Monitoring Calico Components

Thursday, Nov 21, 2024

Prepared by Davide Sellitri,

Customer Success Engineer @ Tigera

Agenda

- Introduction
- Observability VS Monitoring
- Calico Typha & Felix
- Lab description
- Example of Grafana Dashboard
- Top Metrics & Demo
- Top Insightful metrics

TIGERA

Introduction

TIGERA

Introduction: Calico Open Source

8M+
Nodes

1M+
Clusters

1.4B+
Docker Pulls

50k+
Enterprises

>50%
of Fortune 100

166
Countries

Most adopted container networking and security solution

Introduction: Calico Open Source

Choice of Data Plane

- Pluggable Data Plane
- eBPF, Linux, Windows, VPP

Best in class performance

- Blazing fast performance
- Minimal CPU usage & occupancy
- Lower costs

Full Kubernetes Network policy support

- Full implementation Kubernetes network policies
- Additional support for policies across namespaces

Workload Interoperability

- Unified policy across hosts, bare-metal, VMs, and containers
- Mix and match workload types

Kubernetes Native Security Policy Model

- Declarative security policies
- Unified model from host to application layers

Scalable Networking with Encryption

- Exceptional scalability
- Advanced IP Address Management

Introduction: Prometheus & Grafana

Metrics Collection
pull-based model to scrape metrics over HTTP from applications and services

Time-Series Database
Store data as time-series, for understanding trends and performance.

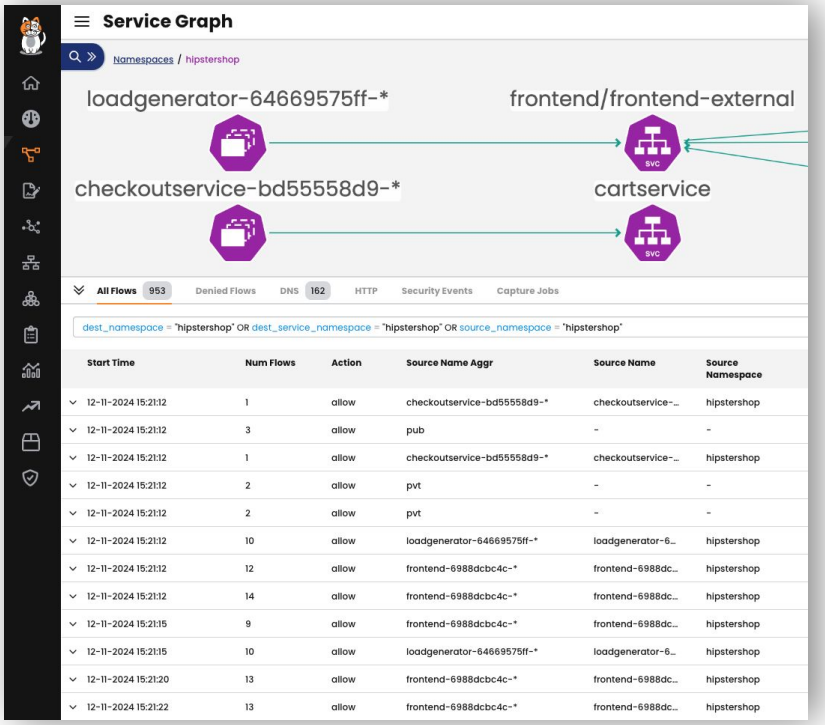
Alerting and Query Language (PromQL)
Built-in alerting system and a query language called PromQL, for custom alerts and complex queries

Data Visualization
Visualize data providing dashboards, graphs, and other visual elements for an intuitive view of metrics.

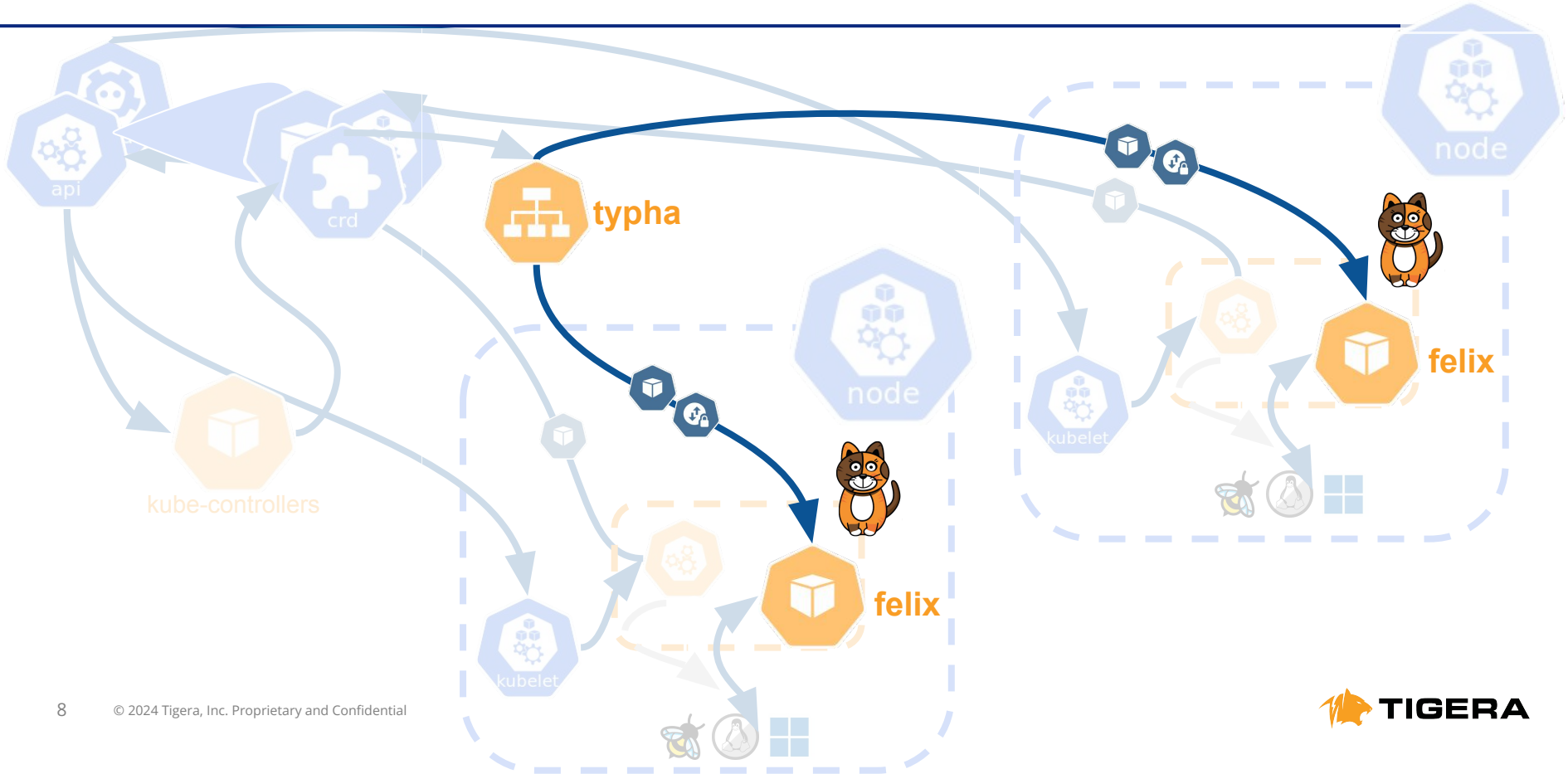
Multi-Source Integration
Support a wide range of data sources (Prometheus, Elasticsearch, AWS, MySQL, etc.)

Alerting and Collaboration
Offers alerting features and team collaboration capabilities for incident response.

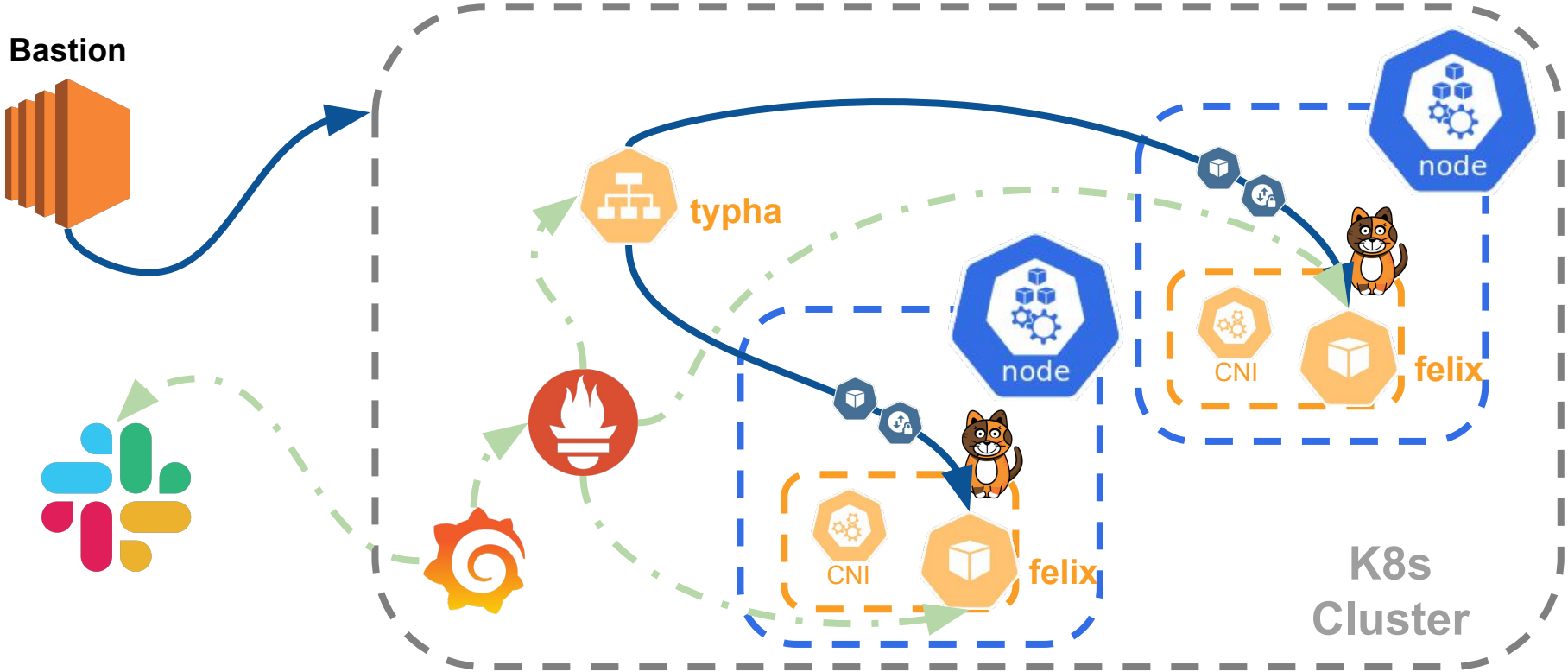
Observability VS Monitoring



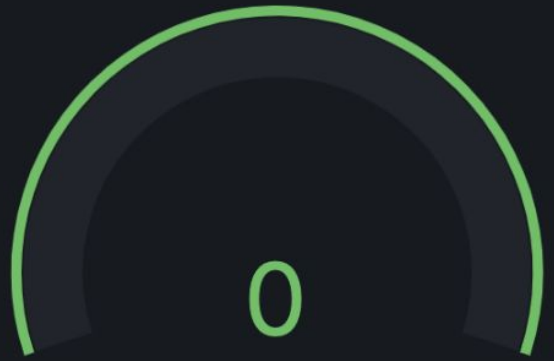
Calico Typha & Felix



Lab description



Nodes without Active Calico nodes



Node Count

FELIX - Active Calico nodes

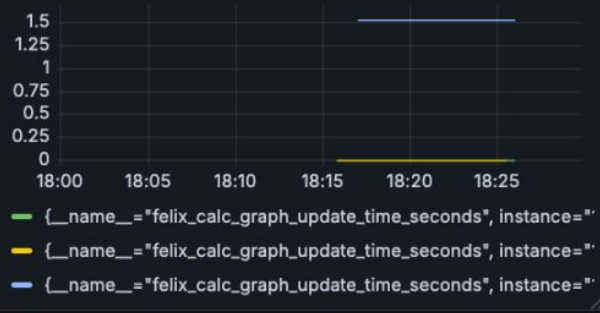
TYPHA - CPU Usage

TYPHA - 99%ile Breadcrumb Size

TYPHA - Memory Usage

TYPHA - Non-blocking breadcrumb fraction

FELIX graph update time (Quantile 0.99)



FELIX - Dataplane apply time (Quantile 0.99)

FELIX - Log Errors

Top Metrics

- CPU Usage
- Memory Usage

- TYPHA - breadcrumb size(Quantile 0.99)
- TYPHA - Non-blocking breadcrumb fraction
- TYPHA - Client fall-behind(Quantile 0.99)
- TYPHA - Client Ping Latency(Quantile 0.99)

- FELIX - Active calico nodes
- FELIX - resync started
- FELIX - Error Plot
- FELIX - Graph update time (Quantile 0.99)
- FELIX - Dataplane apply time (Quantile 0.99)

Top Metrics

- CPU Usage: `rate(process_cpu_seconds_total[30s]) * 100`
- Memory Usage: `rate(process_resident_memory_bytes[30s])`

- FELIX - Active calico nodes: `count(felix_cluster_num_hosts)`
- FELIX - resync started: `sum(rate(felix_resyncs_started[5m]))`
- FELIX - Error Plot:`rate(felix_ipset_errors[5m]) || rate(felix_iptables_restore_errors[5m]) || rate(felix_iptables_save_errors[5m]) || rate(felix_log_errors[5m])`
- FELIX - Graph update time (Quantile 0.99):
  `felix_calc_graph_update_time_seconds{quantile="0.99"}`
- FELIX - Dataplane apply time (Quantile 0.99):
  `felix_int_dataplane_apply_time_seconds{quantile="0.99"}`

- TYPHA - breadcrumb size(Quantile 0.99): `max by (instance) (typha_breadcrumb_size{quantile="0.99"})`
- TYPHA - Non-blocking breadcrumb fraction: `(sum by (instance) (rate(typha_breadcrumb_non_block[30s])))/((sum by (instance) (rate(typha_breadcrumb_non_block[30s])))+(sum by (instance) (rate(typha_breadcrumb_block[30s]))))`
- TYPHA - Client fall-behind(Quantile 0.99):`max by (instance) (typha_client_latency_secs{quantile='0.99'})`
- TYPHA - Client Ping Latency(Quantile 0.99):`max by (instance) (typha_ping_latency{quantile="0.99"})`

Q & A

GitHub

TIGERA

Follow us on:

Thank you!

Feedback

TIGERA

Follow us on:

Additional Slides

Top Metrics Details

TIGERA

Top Metrics: CPU Usage

Metric

*rate(process_cpu_seconds_total[30s]) * 100*

Explanation

CPU in use by typha represented as a percentage of a core

Threshold value recommendation

A spike at startup is normal.

It is recommended to first achieve a baseline and then monitor for any unexpected increases from this baseline

A rule of thumb would be that maintained CPU usage above 90% warrants investigation

Threshold breach symptoms

Unexpected maintained CPU usage could cause Typha to fall behind in updating its clients (e.g. Felix) and could cause delays to policy updates

Threshold breach recommendations

Check CPU usage on k8s nodes. Increase resources if needed, rollout restart typha(s) if needed

Priority Level

Recommended

Top Metrics: Memory Usage

Metric

rate(process_resident_memory_bytes[30s])

Explanation

Amount of memory in use by Typha

Threshold value recommendation

Recommended to first achieve a baseline and then monitor for any unexpected increases from this baseline.

A rule of thumb would be that, maintained memory usage above 90% of what is available from the underlying node warrants investigation

Threshold breach symptoms

Unexpected, maintained, memory usage could cause Typha to fall behind in updating its clients (e.g. Felix) and could cause delays to policy updates

Threshold breach recommendations

Check memory usage on k8s nodes. Increase resources if needed, rollout restart typha(s) if needed

Priority Level

Recommended

Top Metrics: Active Calico nodes

Metric

max(felix_cluster_num_hosts)

Explanation

Total number of nodes in the cluster that have calico-node deployed and running

Threshold value recommendation

This value should be equal to the number of nodes in the cluster

If there are discrepancies here then calico-nodes on some nodes are having issues

Threshold breach symptoms

Calico Network Policies on affected nodes could be out of sync

Threshold breach recommendations

Check Calico-node logs, rollout restart Calico-Node if needed

Priority Level

Recommended

Top Metrics: Felix resync started

Metric

sum(rate(felix_resyncs_started[5m]))

Explanation

This is the number of times that Typha has reported to Felix that it is re-connecting with the datastore

Threshold value recommendation

Occasional resyncs are normal. Resync counters rising rapidly warrants investigation

Threshold breach symptoms

Typha pods may be having issues or experiencing resource contention. Some Calico-nodes that are paired with Typha pods experiencing issues will not be able to sync with the datastore

Threshold breach recommendations

Redeploy Typha, check resource contention and network connectivity from Typha to the datastore

Priority Level

Recommended

Top Metrics: Felix Errors Plot

Metric

rate(felix_ipset_errors[5m]) || rate(felix_iptables_restore_errors[5m]) || rate(felix_iptables_save_errors[5m]) || rate(felix_log_errors[5m])

Explanation

Checks if there have been any iptables-save, iptables-restore or ipset command errors in the past 5 minutes. Keeps track of what node is reporting which error

Threshold value recommendation

Occasional errors are normal. Error counters rising rapidly warrants investigation.

For this specific metric it is worth focusing on the metric that is spiking and referencing that metric information.

Threshold breach symptoms

Dependant on the specific metric that is logging errors

Threshold breach recommendations

If more than one metric is rising it is worth checking if all rising metrics are related to a specific calico-node. If this is the case then the issue is local to that calico-node. Check calico-node logs. Check resource usage for the node and calico-node pod. If more than one metric is rising rapidly across all calico-nodes then it is a cluster-wide issue and cluster health must be checked. Check cluster resource usage, cluster networking/infrastructure health, restart calico-nodes and calico-typha pods

Priority Level

Recommended

Top Metrics: Dataplane apply time (Quantile 0.99)

Metric

felix_int_dataplane_apply_time_seconds{quantile="0.5"}/{quantile="0 .9"}/{quantile="0.99"}

Explanation

Time in seconds that it took to apply a dataplane update, viewed at the median, 90th percentile and 99th percentile

Threshold value recommendation

Thresholds will vary depending on cluster size and rate of churn. It is recommended that a baseline be set to determine a normal threshold value. Examples of what has been seen in the field:

- 3-node test cluster averaging 100ms
- 1000-node 15x federated cluster averaging 30s (potentially larger at start of business day)

Threshold breach symptoms

Large time-to-apply values will cause a delay between Calico Security Policy commits and enforcement in the dataplane. This is dependent on how Calico waiting for kube-proxy to release the iptables lock, which is influenced by the number of services in use

Threshold breach recommendations

Increase cluster resources, reduce the number of kubernetes services if possible

Priority Level

Recommended

Top Metrics: Graph update time (Quantile 0.99)

Metric

felix_calc_graph_update_time_seconds{quantile="0.5"}/{quantile="0. 9"}/{quantile="0.99"}

Explanation

This metric reports the time taken to update the calculation graph for each datastore on an update call, viewed at the median, 90th percentile and 99th percentile. The calculation graph is the Felix component that takes all the policies/workload endpoints/host endpoints information that it has received from Typha, and distills it down to dataplane updates that are relevant for this node.

Threshold value recommendation

After start of day (where we will typically get a large update), then values should be sub 1 second (with occasional blips to 1+ seconds). Should be measured in milliseconds with the occasional blip to a second or two. If this is constantly in values of seconds, this needs to be investigated

Threshold breach symptoms

High values indicate high CPU usage in felix and slow dataplane updates

Threshold breach recommendations

Increase cluster resources. Check calico-node logs. Rollout restart calico-node(s) if needed

Priority Level

Recommended

Top Metrics: breadcrumb size (Quantile 0.99)

Metric

max by (instance) (typha_breadcrumb_size{quantile="0.99"})

Explanation

Typha stores datastore changes as a series of blocks called breadcrumbs. Typha will store updates inside of these breadcrumbs (for example if a pod churned, this would be a single update). Typha can store multiple updates in a single breadcrumb with the default maximum size number being 100

Threshold value recommendation

Typha generating blocks of size 100 during start up is normal. If Typha is consistently generating blocks of size 90+, this can indicative of an overloaded Typha and should be investigated

Threshold breach symptoms

Maintained block of sizes of 100 could indicate that Typha is falling behind on information and updates contained in the datastore. This will lead to Typha clients also falling behind(e.g. Calico Policy object may not be current)

Threshold breach recommendations

Check Typha logs and resource usage. Check if there is a lot of activity within the cluster that would cause Typha to send large breadcrumbs (for example, a huge amount of pod churn)

Priority Level

Recommended

Top Metrics: Non-Blocking Breadcrumb Fraction

Metric

(sum by (instance) (rate(typha_breadcrumb_non_block[30s])))/((sum by (instance) (rate(typha_breadcrumb_non_block[30s])))+(sum by (instance) (rate(typha_breadcrumb_block[30s]))))

Explanation

Typha stores datastore changes as a series of blocks called "breadcrumbs". Each client "follows the breadcrumbs" either by blocking and waiting or skipping straight to the next one (non-blocking) if it is already available. **Non-blocking breadcrumb actions** indicate that Typha is constantly sending breadcrumbs to keep up with the datastore. **Blocking breadcrumb actions** indicate that Typha, and the client, have caught up, are up-to-date, and are waiting on the next breadcrumb. This metric will give a ratio between blocking and non-blocking actions that can indicate the health of Typha, it's clients and the cluster.

Threshold value recommendation

As load on Typha increases, the ratio of skip ahead non-blocking reads increases. If it approaches 100% then Typha may be overloaded (since clients only do non-blocking reads when they're behind).

Threshold breach symptoms

Consistent non-blocking breadcrumbs could indicate that Typha is falling behind on information and updates contained in the datastore. This will lead to Typha clients also being behind (e.g. Calico Policy object may not be current)

Threshold breach recommendations

Check Typha and Felix logs and resource usage. Check if there is a lot of activity within the cluster that would cause Typha to continuously send non-blocking breadcrumbs

Priority Level

Recommended

Top Metrics: Client fall-behind (Quantile 0.99)

Metric

max by (instance) (typha_client_latency_secs{quantile='0.99'})

Explanation

This metric measures how far behind Typha's clients are at reading updates. This metric will increase if:

a) The client (e.g Felix) is slow or overloaded and cannot keep up with what Typha is sending or

b) Typha is overloaded and it cannot keep up with writes to all its clients.

This metric is a good indication of your cluster, Felix and Typha health.

Threshold value recommendation

It is normal for this to spike when new clients connect, they must download and process the snapshot, during which time they will fall slightly behind. Persistent latency warrants investigation

Threshold breach symptoms

Typha clients will be behind in time on updates Typha is sending. Potential symptoms could include Calico Network Policies being out-of-sync

Threshold breach recommendations

Check Typha and Felix logs and resource usage.

It is recommended to focus on Felix logs and resource usage first, as there is generally more overhead with Felix and thus more of a chance of overload. Rollout restart Typha(s) and Calico-node(s) if needed

Priority Level

Recommended

Top Metrics: Client Ping Latency (Quantile 0.99)

Metric

max by (instance) (typha_ping_latency{quantile="0.99"})

Explanation

This metric tracks the round-trip-time from Typha to a client. How long it takes for Typha's clients to respond to pings over the Typha protocol.

Threshold value recommendation

An increase in this metric above 1 second indicates that the clients, network or Typha are more heavily loaded. It is normal for intermittent spikes. Persistent latency above 1 second warrants investigation.

Threshold breach symptoms

Typha clients could be behind in time on updates Typha is sending. Potential symptoms include Calico Network Policies being out-of-sync

Threshold breach recommendations

Check Typha and Felix logs and resource usage. It is recommended to focus on Felix logs and resource usage first, as there is generally more overhead with Felix and thus more of a chance of overload. Rollout restart Typha(s) and Calico-node(s) if needed

Priority Level

Recommended

Top Metrics: Datastore Updates Total

Metric

sum by (instance) (rate(typha_updates_total[30s]))

Explanation

This metric shows the rate of updates from the datastore(s). For example, updates to Pods/Nodes/Policies/etc.

Threshold value recommendation

Intermittent spikes are expected. Constant updates indicates a very busy cluster (e.g. lots of pod churn)

Threshold breach symptoms

Constant updates could lead to overloaded Typhas and thus Typhas clients could fall behind

Threshold breach recommendations

Ensure Typha has enough resources to handle a very dynamic cluster

Priority Level

Optional