https://samber.github.io/awesome-prometheus-alerts/rules#kubernetes

5.1. Kubernetes : [kube-state-metrics](https://github.com/kubernetes/kube-state-metrics/tree/master/docs)(34 rules)[copy section]

$ wget https://raw.githubusercontent.com/samber/awesome-prometheus-alerts/master/dist/rules/kubernetes/kubestate-exporter.yml

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-1) 5.1.1. Kubernetes Node not ready

Node {{ $labels.node }} has been unready for a long time

- alert: KubernetesNodeNotReady

expr: kube\_node\_status\_condition{condition="Ready",status="true"} == 0

for: 10m

labels:

severity: critical

annotations:

summary: Kubernetes Node not ready (instance {{ $labels.instance }})

description: "Node {{ $labels.node }} has been unready for a long time\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-2) 5.1.2. Kubernetes Node memory pressure

Node {{ $labels.node }} has MemoryPressure condition

- alert: KubernetesNodeMemoryPressure

expr: kube\_node\_status\_condition{condition="MemoryPressure",status="true"} == 1

for: 2m

labels:

severity: critical

annotations:

summary: Kubernetes Node memory pressure (instance {{ $labels.instance }})

description: "Node {{ $labels.node }} has MemoryPressure condition\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-3) 5.1.3. Kubernetes Node disk pressure

Node {{ $labels.node }} has DiskPressure condition

- alert: KubernetesNodeDiskPressure

expr: kube\_node\_status\_condition{condition="DiskPressure",status="true"} == 1

for: 2m

labels:

severity: critical

annotations:

summary: Kubernetes Node disk pressure (instance {{ $labels.instance }})

description: "Node {{ $labels.node }} has DiskPressure condition\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-4) 5.1.4. Kubernetes Node network unavailable

Node {{ $labels.node }} has NetworkUnavailable condition

- alert: KubernetesNodeNetworkUnavailable

expr: kube\_node\_status\_condition{condition="NetworkUnavailable",status="true"} == 1

for: 2m

labels:

severity: critical

annotations:

summary: Kubernetes Node network unavailable (instance {{ $labels.instance }})

description: "Node {{ $labels.node }} has NetworkUnavailable condition\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-5) 5.1.5. Kubernetes Node out of pod capacity

Node {{ $labels.node }} is out of pod capacity

- alert: KubernetesNodeOutOfPodCapacity

expr: sum by (node) ((kube\_pod\_status\_phase{phase="Running"} == 1) + on(uid) group\_left(node) (0 \* kube\_pod\_info{pod\_template\_hash=""})) / sum by (node) (kube\_node\_status\_allocatable{resource="pods"}) \* 100 > 90

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes Node out of pod capacity (instance {{ $labels.instance }})

description: "Node {{ $labels.node }} is out of pod capacity\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-6) 5.1.6. Kubernetes Container oom killer

Container {{ $labels.container }} in pod {{ $labels.namespace }}/{{ $labels.pod }} has been OOMKilled {{ $value }} times in the last 10 minutes.

- alert: KubernetesContainerOomKiller

expr: (kube\_pod\_container\_status\_restarts\_total - kube\_pod\_container\_status\_restarts\_total offset 10m >= 1) and ignoring (reason) min\_over\_time(kube\_pod\_container\_status\_last\_terminated\_reason{reason="OOMKilled"}[10m]) == 1

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes Container oom killer (instance {{ $labels.instance }})

description: "Container {{ $labels.container }} in pod {{ $labels.namespace }}/{{ $labels.pod }} has been OOMKilled {{ $value }} times in the last 10 minutes.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-7) 5.1.7. Kubernetes Job failed

Job {{ $labels.namespace }}/{{ $labels.job\_name }} failed to complete

- alert: KubernetesJobFailed

expr: kube\_job\_status\_failed > 0

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes Job failed (instance {{ $labels.instance }})

description: "Job {{ $labels.namespace }}/{{ $labels.job\_name }} failed to complete\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-8) 5.1.8. Kubernetes CronJob suspended

CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is suspended

- alert: KubernetesCronjobSuspended

expr: kube\_cronjob\_spec\_suspend != 0

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes CronJob suspended (instance {{ $labels.instance }})

description: "CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is suspended\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-9) 5.1.9. Kubernetes PersistentVolumeClaim pending

PersistentVolumeClaim {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is pending

- alert: KubernetesPersistentvolumeclaimPending

expr: kube\_persistentvolumeclaim\_status\_phase{phase="Pending"} == 1

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes PersistentVolumeClaim pending (instance {{ $labels.instance }})

description: "PersistentVolumeClaim {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is pending\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-10) 5.1.10. Kubernetes Volume out of disk space

Volume is almost full (< 10% left)

- alert: KubernetesVolumeOutOfDiskSpace

expr: kubelet\_volume\_stats\_available\_bytes / kubelet\_volume\_stats\_capacity\_bytes \* 100 < 10

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes Volume out of disk space (instance {{ $labels.instance }})

description: "Volume is almost full (< 10% left)\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-11) 5.1.11. Kubernetes Volume full in four days

Volume under {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is expected to fill up within four days. Currently {{ $value | humanize }}% is available.

- alert: KubernetesVolumeFullInFourDays

expr: predict\_linear(kubelet\_volume\_stats\_available\_bytes[6h:5m], 4 \* 24 \* 3600) < 0

for: 0m

labels:

severity: critical

annotations:

summary: Kubernetes Volume full in four days (instance {{ $labels.instance }})

description: "Volume under {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is expected to fill up within four days. Currently {{ $value | humanize }}% is available.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-12) 5.1.12. Kubernetes PersistentVolume error

Persistent volume {{ $labels.persistentvolume }} is in bad state

- alert: KubernetesPersistentvolumeError

expr: kube\_persistentvolume\_status\_phase{phase=~"Failed|Pending", job="kube-state-metrics"} > 0

for: 0m

labels:

severity: critical

annotations:

summary: Kubernetes PersistentVolume error (instance {{ $labels.instance }})

description: "Persistent volume {{ $labels.persistentvolume }} is in bad state\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-13) 5.1.13. Kubernetes StatefulSet down

StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} went down

- alert: KubernetesStatefulsetDown

expr: kube\_statefulset\_replicas != kube\_statefulset\_status\_replicas\_ready > 0

for: 1m

labels:

severity: critical

annotations:

summary: Kubernetes StatefulSet down (instance {{ $labels.instance }})

description: "StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} went down\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-14) 5.1.14. Kubernetes HPA scale inability

HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is unable to scale

- alert: KubernetesHpaScaleInability

expr: (kube\_horizontalpodautoscaler\_spec\_max\_replicas - kube\_horizontalpodautoscaler\_status\_desired\_replicas) \* on (horizontalpodautoscaler,namespace) (kube\_horizontalpodautoscaler\_status\_condition{condition="ScalingLimited", status="true"} == 1) == 0

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes HPA scale inability (instance {{ $labels.instance }})

description: "HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is unable to scale\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-15) 5.1.15. Kubernetes HPA metrics unavailability

HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is unable to collect metrics

- alert: KubernetesHpaMetricsUnavailability

expr: kube\_horizontalpodautoscaler\_status\_condition{status="false", condition="ScalingActive"} == 1

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes HPA metrics unavailability (instance {{ $labels.instance }})

description: "HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is unable to collect metrics\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-16) 5.1.16. Kubernetes HPA scale maximum

HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} has hit maximum number of desired pods

- alert: KubernetesHpaScaleMaximum

expr: (kube\_horizontalpodautoscaler\_status\_desired\_replicas >= kube\_horizontalpodautoscaler\_spec\_max\_replicas) and (kube\_horizontalpodautoscaler\_spec\_max\_replicas > 1) and (kube\_horizontalpodautoscaler\_spec\_min\_replicas != kube\_horizontalpodautoscaler\_spec\_max\_replicas)

for: 2m

labels:

severity: info

annotations:

summary: Kubernetes HPA scale maximum (instance {{ $labels.instance }})

description: "HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} has hit maximum number of desired pods\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-17) 5.1.17. Kubernetes HPA underutilized

HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is constantly at minimum replicas for 50% of the time. Potential cost saving here.

- alert: KubernetesHpaUnderutilized

expr: max(quantile\_over\_time(0.5, kube\_horizontalpodautoscaler\_status\_desired\_replicas[1d]) == kube\_horizontalpodautoscaler\_spec\_min\_replicas) by (horizontalpodautoscaler) > 3

for: 0m

labels:

severity: info

annotations:

summary: Kubernetes HPA underutilized (instance {{ $labels.instance }})

description: "HPA {{ $labels.namespace }}/{{ $labels.horizontalpodautoscaler }} is constantly at minimum replicas for 50% of the time. Potential cost saving here.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-18) 5.1.18. Kubernetes Pod not healthy

Pod {{ $labels.namespace }}/{{ $labels.pod }} has been in a non-running state for longer than 15 minutes.

- alert: KubernetesPodNotHealthy

expr: sum by (namespace, pod) (kube\_pod\_status\_phase{phase=~"Pending|Unknown|Failed"}) > 0

for: 15m

labels:

severity: critical

annotations:

summary: Kubernetes Pod not healthy (instance {{ $labels.instance }})

description: "Pod {{ $labels.namespace }}/{{ $labels.pod }} has been in a non-running state for longer than 15 minutes.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-19) 5.1.19. Kubernetes pod crash looping

Pod {{ $labels.namespace }}/{{ $labels.pod }} is crash looping

- alert: KubernetesPodCrashLooping

expr: increase(kube\_pod\_container\_status\_restarts\_total[1m]) > 3

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes pod crash looping (instance {{ $labels.instance }})

description: "Pod {{ $labels.namespace }}/{{ $labels.pod }} is crash looping\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-20) 5.1.20. Kubernetes ReplicaSet replicas mismatch

ReplicaSet {{ $labels.namespace }}/{{ $labels.replicaset }} replicas mismatch

- alert: KubernetesReplicasetReplicasMismatch

expr: kube\_replicaset\_spec\_replicas != kube\_replicaset\_status\_ready\_replicas

for: 10m

labels:

severity: warning

annotations:

summary: Kubernetes ReplicaSet replicas mismatch (instance {{ $labels.instance }})

description: "ReplicaSet {{ $labels.namespace }}/{{ $labels.replicaset }} replicas mismatch\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-21) 5.1.21. Kubernetes Deployment replicas mismatch

Deployment {{ $labels.namespace }}/{{ $labels.deployment }} replicas mismatch

- alert: KubernetesDeploymentReplicasMismatch

expr: kube\_deployment\_spec\_replicas != kube\_deployment\_status\_replicas\_available

for: 10m

labels:

severity: warning

annotations:

summary: Kubernetes Deployment replicas mismatch (instance {{ $labels.instance }})

description: "Deployment {{ $labels.namespace }}/{{ $labels.deployment }} replicas mismatch\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-22) 5.1.22. Kubernetes StatefulSet replicas mismatch

StatefulSet does not match the expected number of replicas.

- alert: KubernetesStatefulsetReplicasMismatch

expr: kube\_statefulset\_status\_replicas\_ready != kube\_statefulset\_status\_replicas

for: 10m

labels:

severity: warning

annotations:

summary: Kubernetes StatefulSet replicas mismatch (instance {{ $labels.instance }})

description: "StatefulSet does not match the expected number of replicas.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-23) 5.1.23. Kubernetes Deployment generation mismatch

Deployment {{ $labels.namespace }}/{{ $labels.deployment }} has failed but has not been rolled back.

- alert: KubernetesDeploymentGenerationMismatch

expr: kube\_deployment\_status\_observed\_generation != kube\_deployment\_metadata\_generation

for: 10m

labels:

severity: critical

annotations:

summary: Kubernetes Deployment generation mismatch (instance {{ $labels.instance }})

description: "Deployment {{ $labels.namespace }}/{{ $labels.deployment }} has failed but has not been rolled back.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-24) 5.1.24. Kubernetes StatefulSet generation mismatch

StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} has failed but has not been rolled back.

- alert: KubernetesStatefulsetGenerationMismatch

expr: kube\_statefulset\_status\_observed\_generation != kube\_statefulset\_metadata\_generation

for: 10m

labels:

severity: critical

annotations:

summary: Kubernetes StatefulSet generation mismatch (instance {{ $labels.instance }})

description: "StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} has failed but has not been rolled back.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-25) 5.1.25. Kubernetes StatefulSet update not rolled out

StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} update has not been rolled out.

- alert: KubernetesStatefulsetUpdateNotRolledOut

expr: max without (revision) (kube\_statefulset\_status\_current\_revision unless kube\_statefulset\_status\_update\_revision) \* (kube\_statefulset\_replicas != kube\_statefulset\_status\_replicas\_updated)

for: 10m

labels:

severity: warning

annotations:

summary: Kubernetes StatefulSet update not rolled out (instance {{ $labels.instance }})

description: "StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} update has not been rolled out.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-26) 5.1.26. Kubernetes DaemonSet rollout stuck

Some Pods of DaemonSet {{ $labels.namespace }}/{{ $labels.daemonset }} are not scheduled or not ready

- alert: KubernetesDaemonsetRolloutStuck

expr: kube\_daemonset\_status\_number\_ready / kube\_daemonset\_status\_desired\_number\_scheduled \* 100 < 100 or kube\_daemonset\_status\_desired\_number\_scheduled - kube\_daemonset\_status\_current\_number\_scheduled > 0

for: 10m

labels:

severity: warning

annotations:

summary: Kubernetes DaemonSet rollout stuck (instance {{ $labels.instance }})

description: "Some Pods of DaemonSet {{ $labels.namespace }}/{{ $labels.daemonset }} are not scheduled or not ready\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-27) 5.1.27. Kubernetes DaemonSet misscheduled

Some Pods of DaemonSet {{ $labels.namespace }}/{{ $labels.daemonset }} are running where they are not supposed to run

- alert: KubernetesDaemonsetMisscheduled

expr: kube\_daemonset\_status\_number\_misscheduled > 0

for: 1m

labels:

severity: critical

annotations:

summary: Kubernetes DaemonSet misscheduled (instance {{ $labels.instance }})

description: "Some Pods of DaemonSet {{ $labels.namespace }}/{{ $labels.daemonset }} are running where they are not supposed to run\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-28) 5.1.28. Kubernetes CronJob too long

CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is taking more than 1h to complete.

*# Threshold should be customized for each cronjob name.*

- alert: KubernetesCronjobTooLong

expr: time() - kube\_cronjob\_next\_schedule\_time > 3600

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes CronJob too long (instance {{ $labels.instance }})

description: "CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is taking more than 1h to complete.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-29) 5.1.29. Kubernetes Job slow completion

Kubernetes Job {{ $labels.namespace }}/{{ $labels.job\_name }} did not complete in time.

- alert: KubernetesJobSlowCompletion

expr: kube\_job\_spec\_completions - kube\_job\_status\_succeeded - kube\_job\_status\_failed > 0

for: 12h

labels:

severity: critical

annotations:

summary: Kubernetes Job slow completion (instance {{ $labels.instance }})

description: "Kubernetes Job {{ $labels.namespace }}/{{ $labels.job\_name }} did not complete in time.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-30) 5.1.30. Kubernetes API server errors

Kubernetes API server is experiencing high error rate

- alert: KubernetesApiServerErrors

expr: sum(rate(apiserver\_request\_total{job="apiserver",code=~"(?:5..)"}[1m])) by (instance, job) / sum(rate(apiserver\_request\_total{job="apiserver"}[1m])) by (instance, job) \* 100 > 3

for: 2m

labels:

severity: critical

annotations:

summary: Kubernetes API server errors (instance {{ $labels.instance }})

description: "Kubernetes API server is experiencing high error rate\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-31) 5.1.31. Kubernetes API client errors

Kubernetes API client is experiencing high error rate

- alert: KubernetesApiClientErrors

expr: (sum(rate(rest\_client\_requests\_total{code=~"(4|5).."}[1m])) by (instance, job) / sum(rate(rest\_client\_requests\_total[1m])) by (instance, job)) \* 100 > 1

for: 2m

labels:

severity: critical

annotations:

summary: Kubernetes API client errors (instance {{ $labels.instance }})

description: "Kubernetes API client is experiencing high error rate\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-32) 5.1.32. Kubernetes client certificate expires next week

A client certificate used to authenticate to the apiserver is expiring next week.

- alert: KubernetesClientCertificateExpiresNextWeek

expr: apiserver\_client\_certificate\_expiration\_seconds\_count{job="apiserver"} > 0 and histogram\_quantile(0.01, sum by (job, le) (rate(apiserver\_client\_certificate\_expiration\_seconds\_bucket{job="apiserver"}[5m]))) < 7\*24\*60\*60

for: 0m

labels:

severity: warning

annotations:

summary: Kubernetes client certificate expires next week (instance {{ $labels.instance }})

description: "A client certificate used to authenticate to the apiserver is expiring next week.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-33) 5.1.33. Kubernetes client certificate expires soon

A client certificate used to authenticate to the apiserver is expiring in less than 24.0 hours.

- alert: KubernetesClientCertificateExpiresSoon

expr: apiserver\_client\_certificate\_expiration\_seconds\_count{job="apiserver"} > 0 and histogram\_quantile(0.01, sum by (job, le) (rate(apiserver\_client\_certificate\_expiration\_seconds\_bucket{job="apiserver"}[5m]))) < 24\*60\*60

for: 0m

labels:

severity: critical

annotations:

summary: Kubernetes client certificate expires soon (instance {{ $labels.instance }})

description: "A client certificate used to authenticate to the apiserver is expiring in less than 24.0 hours.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-kubernetes-1-34) 5.1.34. Kubernetes API server latency

Kubernetes API server has a 99th percentile latency of {{ $value }} seconds for {{ $labels.verb }} {{ $labels.resource }}.

- alert: KubernetesApiServerLatency

expr: histogram\_quantile(0.99, sum(rate(apiserver\_request\_duration\_seconds\_bucket{verb!~"(?:CONNECT|WATCHLIST|WATCH|PROXY)"} [10m])) WITHOUT (subresource)) > 1

for: 2m

labels:

severity: warning

annotations:

summary: Kubernetes API server latency (instance {{ $labels.instance }})

description: "Kubernetes API server has a 99th percentile latency of {{ $value }} seconds for {{ $labels.verb }} {{ $labels.resource }}.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

Docker containers : [google/cAdvisor](https://github.com/google/cadvisor)(8 rules)

$ wget https://raw.githubusercontent.com/samber/awesome-prometheus-alerts/master/dist/rules/docker-containers/google-cadvisor.yml

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-1) 1.4.1. Container killed

A container has disappeared

*# This rule can be very noisy in dynamic infra with legitimate container start/stop/deployment.*

- alert: ContainerKilled

expr: time() - container\_last\_seen > 60

for: 0m

labels:

severity: warning

annotations:

summary: Container killed (instance {{ $labels.instance }})

description: "A container has disappeared\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-2) 1.4.2. Container absent

A container is absent for 5 min

*# This rule can be very noisy in dynamic infra with legitimate container start/stop/deployment.*

- alert: ContainerAbsent

expr: absent(container\_last\_seen)

for: 5m

labels:

severity: warning

annotations:

summary: Container absent (instance {{ $labels.instance }})

description: "A container is absent for 5 min\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-3) 1.4.3. Container High CPU utilization

Container CPU utilization is above 80%

- alert: ContainerHighCpuUtilization

expr: (sum(rate(container\_cpu\_usage\_seconds\_total{container!=""}[5m])) by (pod, container) / sum(container\_spec\_cpu\_quota{container!=""}/container\_spec\_cpu\_period{container!=""}) by (pod, container) \* 100) > 80

for: 2m

labels:

severity: warning

annotations:

summary: Container High CPU utilization (instance {{ $labels.instance }})

description: "Container CPU utilization is above 80%\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-4) 1.4.4. Container High Memory usage

Container Memory usage is above 80%

*# See https://medium.com/faun/how-much-is-too-much-the-linux-oomkiller-and-used-memory-d32186f29c9d*

- alert: ContainerHighMemoryUsage

expr: (sum(container\_memory\_working\_set\_bytes{name!=""}) BY (instance, name) / sum(container\_spec\_memory\_limit\_bytes > 0) BY (instance, name) \* 100) > 80

for: 2m

labels:

severity: warning

annotations:

summary: Container High Memory usage (instance {{ $labels.instance }})

description: "Container Memory usage is above 80%\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-5) 1.4.5. Container Volume usage

Container Volume usage is above 80%

- alert: ContainerVolumeUsage

expr: (1 - (sum(container\_fs\_inodes\_free{name!=""}) BY (instance) / sum(container\_fs\_inodes\_total) BY (instance))) \* 100 > 80

for: 2m

labels:

severity: warning

annotations:

summary: Container Volume usage (instance {{ $labels.instance }})

description: "Container Volume usage is above 80%\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-6) 1.4.6. Container high throttle rate

Container is being throttled

- alert: ContainerHighThrottleRate

expr: sum(increase(container\_cpu\_cfs\_throttled\_periods\_total{container!=""}[5m])) by (container, pod, namespace) / sum(increase(container\_cpu\_cfs\_periods\_total[5m])) by (container, pod, namespace) > ( 25 / 100 )

for: 5m

labels:

severity: warning

annotations:

summary: Container high throttle rate (instance {{ $labels.instance }})

description: "Container is being throttled\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-7) 1.4.7. Container Low CPU utilization

Container CPU utilization is under 20% for 1 week. Consider reducing the allocated CPU.

- alert: ContainerLowCpuUtilization

expr: (sum(rate(container\_cpu\_usage\_seconds\_total{container!=""}[5m])) by (pod, container) / sum(container\_spec\_cpu\_quota{container!=""}/container\_spec\_cpu\_period{container!=""}) by (pod, container) \* 100) < 20

for: 7d

labels:

severity: info

annotations:

summary: Container Low CPU utilization (instance {{ $labels.instance }})

description: "Container CPU utilization is under 20% for 1 week. Consider reducing the allocated CPU.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

* [#](https://samber.github.io/awesome-prometheus-alerts/rules#rule-docker-containers-1-8) 1.4.8. Container Low Memory usage

Container Memory usage is under 20% for 1 week. Consider reducing the allocated memory.

- alert: ContainerLowMemoryUsage

expr: (sum(container\_memory\_working\_set\_bytes{name!=""}) BY (instance, name) / sum(container\_spec\_memory\_limit\_bytes > 0) BY (instance, name) \* 100) < 20

for: 7d

labels:

severity: info

annotations:

summary: Container Low Memory usage (instance {{ $labels.instance }})

description: "Container Memory usage is under 20% for 1 week. Consider reducing the allocated memory.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"