

Daemonset yaml and metrics

```
deepa@ubuntu:~/devops/kind/daemonset$ kubectl get daemonset node-monitor
NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
node-monitor  1         1         1       1             1           <none>          4m10s
deepa@ubuntu:~/devops/kind/daemonset$
```

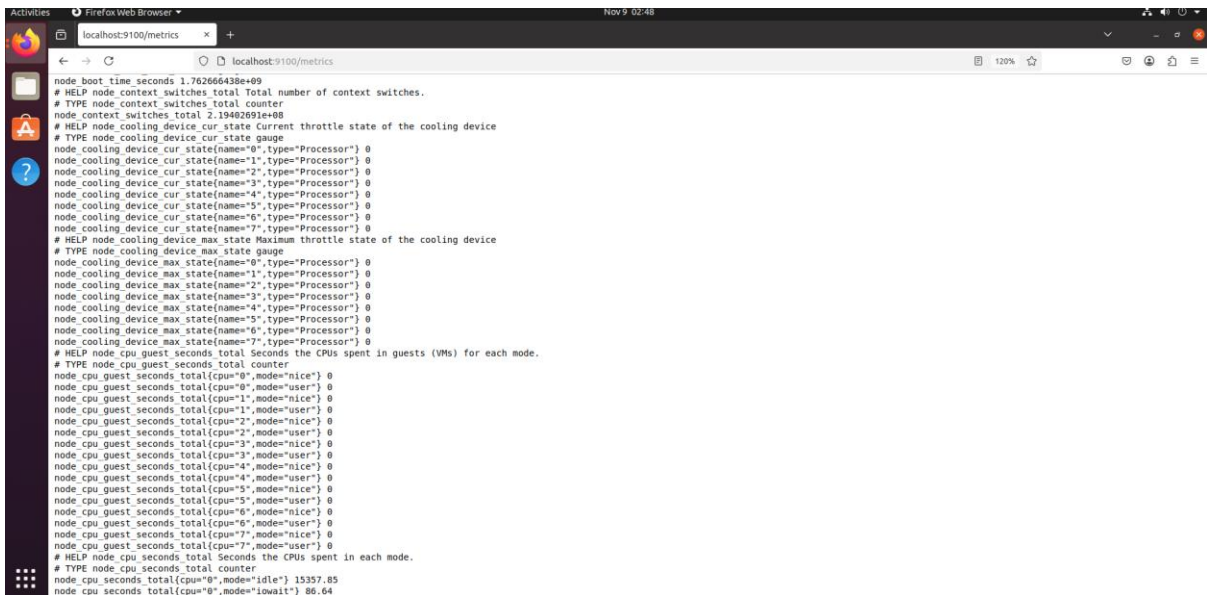
```
deepa@ubuntu:~/devops/kind/daemonset$ kubectl get pods -o wide -l app=node-monitor
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE               NOMINATED NODE   READINESS GATES
node-monitor-2fc4q  1/1     Running   0          7ms   10.244.0.17   dev-cluster-control-plane  <none>          <none>
deepa@ubuntu:~/devops/kind/daemonset$
```

To see metrics: port forward

```
deepa@ubuntu:~/devops/kind/daemonset$ #port forwarding to local system
deepa@ubuntu:~/devops/kind/daemonset$
deepa@ubuntu:~/devops/kind/daemonset$ kubectl port-forward pod/node-monitor-2fc4q 9100:9100
Forwarding from 127.0.0.1:9100 → 9100
Forwarding from [::1]:9100 → 9100

```

<http://localhost:9100/metrics>



```
node boot_time_seconds 1.76266438e+09
# HELP node_context_switches_total Total number of context switches.
# TYPE node_context_switches_total counter
node_context_switches_total 2.19402691e+08
# HELP node_cooling_device_cur_state Current throttle state of the cooling device
# TYPE node_cooling_device_cur_state gauge
node_cooling_device_cur_state{name="0",type="Processor"} 0
node_cooling_device_cur_state{name="1",type="Processor"} 0
node_cooling_device_cur_state{name="2",type="Processor"} 0
node_cooling_device_cur_state{name="3",type="Processor"} 0
node_cooling_device_cur_state{name="4",type="Processor"} 0
node_cooling_device_cur_state{name="5",type="Processor"} 0
node_cooling_device_cur_state{name="6",type="Processor"} 0
node_cooling_device_cur_state{name="7",type="Processor"} 0
# HELP node_cooling_device_max_state Maximum throttle state of the cooling device
# TYPE node_cooling_device_max_state gauge
node_cooling_device_max_state{name="0",type="Processor"} 0
node_cooling_device_max_state{name="1",type="Processor"} 0
node_cooling_device_max_state{name="2",type="Processor"} 0
node_cooling_device_max_state{name="3",type="Processor"} 0
node_cooling_device_max_state{name="4",type="Processor"} 0
node_cooling_device_max_state{name="5",type="Processor"} 0
node_cooling_device_max_state{name="6",type="Processor"} 0
node_cooling_device_max_state{name="7",type="Processor"} 0
# HELP node_cpu_guest_seconds_total Seconds the CPUs spent in guests (VMs) for each node.
# TYPE node_cpu_guest_seconds_total counter
node_cpu_guest_seconds_total{cpu="0",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="0",mode="user"} 0
node_cpu_guest_seconds_total{cpu="1",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="1",mode="user"} 0
node_cpu_guest_seconds_total{cpu="2",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="2",mode="user"} 0
node_cpu_guest_seconds_total{cpu="3",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="3",mode="user"} 0
node_cpu_guest_seconds_total{cpu="4",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="4",mode="user"} 0
node_cpu_guest_seconds_total{cpu="5",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="5",mode="user"} 0
node_cpu_guest_seconds_total{cpu="6",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="6",mode="user"} 0
node_cpu_guest_seconds_total{cpu="7",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="7",mode="user"} 0
# HELP node_cpu_seconds_total Seconds the CPUs spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 15357.85
node_cpu_seconds_total{cpu="0",mode="iowait"} 86.64
```

Use an ephemeral debug container:

`kubectl debug -it node-monitor-2fc4q --image=busybox --target=node-exporter`

inside the container:

`wget -qO- localhost:9100/metrics | head`

```
deepa@ubuntu:~/devops/kind/daemonset$
deepa@ubuntu:~/devops/kind/daemonset$ kubectl debug -it node-monitor-2fc4q --image=busybox --target=node-exporter
Targeting container "node-exporter". If you don't see processes from this container it may be because the container runtime doesn't support this feature.
--profile=legacy is deprecated and will be removed in the future. It is recommended to explicitly specify a profile, for example "--profile=general".
Defaulting debug container name to debugger-rd78p.
All commands and output from this session will be recorded in container logs, including credentials and sensitive information passed through the command prompt.
If you don't see a command prompt, try pressing enter.
/ # wget -qO- localhost:9100/metrics | head
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 4.0193e-05
go_gc_duration_seconds{quantile="0.25"} 6.3834e-05
go_gc_duration_seconds{quantile="0.5"} 0.00012521
go_gc_duration_seconds{quantile="0.75"} 0.000226406
go_gc_duration_seconds{quantile="1"} 0.000371654
go_gc_duration_seconds_sum 0.000931788
go_gc_duration_seconds_count 6
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environment variable, and the runtime/debug.SetGCPercent function. Sourced from /gc/gogc:percent.
/ #
```