level:ALPHA
- Type: Counter
- Labels:level

apiserver_audit_requests_rejected_total
Counter of apiserver requests rejected due to an error in audit logging backend.

- Stability Level:ALPHA
- Type: Counter

apiserver_authentication_config_controller_last_config_info
Information about the last applied authentication configuration with hash as label, split by apiserver identity.

- Stability Level:ALPHA
- Type: Custom
- Labels:apiserver_id_hashhash

apiserver_authentication_jwt_authenticator_latency_seconds
Latency of jwt authentication operations in seconds. This is the time spent authenticating a token for cache miss only (i.e. when the token is not found in the cache).

- Stability Level:ALPHA
- Type: Histogram
- Labels:jwt_issuer_hashresult

apiserver_authorization_config_controller_last_config_info
Information about the last applied authorization configuration with hash as label, split by apiserver identity.

- Stability Level:ALPHA
- Type: Custom
- Labels:apiserver_id_hashhash

apiserver_authorization_decisions_total
Total number of terminal decisions made by an authorizer split by authorizer type, name, and decision.

- Stability Level:ALPHA
- Type: Counter
- Labels:decisionnametype

apiserver_authorization_match_condition_evaluation_errors_total
Total number of errors when an authorization webhook encounters a match condition error split by authorizer type and name.

- Stability Level:ALPHA
- Type: Counter
- Labels:nametype

apiserver_authorization_match_condition_evaluation_seconds
Authorization match condition evaluation time in seconds, split by authorizer type and name.

- Stability Level:ALPHA
- Type: Histogram
- Labels:nametype

apiserver_authorization_match_condition_exclusions_total
Total number of exclusions when an authorization webhook is skipped because match conditions exclude it.

- Stability Level:ALPHA
- Type: Counter
- Labels:nametype

apiserver_authorization_webhook_duration_seconds
Request latency in seconds.

- Stability Level:ALPHA
- Type: Histogram
- Labels:nameresult

apiserver_authorization_webhook_evaluations_fail_open_total
NoOpinion results due to webhook timeout or error.

- Stability Level:ALPHA
- Type: Counter
- Labels:nameresult

apiserver_authorization_webhook_evaluations_total
Round-trips to authorization webhooks.

- Stability Level:ALPHA
- Type: Counter
- Labels:nameresult

apiserver_cache_list_fetched_objects_total
Number of objects read from watch cache in the course of serving a LIST request

- Stability Level:ALPHA
- Type: Counter
- Labels:groupindexresource

apiserver_cache_list_returned_objects_total
Number of objects returned for a LIST request from watch cache

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_cache_list_total
Number of LIST requests served from watch cache

- Stability Level:ALPHA
- Type: Counter
- Labels:groupindexresource

apiserver_certificates_registry_csr_honored_duration_total
Total number of issued CSRs with a requested duration that was honored, sliced by signer (only kubernetes.io signer names are specifically identified)

- Stability Level:ALPHA
- Type: Counter
- Labels:signerName

apiserver_certificates_registry_csr_requested_duration_total
Total number of issued CSRs with a requested duration, sliced by signer (only kubernetes.io signer names are specifically identified)

- Stability Level:ALPHA
- Type: Counter
- Labels:signerName

apiserver_client_certificate_expiration_seconds
Distribution of the remaining lifetime on the certificate used to authenticate a request.

- Stability Level:ALPHA
- Type: Histogram

apiserver_clusterip_repair_ip_errors_total
Number of errors detected on clusterips by the repair loop broken down by type of error: leak, repair, full, outOfRange, duplicate, unknown, invalid

- Stability Level:ALPHA
- Type: Counter
- Labels:type

apiserver_clusterip_repair_reconcile_errors_total
Number of reconciliation failures on the clusterip repair reconcile loop

- Stability Level:ALPHA
- Type: Counter

apiserver_conversion_webhook_duration_seconds
Conversion webhook request latency

- Stability Level:ALPHA
- Type: Histogram
- Labels:failure_typeresult

apiserver_conversion_webhook_request_total
Counter for conversion webhook requests with success/failure and failure error type

- Stability Level:ALPHA
- Type: Counter
- Labels:failure_typeresult

apiserver_crd_conversion_webhook_duration_seconds
CRD webhook conversion duration in seconds

- Stability Level:ALPHA
- Type: Histogram
- Labels:crd_namefrom_versionsucceededto_version

apiserver_current_inqueue_requests
Maximal number of queued requests in this apiserver per request kind in last second.

- Stability Level:ALPHA
- Type: Gauge
- Labels:request_kind

apiserver_delegated_authn_request_duration_seconds
Request latency in seconds. Broken down by status code.

- Stability Level:ALPHA
- Type: Histogram
- Labels:code

apiserver_delegated_authn_request_total
Number of HTTP requests partitioned by status code.

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_delegated_authz_request_duration_seconds
Request latency in seconds. Broken down by status code.

- Stability Level:ALPHA
- Type: Histogram
- Labels:code

apiserver_delegated_authz_request_total
Number of HTTP requests partitioned by status code.

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_egress_dialer_dial_duration_seconds
Dial latency histogram in seconds, labeled by the protocol (http-connect or grpc), transport (tcp or uds)

- Stability Level:ALPHA
- Type: Histogram

- Labels:protocoltransport

apiserver_egress_dialer_dial_failure_count
Dial failure count, labeled by the protocol (http-connect or grpc), transport (tcp or uds), and stage (connect or proxy). The stage indicates at which stage the dial failed

- Stability Level:ALPHA
- Type: Counter
- Labels:protocolstagetransport

apiserver_egress_dialer_dial_start_total
Dial starts, labeled by the protocol (http-connect or grpc) and transport (tcp or uds).

- Stability Level:ALPHA
- Type: Counter
- Labels:protocoltransport

apiserver_encryption_config_controller_automatic_reload_last_timestamp_seconds
Timestamp of the last successful or failed automatic reload of encryption configuration split by apiserver identity.

- Stability Level:ALPHA
- Type: Gauge
- Labels:apiserver_id_hashstatus

apiserver_encryption_config_controller_automatic_reloads_total
Total number of reload successes and failures of encryption configuration split by apiserver identity.

- Stability Level:ALPHA
- Type: Counter
- Labels:apiserver_id_hashstatus

apiserver_encryption_config_controller_last_config_info
Information about the last applied encryption configuration with hash as label, split by apiserver identity.

- Stability Level:ALPHA
- Type: Custom
- Labels:apiserver_id_hashhash

apiserver_envelope_encryption_dek_cache_fill_percent
Percent of the cache slots currently occupied by cached DEKs.

- Stability Level:ALPHA
- Type: Gauge

apiserver_envelope_encryption_dek_cache_inter_arrival_time_seconds
Time (in seconds) of inter arrival of transformation requests.

- Stability Level:ALPHA
- Type: Histogram
- Labels:transformation_type

apiserver_envelope_encryption_dek_source_cache_size
Number of records in data encryption key (DEK) source cache. On a restart, this value is an approximation of the number of decrypt RPC calls the server will make to the KMS plugin.

- Stability Level:ALPHA
- Type: Gauge
- Labels:provider_name

apiserver_envelope_encryption_invalid_key_id_from_status_total
Number of times an invalid keyID is returned by the Status RPC call split by error.

- Stability Level:ALPHA
- Type: Counter
- Labels:errorprovider_name

apiserver_envelope_encryption_key_id_hash_last_timestamp_seconds
The last time in seconds when a keyID was used.

- Stability Level:ALPHA
- Type: Gauge
- Labels:apiserver_id_hashkey_id_hashprovider_nametransformation_type

apiserver_envelope_encryption_key_id_hash_status_last_timestamp_seconds
The last time in seconds when a keyID was returned by the Status RPC call.

- Stability Level:ALPHA
- Type: Gauge
- Labels:apiserver_id_hashkey_id_hashprovider_name

apiserver_envelope_encryption_key_id_hash_total
Number of times a keyID is used split by transformation type, provider, and apiserver identity.

- Stability Level:ALPHA

- Type: Counter
- Labels:apiserver_id_hashkey_id_hashprovider_nametransformation_type

apiserver_envelope_encryption_kms_operations_latency_seconds
KMS operation duration with gRPC error code status total.

- Stability Level:ALPHA
- Type: Histogram
- Labels:grpc_status_codemethod_nameprovider_name

apiserver_externaljwt_fetch_keys_data_timestamp
Unix Timestamp in seconds of the last successful FetchKeys data_timestamp value returned by the external signer

- Stability Level:ALPHA
- Type: Gauge

apiserver_externaljwt_fetch_keys_request_total
Total attempts at syncing supported JWKs

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_externaljwt_fetch_keys_success_timestamp
Unix Timestamp in seconds of the last successful FetchKeys request

- Stability Level:ALPHA
- Type: Gauge

apiserver_externaljwt_request_duration_seconds
Request duration and time for calls to external-jwt-signer

- Stability Level:ALPHA
- Type: Histogram
- Labels:codemethod

apiserver_externaljwt_sign_request_total
Total attempts at signing JWT

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_flowcontrol_current_inqueue_seats
Number of seats currently pending in queues of the API Priority and Fairness subsystem

- Stability Level:ALPHA
- Type: Gauge
- Labels:flow_schemapriority_level

apiserver_flowcontrol_current_limit_seats
current derived number of execution seats available to each priority level

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_current_r
R(time of last change)

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_demand_seats
Observations, at the end of every nanosecond, of (the number of seats each priority level could use) / (nominal number of seats for that level)

- Stability Level:ALPHA
- Type: TimingRatioHistogram
- Labels:priority_level

apiserver_flowcontrol_demand_seats_average
Time-weighted average, over last adjustment period, of demand_seats

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_demand_seats_high_watermark
High watermark, over last adjustment period, of demand_seats

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_demand_seats_smoothed
Smoothed seat demands

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_demand_seats_stdev
Time-weighted standard deviation, over last adjustment period, of demand_seats

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_dispatch_r
R(time of last dispatch)

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_epoch_advance_total
Number of times the queueset's progress meter jumped backward

- Stability Level:ALPHA
- Type: Counter
- Labels:priority_levelsuccess

apiserver_flowcontrol_latest_s
S(most recently dispatched request)

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_lower_limit_seats
Configured lower bound on number of execution seats available to each priority level

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_next_discounted_s_bounds
min and max, over queues, of S(oldest waiting request in queue) - estimated work in progress

- Stability Level:ALPHA
- Type: Gauge
- Labels:boundpriority_level

apiserver_flowcontrol_next_s_bounds
min and max, over queues, of S(oldest waiting request in queue)

- Stability Level:ALPHA
- Type: Gauge
- Labels:boundpriority_level

apiserver_flowcontrol_priority_level_request_utilization
Observations, at the end of every nanosecond, of number of requests (as a fraction of the relevant limit) waiting or in any stage of execution (but only initial stage for WATCHes)

- Stability Level:ALPHA
- Type: TimingRatioHistogram
- Labels:phasepriority_level

apiserver_flowcontrol_priority_level_seat_utilization
Observations, at the end of every nanosecond, of utilization of seats for any stage of execution (but only initial stage for WATCHes)

- Stability Level:ALPHA
- Type: TimingRatioHistogram
- Labels:priority_level
- Const Labels:phase:executing

apiserver_flowcontrol_read_vs_write_current_requests
Observations, at the end of every nanosecond, of the number of requests (as a fraction of the relevant limit) waiting or in regular stage of execution

- Stability Level:ALPHA
- Type: TimingRatioHistogram
- Labels:phaserequest_kind

apiserver_flowcontrol_request_concurrency_in_use
Concurrency (number of seats) occupied by the currently executing (initial stage for a WATCH, any stage otherwise) requests in the API Priority and Fairness subsystem

- Stability Level:ALPHA
- Type: Gauge
- Labels:flow_schemapriority_level
- Deprecated Versions:1.31.0

apiserver_flowcontrol_request_concurrency_limit
Nominal number of execution seats configured for each priority level

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level
- Deprecated Versions:1.30.0

apiserver_flowcontrol_request_dispatch_no_accommodation_total
Number of times a dispatch attempt resulted in a non accommodation due to lack of available seats

- Stability Level:ALPHA
- Type: Counter
- Labels:flow_schemapriority_level

apiserver_flowcontrol_request_execution_seconds
Duration of initial stage (for a WATCH) or any (for a non-WATCH) stage of request execution in the API Priority and Fairness subsystem

- Stability Level:ALPHA
- Type: Histogram
- Labels:flow_schemapriority_leveltype

apiserver_flowcontrol_request_queue_length_after_enqueue
Length of queue in the API Priority and Fairness subsystem, as seen by each request after it is enqueued

- Stability Level:ALPHA
- Type: Histogram
- Labels:flow_schemapriority_level

apiserver_flowcontrol_seat_fair_frac
Fair fraction of server's concurrency to allocate to each priority level that can use it

- Stability Level:ALPHA
- Type: Gauge

apiserver_flowcontrol_target_seats
Seat allocation targets

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_upper_limit_seats
Configured upper bound on number of execution seats available to each priority level

- Stability Level:ALPHA
- Type: Gauge
- Labels:priority_level

apiserver_flowcontrol_watch_count_samples
count of watchers for mutating requests in API Priority and Fairness

- Stability Level:ALPHA
- Type: Histogram
- Labels:flow_schemapriority_level

apiserver_flowcontrol_work_estimated_seats
Number of estimated seats (maximum of initial and final seats) associated with requests in API Priority and Fairness

- Stability Level:ALPHA
- Type: Histogram
- Labels:flow_schemapriority_level

apiserver_init_events_total
Counter of init events processed in watch cache broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_kube_aggregator_x509_insecure_sha1_total
Counts the number of requests to servers with insecure SHA1 signatures in their serving certificate OR the number of connection failures due to the insecure SHA1 signatures (either/or, based on the runtime environment)

- Stability Level:ALPHA
- Type: Counter

apiserver_kube_aggregator_x509_missing_san_total

Counts the number of requests to servers missing SAN extension in their serving certificate OR the number of connection failures due to the lack of x509 certificate SAN extension missing (either/or, based on the runtime environment)

- Stability Level:ALPHA
- Type: Counter

apiserver_mutating_admission_policy_check_duration_seconds
Mutation admission latency for individual mutation expressions in seconds, labeled by policy and binding.

- Stability Level:ALPHA
- Type: Histogram
- Labels:error_typepolicypolicy_binding

apiserver_mutating_admission_policy_check_total
Mutation admission policy check total, labeled by policy and further identified by binding.

- Stability Level:ALPHA
- Type: Counter
- Labels:error_typepolicypolicy_binding

apiserver_nodeport_repair_port_errors_total
Number of errors detected on ports by the repair loop broken down by type of error: leak, repair, full, outOfRange, duplicate, unknown

- Stability Level:ALPHA
- Type: Counter
- Labels:type

apiserver_nodeport_repair_reconcile_errors_total
Number of reconciliation failures on the nodeport repair reconcile loop

- Stability Level:ALPHA
- Type: Counter

apiserver_request_aborts_total
Number of requests which apiserver aborted possibly due to a timeout, for each group, version, verb, resource, subresource and scope

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresourcescopesubresourceverbversion

apiserver_request_body_size_bytes
Apiserver request body size in bytes broken out by resource and verb.

- Stability Level:ALPHA
- Type: Histogram
- Labels:groupresourceverb

apiserver_request_filter_duration_seconds
Request filter latency distribution in seconds, for each filter type

- Stability Level:ALPHA
- Type: Histogram
- Labels:filter

apiserver_request_post_timeout_total
Tracks the activity of the request handlers after the associated requests have been timed out by the apiserver

- Stability Level:ALPHA
- Type: Counter
- Labels:sourcestatus

apiserver_request_sli_duration_seconds
Response latency distribution (not counting webhook duration and priority & fairness queue wait times) in seconds for each verb, group, version, resource, subresource, scope and component.

- Stability Level:ALPHA
- Type: Histogram
- Labels:componentgroupresourcescopesubresourceverbversion

apiserver_request_slo_duration_seconds
Response latency distribution (not counting webhook duration and priority & fairness queue wait times) in seconds for each verb, group, version, resource, subresource, scope and component.

- Stability Level:ALPHA
- Type: Histogram
- Labels:componentgroupresourcescopesubresourceverbversion
- Deprecated Versions:1.27.0

apiserver_request_terminations_total
Number of requests which apiserver terminated in self-defense.

- Stability Level:ALPHA
- Type: Counter
- Labels:codecomponentgroupresourcescopesubresourceverbversion

apiserver_request_timestamp_comparison_time
Time taken for comparison of old vs new objects in UPDATE or PATCH requests

- Stability Level:ALPHA
- Type: Histogram
- Labels:code_path

apiserver_rerouted_request_total
Total number of requests that were proxied to a peer kube apiserver because the local apiserver was not capable of serving it

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_resource_objects
Number of stored objects at the time of last check split by kind. In case of a fetching error, the value will be -1.

- Stability Level:ALPHA
- Type: Gauge
- Labels:groupresource

apiserver_resource_size_estimate_bytes
Estimated size of stored objects in database. Estimate is based on sum of last observed sizes of serialized objects. In case of a fetching error, the value will be -1.

- Stability Level:ALPHA
- Type: Gauge
- Labels:groupresource

apiserver_selfrequest_total
Counter of apiserver self-requests broken out for each verb, API resource and subresource.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresourcesubresourceverb

apiserver_storage_consistency_checks_total
Counter for status of consistency checks between etcd and watch cache

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresourcestatus

apiserver_storage_data_key_generation_duration_seconds
Latencies in seconds of data encryption key(DEK) generation operations.

- Stability Level:ALPHA
- Type: Histogram

apiserver_storage_data_key_generation_failures_total
Total number of failed data encryption key(DEK) generation operations.

- Stability Level:ALPHA
- Type: Counter

apiserver_storage_db_total_size_in_bytes
Total size of the storage database file physically allocated in bytes.

- Stability Level:ALPHA
- Type: Gauge
- Labels:endpoint
- Deprecated Versions:1.28.0

apiserver_storage_decode_errors_total
Number of stored object decode errors split by object type

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_envelope_transformation_cache_misses_total
Total number of cache misses while accessing key decryption key(KEK).

- Stability Level:ALPHA
- Type: Counter

apiserver_storage_events_received_total
Number of etcd events received split by kind.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_list_evaluated_objects_total
Number of objects tested in the course of serving a LIST request from storage

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_list_fetched_objects_total
Number of objects read from storage in the course of serving a LIST request

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_list_returned_objects_total
Number of objects returned for a LIST request from storage

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_list_total
Number of LIST requests served from storage

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_storage_transformation_duration_seconds
Latencies in seconds of value transformation operations.

- Stability Level:ALPHA
- Type: Histogram
- Labels:transformation_typetransformer_prefix

apiserver_storage_transformation_operations_total
Total number of transformations. Successful transformation will have a status 'OK' and a varied status string when the transformation fails. The status, resource, and transformation_type fields can be used for alerting purposes. For example, you can monitor for encryption/decryption failures using the transformation_type (e.g., from_storage for decryption and to_storage for encryption). Additionally, these fields can be used to ensure that the correct transformers are applied to each resource.

- Stability Level:ALPHA
- Type: Counter
- Labels:resourcestatustransformation_typetransformer_prefix

apiserver_stream_translator_requests_total
Total number of requests that were handled by the StreamTranslatorProxy, which processes streaming RemoteCommand/V5

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_stream_tunnel_requests_total
Total number of requests that were handled by the StreamTunnelProxy, which processes streaming PortForward/V2

- Stability Level:ALPHA
- Type: Counter
- Labels:code

apiserver_terminated_watchers_total
Counter of watchers closed due to unresponsiveness broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_tls_handshake_errors_total
Number of requests dropped with 'TLS handshake error from' error

- Stability Level:ALPHA
- Type: Counter

apiserver_watch_cache_consistent_read_total
Counter for consistent reads from cache.

- Stability Level:ALPHA
- Type: Counter
- Labels:fallbackgroupresourcesuccess

apiserver_watch_cache_events_dispatched_total
Counter of events dispatched in watch cache broken by resource type.

- Stability Level:ALPHA

- Type: Counter
- Labels:groupresource

apiserver_watch_cache_events_received_total
Counter of events received in watch cache broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_watch_cache_initializations_total
Counter of watch cache initializations broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

apiserver_watch_cache_read_wait_seconds
Histogram of time spent waiting for a watch cache to become fresh.

- Stability Level:ALPHA
- Type: Histogram
- Labels:groupresource

apiserver_watch_cache_resource_version
Current resource version of watch cache broken by resource type.

- Stability Level:ALPHA
- Type: Gauge
- Labels:groupresource

apiserver_watch_events_sizes
Watch event size distribution in bytes

- Stability Level:ALPHA
- Type: Histogram
- Labels:groupresourceversion

apiserver_watch_events_total
Number of events sent in watch clients

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresourceversion

apiserver_watch_list_duration_seconds
Response latency distribution in seconds for watch list requests broken by group, version, resource and scope.

- Stability Level:ALPHA
- Type: Histogram
- Labels:groupresourcescopeversion

apiserver_webhooks_x509_insecure_sha1_total
Counts the number of requests to servers with insecure SHA1 signatures in their serving certificate OR the number of connection failures due to the insecure SHA1 signatures (either/or, based on the runtime environment)

- Stability Level:ALPHA
- Type: Counter

apiserver_webhooks_x509_missing_san_total
Counts the number of requests to servers missing SAN extension in their serving certificate OR the number of connection failures due to the lack of x509 certificate SAN extension missing (either/or, based on the runtime environment)

- Stability Level:ALPHA
- Type: Counter

attach_detach_controller_attachdetach_controller_forced_detaches
Number of times the A/D Controller performed a forced detach

- Stability Level:ALPHA
- Type: Counter
- Labels:reason

attachdetach_controller_total_volumes
Number of volumes in A/D Controller

- Stability Level:ALPHA
- Type: Custom
- Labels:plugin_namestate

authenticated_user_requests
Counter of authenticated requests broken out by username.

- Stability Level:ALPHA

- Type: Counter
- Labels:username

authentication_attempts
Counter of authenticated attempts.

- Stability Level:ALPHA
- Type: Counter
- Labels:result

authentication_duration_seconds
Authentication duration in seconds broken out by result.

- Stability Level:ALPHA
- Type: Histogram
- Labels:result

authentication_token_cache_active_fetch_count

- Stability Level:ALPHA
- Type: Gauge
- Labels:status

authentication_token_cache_fetch_total

- Stability Level:ALPHA
- Type: Counter
- Labels:status

authentication_token_cache_request_duration_seconds

- Stability Level:ALPHA
- Type: Histogram
- Labels:status

authentication_token_cache_request_total

- Stability Level:ALPHA
- Type: Counter
- Labels:status

authorization_attempts_total
Counter of authorization attempts broken down by result. It can be either 'allowed', 'denied', 'no-opinion' or 'error'.

- Stability Level:ALPHA
- Type: Counter
- Labels:result

authorization_duration_seconds
Authorization duration in seconds broken out by result.

- Stability Level:ALPHA
- Type: Histogram
- Labels:result

cloud_provider_webhook_request_duration_seconds
Request latency in seconds. Broken down by status code.

- Stability Level:ALPHA
- Type: Histogram
- Labels:codewebhook

cloud_provider_webhook_request_total
Number of HTTP requests partitioned by status code.

- Stability Level:ALPHA
- Type: Counter
- Labels:codewebhook

clustertrustbundle_publisher_sync_duration_seconds
The time it took to sync a cluster trust bundle.

- Stability Level:ALPHA
- Type: Histogram
- Labels:code

clustertrustbundle_publisher_sync_total
Number of syncs that occurred in cluster trust bundle publisher.

- Stability Level:ALPHA
- Type: Counter
- Labels:code

container_swap_limit_bytes

Current amount of the container swap limit in bytes. Reported only on non-windows systems

- Stability Level:ALPHA
- Type: Custom
- Labels:containerpodnamespace

container_swap_usage_bytes
Current amount of the container swap usage in bytes. Reported only on non-windows systems

- Stability Level:ALPHA
- Type: Custom
- Labels:containerpodnamespace

csi_operations_seconds
Container Storage Interface operation duration with gRPC error code status total

- Stability Level:ALPHA
- Type: Histogram
- Labels:driver_namegrpc_status_codemethod_namemigrated

device_taint_eviction_controller_pod_deletion_duration_seconds
Latency, in seconds, between the time when a device taint effect has been activated and a Pod's deletion via DeviceTaintEvictionController.

- Stability Level:ALPHA
- Type: Histogram

device_taint_eviction_controller_pod_deletions_total
Total number of Pods deleted by DeviceTaintEvictionController since its start.

- Stability Level:ALPHA
- Type: Counter

dra_grpc_operations_duration_seconds
Duration in seconds of the DRA gRPC operations

- Stability Level:ALPHA
- Type: Histogram
- Labels:driver_namegrpc_status_codemethod_name

dra_operations_duration_seconds
Latency histogram in seconds for the duration of handling all ResourceClaims referenced by a pod when the pod starts or stops. Identified by the name of the operation (PrepareResources or UnprepareResources) and separated by the success of the operation. The number of failed operations is provided through the histogram's overall count.

- Stability Level:ALPHA
- Type: Histogram
- Labels:is_erroroperation_name

dra_resource_claims_in_use
The number of ResourceClaims that are currently in use on the node, by driver name (driver_name label value) and across all drivers (special value for driver_name). Note that the sum of all by-driver counts is not the total number of in-use ResourceClaims because the same ResourceClaim might use devices from different drivers. Instead, use the count for the driver_name.

- Stability Level:ALPHA
- Type: Custom
- Labels:driver_name

endpoint_slice_controller_changes
Number of EndpointSlice changes

- Stability Level:ALPHA
- Type: Counter
- Labels:operation

endpoint_slice_controller_desired_endpoint_slices
Number of EndpointSlices that would exist with perfect endpoint allocation

- Stability Level:ALPHA
- Type: Gauge

endpoint_slice_controller_endpoints_added_per_sync
Number of endpoints added on each Service sync

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_controller_endpoints_desired
Number of endpoints desired

- Stability Level:ALPHA
- Type: Gauge

endpoint_slice_controller_endpoints_removed_per_sync
Number of endpoints removed on each Service sync

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_controller_endpointslices_changed_per_sync
Number of EndpointSlices changed on each Service sync

- Stability Level:ALPHA
- Type: Histogram
- Labels:topologytraffic_distribution

endpoint_slice_controller_num_endpoint_slices
Number of EndpointSlices

- Stability Level:ALPHA
- Type: Gauge

endpoint_slice_controller_services_count_by_traffic_distribution
Number of Services using some specific trafficDistribution

- Stability Level:ALPHA
- Type: Gauge
- Labels:traffic_distribution

endpoint_slice_controller_syncs
Number of EndpointSlice syncs

- Stability Level:ALPHA
- Type: Counter
- Labels:result

endpoint_slice_mirroring_controller_addresses_skipped_per_sync
Number of addresses skipped on each Endpoints sync due to being invalid or exceeding MaxEndpointsPerSubset

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_mirroring_controller_changes
Number of EndpointSlice changes

- Stability Level:ALPHA
- Type: Counter
- Labels:operation

endpoint_slice_mirroring_controller_desired_endpoint_slices
Number of EndpointSlices that would exist with perfect endpoint allocation

- Stability Level:ALPHA
- Type: Gauge

endpoint_slice_mirroring_controller_endpoints_added_per_sync
Number of endpoints added on each Endpoints sync

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_mirroring_controller_endpoints_desired
Number of endpoints desired

- Stability Level:ALPHA
- Type: Gauge

endpoint_slice_mirroring_controller_endpoints_removed_per_sync
Number of endpoints removed on each Endpoints sync

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_mirroring_controller_endpoints_sync_duration
Duration of syncEndpoints() in seconds

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_mirroring_controller_endpoints_updated_per_sync
Number of endpoints updated on each Endpoints sync

- Stability Level:ALPHA
- Type: Histogram

endpoint_slice_mirroring_controller_num_endpoint_slices
Number of EndpointSlices

- Stability Level:ALPHA
- Type: Gauge

ephemeral_volume_controller_create_failures_total
Number of PersistentVolumeClaim creation requests

- Stability Level:ALPHA
- Type: Counter

ephemeral_volume_controller_create_total
Number of PersistentVolumeClaim creation requests

- Stability Level:ALPHA
- Type: Counter

etcd_bookmark_counts
Number of etcd bookmarks (progress notify events) split by kind.

- Stability Level:ALPHA
- Type: Gauge
- Labels:groupresource

etcd_lease_object_counts
Number of objects attached to a single etcd lease.

- Stability Level:ALPHA
- Type: Histogram

etcd_request_duration_seconds
Etcd request latency in seconds for each operation and object type.

- Stability Level:ALPHA
- Type: Histogram
- Labels:groupoperationresource

etcd_request_errors_total
Etcd failed request counts for each operation and object type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupoperationresource

etcd_requests_total
Etcd request counts for each operation and object type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupoperationresource

etcd_version_info
Etcd server's binary version

- Stability Level:ALPHA
- Type: Gauge
- Labels:binary_version

field_validation_request_duration_seconds
Response latency distribution in seconds for each field validation value

- Stability Level:ALPHA
- Type: Histogram
- Labels:field_validation

force_cleaned_failed_volume_operation_errors_total
The number of volumes that failed force cleanup after their reconstruction failed during kubelet startup.

- Stability Level:ALPHA
- Type: Counter

force_cleaned_failed_volume_operations_total
The number of volumes that were force cleaned after their reconstruction failed during kubelet startup. This includes both successful and failed cleanups.

- Stability Level:ALPHA
- Type: Counter

garbagecollector_controller_resources_sync_error_total
Number of garbage collector resources sync errors

- Stability Level:ALPHA
- Type: Counter

horizontal_pod_autoscaler_controller_metric_computation_duration_seconds
The time(seconds) that the HPA controller takes to calculate one metric. The label 'action' should be either 'scale_down', 'scale_up', or 'none'. The label 'error' should be either 'spec', 'internal', or 'none'. The label 'metric_type' corresponds to HPA.spec.metrics[*].type

- Stability Level:ALPHA
- Type: Histogram

- Labels:actionerrormetric_type

horizontal_pod_autoscaler_controller_metric_computation_total
Number of metric computations. The label 'action' should be either 'scale_down', 'scale_up', or 'none'. Also, the label 'error' should be either 'spec', 'internal', or 'none'. The label 'metric_type' corresponds to HPA.spec.metrics[*].type

- Stability Level:ALPHA
- Type: Counter
- Labels:actionerrormetric_type

horizontal_pod_autoscaler_controller_reconciliation_duration_seconds
The time(seconds) that the HPA controller takes to reconcile once. The label 'action' should be either 'scale_down', 'scale_up', or 'none'. Also, the label 'error' should be either 'spec', 'internal', or 'none'. Note that if both spec and internal errors happen during a reconciliation, the first one to occur is reported in `error` label.

- Stability Level:ALPHA
- Type: Histogram
- Labels:actionerror

horizontal_pod_autoscaler_controller_reconciliations_total
Number of reconciliations of HPA controller. The label 'action' should be either 'scale_down', 'scale_up', or 'none'. Also, the label 'error' should be either 'spec', 'internal', or 'none'. Note that if both spec and internal errors happen during a reconciliation, the first one to occur is reported in `error` label.

- Stability Level:ALPHA
- Type: Counter
- Labels:actionerror

job_controller_job_finished_indexes_total
`The number of finished indexes. Possible values for the, status label are: "succeeded", "failed". Possible values for the, backoffLimit label are: "perIndex" and "global"`

- Stability Level:ALPHA
- Type: Counter
- Labels:backoffLimitstatus

job_controller_job_pods_creation_total
`The number of Pods created by the Job controller labelled with a reason for the Pod creation., This metric also distinguishes between Pods created using different PodReplacementPolicy settings., Possible values of the "reason" label are:, "new", "recreate_terminating_or_failed", "recreate_failed"., Possible values of the "status" label are:, "succeeded", "failed".`

- Stability Level:ALPHA
- Type: Counter
- Labels:reasonstatus

job_controller_jobs_by_external_controller_total
The number of Jobs managed by an external controller

- Stability Level:ALPHA
- Type: Counter
- Labels:controller_name

job_controller_pod_failures_handled_by_failure_policy_total
`The number of failed Pods handled by failure policy with, respect to the failure policy action applied based on the matched, rule. Possible values of the action label correspond to the, possible values for the failure policy rule action, which are:, "FailJob", "Ignore" and "Count".`

- Stability Level:ALPHA
- Type: Counter
- Labels:action

job_controller_terminated_pods_tracking_finalizer_total
`The number of terminated pods (phase=Failed|Succeeded), that have the finalizer batch.kubernetes.io/job-tracking, The event label can be "add" or "delete".`

- Stability Level:ALPHA
- Type: Counter
- Labels:event

kube_apiserver_clusterip_allocator_allocated_ips
Gauge measuring the number of allocated IPs for Services

- Stability Level:ALPHA
- Type: Gauge
- Labels:cidr

kube_apiserver_clusterip_allocator_allocation_duration_seconds
Duration in seconds to allocate a Cluster IP by ServiceCIDR

- Stability Level:ALPHA
- Type: Histogram
- Labels:cidr

kube_apiserver_clusterip_allocator_allocation_errors_total
Number of errors trying to allocate Cluster IPs

- Stability Level:ALPHA
- Type: Counter
- Labels:cidrscope

kube_apiserver_clusterip_allocator_allocation_total
Number of Cluster IPs allocations

- Stability Level:ALPHA
- Type: Counter
- Labels:cidrscope

kube_apiserver_clusterip_allocator_available_ips
Gauge measuring the number of available IPs for Services

- Stability Level:ALPHA
- Type: Gauge
- Labels:cidr

kube_apiserver_nodeport_allocator_allocated_ports
Gauge measuring the number of allocated NodePorts for Services

- Stability Level:ALPHA
- Type: Gauge

kube_apiserver_nodeport_allocator_allocation_errors_total
Number of errors trying to allocate NodePort

- Stability Level:ALPHA
- Type: Counter
- Labels:scope

kube_apiserver_nodeport_allocator_allocation_total
Number of NodePort allocations

- Stability Level:ALPHA
- Type: Counter
- Labels:scope

kube_apiserver_nodeport_allocator_available_ports
Gauge measuring the number of available NodePorts for Services

- Stability Level:ALPHA
- Type: Gauge

kube_apiserver_pod_logs_backend_tls_failure_total
Total number of requests for pods/logs that failed due to kubelet server TLS verification

- Stability Level:ALPHA
- Type: Counter

kube_apiserver_pod_logs_insecure_backend_total
Total number of requests for pods/logs sliced by usage type: enforce_tls, skip_tls_allowed, skip_tls_denied

- Stability Level:ALPHA
- Type: Counter
- Labels:usage

kube_apiserver_pod_logs_pods_logs_backend_tls_failure_total
Total number of requests for pods/logs that failed due to kubelet server TLS verification

- Stability Level:ALPHA
- Type: Counter
- Deprecated Versions:1.27.0

kube_apiserver_pod_logs_pods_logs_insecure_backend_total
Total number of requests for pods/logs sliced by usage type: enforce_tls, skip_tls_allowed, skip_tls_denied

- Stability Level:ALPHA
- Type: Counter
- Labels:usage
- Deprecated Versions:1.27.0

kubelet_active_pods
The number of pods the kubelet considers active and which are being considered when admitting new pods. static is true if the pod is not from the apiserver.

- Stability Level:ALPHA
- Type: Gauge
- Labels:static

kubelet_admission_rejections_total
Cumulative number pod admission rejections by the Kubelet.

- Stability Level:ALPHA
- Type: Counter

- Labels:reason

kubelet_certificate_manager_client_expiration_renew_errors
Counter of certificate renewal errors.

- Stability Level:ALPHA
- Type: Counter

kubelet_certificate_manager_client_ttl_seconds
Gauge of the TTL (time-to-live) of the Kubelet's client certificate. The value is in seconds until certificate expiry (negative if already expired). If client certificate is invalid or unused, the value will be +INF.

- Stability Level:ALPHA
- Type: Gauge

kubelet_certificate_manager_server_rotation_seconds
Histogram of the number of seconds the previous certificate lived before being rotated.

- Stability Level:ALPHA
- Type: Histogram

kubelet_certificate_manager_server_ttl_seconds
Gauge of the shortest TTL (time-to-live) of the Kubelet's serving certificate. The value is in seconds until certificate expiry (negative if already expired). If serving certificate is invalid or unused, the value will be +INF.

- Stability Level:ALPHA
- Type: Gauge

kubelet_cgroup_manager_duration_seconds
Duration in seconds for cgroup manager operations. Broken down by method.

- Stability Level:ALPHA
- Type: Histogram
- Labels:operation_type

kubelet_cgroup_version
cgroup version on the hosts.

- Stability Level:ALPHA
- Type: Gauge

kubelet_container_aligned_compute_resources_count
Cumulative number of aligned compute resources allocated to containers by alignment type.

- Stability Level:ALPHA
- Type: Counter
- Labels:boundaryscope

kubelet_container_aligned_compute_resources_failure_count
Cumulative number of failures to allocate aligned compute resources to containers by alignment type.

- Stability Level:ALPHA
- Type: Counter
- Labels:boundaryscope

kubelet_container_log_filesystem_used_bytes
Bytes used by the container's logs on the filesystem.

- Stability Level:ALPHA
- Type: Custom
- Labels:uidnamespacepodcontainer

kubelet_container_requested_resizes_total
Number of requested resizes, counted at the container level. Different resources on the same container are counted separately. The 'requirement' label refers to 'memory' or 'limits'; the 'operation' label can be one of 'add', 'remove', 'increase' or 'decrease'.

- Stability Level:ALPHA
- Type: Counter
- Labels:operationrequirementresource

kubelet_containers_per_pod_count
The number of containers per pod.

- Stability Level:ALPHA
- Type: Histogram

kubelet_cpu_manager_allocation_per_numa
Number of CPUs allocated per NUMA node

- Stability Level:ALPHA
- Type: Gauge
- Labels:numa_node

kubelet_cpu_manager_exclusive_cpu_allocation_count

The total number of CPUs exclusively allocated to containers running on this node

- Stability Level:ALPHA
- Type: Gauge

kubelet_cpu_manager_pinning_errors_total
The number of cpu core allocations which required pinning failed.

- Stability Level:ALPHA
- Type: Counter

kubelet_cpu_manager_pinning_requests_total
The number of cpu core allocations which required pinning.

- Stability Level:ALPHA
- Type: Counter

kubelet_cpu_manager_shared_pool_size_millicores
The size of the shared CPU pool for non-guaranteed QoS pods, in millicores.

- Stability Level:ALPHA
- Type: Gauge

kubelet_credential_provider_config_info
Information about the last applied credential provider configuration with hash as label

- Stability Level:ALPHA
- Type: Custom
- Labels:hash

kubelet_credential_provider_plugin_duration
Duration of execution in seconds for credential provider plugin

- Stability Level:ALPHA
- Type: Histogram
- Labels:plugin_name

kubelet_credential_provider_plugin_errors_total
Number of errors from credential provider plugin

- Stability Level:ALPHA
- Type: Counter
- Labels:plugin_name

kubelet_cri_losing_support
the Kubernetes version that the currently running CRI implementation will lose support on if not upgraded.

- Stability Level:ALPHA
- Type: Gauge
- Labels:version

kubelet_desired_pods
The number of pods the kubelet is being instructed to run. static is true if the pod is not from the apiserver.

- Stability Level:ALPHA
- Type: Gauge
- Labels:static

kubelet_device_plugin_alloc_duration_seconds
Duration in seconds to serve a device plugin Allocation request. Broken down by resource name.

- Stability Level:ALPHA
- Type: Histogram
- Labels:resource_name

kubelet_device_plugin_registration_total
Cumulative number of device plugin registrations. Broken down by resource name.

- Stability Level:ALPHA
- Type: Counter
- Labels:resource_name

kubelet_evented_pleg_connection_error_count
The number of errors encountered during the establishment of streaming connection with the CRI runtime.

- Stability Level:ALPHA
- Type: Counter

kubelet_evented_pleg_connection_latency_seconds
The latency of streaming connection with the CRI runtime, measured in seconds.

- Stability Level:ALPHA
- Type: Histogram

kubelet_evented_pleg_connection_success_count

The number of times a streaming client was obtained to receive CRI Events.

- Stability Level:ALPHA
- Type: Counter

kubelet_eviction_stats_age_seconds
Time between when stats are collected, and when pod is evicted based on those stats by eviction signal

- Stability Level:ALPHA
- Type: Histogram
- Labels:eviction_signal

kubelet_evictions
Cumulative number of pod evictions by eviction signal

- Stability Level:ALPHA
- Type: Counter
- Labels:eviction_signal

kubelet_graceful_shutdown_end_time_seconds
Last graceful shutdown end time since unix epoch in seconds

- Stability Level:ALPHA
- Type: Gauge

kubelet_graceful_shutdown_start_time_seconds
Last graceful shutdown start time since unix epoch in seconds

- Stability Level:ALPHA
- Type: Gauge

kubelet_http_inflight_requests
Number of the inflight http requests

- Stability Level:ALPHA
- Type: Gauge
- Labels:long_runningmethodpathserver_type

kubelet_http_requests_duration_seconds
Duration in seconds to serve http requests

- Stability Level:ALPHA
- Type: Histogram
- Labels:long_runningmethodpathserver_type

kubelet_http_requests_total
Number of the http requests received since the server started

- Stability Level:ALPHA
- Type: Counter
- Labels:long_runningmethodpathserver_type

kubelet_image_garbage_collected_total
Total number of images garbage collected by the kubelet, whether through disk usage or image age.

- Stability Level:ALPHA
- Type: Counter
- Labels:reason

kubelet_image_pull_duration_seconds
Duration in seconds to pull an image.

- Stability Level:ALPHA
- Type: Histogram
- Labels:image_size_in_bytes

kubelet_image_volume_mounted_errors_total
Number of failed image volume mounts.

- Stability Level:ALPHA
- Type: Counter

kubelet_image_volume_mounted_succeed_total
Number of successful image volume mounts.

- Stability Level:ALPHA
- Type: Counter

kubelet_image_volume_requested_total
Number of requested image volumes.

- Stability Level:ALPHA
- Type: Counter

kubelet_lifecycle_handler_http_fallbacks_total

The number of times lifecycle handlers successfully fell back to http from https.

- Stability Level:ALPHA
- Type: Counter

kubelet_managed_ephemeral_containers
Current number of ephemeral containers in pods managed by this kubelet.

- Stability Level:ALPHA
- Type: Gauge

kubelet_memory_manager_pinning_errors_total
The number of memory pages allocations which required pinning that failed.

- Stability Level:ALPHA
- Type: Counter

kubelet_memory_manager_pinning_requests_total
The number of memory pages allocations which required pinning.

- Stability Level:ALPHA
- Type: Counter

kubelet_mirror_pods
The number of mirror pods the kubelet will try to create (one per admitted static pod)

- Stability Level:ALPHA
- Type: Gauge

kubelet_node_name
The node's name. The count is always 1.

- Stability Level:ALPHA
- Type: Gauge
- Labels:node

kubelet_node_startup_duration_seconds
Duration in seconds of node startup in total.

- Stability Level:ALPHA
- Type: Gauge

kubelet_node_startup_post_registration_duration_seconds
Duration in seconds of node startup after registration.

- Stability Level:ALPHA
- Type: Gauge

kubelet_node_startup_pre_kubelet_duration_seconds
Duration in seconds of node startup before kubelet starts.

- Stability Level:ALPHA
- Type: Gauge

kubelet_node_startup_pre_registration_duration_seconds
Duration in seconds of node startup before registration.

- Stability Level:ALPHA
- Type: Gauge

kubelet_node_startup_registration_duration_seconds
Duration in seconds of node startup during registration.

- Stability Level:ALPHA
- Type: Gauge

kubelet_orphan_pod_cleaned_volumes
The total number of orphaned Pods whose volumes were cleaned in the last periodic sweep.

- Stability Level:ALPHA
- Type: Gauge

kubelet_orphan_pod_cleaned_volumes_errors
The number of orphaned Pods whose volumes failed to be cleaned in the last periodic sweep.

- Stability Level:ALPHA
- Type: Gauge

kubelet_orphaned_runtime_pods_total
Number of pods that have been detected in the container runtime without being already known to the pod worker. This typically indicates the kubelet was restarted while a pod was force deleted in the API or in the local configuration, which is unusual.

- Stability Level:ALPHA
- Type: Counter

kubelet_pleg_discard_events
The number of discard events in PLEG.

- Stability Level:ALPHA
- Type: Counter

kubelet_pleg_last_seen_seconds
Timestamp in seconds when PLEG was last seen active.

- Stability Level:ALPHA
- Type: Gauge

kubelet_pleg_relist_duration_seconds
Duration in seconds for relisting pods in PLEG.

- Stability Level:ALPHA
- Type: Histogram

kubelet_pleg_relist_interval_seconds
Interval in seconds between relisting in PLEG.

- Stability Level:ALPHA
- Type: Histogram

kubelet_pod_deferred_accepted_resizes_total
Cumulative number of resizes that were accepted after being deferred.

- Stability Level:ALPHA
- Type: Counter
- Labels:retry_trigger

kubelet_pod_in_progress_resizes
Number of in-progress resizes for pods.

- Stability Level:ALPHA
- Type: Gauge

kubelet_pod_infeasible_resizes_total
Number of infeasible resizes for pods.

- Stability Level:ALPHA
- Type: Counter
- Labels:reason_detail

kubelet_pod_pending_resizes
Number of pending resizes for pods.

- Stability Level:ALPHA
- Type: Gauge
- Labels:reason

kubelet_pod_resize_duration_milliseconds
Duration in milliseconds to actuate a pod resize

- Stability Level:ALPHA
- Type: Histogram
- Labels:success

kubelet_pod_resources_endpoint_errors_get
Number of requests to the PodResource Get endpoint which returned error. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_resources_endpoint_errors_get_allocatable
Number of requests to the PodResource GetAllocatableResources endpoint which returned error. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_resources_endpoint_errors_list
Number of requests to the PodResource List endpoint which returned error. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_resources_endpoint_requests_get
Number of requests to the PodResource Get endpoint. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter

- Labels:server_api_version

kubelet_pod_resources_endpoint_requests_get_allocatable
Number of requests to the PodResource GetAllocatableResources endpoint. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_resources_endpoint_requests_list
Number of requests to the PodResource List endpoint. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_resources_endpoint_requests_total
Cumulative number of requests to the PodResource endpoint. Broken down by server api version.

- Stability Level:ALPHA
- Type: Counter
- Labels:server_api_version

kubelet_pod_start_duration_seconds
Duration in seconds from kubelet seeing a pod for the first time to the pod starting to run

- Stability Level:ALPHA
- Type: Histogram

kubelet_pod_start_sli_duration_seconds
Duration in seconds to start a pod, excluding time to pull images and run init containers, measured from pod creation timestamp to when all its containers are reported as started and observed via watch

- Stability Level:ALPHA
- Type: Histogram

kubelet_pod_start_total_duration_seconds
Duration in seconds to start a pod since creation, including time to pull images and run init containers, measured from pod creation timestamp to when all its containers are reported as started and observed via watch

- Stability Level:ALPHA
- Type: Histogram

kubelet_pod_status_sync_duration_seconds
Duration in seconds to sync a pod status update. Measures time from detection of a change to pod status until the API is successfully updated for that pod, even if multiple intevening changes to pod status occur.

- Stability Level:ALPHA
- Type: Histogram

kubelet_pod_worker_duration_seconds
Duration in seconds to sync a single pod. Broken down by operation type: create, update, or sync

- Stability Level:ALPHA
- Type: Histogram
- Labels:operation_type

kubelet_pod_worker_start_duration_seconds
Duration in seconds from kubelet seeing a pod to starting a worker.

- Stability Level:ALPHA
- Type: Histogram

kubelet_preemptions
Cumulative number of pod preemptions by preemption resource

- Stability Level:ALPHA
- Type: Counter
- Labels:preemption_signal

kubelet_restarted_pods_total
Number of pods that have been restarted because they were deleted and recreated with the same UID while the kubelet was watching them (common for static pods, extremely uncommon for API pods)

- Stability Level:ALPHA
- Type: Counter
- Labels:static

kubelet_run_podsandbox_duration_seconds
Duration in seconds of the run_podsandbox operations. Broken down by RuntimeClass.Handler.

- Stability Level:ALPHA
- Type: Histogram
- Labels:runtime_handler

kubelet_run_podsandbox_errors_total
Cumulative number of the run_podsandbox operation errors by RuntimeClass.Handler.

- Stability Level:ALPHA
- Type: Counter
- Labels:runtime_handler

kubelet_running_containers
Number of containers currently running

- Stability Level:ALPHA
- Type: Gauge
- Labels:container_state

kubelet_running_pods
Number of pods that have a running pod sandbox

- Stability Level:ALPHA
- Type: Gauge

kubelet_runtime_operations_duration_seconds
Duration in seconds of runtime operations. Broken down by operation type.

- Stability Level:ALPHA
- Type: Histogram
- Labels:operation_type

kubelet_runtime_operations_errors_total
Cumulative number of runtime operation errors by operation type.

- Stability Level:ALPHA
- Type: Counter
- Labels:operation_type

kubelet_runtime_operations_total
Cumulative number of runtime operations by operation type.

- Stability Level:ALPHA
- Type: Counter
- Labels:operation_type

kubelet_server_expiration_renew_errors
Counter of certificate renewal errors.

- Stability Level:ALPHA
- Type: Counter

kubelet_sleep_action_terminated_early_total
The number of times lifecycle sleep handler got terminated before it finishes

- Stability Level:ALPHA
- Type: Counter

kubelet_started_containers_errors_total
Cumulative number of errors when starting containers

- Stability Level:ALPHA
- Type: Counter
- Labels:codecontainer_type

kubelet_started_containers_total
Cumulative number of containers started

- Stability Level:ALPHA
- Type: Counter
- Labels:container_type

kubelet_started_host_process_containers_errors_total
Cumulative number of errors when starting hostprocess containers. This metric will only be collected on Windows.

- Stability Level:ALPHA
- Type: Counter
- Labels:codecontainer_type

kubelet_started_host_process_containers_total
Cumulative number of hostprocess containers started. This metric will only be collected on Windows.

- Stability Level:ALPHA
- Type: Counter
- Labels:container_type

kubelet_started_pods_errors_total
Cumulative number of errors when starting pods

- Stability Level:ALPHA
- Type: Counter

kubelet_started_pods_total
Cumulative number of pods started

- Stability Level:ALPHA
- Type: Counter

kubelet_started_user_namespaced_pods_errors_total
Cumulative number of errors when starting pods with user namespaces. This metric will only be collected on Linux.

- Stability Level:ALPHA
- Type: Counter

kubelet_started_user_namespaced_pods_total
Cumulative number of pods with user namespaces started. This metric will only be collected on Linux.

- Stability Level:ALPHA
- Type: Counter

kubelet_topology_manager_admission_duration_ms
Duration in milliseconds to serve a pod admission request.

- Stability Level:ALPHA
- Type: Histogram

kubelet_topology_manager_admission_errors_total
The number of admission request failures where resources could not be aligned.

- Stability Level:ALPHA
- Type: Counter

kubelet_topology_manager_admission_requests_total
The number of admission requests where resources have to be aligned.

- Stability Level:ALPHA
- Type: Counter

kubelet_volume_metric_collection_duration_seconds
Duration in seconds to calculate volume stats

- Stability Level:ALPHA
- Type: Histogram
- Labels:metric_source

kubelet_volume_stats_available_bytes
Number of available bytes in the volume

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_capacity_bytes
Capacity in bytes of the volume

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_health_status_abnormal
Abnormal volume health status. The count is either 1 or 0. 1 indicates the volume is unhealthy, 0 indicates volume is healthy

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_inodes
Maximum number of inodes in the volume

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_inodes_free
Number of free inodes in the volume

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_inodes_used
Number of used inodes in the volume

- Stability Level:ALPHA

- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_volume_stats_used_bytes
Number of used bytes in the volume

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacepersistentvolumeclaim

kubelet_working_pods
Number of pods the kubelet is actually running, broken down by lifecycle phase, whether the pod is desired, orphaned, or runtime only (also orphaned), and whether the pod is static. An orphaned pod has been removed from local configuration or force deleted in the API and consumes resources that are not otherwise visible.

- Stability Level:ALPHA
- Type: Gauge
- Labels:configlifecyclestatic

kubeproxy_conntrack_reconciler_deleted_entries_total
Cumulative conntrack flows deleted by conntrack reconciler

- Stability Level:ALPHA
- Type: Counter
- Labels:ip_family

kubeproxy_conntrack_reconciler_sync_duration_seconds
ReconcileConntrackFlowsLatency latency in seconds

- Stability Level:ALPHA
- Type: Histogram
- Labels:ip_family

kubeproxy_iptables_ct_state_invalid_dropped_packets_total
packets dropped by iptables to work around conntrack problems

- Stability Level:ALPHA
- Type: Custom

kubeproxy_iptables_localhost_nodeports_accepted_packets_total
Number of packets accepted on nodeports of loopback interface

- Stability Level:ALPHA
- Type: Custom

kubeproxy_network_programming_duration_seconds
In Cluster Network Programming Latency in seconds

- Stability Level:ALPHA
- Type: Histogram
- Labels:ip_family

kubeproxy_proxy_healthz_total
Cumulative proxy healthz HTTP status

- Stability Level:ALPHA
- Type: Counter
- Labels:code

kubeproxy_proxy_livez_total
Cumulative proxy livez HTTP status

- Stability Level:ALPHA
- Type: Counter
- Labels:code

kubeproxy_sync_full_proxy_rules_duration_seconds
SyncProxyRules latency in seconds for full resyncs

- Stability Level:ALPHA
- Type: Histogram
- Labels:ip_family

kubeproxy_sync_partial_proxy_rules_duration_seconds
SyncProxyRules latency in seconds for partial resyncs

- Stability Level:ALPHA
- Type: Histogram
- Labels:ip_family

kubeproxy_sync_proxy_rules_duration_seconds
SyncProxyRules latency in seconds

- Stability Level:ALPHA

- Type: Histogram
- Labels:ip_family

kubeproxy_sync_proxy_rules_endpoint_changes_pending
Pending proxy rules Endpoint changes

- Stability Level:ALPHA
- Type: Gauge

kubeproxy_sync_proxy_rules_endpoint_changes_total
Cumulative proxy rules Endpoint changes

- Stability Level:ALPHA
- Type: Counter

kubeproxy_sync_proxy_rules_iptables_last
Number of iptables rules written by kube-proxy in last sync

- Stability Level:ALPHA
- Type: Gauge
- Labels:ip_familytable

kubeproxy_sync_proxy_rules_iptables_partial_restore_failures_total
Cumulative proxy iptables partial restore failures

- Stability Level:ALPHA
- Type: Counter
- Labels:ip_family

kubeproxy_sync_proxy_rules_iptables_restore_failures_total
Cumulative proxy iptables restore failures

- Stability Level:ALPHA
- Type: Counter
- Labels:ip_family

kubeproxy_sync_proxy_rules_iptables_total
Total number of iptables rules owned by kube-proxy

- Stability Level:ALPHA
- Type: Gauge
- Labels:ip_familytable

kubeproxy_sync_proxy_rules_last_queued_timestamp_seconds
The last time a sync of proxy rules was queued

- Stability Level:ALPHA
- Type: Gauge
- Labels:ip_family

kubeproxy_sync_proxy_rules_last_timestamp_seconds
The last time proxy rules were successfully synced

- Stability Level:ALPHA
- Type: Gauge
- Labels:ip_family

kubeproxy_sync_proxy_rules_nftables_cleanup_failures_total
Cumulative proxy nftables cleanup failures

- Stability Level:ALPHA
- Type: Counter
- Labels:ip_family

kubeproxy_sync_proxy_rules_nftables_sync_failures_total
Cumulative proxy nftables sync failures

- Stability Level:ALPHA
- Type: Counter
- Labels:ip_family

kubeproxy_sync_proxy_rules_no_local_endpoints_total
Number of services with a Local traffic policy and no endpoints

- Stability Level:ALPHA
- Type: Gauge
- Labels:ip_familytraffic_policy

kubeproxy_sync_proxy_rules_service_changes_pending
Pending proxy rules Service changes

- Stability Level:ALPHA
- Type: Gauge

kubeproxy_sync_proxy_rules_service_changes_total
Cumulative proxy rules Service changes

- Stability Level:ALPHA
- Type: Counter

kubernetes_build_info
A metric with a constant '1' value labeled by major, minor, git version, git commit, git tree state, build date, Go version, and compiler from which Kubernetes was built, and platform on which it is running.

- Stability Level:ALPHA
- Type: Gauge
- Labels:build_datecompilergit_commitgit_tree_stategit_versiongo_versionmajorminorplatform

leader_election_master_status
Gauge of if the reporting system is master of the relevant lease, 0 indicates backup, 1 indicates master. 'name' is the string used to identify the lease. Please make sure to group by name.

- Stability Level:ALPHA
- Type: Gauge
- Labels:name

leader_election_slowpath_total
Total number of slow path exercised in renewing leader leases. 'name' is the string used to identify the lease. Please make sure to group by name.

- Stability Level:ALPHA
- Type: Counter
- Labels:name

node_authorizer_graph_actions_duration_seconds
Histogram of duration of graph actions in node authorizer.

- Stability Level:ALPHA
- Type: Histogram
- Labels:operation

node_collector_unhealthy_nodes_in_zone
Gauge measuring number of not Ready Nodes per zones.

- Stability Level:ALPHA
- Type: Gauge
- Labels:zone

node_collector_update_all_nodes_health_duration_seconds
Duration in seconds for NodeController to update the health of all nodes.

- Stability Level:ALPHA
- Type: Histogram

node_collector_update_node_health_duration_seconds
Duration in seconds for NodeController to update the health of a single node.

- Stability Level:ALPHA
- Type: Histogram

node_collector_zone_health
Gauge measuring percentage of healthy nodes per zone.

- Stability Level:ALPHA
- Type: Gauge
- Labels:zone

node_collector_zone_size
Gauge measuring number of registered Nodes per zones.

- Stability Level:ALPHA
- Type: Gauge
- Labels:zone

node_controller_cloud_provider_taint_removal_delay_seconds
Number of seconds after node creation when NodeController removed the cloud-provider taint of a single node.

- Stability Level:ALPHA
- Type: Histogram

node_controller_initial_node_sync_delay_seconds
Number of seconds after node creation when NodeController finished the initial synchronization of a single node.

- Stability Level:ALPHA
- Type: Histogram

node_ipam_controller_cidrset_allocation_tries_per_request
Number of endpoints added on each Service sync

- Stability Level:ALPHA
- Type: Histogram
- Labels:clusterCIDR

node_ipam_controller_cidrset_cidrs_allocations_total
Counter measuring total number of CIDR allocations.

- Stability Level:ALPHA
- Type: Counter
- Labels:clusterCIDR

node_ipam_controller_cidrset_cidrs_releases_total
Counter measuring total number of CIDR releases.

- Stability Level:ALPHA
- Type: Counter
- Labels:clusterCIDR

node_ipam_controller_cidrset_usage_cidrs
Gauge measuring percentage of allocated CIDRs.

- Stability Level:ALPHA
- Type: Gauge
- Labels:clusterCIDR

node_ipam_controller_cirdset_max_cidrs
Maximum number of CIDRs that can be allocated.

- Stability Level:ALPHA
- Type: Gauge
- Labels:clusterCIDR

node_swap_usage_bytes
Current swap usage of the node in bytes. Reported only on non-windows systems

- Stability Level:ALPHA
- Type: Custom

plugin_manager_total_plugins
Number of plugins in Plugin Manager

- Stability Level:ALPHA
- Type: Custom
- Labels:socket_pathstate

pod_gc_collector_force_delete_pod_errors_total
Number of errors encountered when forcefully deleting the pods since the Pod GC Controller started.

- Stability Level:ALPHA
- Type: Counter
- Labels:namespacereason

pod_gc_collector_force_delete_pods_total
Number of pods that are being forcefully deleted since the Pod GC Controller started.

- Stability Level:ALPHA
- Type: Counter
- Labels:namespacereason

pod_security_errors_total
Number of errors preventing normal evaluation. Non-fatal errors may result in the latest restricted profile being used for evaluation.

- Stability Level:ALPHA
- Type: Counter
- Labels:fatalrequest_operationresourcesubresource

pod_security_evaluations_total
Number of policy evaluations that occurred, not counting ignored or exempt requests.

- Stability Level:ALPHA
- Type: Counter
- Labels:decisionmodepolicy_levelpolicy_versionrequest_operationresourcesubresource

pod_security_exemptions_total
Number of exempt requests, not counting ignored or out of scope requests.

- Stability Level:ALPHA
- Type: Counter
- Labels:request_operationresourcesubresource

pod_swap_usage_bytes
Current amount of the pod swap usage in bytes. Reported only on non-windows systems

- Stability Level:ALPHA

- Type: Custom
- Labels:podnamespace

prober_probe_duration_seconds
Duration in seconds for a probe response.

- Stability Level:ALPHA
- Type: Histogram
- Labels:containernamespacepodprobe_type

pv_collector_bound_pv_count
Gauge measuring number of persistent volume currently bound

- Stability Level:ALPHA
- Type: Custom
- Labels:storage_class

pv_collector_bound_pvc_count
Gauge measuring number of persistent volume claim currently bound

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacestorage_classvolume_attributes_class

pv_collector_total_pv_count
Gauge measuring total number of persistent volumes

- Stability Level:ALPHA
- Type: Custom
- Labels:plugin_namevolume_mode

pv_collector_unbound_pv_count
Gauge measuring number of persistent volume currently unbound

- Stability Level:ALPHA
- Type: Custom
- Labels:storage_class

pv_collector_unbound_pvc_count
Gauge measuring number of persistent volume claim currently unbound

- Stability Level:ALPHA
- Type: Custom
- Labels:namespacestorage_classvolume_attributes_class

reconstruct_volume_operations_errors_total
The number of volumes that failed reconstruction from the operating system during kubelet startup.

- Stability Level:ALPHA
- Type: Counter

reconstruct_volume_operations_total
The number of volumes that were attempted to be reconstructed from the operating system during kubelet startup. This includes both successful and failed reconstruction.

- Stability Level:ALPHA
- Type: Counter

replicaset_controller_sorting_deletion_age_ratio
The ratio of chosen deleted pod's ages to the current youngest pod's age (at the time). Should be <2. The intent of this metric is to measure the rough efficacy of the LogarithmicScaleDown feature gate's effect on the sorting (and deletion) of pods when a replicaset scales down. This only considers Ready pods when calculating and reporting.

- Stability Level:ALPHA
- Type: Histogram

resourceclaim_controller_creates_total
Number of ResourceClaims creation requests, categorized by creation status and admin access

- Stability Level:ALPHA
- Type: Counter
- Labels:admin_accessstatus

resourceclaim_controller_resource_claims
Number of ResourceClaims, categorized by allocation status and admin access

- Stability Level:ALPHA
- Type: Custom
- Labels:allocatedadmin_access

rest_client_dns_resolution_duration_seconds
DNS resolver latency in seconds. Broken down by host.

- Stability Level:ALPHA

- Type: Histogram
- Labels:host

rest_client_exec_plugin_call_total
Number of calls to an exec plugin, partitioned by the type of event encountered (no_error, plugin_execution_error, plugin_not_found_error, client_internal_error) and an optional exit code. The exit code will be set to 0 if and only if the plugin call was successful.

- Stability Level:ALPHA
- Type: Counter
- Labels:call_statuscode

rest_client_exec_plugin_certificate_rotation_age
Histogram of the number of seconds the last auth exec plugin client certificate lived before being rotated. If auth exec plugin client certificates are unused, histogram will contain no data.

- Stability Level:ALPHA
- Type: Histogram

rest_client_exec_plugin_ttl_seconds
Gauge of the shortest TTL (time-to-live) of the client certificate(s) managed by the auth exec plugin. The value is in seconds until certificate expiry (negative if already expired). If auth exec plugins are unused or manage no TLS certificates, the value will be +INF.

- Stability Level:ALPHA
- Type: Gauge

rest_client_rate_limiter_duration_seconds
Client side rate limiter latency in seconds. Broken down by verb, and host.

- Stability Level:ALPHA
- Type: Histogram
- Labels:hostverb

rest_client_request_duration_seconds
Request latency in seconds. Broken down by verb, and host.

- Stability Level:ALPHA
- Type: Histogram
- Labels:hostverb

rest_client_request_retries_total
Number of request retries, partitioned by status code, verb, and host.

- Stability Level:ALPHA
- Type: Counter
- Labels:codehostverb

rest_client_request_size_bytes
Request size in bytes. Broken down by verb and host.

- Stability Level:ALPHA
- Type: Histogram
- Labels:hostverb

rest_client_requests_total
Number of HTTP requests, partitioned by status code, method, and host.

- Stability Level:ALPHA
- Type: Counter
- Labels:codehostmethod

rest_client_response_size_bytes
Response size in bytes. Broken down by verb and host.

- Stability Level:ALPHA
- Type: Histogram
- Labels:hostverb

rest_client_transport_cache_entries
Number of transport entries in the internal cache.

- Stability Level:ALPHA
- Type: Gauge

rest_client_transport_create_calls_total
Number of calls to get a new transport, partitioned by the result of the operation hit: obtained from the cache, miss: created and added to the cache, uncacheable: created and not cached

- Stability Level:ALPHA
- Type: Counter
- Labels:result

retroactive_storageclass_errors_total
Total number of failed retroactive StorageClass assignments to persistent volume claim

- Stability Level:ALPHA
- Type: Counter

retroactive_storageclass_total
Total number of retroactive StorageClass assignments to persistent volume claim

- Stability Level:ALPHA
- Type: Counter

root_ca_cert_publisher_sync_duration_seconds
Number of namespace syncs happened in root ca cert publisher.

- Stability Level:ALPHA
- Type: Histogram
- Labels:code

root_ca_cert_publisher_sync_total
Number of namespace syncs happened in root ca cert publisher.

- Stability Level:ALPHA
- Type: Counter
- Labels:code

running_managed_controllers
Indicates where instances of a controller are currently running

- Stability Level:ALPHA
- Type: Gauge
- Labels:managername

scheduler_async_api_call_execution_duration_seconds
Duration in seconds for executing API call in the async dispatcher.

- Stability Level:ALPHA
- Type: Histogram
- Labels:call_typeresult

scheduler_async_api_call_execution_total
Total number of API calls executed by the async dispatcher.

- Stability Level:ALPHA
- Type: Counter
- Labels:call_typeresult

scheduler_cache_size
Number of nodes, pods, and assumed (bound) pods in the scheduler cache.

- Stability Level:ALPHA
- Type: Gauge
- Labels:type

scheduler_event_handling_duration_seconds
Event handling latency in seconds.

- Stability Level:ALPHA
- Type: Histogram
- Labels:event

scheduler_goroutines
Number of running goroutines split by the work they do such as binding.

- Stability Level:ALPHA
- Type: Gauge
- Labels:operation

scheduler_inflight_events
Number of events currently tracked in the scheduling queue.

- Stability Level:ALPHA
- Type: Gauge
- Labels:event

scheduler_pending_async_api_calls
Number of API calls currently pending in the async queue.

- Stability Level:ALPHA
- Type: Gauge
- Labels:call_type

scheduler_permit_wait_duration_seconds
Duration of waiting on permit.

- Stability Level:ALPHA
- Type: Histogram

- Labels:result

scheduler_plugin_evaluation_total
Number of attempts to schedule pods by each plugin and the extension point (available only in PreFilter, Filter, PreScore, and Score).

- Stability Level:ALPHA
- Type: Counter
- Labels:extension_pointpluginprofile

scheduler_plugin_execution_duration_seconds
Duration for running a plugin at a specific extension point.

- Stability Level:ALPHA
- Type: Histogram
- Labels:extension_pointpluginstatus

scheduler_preemption_goroutines_duration_seconds
Duration in seconds for running goroutines for the preemption.

- Stability Level:ALPHA
- Type: Histogram
- Labels:result

scheduler_preemption_goroutines_execution_total
Number of preemption goroutines executed.

- Stability Level:ALPHA
- Type: Counter
- Labels:result

scheduler_queueing_hint_execution_duration_seconds
Duration for running a queueing hint function of a plugin.

- Stability Level:ALPHA
- Type: Histogram
- Labels:eventhintplugin

scheduler_scheduling_algorithm_duration_seconds
Scheduling algorithm latency in seconds

- Stability Level:ALPHA
- Type: Histogram

scheduler_unschedulable_pods
The number of unschedulable pods broken down by plugin name. A pod will increment the gauge for all plugins that caused it to not schedule and so this metric have meaning only when broken down by plugin.

- Stability Level:ALPHA
- Type: Gauge
- Labels:pluginprofile

scheduler_volume_binder_cache_requests_total
Total number for request volume binding cache

- Stability Level:ALPHA
- Type: Counter
- Labels:operation

scheduler_volume_scheduling_stage_error_total
Volume scheduling stage error count

- Stability Level:ALPHA
- Type: Counter
- Labels:operation

scrape_error
1 if there was an error while getting container metrics, 0 otherwise

- Stability Level:ALPHA
- Type: Custom
- Deprecated Versions:1.29.0

selinux_warning_controller_selinux_volume_conflict
Conflict between two Pods using the same volume

- Stability Level:ALPHA
- Type: Custom
- Labels:propertypod1_namespacepod1_namepod1_valuepod2_namespacepod2_namepod2_value

service_controller_loadbalancer_sync_total
A metric counting the amount of times any load balancer has been configured, as an effect of service/node changes on the cluster

- Stability Level:ALPHA
- Type: Counter

service_controller_nodesync_error_total
A metric counting the amount of times any load balancer has been configured and errored, as an effect of node changes on the cluster

- Stability Level:ALPHA
- Type: Counter

service_controller_nodesync_latency_seconds
A metric measuring the latency for nodesync which updates loadbalancer hosts on cluster node updates.

- Stability Level:ALPHA
- Type: Histogram

service_controller_update_loadbalancer_host_latency_seconds
A metric measuring the latency for updating each load balancer hosts.

- Stability Level:ALPHA
- Type: Histogram

serviceaccount_invalid_legacy_auto_token_uses_total
Cumulative invalid auto-generated legacy tokens used

- Stability Level:ALPHA
- Type: Counter

serviceaccount_legacy_auto_token_uses_total
Cumulative auto-generated legacy tokens used

- Stability Level:ALPHA
- Type: Counter

serviceaccount_legacy_manual_token_uses_total
Cumulative manually created legacy tokens used

- Stability Level:ALPHA
- Type: Counter

serviceaccount_legacy_tokens_total
Cumulative legacy service account tokens used

- Stability Level:ALPHA
- Type: Counter

serviceaccount_stale_tokens_total
Cumulative stale projected service account tokens used

- Stability Level:ALPHA
- Type: Counter

serviceaccount_valid_tokens_total
Cumulative valid projected service account tokens used

- Stability Level:ALPHA
- Type: Counter

storage_count_attachable_volumes_in_use
Measure number of volumes in use

- Stability Level:ALPHA
- Type: Custom
- Labels:nodevolume_plugin

storage_operation_duration_seconds
Storage operation duration

- Stability Level:ALPHA
- Type: Histogram
- Labels:migratedoperation_namestatusvolume_plugin

taint_eviction_controller_pod_deletion_duration_seconds
Latency, in seconds, between the time when a taint effect has been activated for the Pod and its deletion via TaintEvictionController.

- Stability Level:ALPHA
- Type: Histogram

taint_eviction_controller_pod_deletions_total
Total number of Pods deleted by TaintEvictionController since its start.

- Stability Level:ALPHA
- Type: Counter

ttl_after_finished_controller_job_deletion_duration_seconds
The time it took to delete the job since it became eligible for deletion

- Stability Level:ALPHA

- Type: Histogram

version_info
Provides the compatibility version info of the component. The component label is the name of the component, usually kube, but is relevant for aggregated-apiservers.

- Stability Level:ALPHA
- Type: Gauge
- Labels:binarycomponentemulationmin_compat

volume_manager_selinux_container_errors_total
Number of errors when kubelet cannot compute SELinux context for a container. Kubelet can't start such a Pod then and it will retry, therefore value of this metric may not represent the actual nr. of containers.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_mode

volume_manager_selinux_container_warnings_total
Number of errors when kubelet cannot compute SELinux context for a container that are ignored. They will become real errors when SELinuxMountReadWriteOncePod feature is expanded to all volume access modes.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_mode

volume_manager_selinux_pod_context_mismatch_errors_total
Number of errors when a Pod defines different SELinux contexts for its containers that use the same volume. Kubelet can't start such a Pod then and it will retry, therefore value of this metric may not represent the actual nr. of Pods.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_mode

volume_manager_selinux_pod_context_mismatch_warnings_total
Number of errors when a Pod defines different SELinux contexts for its containers that use the same volume. They are not errors yet, but they will become real errors when SELinuxMountReadWriteOncePod feature is expanded to all volume access modes.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_mode

volume_manager_selinux_volume_context_mismatch_errors_total
Number of errors when a Pod uses a volume that is already mounted with a different SELinux context than the Pod needs. Kubelet can't start such a Pod then and it will retry, therefore value of this metric may not represent the actual nr. of Pods.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_modevolume_plugin

volume_manager_selinux_volume_context_mismatch_warnings_total
Number of errors when a Pod uses a volume that is already mounted with a different SELinux context than the Pod needs. They are not errors yet, but they will become real errors when SELinuxMountReadWriteOncePod feature is expanded to all volume access modes.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_modevolume_plugin

volume_manager_selinux_volumes_admitted_total
Number of volumes whose SELinux context was fine and will be mounted with mount -o context option.

- Stability Level:ALPHA
- Type: Gauge
- Labels:access_modevolume_plugin

volume_manager_total_volumes
Number of volumes in Volume Manager

- Stability Level:ALPHA
- Type: Custom
- Labels:plugin_namestate

volume_operation_total_errors
Total volume operation errors

- Stability Level:ALPHA
- Type: Counter
- Labels:operation_nameplugin_name

volume_operation_total_seconds
Storage operation end to end duration in seconds

- Stability Level:ALPHA
- Type: Histogram

- Labels:operation_nameplugin_name

watch_cache_capacity
Total capacity of watch cache broken by resource type.

- Stability Level:ALPHA
- Type: Gauge
- Labels:groupresource

watch_cache_capacity_decrease_total
Total number of watch cache capacity decrease events broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

watch_cache_capacity_increase_total
Total number of watch cache capacity increase events broken by resource type.

- Stability Level:ALPHA
- Type: Counter
- Labels:groupresource

workqueue_adds_total
Total number of adds handled by workqueue

- Stability Level:ALPHA
- Type: Counter
- Labels:name

workqueue_depth
Current depth of workqueue

- Stability Level:ALPHA
- Type: Gauge
- Labels:name

workqueue_longest_running_processor_seconds
How many seconds has the longest running processor for workqueue been running.

- Stability Level:ALPHA
- Type: Gauge
- Labels:name

workqueue_queue_duration_seconds
How long in seconds an item stays in workqueue before being requested.

- Stability Level:ALPHA
- Type: Histogram
- Labels:name

workqueue_retries_total
Total number of retries handled by workqueue

- Stability Level:ALPHA
- Type: Counter
- Labels:name

workqueue_unfinished_work_seconds
How many seconds of work has done that is in progress and hasn't been observed by work_duration. Large values indicate stuck threads. One can deduce the number of stuck threads by observing the rate at which this increases.

- Stability Level:ALPHA
- Type: Gauge
- Labels:name

workqueue_work_duration_seconds
How long in seconds processing an item from workqueue takes.

- Stability Level:ALPHA
- Type: Histogram
- Labels:name

Kubernetes Issue Tracker

To report a security issue, please follow the Kubernetes security disclosure process.

Work on Kubernetes code and public issues are tracked using GitHub Issues.

- Official list of known CVEs (security vulnerabilities) that have been announced by the Security Response Committee
- CVE-related GitHub issues

Security-related announcements are sent to the kubernetes-security-announce@googlegroups.com mailing list.

Kubernetes Security and Disclosure Information

This page describes Kubernetes security and disclosure information.

Security Announcements

Join the kubernetes-security-announce group for emails about security and major API announcements.

Report a Vulnerability

We're extremely grateful for security researchers and users that report vulnerabilities to the Kubernetes Open Source Community. All reports are thoroughly investigated by a set of community volunteers.

To make a report, submit your vulnerability to the Kubernetes bug bounty program. This allows triage and handling of the vulnerability with standardized response times.

You can also email the private security@kubernetes.io list with the security details and the details expected for all Kubernetes bug reports.

You may encrypt your email to this list using the GPG keys of the Security Response Committee members. Encryption using GPG is NOT required to make a disclosure.

When Should I Report a Vulnerability?

- You think you discovered a potential security vulnerability in Kubernetes
- You are unsure how a vulnerability affects Kubernetes
- You think you discovered a vulnerability in another project that Kubernetes depends on
  - For projects with their own vulnerability reporting and disclosure process, please report it directly there

When Should I NOT Report a Vulnerability?

- You need help tuning Kubernetes components for security
- You need help applying security related updates
- Your issue is not security related

Security Vulnerability Response

Each report is acknowledged and analyzed by Security Response Committee members within 3 working days. This will set off the Security Release Process.

Any vulnerability information shared with Security Response Committee stays within Kubernetes project and will not be disseminated to other projects unless it is necessary to get the issue fixed.

As the security issue moves from triage, to identified fix, to release planning we will keep the reporter updated.

Public Disclosure Timing

A public disclosure date is negotiated by the Kubernetes Security Response Committee and the bug submitter. We prefer to fully disclose the bug as soon as possible once a user mitigation is available. It is reasonable to delay disclosure when the bug or the fix is not yet fully understood, the solution is not well-tested, or for vendor coordination. The timeframe for disclosure is from immediate (especially if it's already publicly known) to a few weeks. For a vulnerability with a straightforward mitigation, we expect report date to disclosure date to be on the order of 7 days. The Kubernetes Security Response Committee holds the final say when setting a disclosure date.

Node metrics data

Mechanisms for accessing metrics at node, volume, pod and container level, as seen by the kubelet.

The kubelet gathers metric statistics at the node, volume, pod and container level, and emits this information in the Summary API.

You can send a proxied request to the stats summary API via the Kubernetes API server.

Here is an example of a Summary API request for a node named `minikube`:

```
kubectl get --raw "/api/v1/nodes/minikube/proxy/stats/summary"
```

Here is the same API call using `curl`:

```
# You need to run "kubectl proxy" first
# Change 8080 to the port that "kubectl proxy" assigns
curl http://localhost:8080/api/v1/nodes/minikube/proxy/stats/summary
```

Note:

Beginning with `metrics-server` 0.6.x, `metrics-server` queries the `/metrics/resource` kubelet endpoint, and not `/stats/summary`.

Summary metrics API source

By default, Kubernetes fetches node summary metrics data using an embedded cAdvisor that runs within the kubelet. If you enable the `PodAndContainerStatsFromCRI` feature gate in your cluster, and you use a container runtime that supports statistics access via Container Runtime Interface (CRI), then the kubelet fetches Pod- and container-level metric data using CRI, and not via cAdvisor.

Pressure Stall Information (PSI)

FEATURE STATE: `Kubernetes v1.34 [beta]`

As a beta feature, Kubernetes lets you configure kubelet to collect Linux kernel Pressure Stall Information (PSI) for CPU, memory, and I/O usage. The information is collected at node, pod and container level. See Summary API for detailed schema. This feature is enabled by default, by setting the `KubeletPSI` feature gate. The information is also exposed in Prometheus metrics.

You can learn how to interpret the PSI metrics in Understand PSI Metrics.

Requirements

Pressure Stall Information requires:

- Linux kernel versions 4.20 or later.
- cgroup v2

What's next

The task pages for Troubleshooting Clusters discuss how to use a metrics pipeline that rely on these data.

Instrumentation

Kubernetes Component SLI Metrics

High-level indicators for measuring the reliability and performance of Kubernetes components.

CRI Pod & Container Metrics

Collection of Pod & Container metrics via the CRI.

Node metrics data

Mechanisms for accessing metrics at node, volume, pod and container level, as seen by the kubelet.

Understand Pressure Stall Information (PSI) Metrics

Detailed explanation of Pressure Stall Information (PSI) metrics and how to use them to identify resource pressure in Kubernetes.

Kubernetes z-pages

Provides runtime diagnostics for Kubernetes components, offering insights into component runtime status and configuration flags.

Kubernetes Metrics Reference

Details of the metric data that Kubernetes components export.