Epic Games - Cloud Developer Platform

Kubernetes Infrastructure Metrics

Downloaded from Epic Games Confluence
Date: 2025-07-12 04:08:52
Original URL: https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068879

Document Level Classification

200

Introduction

The OpenTelemetry pipeline in Substrate collects a number of standard Kubernetes metrics using off-the-shelf/OSS tools, including:

- kube-state-metrics

- OpenTelemetry's kubeletstats receiver
- prometheus-node-exporter
- kubelet's cadvisor

With these we've attempted to build a "complete" collection of Kubernetes infrastructure metrics that are useful to most engineers while balancing cost concerns.

Cost Concerns with Kubernetes

Kubernetes has a fairly large metric footprint due to the large number of things it manages and exports metrics for. Pods, nodes, HPAs, etc all have potentially interesting information, so the off-the-shelf tools we use err on the side of exporting everything, even if they aren't particularly useful to engineers at Epic. In addition, the ephemeral nature of Kubernetes resources (especially pods) means there is regular "churn" of metrics, which can cause wild swings in cardinality during scaling or deployment operations.

The Observability team has spent some time analyzing and curating these metrics, dropping some metrics and aggregating others to reduce the overall footprint. For example, we collect only a handful of prometheus-node-exporter metrics, as grabbing all metrics from this service across all clusters would generate millions of additional metrics with very little value to most teams. Over time we expect we will tinker with what's available as needs change. If you are having an issue with available metrics, reach out to #ct-obs-support-ext.

Finding/Querying Available Kubernetes Metrics

Kubernetes infrastructure metrics generally have one of the following prefixes:

- **kube_** - kube-state-metrics
- **k8s_** - kubeletstats
- **node_** - prometheus-node-exporter
- **container_ -** cadvisor

We try to follow OTEL K8s Semantic Conventions for labels where possible. You should use these labels for authoring dashboards, even if you see other labels like **pod** or **podName,** as the semantic labels are stable and consistent across environments. Generally, all Kubernetes metrics should have some combination of the following labels:

- k8s_namespace_name
- k8s_cluster_name
- k8s_pod_name (if applicable)
- k8s_deployment_name (if applicable)
- k8s_statefulset_name (if applicable)
- k8s_cronjob_name (if applicable)
- k8s_daemonset_name (if applicable)
- k8s_container_name (if applicable)

We also add a **service_name** label using the values of (k8s_statefulset_name, k8s_daemonset_name, k8s_deployment_name) to allow easier discovery of service-related metrics without the need to keep track of the specific deployment type.

Examples

Get number of pods for a service by cluster

```
sum(kube_pod_info{service_name="$service"}) by (k8s_cluster_name)
```

Get CPU usage for service by pod:

```
k8s_pod_cpu_utilization{service_name="$service", k8s_cluster_name="$k8s
```

Get the CPU resource limits and requests for a service container

```
kube_pod_container_resource_requests{service_name="$service", k8s_clust
kube_pod_container_resource_limits{service_name="$service", k8s_cluster
```

You can see some practical examples on some existing generic dashboards that the Observability team maintains:

- Kubernetes Service Resource Metrics - gets commonly requested metrics about individual services (CPU, memory, network, capacity, throttling)
- OTEL Unified Dashboard for Java - see the "Kubernetes" section for examples of commonly requested metrics.

Collection of Metrics

We use a combination of OTEL's prometheus and kubeletstats receivers to gather Kubernetes metrics. Prometheus scrapes the following:

- kube-state-metrics - exports metrics based on resources in the Kubernetes API
- prometheus-node-exporter - exports verbose metrics about individual nodes, including information like MAC address ids of ethernet ports.
- cadvisor - we grab specific container-level metrics directly from kubelet which are not available via the kubeletstats receiver

Kubeletstats runs as a daemonset, talking to the node-local kubelet process and gathering select pod, node, and container metrics such as cpu/memory/network.

The **epic-system** and **kube-system** namespaces have a substantial metrics footprint that is largely the responsibility of Cloud Infrastructure Engineering, so we collect and store Kubernetes metrics from these namespaces separately.

Aggregation and Filtering

Aggregated Metrics

Every pod-level metric generates an average of 70k metrics (at the time of writing), and this will grow over time as our Kubernetes fleet grows. We leverage Chronosphere's aggregation rules to drop some high cardinality labels where we do not need pod-level granularity. The following aggregation rules are applied to some k8s metrics to reduce the overall footprint. This list may be out of date in the future, the source of truth for this list resides here.

In all cases we drop the following labels: **instance, k8s_node_name, k8s_pod_ip, k8s_pod_name, k8s_job_name, node, nodeName, pod, podName, service_instance_id**

kube_state_metrics_rollup

Aggregation: SUM

- kube_pod_info
- kube_pod_status_ready
- kube_pod_status_phase
- kube_pod_container_info
- kube_pod_container_status_running
- kube_pod_container_status_restarts_total

kube_state_metrics_rollup_limits

Aggregation: MAX

- kube_pod_container_resource_limits
- kube_pod_container_resource_requests

cadvisor_rollup_counters

Aggregation: SUM

- container_cpu_cfs_periods_total
- container_cpu_cfs_throttled_periods_total
- container_cpu_cfs_throttled_seconds_total

- container_oom_events_total

Filtered Metrics

The following metrics are dropped at the source by our Prometheus collectors. This list may be out of date in the future, the source of truth for this list resides here.

```
              - 'name == "kube_configmap_info"'
              - 'name == "kube_configmap_metadata_resource_version"'
              - 'name == "kube_configmap_created"'
              - 'name == "kube_deployment_status_replicas_updated"'
              - 'name == "kube_deployment_spec_replicas"'
              - 'name == "kube_deployment_status_observed_generation"'
              - 'name == "kube_deployment_metadata_generation"'
              - 'name == "kube_deployment_status_replicas_ready"'
              - 'name == "kube_deployment_spec_paused"'
              - 'name == "kube_deployment_spec_strategy_rollingupdate_max
              - 'name == "kube_deployment_spec_strategy_rollingupdate_max
              - 'name == "kube_endpoint_address"'
              - 'name == "kube_endpoint_address_available"'
              - 'name == "kube_endpoint_address_not_ready"'
              - 'name == "kube_endpoint_created"'
              - 'name == "kube_endpoint_info"'
              - 'name == "kube_endpoint_ports"'
              - 'name == "kube_horizontalpodautoscaler_spec_target_metric
              - 'name == "kube_horizontalpodautoscaler_status_target_metr
              - 'name == "kube_ingress_path"'
              - 'IsMatch(name, "kube_job_.+")'
              - 'name == "kube_lease_owner"'
              - 'name == "kube_lease_renew_time"'
              - 'name == "kube_namespace_created"'
              - 'name == "kube_node_spec_unschedulable"'
              - 'name == "kube_node_created"'
              - 'name == "kube_persistentvolume_status_phase"'
              - 'name == "kube_persistentvolumeclaim_status_phase"'
              - 'name == "kube_persistentvolumeclaim_access_mode"'
```

```
            - 'name == "kube_persistentvolumeclaim_info"'
            - 'name == "kube_persistentvolume_capacity_bytes"'
            - 'name == "kube_persistentvolume_claim_ref"'
            - 'name == "kube_persistentvolumeclaim_created"'
            - 'name == "kube_persistentvolumeclaim_resource_requests_st
            - 'name == "kube_pod_completion_time"'
            - 'name == "kube_pod_container_status_terminated"'
            - 'name == "kube_pod_container_status_waiting"'
            - 'name == "kube_pod_container_state_started"'
            - 'name == "kube_pod_created"'
            - 'name == "kube_pod_init_container_resource_requests"'
            - 'name == "kube_pod_init_container_resource_limits"'
            - 'name == "kube_pod_init_container_status_running"'
            - 'name == "kube_pod_init_container_status_terminated"'
            - 'name == "kube_pod_init_container_status_terminated_reaso
            - 'name == "kube_pod_init_container_status_waiting"'
            - 'name == "kube_pod_init_container_status_ready"'
            - 'name == "kube_pod_init_container_info"'
            - 'name == "kube_pod_init_container_status_restarts_total"'
            - 'name == "kube_pod_ips"'
            - 'name == "kube_pod_labels"'
            - 'name == "kube_pod_owner"'
            - 'name == "kube_pod_restart_policy"'
            - 'name == "kube_pod_start_time"'
            - 'name == "kube_pod_status_scheduled"'
            - 'name == "kube_pod_status_qos_class"'
            - 'name == "kube_pod_status_scheduled_time"'
            - 'name == "kube_pod_status_initialized_time"'
            - 'name == "kube_pod_service_account"'
            - 'name == "kube_pod_status_container_ready_time"'
            - 'name == "kube_pod_status_ready_time"'
            - 'name == "kube_pod_tolerations"'
            - 'name == "kube_poddisruptionbudget_status_observed_genera
            - 'name == "kube_poddisruptionbudget_status_pod_disruptions_
            - 'name == "kube_poddisruptionbudget_status_current_healthy
            - 'name == "kube_poddisruptionbudget_status_expected_pods"'
            - 'name == "kube_poddisruptionbudget_created"'
            - 'name == "kube_poddisruptionbudget_status_desired_healthy
            - 'name == "kube_replicaset_owner"'
            - 'name == "kube_replicaset_metadata_generation"'
            - 'name == "kube_replicaset_spec_replicas"'
            - 'name == "kube_replicaset_status_replicas"'
            - 'name == "kube_replicaset_status_ready_replicas"'
            - 'name == "kube_replicaset_status_fully_labeled_replicas"'
            - 'name == "kube_replicaset_created"'
            - 'name == "kube_replicaset_status_observed_generation"'
            - 'IsMatch(name, "kube_secret_.+")'
            - 'name == "kube_service_created"'
            - 'name == "kube_service_spec_type"'
            - 'name == "kube_validatingwebhookconfiguration_webhook_cli
            - 'IsMatch(name, "kube_volumeattachment_.+")'
            # cadvisor metrics we don't want
            - 'name ==  "container_blkio_device_usage_total"'
            - 'name ==  "container_file_descriptors"'
            - 'name ==  "container_last_seen"'
            - 'name ==  "container_processes"'
            - 'name ==  "container_start_time_seconds"'
            - 'name ==  "container_tasks_state"'
            - 'name ==  "container_threads"'
            - 'name ==  "container_threads_max"'
            - 'name ==  "container_sockets"'
            - 'name ==  "container_ulimits_soft"'
            # grab select container_cpu_ metrics
            - 'IsMatch(name, "^container_cpu_.+$") and not (name == "co
            - 'IsMatch(name, "^container_fs_.+$")'
            - 'IsMatch(name, "^container_network_.+$")'
            - 'IsMatch(name, "^container_memory_.+$") and not (name ==
            - 'IsMatch(name, "^container_spec_.+$")'
            - 'IsMatch(name, "^machine_.+$")'
            # prometheus-node-exporter metrics we don't want
            # easier to express the metrics we DO want
            - 'IsMatch(name, "^node_.+$") and not (name == "node_memory_
```

Page Information:
Page ID: 81068879
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:08:52