Node Temperature/Power

And why we probably cannot collect it

All series with ‘node’ in the name (in nautilus’ database)

```
[
"kubelet_volume_stats_inodes",
"kubelet_volume_stats_inodes_free",
 "kubelet_volume_stats_inodes_used",
"node_namespace_pod:kube_pod_info:",
 "node_namespace_pod_container:container_cpu_usage_seconds_total:sum_irate",
 "node_namespace_pod_container:container_memory_cache",
 "node_namespace_pod_container:container_memory_rss",
 "node_namespace_pod_container:container_memory_swap",
 "node_namespace_pod_container:container_memory_working_set_bytes"
]
```

(This is for all series that have had some data in the last week)
Collected by querying in node_metrics_retrieval.py

Types of series with node in the name or as a label (K8S metrics)

IPAM (IP Address Manager)

Kube API Server Nodeport Allocator

Kubelet Volume Stats

Service Controller and Node Controller Node Sync Stats

Node Collector Stats (Node Health, Zone Size)

Node Info: Name, CPU Usage, Memory

Other: Pod Resource Limit (node is a label), Node Evictions, Storage Volume in Use (label) , Node Authorizer Graph Actions

Note: nothing on the site has any “power” or “temp”/“temperature” and these series aren’t helpful in collecting those metrics either, so there does not currently appear to be a way to collect stats on node temperature and power.

https://kubernetes.io/docs/reference/instrumentation/metrics/ all these series (next slide) found by searching through this website

All series with node in the name or node as a label (for K8S metrics)

kube_pod_resource_limit (node is a label)

node_collector_evictions_total

node_cpu_usage_seconds_total

node_memory_working_set_bytes

kube_apiserver_nodeport_allocator_allocated_ports

kube_apiserver_nodeport_allocator_available_ports

kubelet_node_name

kubelet_volume_stats_inodes

kubelet_volume_stats_inodes_free

kubelet_volume_stats_inodes_used

node_authorizer_graph_actions_duration_seconds

node_collector_unhealthy_nodes_in_zone

node_collector_update_all_nodes_health_duration_seconds

node_collector_update_node_health_duration_seconds

node_collector_zone_health

node_collector_zone_size

node_controller_cloud_provider_taint_removal_delay_seconds

node_controller_initial_node_sync_delay_seconds

node_ipam_controller_cidrset_allocation_tries_per_request

node_ipam_controller_cidrset_cidrs_allocations_total

node_ipam_controller_cidrset_cidrs_releases_total

node_ipam_controller_cidrset_usage_cidrs

node_ipam_controller_cirdset_max_cidrs

node_ipam_controller_multicidrset_allocation_tries_per_request

node_ipam_controller_multicidrset_cidrs_allocations_total

node_ipam_controller_multicidrset_cidrs_releases_total

node_ipam_controller_multicidrset_usage_cidrs

node_ipam_controller_multicirdset_max_cidrs

node_swap_usage_bytes

service_controller_nodesync_error_total

service_controller_nodesync_latency_seconds

storage_count_attachable_volumes_in_use

Conclusion

Although there are many metrics that can be collected involving nodes, none appear to be useful in finding the temperature or power usage of the node.

For more information on this, look in the further_node_investigation folder.

- To find all recently updated queries collected by nautilus with ‘node’ in the name, run node_metrics_retrieval.py
- To see what data is returned by querying these metrics, run query_investigation.py
- Each of these two python files prints the information to the terminal as well as writing to a text file in the same folder.